MOLCAS version 8.2

User's Manual

Contents

I Introduction 1

1 Introduction to Molcas 3

1.1 MOLCAS Quantum Chemistry Software 3

1.2 The MOLCAS Manual 5

1.3 New features and updates 7

1.4 Parallelization efforts for MOLCAS modules 9

1.5 Acknowledgment 9

1.6 Citation for MOLCAS 9

1.7 Web Site 10

1.8 Disclaimer 10

II Installation Guide 11

2 Installation 13

2.1 Prerequisites 13

2.2 Configuring MOLCAS 15

2.3 Building MOLCAS 19

2.4 GUI and documentation 20

2.5 Parallel Installation 20

3 Maintaining the package 27

3.1 Tailoring 27

III Short Guide to Molcas 31

4 Quickstart Guide for Molcas 33

4.1 Introduction 33

4.2 Molcas Environment Setup 33

4.3 Customization of Molcas Execution 34

4.4 Molcas Command-Line Help System 36

4.5 Input Structure and EMIL Commands 36

4.6 Basic Examples 36

4.7 Analyzing Results: Output Files and the LUSCUS Program 40

5 Problem Based Tutorials 43

5.1 Electronic Energy at Fixed Nuclear Geometry 43

5.2 Optimizing geometries 53

5.3 Computing excited states 59

6 Program Based Tutorials 69

6.1 8.2 Flowchart 70

6.2 Environment and EMIL Commands 70

6.3 GATEWAY - Definition of geometry, basis sets, and symmetry 72

6.4 SEWARD — An Integral Generation Program 75

6.5 SCF — A Self-Consistent Field program and Kohn Sham DFT 75

6.6 MBPT2 — A Second-Order Many-Body PT RHF Program 79

6.7 RASSCF — A Multi Configurational SCF Program 80

6.8 CASPT2 — A Many Body Perturbation Program 84

6.9 RASSI — A RAS State Interaction Program 86

6.10 MOTRA — An Integral Transformation Program 91

6.11 GUGA — CI Coupling Coefficients Program 92

6.12 MRCI — A Configuration Interaction Program 93

6.13 CPF — A Coupled-Pair Functional Program 94

6.14 CCSDT — A Set of Coupled-Cluster Programs 94

6.15 ALASKA and SLAPAF: A Molecular Structure Optimization 97

6.16 MCKINLEY — A Program for Integral Second Derivatives 98

6.17 MCLR — A Program for Linear Response Calculations 99
6.18 GENANO — A Program to Generate ANO Basis Sets 100
6.19 FFPT — A Finite Field Perturbation Program 101
6.20 VIBROT — A Program for Vibration-Rotation on Diatomic Molecules 101
6.21 SINGLE_ANISO — A Magnetism of Complexes Program 101
6.22 GRID_IT: A Program for Orbital Visualization 104
6.23 Writing MOLDEN input 105
6.24 Most frequent error messages found in MOLCAS 106
6.25 Tools for selection of the active space 109
6.26 Some practical HINTS 110

IV User's Guide 117

7 The Molcas environment 119

7.1 Overview 119
7.2 Commands and environment variables 123
7.3 General input structure. EMIL commands 130

8 Programs 139

8.1 ALASKA 139
8.2 AVERD 143
8.3 CASPT2 145
8.4 CASVB 153
8.5 CCSDT 164
8.6 CHCC 169
8.7 CHT3 173
8.8 CMOCORR 175
8.9 CPF 177
8.10 DYNAMIX 179
8.11 EMBQ 183
8.12 ESPF (+ QM/MM INTERFACE) 188
8.13 EXPBAS 196
8.14 FALCON 198

8.15 FFPT . . . 200
8.16 GATEWAY . . . 204
8.17 GENANO . . . 227
8.18 GEO . . . 232
8.19 GRID_IT . . . 237
8.20 GUESSORB . . . 240
8.21 GUGACI . . . 242
8.22 GUGADRT . . . 246
8.23 GUGA . . . 248
8.24 LOCALISATION . . . 251
8.25 LOPROP . . . 256
8.26 MBPT2 . . . 260
8.27 MCKINLEY (a.k.a. DENALI) . . . 265
8.28 MCLR . . . 267
8.29 MCPDFT . . . 271
8.30 MKNEMO . . . 273
8.31 MOTRA . . . 283
8.32 MPPROP . . . 286
8.33 MRCI . . . 289
8.34 MULA . . . 293
8.35 NEMO . . . 298
8.36 NUMERICAL_GRADIENT . . . 309
8.37 POLY_ANISO . . . 310
8.38 QMSTAT . . . 310
8.39 QUATER . . . 321
8.40 RASSCF . . . 323
8.41 RASSI . . . 341
8.42 RPA . . . 350
8.43 SCF . . . 350
8.44 SEWARD . . . 363
8.45 SINGLE_ANISO . . . 378
8.46 SLAPAF . . . 383
8.47 SURFACEHOP . . . 397
8.48 VIBROT . . . 400
8.49 The Basis Set Libraries . . . 404

9 GUI **419**

9.1 Writing LUSCUS/MOLDEN input 419

V Advanced Examples and Annexes **421**

10 Examples **423**

10.1 Computing high symmetry molecules. 423

10.2 Geometry optimizations and Hessians. 451

10.3 Computing a reaction path. 469

10.4 High quality wave functions at optimized structures 476

10.5 Excited states. 490

10.6 Solvent models. 522

10.7 Computing relativistic effects in molecules. 533

10.8 Extra information about basis sets and integrals 540

10.9 Core and Embedding Potentials within the SEWARD Program 543

Part I

Introduction

Section 1

Introduction to Molcas

1.1 Molcas Quantum Chemistry Software

MOLCAS is a Quantum Chemistry software package developed by scientists to be used by scientists. It is neither a commercial product nor is it sold for significant profit gain by its owner, Lund University. The authors of MOLCAS have assembled their collected experience and knowledge in computational Quantum Chemistry to produce a research product which is used as a platform by the scientists in the MOLCAS network to develop new and improved computational tools in Quantum Chemistry. Several of the codes in the MOLCAS software have newly developed leading-edge features. Along with these new capabilities, users should not be surprised to ocasionally discover bugs when using MOLCAS.

The basic philosophy behind MOLCAS is to develop methods that allow accurate *ab initio* treatment of very general electronic structure problems for molecular systems in both ground and excited states which is not an easy task. Nowadays, knowledge about how to obtain accurate properties for single-reference dominated ground states is well developed, and MOLCAS contains a number of codes that can perform such calculations (MP2, CC, CPF, DFT etc). All these methods treat the electron correlation starting from a single determinant (closed or open shell) reference state. Such codes are today's standard in most Quantum Chemistry program.

However, MOLCAS is to be able to treat, highly degenerate states, such as those occurring in excited states, transition states in chemical reactions, diradicaloid systems, heavy metal systems, as well as other chemically important problems, all at the same level of accuracy. This is a much more difficult problem, since a single-determinant approach does not work well in these cases. The key feature of MOLCAS is the multiconfigurational approach. MOLCAS contains codes for general and effective multi-configurational SCF calculations at the Complete Active Space (CASSCF) level, but also employs more restricted MCSCF wave functions such as the Restricted Active Space, RASSCF, and the Generalized Active Space, GASSCF. It is also possible using either CASSCF or RASSCF to employ optimization techniques and obtain equilibrium geometries, transition-state structures, force fields, and vibrational energies using gradient techniques.

Although the RASSCF approach is known to give reasonable structures for degenerate systems both in ground and excited states, in general it is not capable of recovering more than a fraction of the correlation energy. Therefore, it becomes necessary to supplement the multi-configurational SCF treatment with a calculation of dynamic correlation effects. In the earliest version of Molcas, this was achieved by means of the multi-reference (MR) CI method. This method has severe limitations in the number of electrons that can be correlated and the size of the reference space which limits study to excited states of small molecules. However, the MRCI code in Molcas does have the capacity to produce very accurate wave functions and potential energy surfaces, and is used by many groups for this purpose. In fact, it is also possible to run the COLUMBUS MRCI code together with Molcas.

During the period of 1986-90, a new method called CASPT2 was developed, which computes dynamic electron correlation effects for multi-configurational wave functions based on second order perturbation theory and was included into the second version of Molcas. From the beginning it was not clear whether or not the CASPT2 method would be sufficiently accurate to be useful in practice, but it turned out to be surprisingly accurate in a number of different types of chemical applications. The CASPT2 approach has become especially important in studies of excited states and spectroscopic properties of large molecules, where no other *ab initio* method has, so far, been applicable. Since the CASPT2 method is based on second order perturbation theory and has, therefore, limitations in accuracy, the error limits have been investigated in a large number of applications. The relative energy errors are small in almost all cases leading to results which can be used for conclusive predictions of molecular properties in ground and excited states. Important application areas for the CASPT2 method are potential energy surfaces for chemical reactions, photochemistry, transition metal chemistry, and heavy element chemistry.

A multi-state version of CASPT2 is available, which allows for the simultaneous study of several electronic states, including their interaction to second order. This code is especially useful in cases where two or more energy surfaces are close in energy. An analytical CASPT2 gradient code is in the process of development, but this work is as yet unfinished. In place of the analytical gradient capability, the present version (8.2) includes a numerical procedure, which allows automatic geometry optimization at the CASPT2 level of theory. It is applicable to all states and systems for which the CASPT2 energy can be computed including the calculation of vibrational frequencies. It is important to note that the CASPT2 method is under constant development.

In the present version (8.2) the Multiconfiguration Pair-Density Functional Theory, MC-PDFT, is also available to treat dynamical correlation.

If only a few electrons are correlated, the active space can be quite large without too many configurations being generated, but in most cases the number of active electrons is comparable to the number of active orbitals. Occasionally, a larger active space would be preferred, but would result in too many configurations (more than a few million CSF's). The more general RASSCF scheme can be useful, at the price of less efficient calculations, and the risk of bad convergence properties in the orbital optimization. The CASPT2 program handles also such wave functions, but will not include correlation within the active space, i.e., interaction with states that would have been included in the full CASSCF but are excluded by the RASSCF restrictions. The RASSCF wave function is regarded as an accurate approximation to the full CASSCF wave function, and the CASPT2 program evaluates only dynamic correlation that involves at least one non-active orbital.

Molcas not only contains the ability to produce various types of wave functions, but also can compute molecular properties using formulas of expectation values or finite perturbation theory using the RASSI program. The RASSI program has the capacity to compute the interaction between several RASSCF wave functions based on different orbitals which are generally non-orthonormal (i.e. a non-orthogonal CI). RASSI is routinely used to compute transition dipole moments in spectroscopy, to study electron transfer, and to obtain eigenstates of a relativistic Hamiltonian with inclusion of spin-orbit interaction.

Scalar, i.e. spin-averaged, relativistic effects are typically included in any calculations by using Douglas-Kroll-Hess transformation of one-electron integrals. The standard basis set library ANO-RCC [1, 2, 3, 4, 5] is optimized for use with these integrals and to include correlation of semi-core orbitals, and to to have uniform quality across the periodic system up to element 96, Curium. For heavier elements, typical calculations include the spin-orbit interaction by using CASSCF wave functions as a many-electron basis set, letting RASSI compute a Hamiltonian matrix over the set of all spin-components of these functions, correct for dynamic correlation using CASPT2 and include a one-electron spin-orbit Hamiltonian. This procedure has been shown to give accurate results in a number of studies for actinides and other heavy atom systems [6].

It is also possible to model solvent effects by adding a reaction field Hamiltonian (PCM). A new QM/MM model is also included in 8.2.

The release of MOLCAS-7 leads to many important enhancements. The sizes of the systems that can be treated with MOLCAS were previously restricted because of limitations in storing two-electron integrals for large basis sets. This system size limitation has been substantially reduced by the introduction of a Cholesky decomposition of the two-electron integrals. This feature is used in MOLCAS-7 at all levels of theory [7, 8, 9] and speeds up calculations by orders of magnitude, extending the size of basis sets that can be used. Accuracy can be controlled by the threshold used in the decomposition. The same approach can be used to generate RI auxiliary basis sets on the fly, allowing the calculation of energy derivatives for HF, MP2, DFT, and CASSCF levels of theory.

It is important to emphasize that important problems in Quantum Chemistry cannot be solved by simply applying ***black box*** techniques. Nor is MOLCAS a ***black box*** tool. A typical MOLCAS user should be someone with a high degree of chemical insight, who has some knowledge of different Quantum Chemical models in use today, and, most importantly, is able to apply these models to the appropriate chemical problem while understanding the inherent accuracy of these methods. The typical MOLCAS user should also apply critical analysis of results, take nothing for granted, and always check that the results are consistent with the model that was used. The skill to use MOLCAS effectively will not come immediately, but the user has several resources including this manual and examples which explain how different key projects were solved using MOLCAS . Users are certain to find them helpful in their own attempts to master the software for use in chemical applications. The MOLCAS group also arranges regular workshops, which provide a more intimate environment on learning how to use MOLCAS .

1.2 The Molcas Manual

1.2.1 Manual in Four Parts

This manual is designed for use with the *ab initio* Quantum chemistry software package MOLCAS 8.2 developed at the by the world-wide MOLCAS team where its base and origin is the Department of Theoretical Chemistry, Lund University, Sweden. MOLCAS is designed for use by Theoretical Chemists and requires knowledge of the Chemistry involved in the calculations in order to produce and interpret the results correctly. The package can be moderately difficult to use because of this ‘knowledge requirement’, but the results are often more meaningful than those produced by ***blackbox*** packages which may not be sufficiently chemically precise in either input or output.

The MOLCAS manual is divided in four parts to facilitate its use.

1. The **Molcas *Installation Guide*** describes simple and more complex aspects on how to install, tailor, and control the MOLCAS package.

2. The ***Short Guide to* Molcas** is a brief introductory guide which addresses the needs of the novice and intermediate users and is designed for all those who want to start using MOLCAS as soon as possible. Only basic environment definitions, simple input examples, and minimal description of output results are included in the short guide.

   Two types of introductory tutorials are given in the short guide: problem-based and program-specific.

   (a) Problem-based tutorials are exercises focused on solving a simple Quantum Chemical project and contain all the required input files. Examples include computing electronic energy of a molecule at different levels of theory, optimizing the geometry of a molecule, calculating the transition state in the ground state of a chemical system, and computing an excited state. The input files for this section can be found in the directory $MOLCAS/doc/samples/problem_based_tutorials. These examples are also employed in MOLCAS workshops that the MOLCAS team has organized in recent years.

   (b) Another type of tutorial is designed for the first-time user to provide an understanding of program modules contained in MOLCAS include simple, easy-to-follow examples for many of these modules.

   The systems covered in the short guide are not necessarily calculated with most suitable methods or produce highly significant results, but provide both several tips for the beginner and actual input file formats.

   The ***Short Guide to* Molcas** can be independently printed as a booklet.

3. The **Molcas *User’s Guide*** contains a complete listing of the input keywords for each of the program modules and a information regarding files used in each calculation. Here the user will find all keywords that can be used together with a specific program and thus how to set up the input for a MOLCAS run.

4. ***Advanced Examples*** and ***Annexes*** include outlines of actual research performed using MOLCAS.

   The approach to a research project is outlined including input files and shell scripts. More importantly, however, the value of the calculations is evaluated and advanced features of 8.2 are used and explained to improve the value of the results.

The complete manual is available on the net in HTML and PDF formats
(*http://www.molcas.org*).

1.2.2 Notation

For clarity, some words are printed using special typefaces.

- Keywords, i.e. words used in input files, are typeset in the small-caps typeface, for example EndOfInput.
- Programs (or modules) are typeset in the teletype typeface. This will eliminate some potential confusion. For example, when discussing the RASSCF method, regular uppercase letters are used, while the program will look like `RASSCF`.
- Files are typeset in the slanted teletype typeface, like *InpOrb*.
- Commands, unix or other, are typeset in a sans serif typeface, like `ln -fs`.
- Complete examples, like input files, shell scripts, etc, are typeset in the teletype typeface.

1.3 New features and updates

Below is presented a list of the major new features of Molcas. These features comprise a number of new codes and introduction of new methods, but also considerable updates of many of the programs in Molcas. We keep some history, so that people who are using older versions of Molcas can get a feeling for what has happened on later versions

New features in 8.2:

- General improvements:
  - Improved CASPT2 XMS
  - Improved parallelization of CASPT2
- New codes
  - A new, MC-PDFT program is now available, and additional functionals have been included.
- User visible changes:
  - Renamed environment variables:
    - MOLCAS_NPROCS instead of CPU
    - MOLCAS_MEM instead of MOLCASMEM
- Visualisation
  - GRID_IT produces .lus files, for visualisation with LUSCUS version 0.8.3
  - keyword ASCII in GRID_IT is obsolete, and can only be used to generate grid files for GV

New features in 8.0:

- General improvements:
  - includes major bug fixes;
  - enhanced performance;
  - better parallelization;
  - better support for the Intel, and GCC compilers;
- New codes and major updates:
  - MC-PDFT combines multiconfigurational wavefunctions with density functional theory to recover both static and dynamical correlation energy;
  - GASSCF allows for more flexibility in choosing the active space;
  - EMBQ is general purpose embedding technique;
  - FALCON is fragment-based approach for computing an electronic energy of the large systems;
  - GEO/HYPER module for constrained multi-fragment geometry optimisation in internal coordinates;
  - enhanced I/O via the Files In Memory technology;
  - SINGLE_ANISO code received several important updates:
    - CRYS: extraction of the parameters of the multiplet-specific crystal field for lanthanides;
    - UBAR: construction of the blocking barriers of single-molecule magnets;
    - ABCC: magnetic and anisotropy axes are given in the crystallographic *abc* system;
- New features in existing codes:
  - Relativistic exact decoupling (X2C/BSS/infinite-order DKH);
  - Local X2C/BSS/infinite-order DKH;
  - RICD analytical gradients are available for the MBPT2, and CASSCF methods;
  - auto-segmentation in CD-based coupled cluster CHCC and CHT3 modules;
  - Orbital-free density embedding;
  - more robust and efficient SLAPAF module;
  - enhanced EMIL functional;
- Installation and tools:
  - first release of the Global Arrays free MOLCAS; a new parallel framework of MOLCAS requires only MPI-2 library;
  - better support for Mac OS X (including the both serial and parallel installations);

1.4 Parallelization efforts for Molcas modules

Presented below is a table of modules in Molcas that *can* benifit from parallel execution through distribution of work and/or resources. If a module is not listed in this table, and the module-specific documentation does not mention anything about parallelization, then you have to assume the module is not (efficiently) parallelized. This means that even though it will get executed in parallel, all processes will perform the same serial calculation! Be aware that for parallel modules with serial components, the use of the serial components (indirectly or through the use of a keyword) might adversely affect CPU and memory usage for large calculations. In that case, you might have to increase the runtime or memory, or avoid/use keywords that activate/deactivate the serial components.

Table 1.1: Modules in Molcas which benefit from parallel processing.

Module	Parallel speed-up expected for	Notable non-parallel parts
`SEWARD`	conventional 2-el integrals Cholesky vectors	1-el integrals Douglas-Kroll-Hess properties
`SCF`	orbital optimization	properties
`RASSCF`	orbital optimization	CI optimization properties
`MBPT2`		
`CASPT2`	Cholesky vectors	conventional 2-el integrals properties multi-state interaction
`ALASKA`	displacements (if using numerical gradients)	
`GEO`	displacements	

1.5 Acknowledgment

The Molcas project is carried out by the Lund University quantum chemistry group supported by the Swedish Science Research Council (VR).

The MOLCAS group is a member of the Linnaeus project *Organising Molecular Matter, OMM.*

The MOLCAS group acknowledges the contributions from the scientists in the MOLCAS Network, who are members of the MOLCAS developers team.

The Lund University Center for Scientific and Technical Computing (LUNARC) has provided computer resources for the project.

1.6 Citation for Molcas

The recommended citations for Molcas Version 8.2 are:

Molcas 8: F. Aquilante, J. Autschbach, R. K. Carlson, L. F. Chibotaru, M. G. Delcey, L. De Vico, I. Fdez. Galván, N. Ferré, L. M. Frutos, L. Gagliardi, M. Garavelli, A. Giussani, C. E. Hoyer, G. Li Manni, H. Lischka, D. Ma, P. Å. Malmqvist, T. Müller, A. Nenov, M. Olivucci, T. B. Pedersen, D. Peng, F. Plasser, B. Pritchard, M. Reiher, I. Rivalta, I. Schapiro, J. Segarra-Martí, M. Stenrup, D. G. Truhlar, L. Ungur, A. Valentini, S. Vancoillie, V. Veryazov, V. P. Vysotskiy, O. Weingart, F. Zapata, R. Lindh, *Journal of Computational Chemistry*, **37**, 506, (2016). DOI: 10.1002/jcc.24221

Molcas 7: F. Aquilante, L. De Vico, N. Ferré, G. Ghigo, P.-Å. Malmqvist, P. Neogrády, T. B. Pedersen, M. Pitonak, M. Reiher, B. O. Roos, L. Serrano-Andrés, M. Urban, V. Veryazov, R. Lindh, *Journal of Computational Chemistry*, **31**, 224, (2010). DOI: 10.1002/jcc.21318

Code development: V. Veryazov, P.-O. Widmark, L. Serrano-Andrés, R. Lindh, B. O. Roos, *International Journal of Quantum Chemistry*, **100**, 626 (2004). DOI: 10.1002/qua.20166

Molcas 6: G. Karlström, R. Lindh, P.-Å. Malmqvist, B. O. Roos, U. Ryde, V. Veryazov, P.-O. Widmark, M. Cossi, B. Schimmelpfennig, P. Neogrády, L. Seijo, *Computational Material Science*, **28**, 222 (2003). DOI: 10.1016/S0927-0256(03)00109-5

The following persons have contributed to the development of the MOLCAS software:

K. Andersson, F. Aquilante, M. Barysz, A. Bernhardsson, M. R. A. Blomberg, J. Boström, Y. Carissan, L. Chibotaru, D. L. Cooper, M. Cossi, M. Delcey, A. Devarajan, L. De Vico, I. Fdez. Galván, N. Ferré, M. P. Fülscher, A. Gaenko, L. Gagliardi, G. Ghigo, C. de Graaf, S. Gusarov, B. A. Hess, D. Hagberg, J. M. Hermida-Ramón, A. Holt, G. Karlström, J. W. Krogh, R. Lindh, P.-Å. Malmqvist, T. Nakajima, P. Neogrády, J. Olsen, T. B. Pedersen, D. Peng, M. Pitonak, J. Raab, M. Reiher, B. O. Roos, U. Ryde, I. Schapiro, B. Schimmelpfennig, M. Schütz, L. Seijo, L. Serrano-Andrés, P. E. M. Siegbahn, J. Stålring, B. Suo, P. Sushko, T. Thorsteinsson, T. Tsuchiya, L. Ungur, S. Vancoillie, V. Veryazov, V. Vysotskiy, and P.-O. Widmark.

1.7 Web Site

Please contact on the web at :
URL: *`http://www.molcas.org`*

1.8 Disclaimer

MOLCAS is shipped on an "as is" basis without warranties of any kind. The authors of MOLCAS-8 therefore assume no responsibility of any kind from the use of the supplied material. Permission is granted to use the package, but not to reproduce or redistribute any part of this material by any means. Material in the software may not be included in any commercial product. The authors reserve the right to change plans and existing material without notice.

Part II

Installation Guide

Section 2

Installation

The present installation guide describes the necessary steps for installing and tailoring MOLCAS. It also describes the steps for applying updates whenever necessary.

The installation procedure can be reduced to a few simple steps:

1. Extract the contents of the tar
2. Configure the package
3. Build the package
4. Build GUI and documentation (optional)
5. Make the package generally available.

2.1 Prerequisites

2.1.1 Prerequisite hardware

In general, MOLCAS can be built on any hardware that runs under a UNIX operating system. Some of these variants of hardware and software have been tested by us, and you should not have any problems to install MOLCAS on any of these. For other platforms you will most likely need to put some extra effort into the installation. In many cases the only effort on your part is setting some compiler flags, paths to system software etc. For a list of the platforms where we have successfully installed MOLCAS see our homepage: *http://www.molcas.org*.

To load the executables resident, sufficient memory is required. In addition, the programs are enabled to allocate work space dynamically. To avoid excessive paging we recommend that your machine should be equipped with at least 2 GB of memory per running application. Note, that MOLCAS will run faster with more memory.

To build MOLCAS you may need up to 2 GB of free disk space depending on what you choose to install and how (e.g. CMake keeps the object files around). The actual size of a MOLCAS installation is around 300-600 MB. To run the verification tests of MOLCAS you should have a scratch disk with up to 1 GB of free disk space, depending on the suite you run. For the "small" set about 400 MB will suffice. To perform larger calculations, ample amount of scratch disk space is necessary. The exact amount varies with the type of systems studied, but a general recommendation is at least 4 GB of disk space, per production run.

2.1.2 Prerequisite software

If you obtain the source code of Molcas, then you need to make certain that the necessary software is available to build Molcas. The minimum requirements are:

- A Fortran compiler (with support for Fortran 2003 and later)
- A C compiler
- GNU make See URL *`http://www.gnu.org`* and navigate to the gnumake page or go directly to *`https://www.gnu.org/software/make/`*
- Perl (version 5.008 or higher)

Also, you can benefit from following optional dependencies:

- CMake (version 2.8.11 or higher)
- an optimized BLAS/LAPACK library (alternative for the slow built-in library based on Netlib)
- NVBLAS to utilise usage of GPU for some BLAS operations
- MPI-2 (to enable parallelization features of Molcas)
- Global Arrays (version 5 or higher, an alternative to the built-in DGA library)

The Graphical User Interface codes used in Molcas can be downloaded from the following pages: LUSCUS version >= 0.8.3 ($https : //sourceforge.net/projects/luscus/$), and GV ($http : //www.molcas.org/GV/$).

Molcas can be interfaced to other Quantum chemical software such as Columbus, QC-Maquis, Block. However at the moment there is no out-of-the-box installation tools.

Windows

To install Molcas under MS Windows (98/NT/XP/7/8) one should install Cygwin (freeware from RedHat Inc., which can be downloaded from *`http://www.cygwin.com`*). The minimal installation of Cygwin to run Molcas includes:

- check that user name (under Windows) does not contain spaces
- select a disk, which has enough space for installation of Cygwin and Molcas
- install Cygwin to the root of selected disk with all defaults

- run setup again and install the following packages: Devel→gcc-fortran, Devel→make, Devel→gcc-gcc, Utils→time, Perl→perl
- optionally install editors: Editors→mc, Editors→vim
- run cygwin.bat to create Cygwin environment for the user
- copy MOLCAS tar file into your home directory in Cygwin, and proceed with installation in the same way as under Linux.

MacOS

Installation of MOLCAS under MacOS requires installation of the Apple Developer Tools (Xcode) and a Fortran compiler. These programs could be downloaded from:

https://developer.apple.com/xcode/downloads/
http://opensource.apple.com/

http://gcc.gnu.org/wiki/GFortranBinaries#MacOS
http://hpc.sourceforge.net/
http://www.macports.org

However, if you are looking for an out of the box solution, you can download a Free PGI for Mac OS X distribution available at
http://www.pgroup.com/products/freepgi/index.htm

2.1.3 Preparing the installation

In order to install MOLCAS you need to choose a directory where the MOLCAS driver script is to be installed. The driver executes scripts and programs form the MOLCAS package and must be located in a directory included into the `PATH` variable. Usually this will be *`/usr/bin`* when installing as root, and *`/bin`* when installing as an unprivileged user.

The driver script *`molcas`* uses the value of the environment variable `MOLCAS` to identify which version to use. The major advantage with this mechanism is that it is easy to switch between different versions of MOLCAS by simply changing the environment variable `MOLCAS`. However if the current directory is a subdirectory (up to 3rd level) of a MOLCAS tree, the latter will be used regardless of the value of the `MOLCAS` variable.

MOLCAS itself can be located in any place on the disk. The installation can be done by root, or by an unprivileged user. In the later case you can copy the *`molcas`* driver script to an appropriate location, e.g. *`/usr/local/bin`*, after the installation.

2.2 Configuring Molcas

Before you can build MOLCAS you have to configure it. Most common platforms have been setup by the MOLCAS developers, so for a serial installation with default settings for compiler and compiler flags configuration of MOLCAS can be done without specifying any special extra options.

There are two ways to configure MOLCAS: using the configure script (alternative 1) or using cmake (alternative 2). The latter is more recent and does not support all the original platforms, but it supports most commonly used environments and is easier as it is usually able to autodetect the necessary libraries. The CMake alternative also makes it easier to synchronize different installations based on the same source. Therefore, we recommend you to use alternative 2. If you are already familiar with building MOLCAS 8.0 or an earlier version, it might be easier to keep using alternative 1, as you can then port your exisiting configuration options. Note that for certain external software (e.g. DMRG), CMake is mandatory, and thus alternative 2 is needed.

For new users, use of the setup script is recommended.

2.2.1 Simple configuration with the setup script

If you are new to building MOLCAS, it is recommended that you use the ***setup*** script to configure MOLCAS. Advanced users and/or users that need further customization should skip this section and use one of the alternatives in the next sections to configure MOLCAS.

The ***setup*** script will prompt you interactively about the most important settings. To get started, first unpack the MOLCAS source and then run:

```
./setup
```

in the main MOLCAS directory. Answer the questions and then proceed to section 2.3 to build MOLCAS.

2.2.2 Advanced configuration with the configure script (alternative 1)

You can start the configuration by running the ***configure*** script:

```
./configure
```

To know which flags can be used, run ./configure -h. The above command (without any flags) will use a default configuration, i.e. a serial MOLCASinstallation using built-in BLAS/LAPACK.

When configuration is finished, you should review the log file ***configure.log*** to see if everything is ok. There is no harm in running the configuration script even if it should fail, you simply rerun it with correct parameters.

If the configuration step was not successful, you probably are missing some prerequisite software, or this software is located in an unusual location on the disk. In the later case you might need to update your PATH, or use flag -path in configure.

MOLCAS supports out-of-source installation. If for some reason, you would like to install molcas under a separate tree, you can create a directory, and call configure with appropriate flags, e.g.

```
mkdir $HOME/molcas
cd $HOME/molcas
/sw/molcas_dist/configure -speed safe
```

The list all the options for configure, run

```
./configure -help
```

2.2.3 Advanced configuration with CMake (alternative 2)

Start configuration by creating a build directory and then running the cmake program with the location of the Molcas source. You can create the build directory as a subdirectory of the Molcas root directory, but we recommend you create it outside. For example, suppose the Molcas root is located at ***molcas*** in your home directory and you want to build it at ***molcas-build***:

```
mkdir ~/molcas-build
cd ~/molcas-build/
cmake ~/molcas/
```

After the first run, CMake keeps a cache of the configuration around as a file ***CMakeCache.txt***. If you want to change certain options (e.g. use a different compiler), you need to remove this file first. As an example, if we wish to reconfigure with the intel compilers instead we do:

```
rm CMakeCache.txt
CC=icc FC=ifort cmake ~/molcas/
```

Once the cache file is there, subsequent additional options do not require pointing to the Molcas source, just to the build directory itself. Here we add an option to build parallel Molcas:

```
cmake -DMPI=ON .
```

When using MPI, CMake will pick up the correct compiler and MPI library from the wrappers. So if you configure with MPI or you later wish to use a different MPI library/compiler wrapper, it is better to remove the cache file first, then (re)configure pointing to the wrappers:

```
rm CMakeCache.txt
CC=mpicc FC=mpifort cmake -DMPI=ON ~/molcas/
```

Summary of main options for CMake

To see all the CMake options, you can run the ccmake command to interactively select the options and their values. The following is a list of some of the most important options.

- `BUILD_SHARED_LIBS` — (`ON`/`OFF`): Enable building shared libraries (reduced disk space).

- CMAKE_BUILD_TYPE — (Debug/Garble/RelWithDebInfo/Release/Fast): Normally use Release. RelWithDebInfo may be useful for reporting a problem. Fast in unsupported and may give wrong results in some cases.
- CMAKE_INSTALL_PREFIX — Specify the directory where Molcas will be installed when running make install.
- GA — (ON/OFF): Enable interface with Global Arrays (see section 2.5).
- HDF5 — (ON/OFF): Enable HDF5 files (portable binary format) in some programs.
- LINALG — (Internal/Runtime/MKL/ACML/OpenBLAS): Select the linear algebra library (BLAS + LAPACK) against which to link Molcas. Internal uses the default (and slow) Netlib version included with Molcas. Runtime (experimental) offers the possibility of choosing the library at run time.
- MPI — (ON/OFF): Enable multi-process parallelization.
- OPENMP — (ON/OFF): Enable multi-thread parallelization (usually restricted to using a multi-threaded version of linear algebra libraries.
- TOOLS — (ON/OFF): Compile the tools that have CMake support.

Example 1: GCC C/Fortran compilers with GA/OpenMPI and OpenBLAS

- 64-bit Linux OS
- MPI preinstalled (install OpenMPI or MPICH with package manager)

```
# OpenBLAS
tar zxvf OpenBLAS-v0.2.15.tar.gz
cd OpenBLAS-0.2.15/
make USE_OPENMP=1 NO_LAPACK=0 INTERFACE64=1 BINARY=64 DYNAMIC_ARCH=1 \
      libs netlib shared
[sudo] make PREFIX=/opt/openblas-lapack-ilp64 install
# GA
tar zxvf /path/to/ga-5-4b.tgz
cd ga-5-4b/
./configure --enable-i8 --with-blas8 --with-lapack8 --with-scalapack8 \
      --prefix=/opt/ga54b-ilp64.OpenMPI
make
[sudo] make install
# Molcas
tar zxvf molcas.tgz
cd molcas
mkdir build && cd build/
export GA=/opt/ga54b-ilp64.OpenMPI
export OPENBLASROOT=/opt/openblas-lapack-ilp64
CC=mpicc FC=mpifort cmake -DMPI=ON -DGA=ON -DLINALG=OpenBLAS ../
make
```

Any other configurations can be easily derived from the above by simply removing the unneeded optional stuff.

Example 2: Intel C/Fortran compilers with GA/IntelMPI and MKL

- 64-bit Linux OS
- Intel Compilers/MPI preinstalled (usually on a cluster as a module)
- Infiniband (OpenIB)

```
# make sure Intel compilers/MPI are loaded
module load ...
# check compilers/environment
type mpiicc
type mpiifort
echo $MKLROOT
# GA
tar zxvf /path/to/ga-5-4b.tgz
cd ga-5-4b/
./configure --prefix=/opt/ga54b-ilp64.IntelMPI --enable-i8     \
            --with-openib --with-blas8="-L$MKLROOT/lib/intel64 \
            -lmkl_intel_ilp64 -lmkl_intel_thread -lmkl_core    \
            -liomp5 -lpthread -lm"                             \
            --with-scalapack8="-L$MKLROOT/lib/intel64          \
            -lmkl_scalapack_ilp64 -lmkl_intel_ilp64            \
            -lmkl_intel_thread -lmkl_core                      \
            -lmkl_blacs_intelmpi_ilp64 -liomp5 -lpthread -lm"
make
[sudo]make install
# Molcas
tar zxvf molcas.tgz
cd molcas
mkdir build && cd build/
export GA=/opt/ga54b-ilp64.IntelMPI
CC=mpiicc FC=mpiifort cmake -DMPI=ON -DGA=ON -DLINALG=MKL ../
make
```

2.3 Building Molcas

When the configuration step (section 2.2) is completed successfully, you can build MOLCAS. This is simply done by typing make in the MOLCAS root directory. It is recommended that you save the output from make in a log file for tracing of potential problems.

```
make > make.log 2>&1
```

After MOLCAS has been built correctly, you should absolutely run a basic verification to ensure that the installation is sane. See the next section 2.3.1 for details on verification.

2.3.1 Verifying the Molcas installation

After a successful build of MOLCAS you should verify that the various modules run correctly. Directory *Test/* contains various subdirectories with test inputs for MOLCAS. Use the command molcas verify [parameters] to start verification. Running this command without parameters will check the main modules and features of MOLCAS and we recommend this default for verifying the installation. You can also specify a keyword as argument that translates into a sequence of test jobs, or you can specify a list of test jobs yourself. The script has extended documentation, just run molcas verify –help.

To generate a report after performance tests you should execute a command molcas timing. The report is then located in the file *test/timing/user.timing*. The results of benchmark tests for some machines are collected on our webpage
http://www.molcas.org/benchmark.html At the completion of the test suite a log of the results is generated in the file *Test/results*. If installation was performed by another user (e.g. root), you can redefine the location of output files by adding the flag -path PATH. Each test job is signaled as either ok of failed. If there are any failed jobs, the outputs are saved in *Test/Failed_Tests*. Each test job tests for a resulting checksum for the modules tested. This checksum is typically the energy for a wavefunction program such as RASSCF, whereas other types of codes use other checksums.

The checksums will not match exactly with our reference values since different machines use different arithmetics. We have tried to make the acceptable tolerances as small as possible and at the same time make all tests pass successfully. It might be the case that your particular platform will produce one or more results that are just outside our tolerances, and in such a case the test is most likely ok.

2.4 GUI and documentation

You can download executables for the GUI from the MOLCAS webpage (*http://www.molcas.org*).

In order to build documentation in various formats, use the command make doc. Alternatively the documentation is also available no the web page (*http://www.molcas.org*).

2.5 Parallel Installation

Installation of MOLCAS for execution in multi-processor environments can be a bit more involved than the standard installation, this chapter considers those particulars not covered previously.

The parallelization of MOLCAS is achieved through the use of the Global Arrays (GA) API and some direct MPI calls. The API can be linked to an external GA library or to our own DGA library (an internal PGAS framework built upon MPI-2).

When using DGA (the default), the current list of supported MPI-2.2 implementations is given below: MPICH2/MPICH3, MVAPICH2, OpenMPI, Intel MPI.

When one wants to use an external GA library, it has to be configured and compiled separately. In that case, please read the section on using an external GA installation to properly configure and install GA first!!!

IMPORTANT: not all modules support distribution of work and/or resources through parallel execution, and even if they do it might be that some functionality is limited to serial performance. This is a list of core modules which *can*benefit from parallel execution: gateway, seward, scf, rasscf, caspt2. More detailed information regarding parallel behaviour can be found in the documentation of the respective module and in the table at the beginning of the manual about supported parallelism. If no information is available, you should conclude that there is nothing to be gained from parallel execution.

2.5.1 Supported MPI implementations

Most probably, you will use a free MPI-2 implementation such as MPICH2/MPICH3, MVAPICH2, or Open MPI.

MPICH2: *http://www.mcs.anl.gov/research/projects/mpich2/*
MPICH3: *http://www.mcs.anl.gov/research/projects/mpich3/*
MVAPICH2: *http://www.mvapich.cse.ohio-state.edu*
Open MPI: *http://www.open-mpi.org/*

NOTE: Open MPI versions older than v1.6.5 are not supported. More specifically, only Open MPI v1.6.5, and v1.8.1 are tested and known to work correctly with MOLCAS.

It is a very good idea to verify that the correct compiler environment is present before configuring MOLCAS. You should therefore check that the backend compiler of the wrappers is correct by running /path/to/mpif77 -show (MPICH2/MPICH3 and MVAPICH2) or /path/to/mpif77 –showme (Open MPI), which will list the actual executed command. If the backend compiler seems to be correct, also try to run it to see if it is properly detected (on some clusters you will need to load the appropriate module for the compiler). If all is well, you should be able to configure MOLCAS without any problems.

It is highly recommended to use the compiler that was used for the MPI library to build GA (optional) and MOLCAS to avoid compatibility issues. However, if you really want to use a different compiler than the compiler which was used for building the MPI library, you can do so by passing the -fc and -cc command line arguments (MPICH2/MPICH3 and MVAPICH2) to the wrappers, or setting the environment variables OMPI_F77/OMPI_F90 and OMPI_CC (Open MPI).

Several commercial MPI implementations exist such as HP-MPI, IBM's MPI-F, Intel MPI, SGI's MPT. Currently we only support Intel MPI. For the others that are not (yet) supported, it is recommended to either configure MOLCAS without parallel options and change the Symbols file after the serial configuration, or rely on cmake to use the correct options.

Please refer to the documentation of your MPI implementation for details on how to build programs, i.e. which wrappers to use and if necessary what libraries you need to link in.

2.5.2 Using an external Global Arrays installation (optional step)

If you wish to use an external GA library, it has to be installed before you build MOLCAS. You could e.g. use this if you have trouble with the built-in DGA solution. The installation instructions may be found at the Global Arrays home page:

http://hpc.pnl.gov/globalarrays/

Note that any problems with installation or other issues specific to GA are best resolved by contacting the GA authors directly, rather than the MOLCAS group. It is therefore a very good idea to run the GA testing code as a job on the cluster where you want to use MOLCAS to make sure that it works properly before continuing to install MOLCAS.

Global Arrays needs to be installed with 8-byte integer support using the flag(s) –enable-i8 –with-blas8[=...] [–with-scalapack8[=...]], and for infiniband clusters you probably need to use the –with-openib flag. When linking to an external library, e.g. the Intel MKL, do not forget to include the proper ilp64 library versions.

Please read the documentation of GA for more details about installation.

2.5.3 General overview of the procedure with the configure script (alternative 1)

In the simplest case, the parallel version of MOLCAS may be installed simply by specifying an flag -PARALLEL to *configure*. For example:

```
./configure -parallel
```

When using an external GA, pass the location of the installation to MOLCAS configure:

```
./configure -parallel -garoot /opt/ga-5.1
```

When the locations of the MPI *lib* and *include* directories is set incorrectly, you can specify them by setting their common root directory with the mpiroot flag:

```
./configure -parallel -mpiroot /usr/lib/openmpi
```

More likely, some individual tailoring will be required, the following summarizes the necessary steps:

1. Check that the correct wrapper compilers were detected, as specified in *$MOLCAS/Symbols*.
2. If needed, change the F77/F90 and CC variables in the Symbols file for any custom modifications you made to the wrappers.
3. Optionally install (and test) the external Global Arrays library.
4. Check the command for executing binaries in parallel, as specified by $RUNBINARY in *$MOLCAS/molcas.rte*.
5. Install (and test) MOLCAS.

Provided that steps 1–4 can be successfully accomplished, the installation of MOLCAS itself is unlikely to present many difficulties.

2.5.4 General overview of the procedure with cmake (alternative 2)

CMake accepts two main flags for parallel installation, one to specify the use of parallelization -DMPI=ON, and one to speficy an external GA library -DGA=ON instead of DGA (the default is -DGA=OFF, meaning no external GA is used, so do not confuse the option "-DGA" which means "define GA" with DGA). When using the latter -DGA=ON flag, make sure the GAROOT environment variable is exported and contains the path of the GA installation, before running cmake.

CMake will determine an appropriate MPI libary based on the compiler it finds, so in order to use a specific MPI library, just make sure the CC and FC variables point to the correct MPI wrappers!

The whole procedure is summarized below (square brackets showing optional commands):

```
[export GAROOT=/path/to/external/GA]
[CC=/path/to/mpicc] [FC=/path/to/mpifort] cmake -DMPI=ON [-DGA=ON] /path/to/molcas
make [-j4]
```

2.5.5 Running Molcas in parallel

A few comments on running on a cluster:

The very old MPICH versions sometimes need a file with a list of the nodes the job at hand is allowed to use. At default the file is static and located in the MPICH installation tree. This will not work on a workstation cluster, though, because then all jobs would use the same nodes.

Instead the queue system sets up a temporary file, which contains a list of the nodes to be used for the current task. You have to make sure that this filename is transfered to $mpirun. This is done with the -machinefile flag. On a Beowulf cluster using PBS as queue system the $RUNBINARY variable in *$MOLCAS/molcas.rte* should look something like:

RUNBINARY='/path/to/mpirun -machinefile $PBS_NODEFILE -np $MOLCAS_NPROCS $program'

The newer MPICH2/MPICH3 as well as MVAPICH2, which works through the use of the HYDRA daemons and does not need this command line argument, as well as Open MPI most likely only need the -np $MOLCAS_NPROCS command line option. They use mpiexec instead of mpirun.

Parallel execution of MOLCAS is achieved by exporting the environment variable MOLCAS_NPROCS, for example when running on 4 nodes use:

```
export MOLCAS_NPROCS=4
```

and continuing as usual.

In this section, we assume you will be using PBS on a cluster in order to submit jobs. If you don't use PBS, please ask your system administrator or consult the cluster documentation for equivalent functionality.

Example of a submit script

```
#!/bin/sh
#PBS -l walltime=10:00:00
#PBS -l nodes=4
#PBS -l pmem=3000mb

######## Job settings ###########
export MOLCAS_MEM=800
export SUBMIT=/home/molcasuser/project/test/
export Project=test000
export MOLCAS_NPROCS=4

######## modules ###########
. use_modules
module load intel/11.1
module load openmpi/1.4.1/intel/11.1


######## molcas settings ###########
export MOLCAS=/usr/local/molcas80.par/
export WorkDir=/disk/local/

######## run ###########
cd $SUBMIT
molcas $Project.input -f
```

Memory The maximum available memory is set using the PBS option pmem. Typically, MOLCAS_MEM will then be set to around 75% of the available physical memory. So for a parallel run, just divide the total physical memory by the number of processes you will use and take a bit less. For example, for a system with 2 sockets per node and 64 GB of memory, running 1 process per socket, we would set pmem to 30000 MB.

I/O The important thing to consider for I/O is to have enough scratch space available and enough bandwidth to the scratch space. If local disk is large enough, this is usually preferred over network-attached storage. MOLCAS requires the absolute pathname of the scratch directory to be the same across nodes.

Pinning Process pinning is sometimes required to achieve maximum performance. For CASPT2 for example, processes need to be pinned to their socket or NUMA domain.

The pinning configuration can usually be given as an option to the MPI runtime. With Intel MPI for example, one would set the I_MPI_PIN_DOMAIN variable to socket. Alternatively, you can use a third-party program to intervene on your behalf, e.g. *https://code.google.com/p/likwid/*. Please ask your system administrator how to correctly pin your processes.

GA specific issues When using GA, several problems can occur when trying to run jobs with a large amount of memory per process. A few example error messages are given here with their proposed solution.

```
(rank:0 hostname:node1011 pid:65317):ARMCI DASSERT fail.
 src/devices/openib/openib.c:armci_pin_contig_hndl():1142
 cond:(memhdl->memhndl!=((void *)0))
```

The error output in the Molcas errfile (stderr) then says:

```
Last System Error Message from Task 2:: Cannot allocate memory
```

Related messages that display a problem with *armci_server_register_region* instead of *armci_pin_contig_hndl* can also occur, and point to similar problems.

This can have two causes:

- Some parameters of the Mellanox *mlx4_core* kernel module were set too low, i.e., *log_num_mtt* and *log_mtts_per_seg*. These should be set according to the instructions on *http://community.mellanox.com/docs/DOC-1120*. Values of 25 and 0 respectively, or 24 and 1 should be fine.
- The 'max locked memory' process limit was set too low. You can check this value by running ulimit -a or ulimit -l. Make sure you check this through an actual job! Easiest is to start an interactive job and then execute the command. The value should be set to unlimited, or at least to the amount of physical memory available.

```
0: error ival=4 (rank:0 hostname:node1011 pid:19142):ARMCI DASSERT fail.
 src/devices/openib/openib.c:armci_call_data_server():2193
 cond:(pdscr->status==IBV_WC_SUCCESS)
```

This error is related to the value of the variable ARMCI_DEFAULT_SHMMAX, try setting it at least to 2048. If this is still too low, you should consider patching GA to allow higher values.

Section 3

Maintaining the package

3.1 Tailoring

Molcas, as shipped, is configured with some default settings. You can change some of these easily. You can change default settings used in Molcas (like memory usage, default scratch area, policy in saving files, etc.) by editing Molcas resource file: global resource file *$MOLCAS/molcasrc* or user resource file *$HOME/.Molcas/molcasrc*.

3.1.1 Dynamic memory

Most modules in Molcas utilize dynamic memory allocation. The amount of memory each module allocate is controlled by the environment variable `MOLCAS_MEM`. The amount of memory allocated is

- `MOLCAS_MEM` is undefined — 2048Mb of memory is allocated
- `MOLCAS_MEM=nn` — `nn`Mb is allocated. If this amount cannot be allocated, the module stops.

3.1.2 Disk usage

Today many workstations utilize 64-bit integers and addressing. However, old UNIX workstations and PC's had 32-bit integers resulting in a file size limit of 2GB. To circumvent these limitations, the I/O routines of Molcas support multifile files, where a "file" is in reality a logical file consisting of several physical files. The size limit of these physical files is controlled by the environment variable `MOLCAS_DISK`. The use of `MOLCAS_DISK` is not recommended.

3.1.3 Improving CPU performance

Molcas is shipped with a number of default setup files located in directory *cfg/*. The defaults in these files are set to a fairly safe level, but not necessary optimal. What you can change to improve performance is

- Compiler flags
- Mathematical (blas) libraries

The simplest way to set up optimization level, and/or compile MOLCAS with various BLAS libraries is to use configure -setup. This interactive script helps to make a proper selection of flags for improvement of MOLCAS performance.

If you do decide to try to improve the performance we recommend that you create a new setup file, for example, *cfg/local.cfg* and modify this file. It is not unlikely that your attempts to optimize the codes will lead you to a case where some modules work and others do not. In such a scenario it can be fruitful to have two copies of MOLCAS, one "safe" where all modules work and one "fast" where some modules do not function properly.

Changing the compiler flags is the easiest. Using the most aggressive optimization flags do sometimes lead to problems for some of the modules. We have tried to choose an optimization level that yields functioning code, but still reasonable fast. For some systems there is a predefined set of compiler flags for aggressive optimization. To compile MOLCAS with these flags you should run configure with flag -speed fast. Note that this aggressive optimization level is not supported by the MOLCAS team. In other words, you are using it at your own risk.

For some platforms you can utilize the vendor blas libraries. This will certainly yield better performance, but may not work on all platforms.

During configuration of MOLCAS it is possible to specify an external BLAS/LAPACK library. Use a flag -blas TYPE to specify the type of BLAS libary: lapack (for a standard lapack library), Goto (for GotoBLAS), Atlas (for ATLAS), MKL (for Intel MKL). You should also specify a flag -blas_lib -Wl,--start-group -L/path/to/blas -lmy-blas -Wl,--end-group specifying the link options. For example, to configure MOLCAS with Intel MKL library, you should issue a command ./configure -compiler intel -blas MKL -blas_lib -Wl,--start-group /opt/intel/mkl/lib/intel64 -lmkl_gf_ilp64 -lmkl_sequential -lmkl_core -Wl,--end-group

To compile MOLCAS with CUDA BLAS library, first, you have to compile the fortran wrapper provided by nVIDIA:

```
CUDA=/path/to/cuda/
FLAGS=-m64
gcc $FLAGS -I$CUDA/include/ -I$CUDA/src/ -c $CUDA/src/fortran_thunking.c -o \
$MOLCAS/lib/fortran.o
./configure -blas CUDA -blas_dir $CUDA/lib
```

or, if on a 64bit system:

```
./configure -blas CUDA -blas_dir $CUDA/lib64
```

After making changes to the setup files you have to issue the commands make veryclean, ./configure and make in the MOLCAS root directory. It is highly recommended to run the verification suite after any changes in configuration file.

3.1.4 Customizing handling of files

The location and attributes of files used by Molcas are defined in PRGM files. The master copy of these files is located at the `data` directory in the Molcas root. A user can copy these files and modify them. The highest priority is given to the files located in the subdirectory *prgm* in the current (CurrDir) directory, next the *$HOME/.Molcas* and finally the original location at *$MOLCAS/data*.

The simplest way to manipulate prgm files is to use the molcas prgm command. A command molcas prgm init global makes a copy of the prgm files in the *.Molcas* directory. molcas prgm init local creates a *prgm* subdirectory, and copies the prgm files into it. The editing of the files can be performed by your favourite editor, or by the molcas prgm script.

The structure of PRGM files is simple. The field `(file)` is followed by the FORTRAN name (as this file is known for Molcas), real file name (as this file is known by the Operating system), and finally the attributes.

The attributes are listed here:

ro	the file is accessed for reading
rw	the file is accessed for reading and writing
s	the file will be saved (copied) from the scratch area
m	the file will be saved (moved) from the scratch area
g	the file can be visualized by program ”gv/luscus”
t	the file is an ASCII text
	multifile
.	multifile (for internal use)
l	use lustre filesystem in parallel run
p	the file will be deleted
e	the file will be placed to memory (see FiM)
f	use an alternative file location (FastDir)

A command molcas prgm +x ScfOrb will add the attribute 'x' to all ScfOrb files. A regexp can be used for a filename. An opposite command molcas prgm -x ScfOrb will remove the attribute 'x'. A command moclas prgm comp [global] shows the list of modifications in the local (*/prgm*), or global (*.Molcas*) directories.

3.1.5 Improving I/O performance

In order to activate this technology for a MOLCAS scratch file, one needs to do three things. First, please edit an external resource **.prgm* (for example, *$MOLCAS/data/seward.prgm*) from the *$MOLCAS/data/* directory. If you don't have access to the root MOLCAS directory, then you can simply copy the needed resource file into your home *$HOME/.Molcas/* directory and edit it there. The editing of the file consists in adding the 'e' character to its attributes:

```
original: (file) ORDINT "$WorkDir/$Project."OrdInt rw*
modified: (file) ORDINT "$WorkDir/$Project."OrdInt rw*e
```

Second, you need to set up the MOLCAS_FIM environment variable to 1 i.e.:

```
export MOLCAS_FIM=1
```

The third and the final step is to specify the MOLCAS_MAXMEM (≥MOLCAS_MEM) parameter such that the MOLCAS_MAXMEM-MOLCAS_MEM difference (in MW) is sufficient to host an entire file in RAM. In other words, the MOLCAS_MAXMEM-MOLCAS_MEM difference should exceed the original filesize.

In general, not all MOLCAS files are suitable for placing in RAM. In particular, it is a bad idea to activate FiM for *RUNFILE*. In order to identify which MOLCAS's files are proper candidates for FiM, you can simply inspect the section *”II. I/O Access Patterns”* from a MOLCAS's output. All files with high ratio of I/O random Write/Read calls are good candidates for FiM. In particular case of the SEWARD module, the *ORDINT* file is a very good candidate for FiM:

```
II. I/O Access Patterns
- - - - - - - - - - - - - - - - - - - -
Unit  Name              % of random
                       Write/Read calls
- - - - - - - - - - - - - - - - - - - -
 1  RUNFILE             28.6/  11.5
 2  ORDINT             100.0/  24.0
 3  DNSMAT               0.0/   0.0
 4  TWOHAM               0.0/   0.0
 5  GRADIENT            88.9/   0.0
 6  DNSMAX               0.0/   0.0
 7  TWOHAX               0.0/   0.0
 8  SODGRAD             85.7/   0.0
 9  SOXVEC              85.7/   0.0
10  SODELTA             88.9/   0.0
11  SOYVEC              88.9/   0.0
12  ONEINT             100.0/  53.3
- - - - - - - - - - - - - - - - - - - -
```

3.1.6 Local modifications

Molcas is shipped with source code so you can make modifications yourself. You are, of course, responsible for the correctness of any such modification.

If you do make changes/additions to the source code that you feel is of interest to other users, we encourage you to make these available. Perhaps the best mechanism is to use the bulletin board on out homepage: *http://www.molcas.org*.

Check Molcas Programming Guide for a detailed description of development and distribution of modified code in Molcas .

Part III

Short Guide to Molcas

Section 4

Quickstart Guide for Molcas

4.1 Introduction

Running MOLCAS 8.2 requires a small number of operations. This section of the manual, entitled **Quickstart Guide for Molcas** is aimed at those users who want to immediately run a simple MOLCAS calculation in order to become familiar with the program. Basic hints are included which set the proper environment, build simple input files, run a calculation, and subsequently extract information from the resulting output.

4.2 Molcas Environment Setup

The environment variable (MOLCAS) and MOLCAS driver (*molcas*) must be defined in order to run MOLCAS. The MOLCAS environment variable points to the root directory of the MOLCAS installation and can be defined by the bash shell command

```
export MOLCAS=/home/molcas/molcas.version
```

The location of the MOLCAS driver is defined at installation time and is typically located in */usr/local/bin* or *$HOME/bin*. Check to ensure that this directory is included in your path. Otherwise, the path can be extended by the following command:

```
export PATH=$PATH:$HOME/bin
```

In addition, the variable MOLCAS_NPROCS is needed to run MOLCAS in parallel. This specifies the number of MPI processes that will be used.

It may be also convenient to define environment variables such as WorkDir which points to a directory for intermediate files and Project to define the name of a project:

```
export Project=MyMolecule
```

MOLCAS will provide default values if they are not explicitly defined. For a discussion of other MOLCAS environment variables, please see the section on **Environment and EMIL Command** . All environment variables can either be defined explicitly or entered in a shell script which can be subsequently executed.

4.3 Customization of Molcas Execution

MOLCAS has flexible control of organizing filenames and directories used during a calculation. The default values used for customization can be altered either by shell variables or a resource file *molcasrc* which is more preferable. A command molcas setuprc provides guided help if to create such file.

The terminology used in this chapter:

- LOG: the output and error files produced by MOLCAS .
- ProjectName: the Project name used for file naming.
- RUNFILE: a file used in a calculation will be named as ProjectName.Runfile,
- WorkDirName: the WorkDir name used as the directory for temporary/binary files produced by MOLCAS .
- Scratch: the scratch disk area which provides a path to a parent directory for WorkDirNames.

  The *WorkDir* variable used in the MOLCAS manual is constructed as Scratch/WorkDirName,
- CurrDir: the submit directory where the MOLCAS command was issued.

  Note, that in this tutorial, it is assumed that the input file is located in CurrDir,
- OutputDir: the output directory which is used for storage of extra output files, such as Orbital files and molden files.

It is quite important to understand, that if a user performs two consecutive runs of molcas, using the same scratch area (WorkDir) and project name, MOLCAS will try to reuse intermediate data, e.g. integrals and orbitals, in order to make a restart of a calculation. This can save time, but can also be can be dangerous if two consecutive calculations are not compatible.

Assuming that *molcasrc* does not exist, and no environment is set, the command molcas inputfile will use the following defaults:

- LOG is printed to the screen,
- OutputDir and CurrDir are defined to be the same directory,
- ProjectName is s taken as the name of ***inputfile*** by removing the suffix (before the last . (dot) character),
- Scratch is defined as */tmp/*,
- and WorkDirName is defined from the ProjectName plus a random suffix.

For example, when a user issues the following commands:

```
cd /home/joe/projects/water
vi H2O.DFT.input
molcas H2O.DFT.input
```

the following files will be generated:

```
/home/joe/projects/water/H2O.DFT.ScfOrb
/home/joe/projects/water/H2O.DFT.scf.molden
...
/tmp/H2O.DFT.15014/H2O.DFT.RunFile
...
```

If a flag -f is used in a Molcas command, `LOG` files will be stored in the `CurrDir` directory with a name `ProjectName`.log and `ProjectName`.err.

`ProjectName` can either be set in a shell script running Molcas or included directly into the Molcas command:

molcas Project=water H2O.DFT.input

will change the default value for `ProjectName` to water.

If the `MOLCAS_WORKDIR` environment variable is set either as part of Molcas command or is included in the ***molcasrc*** file), the name of WorkDir will NOT be random, but determined by the `ProjectName`.

Example:

```
cd /home/joe/projects/water
vi H2O.DFT.input
molcas MOLCAS_WORKDIR=/tmp Project=water -f H2O.DFT.input
```

will generate the following files:

```
/home/joe/projects/water/water.log
/home/joe/projects/water/water.ScfOrb
...
/tmp/water/water.RunFile
...
```

For More options to control the behavior of Molcas , run the command molcas setuprc script. The file ***molcasrc*** can be used to set global preferences for the Molcas package and/or to set user preferences. The ***setuprc*** script creates a ***molcasrc*** file (***HOME/.Molcas***) in a users home directory.

The following ***molcasrc*** file for uses the ***/scratch*** area as a parent for WorkDirs and Project name generated for the the name of the input file, then removes WorkDir before a calculation followed by subsequent retains of this file when the calculation finished:

```
# Version 1.0
MOLCAS_MEM=256
MOLCAS_WORKDIR=/scratch
MOLCAS_NEW_WORKDIR=YES
MOLCAS_KEEP_WORKDIR=YES
MOLCAS_PROJECT=NAME
```

Once the ***molcasrc*** is created, it is usually not necessary to use shell script or environment variables to run Molcas.

4.4 Molcas Command-Line Help System

Just by typing molcas help you get access to Molcas Command-Line Help System. There are different options:

- molcas help produces a list of available programs and utilities.
- molcas help module yields the list of keywords of the program MODULE.
- molcas help module keyword offers the detailed description of the keyword.
- molcas help -t text displays a list of keywords that contain the text word in their description.

4.5 Input Structure and EMIL Commands

Molcas has a modular program structure. The easiest way to run calculations is to prepare an input file in which the different programs are executed sequentially when the the module name (&module) is provided. The keywords of module name then follow, with each entry on a separate line or several entries on one line, separated by ;. In addition to specific program module keywords, Molcas incorporates certain commands (See section on EMIL Commands.) that allow operations such as looping over the modules, allowing partial execution, changing variables, and substituting certain Unix commands.

4.6 Basic Examples

4.6.1 Simple Calculation on Water

Start by preparing a file containing the cartesian coordinates of a water molecule.

```
 3
Angstrom
 O       0.000000  0.000000  0.000000
 H       0.758602  0.000000  0.504284
 H       0.758602  0.000000 -0.504284
```

which is given the name *water.xyz*. In the same directory we prepare the input for the Molcas run. We can name it *water.input*:

In addition to using an editor to insert atomic coordinates into a file, a coordinate file can be obtained by using a graphical interface program, for example, the LUSCUS module as shown later in this guide.

```
&GATEWAY
 coord=water.xyz
 basis=sto-3g
&SEWARD
&SCF
```

The GATEWAY program module combines the molecular geometric of water (In this case, from the external file, water.xyz) and the basis set definition. The SEWARD program module then computes the integrals, and SCF program modules completer the calculation by computing the Hartree-Fock wave function.

To run the calculation, the following command is used:

```
molcas water.input -f
```

The file ***water.log*** now contains output from the calculation, and the ***water.err*** includes any error messages. In the same directory, other files, including ***water.scf.molden*** or ***water.lus*** (if the keyword GRID_IT is added at end of input file) that help to analyze the results graphically with the external graphical viewer LUSCUS or MOLDEN program. Examples of their use are demonstrated below.

In the case of an open-shell calculation (UHF or UDFT), the SCF program is again used. Below, two examples are shown:

(a) A UDFT calculation yielding an approximate doublet by setting the charge to +1, even if they are not pure spin functions:

```
&GATEWAY
 coord=water.xyz
 basis=sto-3g
&SEWARD
&SCF
 charge=+1
 uhf; ksdft=b3lyp
```

(b) A triplet state (using keyword ZSPIN to specify that there are two more α than β electrons) states

```
&GATEWAY
 coord=water.xyz
 basis=sto-3g
&SEWARD
&SCF
 zspin=2
 uhf; ksdft=b3lyp
```

4.6.2 Geometry Optimization

In the next example, a DFT/B3LYP geometry optimization is performed on the ground state of the water molecule. Notice that, after &gateway has defined the coordinates and basis set definition, the EMIL commands >>> Do while and >>> EndDo are employed to form a loop with the SEWARD, SLAPAF, and SCF programs until convergence of geometry optimization is reached. Program SEWARD computes the integrals in atomic basis, SCF computes the DFT energy, and the program SLAPAF controls the geometry optimization and uses the module ALASKA to compute the gradients of the energy with respect to the degrees of freedom. SLAPAF generates the new geometry to continue the iterative structure optimization process and checks to determine convergence parameters are satisfied notifying MOLCAS and stopping the loop.

```
&GATEWAY
 coord=water.xyz
```

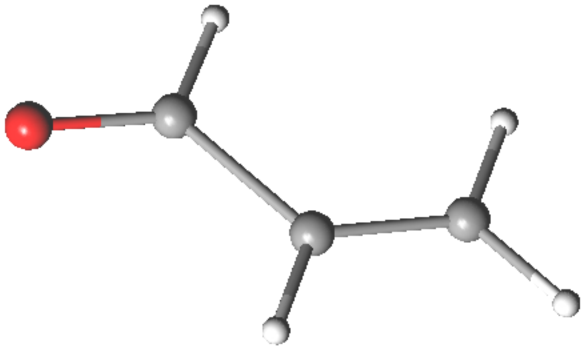


Figure 4.1: *The acrolein molecule.*

```
  basis=ANO-S-MB
 >> Do While
   &SEWARD
   &SCF
     ksdft=b3lyp
   &SLAPAF
 >>> EndDo
```

The above example illustrates the default situation of optimizing to a minimum geometry without any further constraint. If other options are required such as determining a transition state, obtaining a states crossing, or imposing a geometry constraint, specific input should be added to program SLAPAF.

One of the most powerful aspects of MOLCAS is the possibility of computing excited states with multiconfigurational approaches. The next example demonstrates a calculation of the five lowest singlet roots in a State-Average (SA) CASSCF calculation using the RASSCF program. It also illustrates the addition of the CASPT2 program to determine dynamical correlation which provides accurate electronic energies at the CASPT2 level. The resulting wave functions are used in the RASSI module to calculate state-interaction properties such as oscillator strengths and other properties.

```
 &gateway
 Coord
  8
 Acrolein coordinates in Angstrom
  O     -1.808864   -0.137998    0.000000
  C      1.769114    0.136549    0.000000
  C      0.588145   -0.434423    0.000000
  C     -0.695203    0.361447    0.000000
  H     -0.548852    1.455362    0.000000
  H      0.477859   -1.512556    0.000000
  H      2.688665   -0.434186    0.000000
  H      1.880903    1.213924    0.000000
 Basis=ANO-S-MB
 Group=Nosym
 &SEWARD
 &RASSCF
   nactel  = 6 0 0
   inactive= 12
   ras2    = 5
   ciroot  = 5 5 1
 &CASPT2
   multistate=5 1 2 3 4 5
 &RASSI
   Nr of Job=1 5; 1 2 3 4 5
   EJob
```

Notice that the GROUP with the option NOSYM has been used to prevent GATEWAY from identifying the symmetry of the molecule (C_s in this case). Otherwise, the input of the RASSCF program will have to change to incorporate the classification of the active space into the corresponding symmetry species. Working with symmetry will be skipped at this stage, although its use is very convenient in many cases. A good strategy is to run only GATEWAY and let the program guide you.

The RASSCF input describes the active space employed, composed by six active electrons distributed in five active orbitals. By indicating twelve inactive orbitals (always doubly occupied), information about the total number of electrons and the distribution of the orbitals is then complete. Five roots will be obtained in the SA-CASSCF procedurei, and all them will be computed at the CASPT2 level to obtain the transition energies at the higher level of theory. Further, the RASSI will compute the transition properties, in particular, transition dipole moments and oscillator strengths.

4.6.3 GASSCF method

In certain cases it is useful/necessary to enforce restrictions on electronic excitations within the active space beyond the ones accessible by RASSCF. These restrictions are meant to remove configurations that contribute only marginally to the total wave function. In MOLCAS this is obtained by the GASSCF approach [10]. In GASSCF an arbitrary number of active spaces may be chosen. All intra-space excitations are allowed (Full-CI in subspaces). Constraints are imposed by user choice on inter-space excitations. This method, like RASSCF, allows restrictions on the active space, but they are more flexible than in RASSCF. These restrictions are particularly useful when the cost of using the full CI expansion of the active space is beyond reach. These restrictions allow GASSCF to be applied to larger and more complex systems at affordable cost. Instead of a maximum number of holes in RAS1 and particles in RAS3, accumulated minimum and maximum numbers of electrons are specified for GAS1, GAS1+GAS2, GAS1+GAS2+GAS3, etc. in order to define the desired CI expansion. The GAS scheme reduces to CAS or RAS when one or three spaces are chosen and restrictions on electron excitations are adequately imposed. When and how to use the GAS approach? We consider three examples: (1) an organometallic material with separated metal centers and orbitals not delocalized across the metal centers. One can include the near degenerate orbitals of each center in its own GAS space. This implies that one may choose as many GAS spaces as the number of multiconfigurational centers. (2) Lanthanide or actinide metal compounds where the f-electrons require a MC treatment but they do not participate in bonding neither mix with d orbitals. In this case one can put the f orbitals and their electrons into one or more separated GAS spaces and not allow excitations from and/or to other GAS spaces. (3) Molecules where each bond and its correlating anti-bonding orbital could form a separate GAS space as in GVB approach. Finally, if a wave function with a fixed number of holes in one or more orbitals is desired, without interference of configurations where those orbitals are fully occupied the GAS approach is the method of choice instead of the RAS approach. There is no rigorous scheme to choose a GAS partitioning. The right GAS strategy is system-specific. This makes the method versatile but at the same time it is not a black box method. An input example follows:

```
&RASSCF
nActEl
 6 0 0
FROZen
0 0 0 0 0 0 0 0
INACTIVE
2 0 0 0 2 0 0 0
GASScf
3
 1 0 0 0 1 0 0 0
2 2
 0 1 0 0 0 1 0 0
4 4
 0 0 1 0 0 0 1 0
6 6
DELEted
0 0 0 0 0 0 0 0
```

In this example the entire active space counts six active electrons and six active orbitals. These latter are partitioned in three GAS spaces according to symmetry consideration and in the spirit of the GVB strategy. Each subspace has a fixed number of electrons, TWO, and no interspace excitations are allowed. This input shows clearly the difference with the RAS approach.

4.6.4 Solvation Effects

MOLCAS incorporates the effects of the solvent using several models. The most common is the cavity-based reaction-field Polarizable Continuum Model (PCM) which is invoked by adding the keyword RF-INPUT to the SEWARD code and is needed to compute the proper integrals.

```
&GATEWAY
  coord=CH4.xyz
  Basis=ANO-S-MB
&SEWARD
  RF-Input
   PCM-Model
   Solvent=Water
  End of RF-Input
&RASSCF
  Nactel=8 0 0
  Inactive=1
  Ras2=8
&CASPT2
  rfpert
```

The reaction field is computed in a self-consistent manner by the SCF or RASSCF codes and added as a perturbation to the Hamiltonian in the CASPT2 method with the keyword RFPERT.

4.7 Analyzing Results: Output Files and the LUSCUS Program

MOLCAS provides a great deal of printed information in output files, and the printing level is controlled by the environmental variable MOLCAS_PRINT. By default this value is set to TWO, but can be modified by environmental variable MOLCAS_PRINT Typical MOLCAS output contains the program header and input information, conditions of the calculation, the number of steps to achieve convergence, the energies and wave functions, and final results, including in many cases the molecular orbital coefficients as well as an analysis of the properties for the computed states.

4.7.1 LUSCUS: Grid and Geometry Visualization

Molcas developers have developed a graphical interface that can be used both to create input for the Molcas program and to analyze the results in a graphical manner by visualizing molecular orbitals, density plots, and other output properties.

The first version of the code has the name GV (stands for Grid Viewer, or Geometry Visualization. By an accident, the name also matches the nicknames of the main developers). GV program uses a very limited set of graphic libraries, and thus has very primitive user interface.

The next generation of GV program has the name LUSCUS. Luscus re-uses the code of GV, and so GV users can use the same key combinations to operate with LUSCUS. At the same time, LUSCUS provides a user-friendly interface, and contains many new options, compared to GV.

LUSCUS can be obtained from https://sourceforge.net/projects/luscus/.

LUSCUS can read the files only in one format: Luscus internal format (*.lus*). This format contains two sections: XYZ cartesian coordinates, and XML formated data. It means that a standard XYZ file is a valid file in LUSCUS format.

Files with different formats, e.g. molden files, can be understood by LUSCUS since they can be converted to LUSCUS format by a corresponding plug-in. For instance, opening a file with the extension *.molden*, LUSCUS automatically runs a plug-in to convert a file from molden format to LUSCUS format. Saving a LUSCUS file as a Molcas orbital file will automatically run a converter from LUSCUS format to Orbital format.

- luscus xyz_file: reads coordinates from a cartesian coordinate file.

  A molecule can be visualized and modified with the use of the left-button of the mouse and the keyboard. Below are some of the most useful commands.

Left mouse click:	Select atoms (if two, a bond is selected, if three a bond angle, if four a dihedral angle
Left mouse + Shift click:	Mark/unmark atoms to/from the group
Middle mouse/Space:	Remove selection, or marking
Insert key:	Insert atom
PageUp, PageDown:	Alter type of selected atom or bond
Delete/Supress key:	Delete a selected atom
+/-:	Change a value of selected bond/angle in steps
Backspace:	Undo last action
Home:	Set selected atom to center of coordinates
F8 key:	Find or apply symmetry

- luscus molden_file: reads (check the comment about plug-in) from MOLDEN files such as *wavefunction.molden*, *freq.molden*, and *geo.molden*.

  Note that MOLCAS produces molden files with several extensions, so it is recommended to visualize these files by using LUSCUS.

- luscus grid_file: reads coordinates and densities and molecular orbitals from a binary *grid_file*.

  This file is generated by GRID_IT and, by default, placed in the *$WorkDir* directory with the name *$Project.lus*. The program allows displaying total densities, molecular orbitals, and charge density differences.

  If MOLCAS and Luscus are installed locally, LUSCUS can also be called from user input as shown in the following example:

```
&GATEWAY
     coord = acrolein.xyz
     basis = ANO-L-MB
&SEWARD
&SCF
&GRID_IT
ALL
* running external GUI program luscus
! luscus $Project.lus
* User has to select active space and save GvOrb file!
&RASSCF
Fileorb=$CurrDir/$Project.GvOrb
```

  Note, that in the example above, the GRID_IT program will generate a *$Project.lus* file which LUSCUS then uses, eliminating the need for defining *$Project.lus* and allowing this file to be overwritten. RASSCF will read starting orbitals from the *$Project.GvOrb* file.

Section 5

Problem Based Tutorials

5.1 Electronic Energy at Fixed Nuclear Geometry

The Molcas 8.2 suite of Quantum Chemical programs is modular in design, and a desired calculation is achieved by executing a list of Molcas program modules in succession, occasionally manipulating the program information files. If the information files from a previous calculation are saved, then a subsequent calculation need not recompute them. This is dependent on the correct information being preserved in the information files for the subsequent calculations. Each module has keywords to specify the functions to be carried out, and many modules rely on the specification of keywords in previous modules.

In the present examples the calculations will be designed by preparing a single file in which the input for the different programs is presented sequentially. The initial problem will be to compute an electronic energy at a fixed geometry of the nuclei, and this will be performed using different methods and thus requiring different Molcas program modules.

First, the proper Molcas environment has to be set up which requires that following variables must be properly defined, for instance:

```
export MOLCAS=/home/molcas/molcas.8.2
export Project=CH4
export WorkDir=/home/user/tmp
```

If not defined, Molcas provides default values for the above environment variables:

- The MOLCAS variable will be set to the latest implemented version of the code.

  This variable is set directly in the Molcas home directory
- Project and WorkDir have the default values None and $PWD, respectively.

  It is very important that the molcas driver, called by command molcas, and built during the installation of the code, is included in the $PATH.

The first run involves a calculation of the SCF energy of the methane (CH_4) molecule. Three programs should be used: `GATEWAY` to specify information about the system, `SEWARD` to compute and store the one- and two-electron integrals, and `SCF` to obtain the Hartree-Fock SCF wave function and energy.

The three MOLCAS programs to be used leads to three major entries in the input file: `GATEWAY`, `SEWARD`, and `SCF`. The `GATEWAY` program contains the nuclear geometry in cartesian coordinates and the label for the one-electron basis set. The keyword COORD allows automatic insertion of `GATEWAY` input from a standard file containing the cartesian coordinates in Angstrom which can be generated by programs like `LUSCUS` or `MOLDEN`). No symmetry is being considered so the keyword GROUP=C1 is used to force the program not to look for symmetry in the CH_4 molecule, and ,thus, input for `SEWARD` is not required. In closed-shell cases, like CH_4, input for `SCF` is not required. All the input files discussed here can be found at *$MOLCAS/doc/samples/problem_based_tutorials*, including the file *SCF.energy.CH4* described below.

```
*SCF energy for CH4 at a fixed nuclear geometry.
*File: SCF.energy.CH4
*
&GATEWAY
 Title = CH4 molecule
 coord  = CH4.xyz
 basis = STO-3G
 group = C1
&SEWARD
&SCF
```

where the content of the *CH4.xyz* file is:

```
5
distorted CH4 coordinates in Angstroms
C    0.000000     0.000000     0.000000
H    0.000000     0.000000     1.050000
H    1.037090     0.000000    -0.366667
H   -0.542115    -0.938971    -0.383333
H   -0.565685     0.979796    -0.400000
```

To run MOLCAS , simply execute the command

```
molcas SCF.energy.CH4.input > SCF.energy.CH4.log 2 > SCF.energy.CH4.err
```

where the main output is stored in file *SCF.energy.CH4.log*

or

```
molcas -f SCF.energy.CH4.input
```

where the main output is stored in *SCF.energy.CH4.log*, and the default error file in *SCF.energy.CH4.err*.

The most relevant information is contained in the output file, where the `GATEWAY` program information describing the nuclear geometry, molecular symmetry, and the data regarding the one-electron basis sets and the calculation of one- and two-electron integrals, as described in section 6.4. Next, comes the output of program `SCF` with information of the electronic energy, wave function, and the Hartree-Fock (HF) molecular orbitals (see section 6.5).

Files containing intermediate information, integrals, orbitals, etc, will be kept in the $WorkDir directory for further use. For instance, files *$Project.OneInt* and *$Project.OrdInt* contain the one- and two-electron integrals stored in binary format. File *$Project.ScfOrb* stores the HF molecular orbitals in ASCII format, and *$Project.RunFile* is a communication file between programs. All these files can be used later for more advanced calculations avoiding a repeat of certain calculations.

There are graphical utilities that can be used for the analysis of the results. By default, Molcas generates files which can be read with the MOLDEN program and are found in the $WorkDir including the fileCH4.scf.molden. This file contains information about molecular geometry and molecular orbitals, and requires the use if *Density Mode* in MOLDEN. However, Molcas has its own graphical tool, program LUSCUS, which is a viewer based on openGL and allows the visualization of molecular geometries, orbitals, densities, and density differences. For example, a graphical display of the CH_4 molecule can be obtained from a standard coordinate file by the following command:

```
luscus CH4.xyz
```

In order to obtain the information for displaying molecular orbitals and densities, it is necessary to run the Molcas program called GRID_IT:

```
*SCF energy for CH4 at a fixed nuclear geometry plus a grid for visualization.
*File: SCF.energy_grid.CH4
*
&GATEWAY
 Title = CH4 molecule
 coord = CH4.xyz
 basis = STO-3G
 Group = C1
&SEWARD; &SCF
&GRID_IT
 All
```

Now, execcute the Molcas program:

```
molcas SCF.energy_grid.CH4.input -f
```

In the $WorkDir and $PWD directories a new file is generated, *CH4.lus* which contains the information required by the GRID_IT input. The file can be visualized by LUSCUS (Open source program, which can be downloaded and installed to your Linux, Windows, or MacOS workstation or laptop). By typing the command:

```
luscus CH4.lus
```

a window will be opened displaying the molecule and its charge density. By proper selection of options with the mouse buttons, the shape and size of several molecular orbitals can be visualized.

GRID_IT can also be run separately, if an orbital file is specified in the input, and the $WorkDir directory is available.

More information can be found in section 8.19.

As an alternative to running a specific project, the short script provided below can be placed in the directory $MOLCAS/doc/samples/problem_based_tutorials with the name *project.sh*. Simply execute the shell script, project.sh $Project, where $Project is the MOLCAS input, and output files, error files, and a $WorkDir directory called $Project.work will be obtained.

```
#!/bin/bash
export MOLCAS_MEM=1Gb
export MOLCAS_PRINT=3
molcas $1 -f
exit
```

In order to run a Kohn-Sham density functional calculation, Molcas uses the same SCF module, and, therefore, the only change needed are the specification of the DFT option and required functional (e.g. B3LYP) in the SCF input:

```
*DFT energy for CH4 at a fixed nuclear geometry plus a grid for visualization.
*File: DFT.energy.CH4
*
&GATEWAY
 Title = CH4 molecule
 coord = CH4.xyz
 basis = STO-3G
 group = C1
&SEWARD
&SCF
 KSDFT = B3LYP
&GRID_IT
 All
```

Similar graphical files can be found in $WorkDir and $PWD.

The next step is to obtain the second-order Møller–Plesset perturbation (MP2) energy for methane at the same molecular geometry using the same one-electron basis set. Program MBPT2 is now used, and it is possible to take advantage of having previously computed the proper integrals with SEWARD and the reference closed-shell HF wave function with the SCF program. In such cases, it is possible to keep the same definitions as before and simply prepare a file containing the MBPT2 input and run it using the molcas command.

The proper intermediate file will be already in $WorkDir. On the other hand, one has to start from scratch, all required inputs should be placed sequentially in the *MP2.energy.CH4* file. If the decision is to start the project from the beginning, it is probably a good idea to remove the entire $WorkDir directory, unless it is known for certain the exact nature of the files contained in this directory.

```
*MP2 energy for CH4 at a fixed nuclear geometry.
*File: MP2.energy.CH4
*
&GATEWAY
 Title = CH4 molecule
 coord = CH4.xyz
 basis = STO-3G
 group = C1
&SEWARD
&SCF
&MBPT2
 Frozen = 1
```

In addition to the calculation of a HF wave function, an MP2 calculation has been performed with a frozen deepest orbital, the carbon 1s, of CH_4. Information about the output of the MBPT2 program can be found on section 6.6.

The SCF program works by default with closed-shell systems with an even number of electrons at the Restricted Hartee-Fock (RHF) level. If, instead there is a need to use the Unrestricted

Hartree Fock (UHF) method, this can be schieved by invoking the keyword UHF. This is possible for both even and odd electron systems. For instance, in a system with an odd number of electrons such as the CH_3 radical, with the following Cartesian coordinates

```
4
CH3 coordinates in Angstrom
C    0.000000     0.000000     0.000000
H    0.000000     0.000000     1.050000
H    1.037090     0.000000    -0.366667
H   -0.542115    -0.938971    -0.383333
```

the input to run an open-shell UHF calculation is easily obtained:

```
*SCF/UHF energy for CH3 at a fixed nuclear geometry
*File: SCF.energy_UHF.CH3
*
&GATEWAY
 Title = CH3 molecule
 coord = CH3.xyz
 basis = STO-3G
 group = C1
&SEWARD
&SCF
 UHF
```

If the system is charged, this must be indicated in the SCF input, for example, by computing the cation of the CH_4 molecule at the UHF level:

```
*SCF/UHF energy for CH4+ at a fixed nuclear geometry
*File: SCF.energy_UHF.CH4plus
*
&GATEWAY
 Title = CH4+ molecule
 coord = CH4.xyz
 basis = STO-3G
 group = c1
&SEWARD
&SCF
 UHF
 Charge = +1
```

The Kohn-Sham DFT calculation can be also run using the UHF algorithm:

```
*DFT/UHF energy for CH4+ at a fixed nuclear geometry
*File: DFT.energy.CH4plus
*
&GATEWAY
 Title = CH4+ molecule
 coord = CH4.xyz
 basis = STO-3G
 group = C1
&SEWARD
&SCF
 KSDFT = B3LYP
 UHF
 Charge = +1
```

For the UHF and UHF/DFT methods it is also possible to specify α and β orbital occupations in two ways.

1. First, the keyword ZSPIn can be invoked in the SCF program, which represents the difference between the number of α and β electrons.

   For example, setting the keyword to 2 forces the program to converge to a result with two more α than β electrons.

   ```
   *DFT/UHF energy for different electronic occupation in CH4 at a fixed nuclear geometry
   *File: DFT.energy_zspin.CH4
   *
   &GATEWAY
    Title = CH4 molecule
    coord = CH4.xyz
    basis = STO-3G
    group = c1
   &SEWARD
   &SCF
    Title = CH4 molecule zspin 2
    UHF; ZSPIN =  2
    KSDFT =  B3LYP
   ```

   The final occupations in the output will show six α and four β orbitals.

2. Alternatively, instead of ZSPIn, it is possible to specify occupation numbers with keyword Occupation at the beginning of the SCF calculation.

   This requires an additional input line containing the occupied α orbitals (e.g. 6 in this case), and a second line with the β orbitals (e.g. 4 in this case). Sometimes, SCF convergence may be improved by using this option.

Different sets of methods use other Molcas modules. For example, to perform a Complete Active Space (CAS) SCF calculation, the RASSCF program has to be used. This module requires starting trial orbitals, which can be obtained from a previous SCF calculation or, automatically, from the SEWARD program which provides trial orbitals by using a model Fock operator.

Recommended keywords are

- Nactel defines the total number of active electrons, holes in Ras1, and particles in Ras3, respectively. The last two values are only for RASSCF-type calculations.

- Inactive indicates the number of inactive orbitals where the occupation is always 2 in the CASSCF reference, and

- Ras2 defines the number of active orbitals.

  By default, the wave function for the lowest state corresponds to the symmetry with spin multiplicity of 1. Most of the input may not be necessary, if one has prepared and linked an INPORB file with the different orbital types defined by a program like LUSCUS.

```
*CASSCF energy for CH4 at a fixed nuclear geometry
*File: CASSCF.energy.CH4
*
&GATEWAY
 coord = CH4.xyz
 basis = STO-3G
 group = C1
&SEWARD
```

```
&RASSCF
 Title = CH4 molecule
 Spin = 1; Nactel = 8 0 0; Inactive = 1; Ras2 = 8
&GRID_IT
 All
```

In this case, the lowest singlet state (i.e. the ground dstate) is computed, since this is a closed-shell situation with an active space of eight electrons in eight orbitals and with an inactive C 1s orbital, the lowest orbital of the CH_4 molecule. This is a CASSCF example in which all the valence orbitals and electrons (C 2s, C 2p and 4 x H 1s) are included in the active space and allows complete dissociation into atoms. If this is not the goal, then the three almost degenerate highest energy occupied orbitals and the corresponding antibonding unoccupied orbitalsmust be active, leading to a 6 in 6 active space.

Using the CASSCF wave function as a reference, it is possible to perform a second-order perturbative, CASPT2, correction to the electronic energy by employing the `CASPT2` program. If all previously calculated files are retained in the $WorkDir directory, in particular, integral files (*CH4.OneInt*,*CH4.OrdInt*), the CASSCF wave function information file (*CH4.JobIph*), and communication file *CH4.RunFile*), it will not be necessary to re-run programs `SEWARD`, and `RASSCF`. In this case case, it is enough to prepare a file containing input only for the `CASPT2` program followed be execution. Here, however, for the sake of completness, input to all MOLCAS moddules is provided:

```
*CASPT2 energy for CH4 at a fixed nuclear geometry
*File: CASPT2.energy.CH4
*
&GATEWAY
 coord = CH4.xyz; basis = STO-3G; group = C1
&SEWARD
&RASSCF
   LumOrb
   Spin = 1; Nactel = 8 0 0; Inactive = 1; Ras2 = 8
&CASPT2
 Multistate = 1 1
```

In most of casesi, the Hartree-Fock orbitals will be a better choice as starting orbitals. In that case, the `RASSCF` input has to include keyword LUMORB to read from any external source of orbitals other than those generated by the `SEWARD` program. By modifying input to the `SCF` program, it is possible to generate alternative trial orbitals for the `RASSCF` program. Since a new set of trial orbitals is used, the input to the `RASSCF` program is also changed. Now, the number of active orbitals, as well as the number of active electrons, are 6.

The two lowest orbitals (INACTIVE 2) are excluded from the active space and one other orbital is placed in the secondary space. If the previous (8,8) full valence space was used, the `CASPT2` program would not be able to include more electronic correlation energy, considering that the calculation involves a minimal basis set. The input for the `CASPT2` program includes a frozen C 1s orbital, the lowest orbital in the CH_4 molecule.

The charge and multiplicity of our wave function can be changed by computing the CH_4^+ cation with the same methods. The `RASSCF` program defines the character of the problem by specifying the number of electrons, the spin multiplicity, and the spatial symmetry. In the example below, there is one less electron giving rise to doublet multiplicity:

```
*CASSCF energy for CH4+ at a fixed nuclear geometry
*File: CASSCF.energy.CH4plus
```

```
*
&GATEWAY
 Title = CH4+ molecule
 coord = CH4.xyz; basis = STO-3G; Group = C1
&SEWARD
&RASSCF
   Spin = 2; Nactel = 7 0 0; Inactive = 1; Ras2 = 8
```

No further modification is needed to the CASPT2 input:

```
*CASPT2 energy for CH4+ at a fixed nuclear geometry
*File: CASPT2.energy.CH4plus
*
&GATEWAY
 coord = CH4.xyz; basis = STO-3G; group = C1
&SEWARD
&RASSCF
   Title = CH4+ molecule
   Spin = 2; Nactel = 1 0 0; Inactive = 4; Ras2 = 1
&CASPT2
```

A somewhat more sophisticated calculation can be performed at the Restricted Active Space (RAS) SCF level. In such a situation, the level of excitation in the CI expansion can be controlled by restricting the number of holes and particles present in certain orbitals.

```
*RASSCF energy for CH4 at a fixed nuclear geometry
*File: RASSCF.energy.CH4
*
&GATEWAY
 coord = CH4.xyz; basis = STO-3G; group = C1
&SEWARD
&RASSCF
   Title = CH4 molecule
   Spin = 1; Nactel = 8 1 1
   Inactive = 1; Ras1 = 1; Ras2 = 6; Ras3 = 1
```

In particular, the previous calculation includes one orbital within the Ras1 space and one orbital within the Ras3 space. One hole (single excitation) at maximum is allowed from Ras1 to Ras2 or Ras3, while a maximum of one particle is allowed in Ras3, derived from either Ras1 or Ras2. Within Ras2, all types of orbital occupations are allowed. The RASSCF wave functions can be used as reference for multiconfigurational perturbation theory (RASPT2), but this approach has not been as extensively tested as CASPT2, and, so experience is still somewhat limited.

MOLCAS also has the possibility of computing electronic energies at different CI levels by using the MRCI program. The input provided below involves a Singles and Doubles Configuration Interaction (SDCI) calculation on the CH_4 molecule. To set up the calculations, program MOTRA which transforms the integrals to molecular basis, and program GUGA which computes the coupling coefficients, must be run before the MRCI program. In MOTRA the reference orbitals are specifiedi, and those employed here are from an HF SCF calculation including frozen orbitals. In GUGA the reference for the CI calculation is described by the number of correlated electrons, the spatial and spin symmetry, the inactive orbitals always occupation 2 in the reference space, and the type of CI expansion.

```
*SDCI energy for CH4 at a fixed nuclear geometry
*File: SDCI.energy.CH4
*
&GATEWAY
```

```
 coord = CH4.xyz; basis = STO-3G; group = c1
&SEWARD
&SCF
&MOTRA
 Lumorb
 Frozen= 1
&GUGA
 Electrons = 8
 Spin = 1
 Inactive= 4
 Active= 0
 Ciall= 1
&MRCI
 SDCI
```

To use reference orbitals from a previous CASSCF calculation, the RASSCF program will have to be run before the MOTRA module. Also, if the spatial or spin symmetry are changed for the CI calculation, the modifications will be introduced in the input to GUGA program. Many alternatives are possible for performing an MRCI calculation as shown in the next example below, in which the reference space to perform the CI is multiconfigurational:

```
*MRCI energy for CH4 at a fixed nuclear geometry
*File: MRCI.energy.CH4
*
&GATEWAY
 Title = CH4 molecule
 coord = CH4.xyz; basis = STO-3G; group = c1
&SEWARD
&SCF
&RASSCF
 LumOrb
 Spin= 1; Nactel= 6 0 0; Inactive= 2; Ras2= 6
&MOTRA
 Lumorb
 Frozen= 1
&GUGA
 Electrons= 8
 Spin= 1
 Inactive= 2
 Active= 3
 Ciall= 1
&MRCI
 SDCI
```

The MRCI program also allows the calculation of electronic energies using the ACPF method. Another Molcas program, CPF, offers the possibility to use the CPF, MCPF, and ACPF methods with a single reference function. The required input is quite similar to that for the MRCI program:

```
*CPF energy for CH4 at a fixed nuclear geometry
*File: CPF.energy.CH4
*
&GATEWAY
 Title= CH4 molecule
 coord = CH4.xyz; basis = STO-3G; group = c1
&SEWARD
&SCF
&MOTRA
 Lumorb
 Frozen= 1
&GUGA
 Electrons= 8
```

```
  Spin = 1
  Inactive = 4
  Active = 0
  Ciall= 1
 &CPF
  CPF
 End Of Input
```

Finally, MOLCAS can also perform closed- and open-shell coupled cluster calculations at the CCSD and CCSD(T) levels. These calculations are controlled by the CCSDT program, whose main requirement is that the reference function has to be generated with the RASSCF program. The following input is required to obtain a CCSD(T) energy for the CH_4 molecule:

```
 *CCSDT energy for CH4 at a fixed nuclear geometry
 *File: CCSDT.energy.CH4
 *
 &GATEWAY
  Title= CH4 molecule
  coord = CH4.xyz; basis = STO-3G; group = c1
 &SEWARD
 &RASSCF
  Spin= 1; Nactel= 0 0 0; Inactive= 5; Ras2= 0
  OutOrbitals
  Canonical
 &MOTRA
  JobIph
  Frozen= 1
 &CCSDT
  CCT
```

Since this is a closed-shell calculation, the RASSCF input computes a simple RHF wave function with zero active electrons and orbitals using keywords OUTORBITALS and CANONICAL. The MOTRA must include the keyword JOBIPH to extract the wave function information from file *JOBIPH* which is automatically generated by RASSCF. Finally, the keywork CCT in program CCSDT leads to the calculation of the CCSD(T) energy using the default algorithms.

The CCSDT program in MOLCAS is especially suited to compute open-shell cases. The input required to obtain the electronic energy of the CH_4^+ cation with the CCSD(T) method is:

```
 *CCSDT energy for CH4+ at a fixed nuclear geometry
 *File: CCSDT.energy.CH4plus
 *
 &GATEWAY
  Title= CH4+ molecule
  coord = CH4.xyz; basis = STO-3G; group = c1
 &SEWARD
 &RASSCF
  Spin= 2; Nactel= 1 0 0; Inactive= 4; Ras2= 1
  OutOrbitals
  Canonical
 &MOTRA
  JobIph
  Frozen= 1
 &CCSDT
  CCT
```

where the RASSCF program generated the proper Restricted Open-Shell Hartree-Fock (ROHF) reference. Different levels of spin adaptation are also available.

If solvent effects are desired, MOLCAS includes two models: Kirkwood and PCM. Adding a solvent effect to a ground state at HF, DFT, or CASSCF levels, simply requires the inclusion of the keyword RF-input within the input for the SEWARD which calculates a self-consistend reaction field.

```
*DFT energy for CH4 in water at a fixed nuclear geometry
*File: DFT.energy_solvent.CH4
*
&GATEWAY
 Title= CH4 molecule
 coord = CH4.xyz; basis = STO-3G; group = c1
 RF-input
   PCM-model; solvent= water
 End of RF-input
&SEWARD
&SCF
KSDFT= B3LYP
```

Other programs such as CASPT2, RASSI, and MOTRA require that the reaction field is included as a perturbation with keyword RFPerturbation. In the next example the correction is added at both the CASSCF and CASPT2 levels.

```
*CASPT2 energy for CH4 in acetone at a fixed nuclear geometry
*File: CASPT2.energy_solvent.CH4
*
&GATEWAY
 Title= CH4 molecule
 coord = CH4.xyz; basis = STO-3G; group = c1
  RF-input
   PCM-model; solvent= acetone; AAre= 0.2
  End of RF-input
&SEWARD
&RASSCF
  Spin= 1; Nactel= 6 0 0; Inactive= 2; Ras2= 6
&CASPT2
 Frozen= 1
 Multistate= 1 1
 RFPert
```

Notice that the tesserae of the average area in the PCM model (keyword has been changed to the value required for acetone by the keyword Aare, while the default is 0.4 Å^2 for water (see section 8.16.1). More detailed examples can be found in section 10.6.

5.2 Optimizing geometries

It is now useful to explore potential energy surfaces (PES) and optimize the molecular geometry for specific points along the PES. Different cases are discussed including a way to obtain the optimal geometry in a minimum energy search, to obtain a transition-state structure connecting different regions of the PES, to find the crossing between two PES where the energy becomes degenerate, or to map the minimum steepest-descent energy path (MEP) from an initial point to the final a minimum energy geometry as the PES progresses in a downward manner.

All these types of searches can be performed either by fully optimizing all degrees of freedom of the system or by introducing certain restrictions. Molcas 8.2 can perform geometry optimizations at the SCF (RHF and UHF), DFT (RHF and UHF based), CASSCF (CASSCF and RASSCF) levels of theory, where efficient analytical gradients are available and at the CASPT2 and other correlated levels where numerical gradients are used.

Geometry optimizations require many cycles, in which the electronic energy is estimated at a specific level of calculation followed by calculation of the gradient of the energy with respect to the geometric degrees of freedom (DOF). With this information at hand, the program must decide if the molecule is already at the final required geometry (i.e. gradient ∼ 0 for all DOF) indicating a minimum in the PES or if the geometry must be modified and continue the cycle. The input file should, therefore, be built in a way that allows a loop over the different programs.

The general input commands Do while and Enddo control the loop and program input is inserted within these commands. Instructions for the number of maximum iterations allowed and the type of output required can also be added. (see section 7.2.5)

All examples previously discussed, use COORD keyword, but it also possible to use *native format*, where symmetry unique atoms are specified (SYMMETRY) and provide generators to construct all atoms in the molecule.

The selected example describes geometry optimization of the water molecule at the SCF RHF level of calculation:

```
3
 coordinates for water molecule NOT in equilibrium
O 0.000000  0.000000  0.000000
H 0.758602  0.000000  0.504284
H 0.758602  0.000000 -0.504284
```

```
*SCF minimum energy optimization for H2O
*File: SCF.minimum_optimization.H2O
*
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
>>> Do while
 &SEWARD ;&SCF; &SLAPAF
>>> EndDo
```

The sequence of programs employed includes `GATEWAY` which is external to the loop, followed by `SEWARD`, `SCF`, and `SLAPAF`. `SEWARD` computes the integrals, `SCF` program computes the RHF energy and wave function, and `SLAPAF` will control the calculation of gradients and estimate if the calculation has already finished or needs to proceed to a new nuclear geometry for the next iteration. Automatically, a file named *$Project.geo.molden* will be generated in $WorkDir containing all the geometric steps contained in the optimization process. `MOLDEN` or `LUSCUS` can then read this file to display the individual molecular geometries which form the optimization cycle.

Using another reference wave function can be simply performed by changing the sequence of programs. For instance, we can perform an UHF calculation of the H_2O^+ cation:

```
*UHF minimum energy optimization for H2O+
*File: UHF.minimum_optimization.H2Oplus
*
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
```

```
>> Do while
 &SEWARD
 &SCF; Title="H2O minimum optimization"; UHF; Charge=1
 &SLAPAF
>> EndDo
```

The same procedure can be followed if we pretend to perform a DFT geometry optimization:

```
*DFT minimum energy optimization for H2O
*File: DFT.minimum_optimization.H2O
*
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
>>> Export MOLCAS_MAXITER=100
>>> Do while
 &SEWARD
 &SCF ; Title="H2O minimum optimization"; KSDFT=B3LYP
 &SLAPAF &END
>>> EndDo
```

Once an energy minimum is found based on the calculation of gradients, it is necessary to ensure that the geometry really is a minimum energy point. This can be only accomplished by computing second derivatives of the energy (i.e. the Hessian). MOLCAS can compute analytical Hessians for SCF and single state CASSCF wave functions. For other methods, numerical procedures can be used to compute the Hessian. Once the Hessian is computed, vibrational frequencies are calculated, and Statistical Mechanics is used to obtain thermodynamic properties. At a true energy minimum, there will be 3N-6 real frequencies Program MCKINLEY computes second derivatives of a predefined (SCF or CASSCF) wave function, while MCLR performs the vibrational and statistical analyses. MOLCAS simply requires input for the MCKINLEY program to perform the entire calculation by using keywords PERTURBATION and HESSIAN, while program MCLR will be called automatically but requires no input. The full set of calculationsi is included below first a geometry optimization followed by the calculation of a Hessian.

```
*SCF minimum energy optimization plus hessian of the water molecule
*File: SCF.minimization_plus_hessian.H2O
*
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
>>> Export MOLCAS_MAXITER=100
>>> Do while
 &SEWARD
 &SCF; Title="H2O minimum optimization"
 &SLAPAF
>>> EndDo
&MCKINLEY
```

Note that MCKINLEY input above is placed after EndDo, and, therefore, is external to the looping scheme. Once the geometry optimization at the desired level of theory has finished, the Hessian will be computed at the final geometry. In general, any calculation performed using a $WorkDir directory where a previous geometry optimization has taken place will use the last geomtry calculated from that optimization as the input geometry even if SEWARD input is present. To avoid that, the only solution is to remove the communication file *RUNFILE* where the geometry is stored. Note also, that the frequencies are computed in a cartesian basis, and that three translational and three rotational frequencies which should be very close to zero are included in the output file. This is not the case when numerical gradients and Hessians are used. In particular, for water at its minimum energy structure three (3N-6) real vibrational frequencies. By default, in $WorkDir a file *$Project.freq.molden* is generated containing the vibrational frequencies and modes, which can be visualized by MOLDEN.

A new level of theory, CASSCF, is introduced here which is especially suited for geometry optimizations of excited states discussed in the next chapter. A geometry optimization is performed to illustrate a broader range of possibilities including the imposition of a geometric restrain that the HOH angle in water should be constrained to 120^o during the optimization. This means that only the O-H bond distances be optimized in this partial minimization. The restriction is indicated in in GATEWAY by invoking the keyword Constraints and ending with the keyword End of Constraints. The names of variables corresponding to geometrical variables in either internal or Cartesian coordinates that are to be constrained are placed between these two keywords. (see nomenclature in section 8.46.4) In the case of H_2O, the H1-O-H2 angle is fixed at 120^o, so a variable, a, is first defined with the keywork (Angle), which relates it to the H1-O1-H2 angle, followed by the second keyword, Value, where the variable a is specified as 120^o. It is not required that the initial geometry is 120^o, only that the final result for the calculation will become 120^o.

Note that the RASSCF program requires initial trial orbitals, and those which are automatically generated by SEWARD are used. The resulting CASSCF wave function includes all valence orbitals and electrons.

```
*CASSCF minimum energy optimization of the water molecule with geometrical restrictions
*File: CASSCF.minimum_optimization_restricted.H2O
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
Constraint
   a = Angle H2 O1 H3
  Value
   a = 90. degree
End of Constraints
>>> Do while
 &SEWARD
 &RASSCF; nActEl=8 0 0; Inactive=1; Ras2=6
 &SLAPAF
>>> EndDo
```

Other more flexible ways to impose geometric restrictions involve the specification of which internal coordinates should remain fixed and which should change. In the next example, the bond lengths are forced to remain fixed at their initial distance (here 0.91 Å), while the bond angle, having an initial of 81°, is optimized.

```
*DFT minimum energy optimization of the angle in the water molecule at fixed bond lengths
*File: DFT.minimum_optimization_restricted.H2O
*
&GATEWAY
 Title= H2O minimum optimization
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
```

```
>>> EXPORT MOLCAS_MAXITER=100
>>> Do while
 &SEWARD; &SCF; Title="H2O restricted minimum"; KSDFT=B3LYP
 &SLAPAF
  Internal Coordinates
     b1 = Bond O1 H2
     b2 = Bond O1 H3
     a1 = Angle H2 O1 H3
  Vary
     a1
  Fix
     b1
     b2
  End of Internal
>>> EndDo
```

In the final output, the two O-H bond lengths remain at the initia values, while the H1-O1=H2 angle is optimized to a final angle of 112^o.

The next step entails the computation of a transition state, a structure connecting different regions of the potential energy hypersurface, and is a maximum for only one degree of freedom. The most common saddle points have order one, that is, they are maxima for one of one displacement and minima for the others. The simplest way to search for a transition state in MOLCAS is to add the keyword TS to the SLAPAF input. Keyword PRFC is suggested in order to verify the nature of the transition structure. Searching for transition states is, however, not an easy task. An illustration of the input required for transition state optimization for water at the DFT level is given below:

```
3
water in Transition state in bohr
O1            0.750000        0.000000        0.000000
H2            1.350000        0.000000        1.550000
H3            1.350000        0.000000       -1.550000


*DFT transition state optimization of the water molecule
*File: DFT.transition_state.H2O
*
&Gateway
 Coord=Water_TS.xyz
 Basis=ANO-S-VDZ
 Group=C1
>>> Do while
 &SEWARD
 &SCF; Title="H2O TS optimization"; KSDFT=B3LYP
 &SLAPAF ; ITER=20 ; TS
>>> EndDo
```

The initial coordinates were chosen in units of Bohr, to illustrare that this is the default case. The optimal geometry for ground state of water is a structure with C_{2v} symmetry. A transition state has been found with a linear HOH angle of 180^o. In many cases, there may be a clue along the energy pathway for a chemical reaction about the nature of the transition state structure, which typically represents an intermediate conformation between reactants and products. If this turns out to be the case, it is possible to help the optimization process proceed toward an informed guess, by invoking the keyword FINDTS in SLAPAF. FINDTS must to be accompanied with a definition of constrained geometric definitions. SLAPAF will guide the optimization of the transition state towards a region in which the restriction is fulfilled. Once there, the restriction will be released and a free search of the transition state will be performed. This technique is frequently quite effective and makes it possible to find difficult transition states or reduce the number of required iterations. Here, an example is provided, in which the initial geometry of water is clearly bent, and a trial restraint is imposed such that the angle for the transition state should be near 180^o. The final transition state will, however, be obtained without any type of geometrical restriction.

```
*DFT transition state optimization of the water molecule with geometrical restrictions
*File: DFT.transition_state_restricted.H2O
*
&Gateway
 Coord=Water_TS.xyz
 Basis=ANO-S-VDZ
 Group=C1
 Constraints
   a = Angle H2 O1 H3
 Value
   a = 180.0 degree
 End of Constraints
>>> Do while
 &SEWARD
 &SCF; Title="H2O TS optimization"; KSDFT=B3LYP
 &SLAPAF ;FindTS
>>> EndDo
```

The CASPT2 geometry optimizations are somewhat different because ALASKA is not suited to compute CASPT2 analytical gradients. Therefore the ALASKA program is automatically substituted by program NUMERICAL_GRADIENT, which will take care of performing numerical gradients. From the user point of view the only requirement is to place the CASPT2 input after the RASSCF input. The CASSCF wave function has of course to be generated in each step before performing CASPT2. To compute a numerical gradient can be quite time consuming, although it is a task that can be nicely parallelized. In a double-sided gradient algorithm like here a total of 6N-12+1 CASPT2 calculations are performed each pass of the optimization, where N is the number of atoms.

```
*CASPT2 minimum energy optimization for water
*File: CASPT2.minimum_optimization.H2O
*
&GATEWAY
 coord=Water_distorted.xyz
 basis=ANO-S-MB
 group=C1
>>> Do while
 &SEWARD
 &RASSCF; Title="H2O restricted minimum"; nActEl=8 0 0; Inactive=1; Ras2=6
 &CASPT2; Frozen=1
 &SLAPAF
>>> EndDo
```

The use of spatial symmetry makes the calculations more efficient, although they may again complicate the preparation of input files. We can repeat the previous CASPT2 optimization by restricting the molecule to work in the C_{2v} point group, which, by the way, is the proper symmetry for water in the ground state. The GATEWAY program (as no symmetry has been specified) will identify and work with the highest available point group, C_{2v}. Here the molecule is placed with YZ as the molecular plane. By adding keyword SYMMETRY containing as elements of symmetry the YZ (symbol X) and YX (symbol Z), the point group is totally defined and the molecule properly generated. From that point the calculations will be restricted to use symmetry restrictions. For instance, the molecular orbitals will be classified in the four elements of symmetry of the group, a_1, b_1, b_2, and a_2, and most of the programs will require to define the selection of the orbitals in the proper order. The order of the symmetry labels is determined by SEWARD and must be checked before proceeding, because from that point the elements of symmetry will be known by their order in SEWARD: a_1, b_1, b_2, and a_2, for instance, will be symmetries 1, 2, 3, and 4, respectively. SCF does not require to specify the class of orbitals and it can be used as a learning tool.

```
*CASPT2 minimum energy optimization for water in C2v
*File: CASPT2.minimum_optimization_C2v.H2O
*
 &GATEWAY
Title= H2O caspt2 minimum optimization
Symmetry= X Z
Basis set
O.ANO-S...2s1p.
O        0.000000  0.000000  0.000000 Angstrom
End of basis
Basis set
H.ANO-S...1s.
H1       0.000000  0.758602  0.504284 Angstrom
End of basis
>>> EXPORT MOLCAS_MAXITER=100
>>> Do while
 &SEWARD
 &RASSCF; nActEl=8 0 0; Inactive=1 0 0 0; Ras2=3 1 2 0
 &CASPT2; Frozen=1 0 0 0
 &SLAPAF &END
>>> EndDo
```

Thanks to symmetry restrictions the number of iterations within NUMERICAL_GRADIENT has been reduced to five instead of seven, because many of the deformations are redundant within the C_{2v} symmetry. Also, symmetry considerations are important when defining geometrical restrictions (see sections 8.46.4 and 10.2).

5.3 Computing excited states

The calculation of electronic excited states is typically a multiconfigurational problem, and therefore it should preferably be treated with multiconfigurational methods such as CASSCF and CASPT2. We can start this section by computing the low-lying electronic states of the acrolein molecule at the CASSCF level and using a minimal basis set. The standard file with cartesian coordinates is:

```
 8
Angstrom
 O      -1.808864   -0.137998    0.000000
 C       1.769114    0.136549    0.000000
 C       0.588145   -0.434423    0.000000
 C      -0.695203    0.361447    0.000000
 H      -0.548852    1.455362    0.000000
 H       0.477859   -1.512556    0.000000
 H       2.688665   -0.434186    0.000000
 H       1.880903    1.213924    0.000000
```

We shall carry out State-Averaged (SA) CASSCF calculations, in which one single set of molecular orbitals is used to compute all the states of a given spatial and spin symmetry. The obtained density matrix is the average for all states included, although each state will have its own set of optimized CI coefficients. Different weights can be considered for each of the states, but this should not be used except in very special cases by experts. It is better to let the CASPT2 method to handle that. The use of a SA-CASSCF procedure has an great advantage. For example, all states in a SA-CASSCF calculation are orthogonal to each other, which is not necessarily true for state specific calculations. Here, we shall include five states of singlet character the calculation. As no symmetry is invoked all the states belong by default to the first symmetry, including the ground state.

```
*CASSCF SA calculation on five singlet excited states in acrolein
*File: CASSCF.excited.acrolein
*
&GATEWAY
  Title= Acrolein molecule
  coord = acrolein.xyz; basis = STO-3G; group = c1
&SEWARD; &SCF
&RASSCF
  LumOrb
  Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
  CiRoot= 5 5 1
&GRID_IT
  All
```

We have used as active all the π and π^* orbitals, two bonding and two antibonding π orbitals with four electrons and in addition the oxygen lone pair (n). Keyword CiRoot informs the program that we want to compute a total of five states, the ground state and the lowest four excited states at the CASSCF level and that all of them should have the same weight in the average procedure. Once analyzed we find that the calculation has provided, in this order, the ground state, two $n \rightarrow \pi^*$ states, and two $\pi \rightarrow \pi^*$ states. It is convenient to add the GRID_IT input in order to be able to use the LUSCUS interface for the analysis of the orbitals and the occupations in the different electronic states. Such an analysis should always be made in order to understand the nature of the different excited states. In order to get a more detailed analysis of the nature of the obtained states it is also possible to obtain in a graphical way the charge density differences between to states, typically the difference between the ground and an excited state. The following example creates five different density files:

```
*CASSCF SA calculation on five singlet excited states in acrolein
*File: CASSCF.excited_grid.acrolein
*
&GATEWAY
  Title= Acrolein molecule
  coord= acrolein.xyz; basis= STO-3G; group= c1
&SEWARD; &SCF
&RASSCF
 LumOrb
 Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
 CiRoot= 5 5 1
 OutOrbital
 Natural= 5
&GRID_IT
 FILEORB = $Project.RasOrb.1
 NAME = 1; All
&GRID_IT
 FILEORB = $Project.RasOrb.2
 NAME = 2; All
&GRID_IT
 FILEORB = $Project.RasOrb.3
 NAME = 3; All
&GRID_IT
 FILEORB = $Project.RasOrb.4
 NAME = 4; All
&GRID_IT
 FILEORB = $Project.RasOrb.5
 NAME = 5; All
```

In GRID_IT input we have included all orbitals. It is, however, possible and in general recommended to restrict the calculation to certain sets of orbitals. How to do this is described in the input manual for GRID_IT.

Simple math operations can be performed with grids of the same size, for example, LUSCUS can be used to display the difference between two densities.

CASSCF wave functions are typically good enough, but this is not the case for electronic energies, and the dynamic correlation effects have to be included, in particular here with the CASPT2 method. The proper input is prepared, again including SEWARD and RASSCF (unnecessary if they were computed previously), adding a CASPT2 input with the keyword MultiState set to 5 1 2 3 4 5. The CASPT2 will perform four consecutive single-state (SS) CASPT2 calculations using the SA-CASSCF roots computed by the RASSCF module. At the end, a multi-state CASPT2 calculation will be added in which the five SS-CASPT2 roots will be allowed to interact. The final MS-CASPT2 solutions, unlike the previous SS-CASPT2 states, will be orthogonal. The FROZen keyword is put here as a reminder. By default the program leaves the core orbitals frozen.

```
*CASPT2 calculation on five singlet excited states in acrolein
*File: CASPT2.excited.acrolein
*
&GATEWAY
 Title= Acrolein molecule
 coord = acrolein.xyz; basis = STO-3G; group= c1
&SEWARD; &SCF
&RASSCF
 Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
 CiRoot= 5 5 1
&GRID_IT
 All
&CASPT2
 Multistate= 5 1 2 3 4 5
 Frozen= 4
```

Apart from energies and state properties it is quite often necessary to compute state interaction properties such as transition dipole moments, Einstein coefficients, and many other. This can be achieved with the RASSI module, a powerful program which can be used for many purposes (see section 8.41) . We can start by simply computing the basic interaction properties

```
*RASSI calculation on five singlet excited states in acrolein
*File: RASSI.excited.acrolein
*
&GATEWAY
 Title= Acrolein molecule
 coord = acrolein.xyz; basis = STO-3G; group = c1
&SEWARD; &SCF
&RASSCF
 LumOrb
 Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
 CiRoot= 5 5 1
&CASPT2
 Frozen = 4
 MultiState= 5 1 2 3 4 5
```

```
>>COPY $Project.JobMix JOB001
&RASSI
 Nr of JobIph
 1 5
 1 2 3 4 5
 EJob
```

Oscillator strengths for the computed transitions and Einstein coefficients are compiled at the end of the `RASSI` output file. To obtain these values, however, energy differences have been used which are obtained from the previous CASSCF calculation. Those energies are not accurate because they do not include dynamic correlation energy and it is better to substitute them by properly computed values, such those at the CASPT2 level. This is achieved with the keyword EJob. More information is available in section 10.5.1.

Now a more complex case. We want to compute vertical singlet-triplet gaps from the singlet ground state of acrolein to different, up to five, triplet excited states. Also, interaction properties are requested. Considering that the spin multiplicity differs from the ground to the excited states, the spin Hamiltonian has to be added to our calculations and the `RASSI` program takes charge of that. It is required first, to add in the `SEWARD` input the keyword AMFI, which introduces the proper integrals required, and to the `RASSI` input the keyword SpinOrbit. Additionally, as we want to perform the calculation sequentially and `RASSI` will read from two different wave function calculations, we need to perform specific links to save the information. The link to the first `CASPT2` calculation will saved in file *$Project.JobMix.S* the data from the `CASPT2` result of the ground state, while the second link before the second `CASPT2` run will do the same for the triplet states. Later, we link these files as *JOB001* and *JOB002* to become input files for `RASSI`. In the `RASSI` input NrofJobIph will be set to two, meaning two *JobIph* or *JobMix* files, the first containing one root (the ground state) and the second five roots (the triplet states). Finally, we have added EJob, which will read the CASPT2 (or MS-CASPT2) energies from the *JobMix* files to be incorporated to the `RASSI` results. The magnitude of properties computed with spin-orbit coupling (SOC) depends strongly on the energy gap, and this has to be computed at the highest possible level, such as CASPT2.

```
*CASPT2/RASSI calculation on singlet-triplet gaps in acrolein
*File: CASPT2.S-T_gap.acrolein
*
&GATEWAY
 Title= Acrolein molecule
 coord = acrolein.xyz; basis = STO-3G; group= c1
 AMFI
&SEWARD
&SCF
&RASSCF
 Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
 CiRoot= 1 1 1
&CASPT2
 Frozen= 4
 MultiState= 1 1
>>COPY $Project.JobMix JOB001
&RASSCF
 LumOrb
 Spin= 3; Nactel= 6 0 0; Inactive= 12; Ras2= 5
 CiRoot= 5 5 1
&CASPT2
 Frozen= 4
 MultiState= 5 1 2 3 4 5
>>COPY $Project.JobMix JOB002
&RASSI
```

```
 Nr of JobIph= 2 1 5; 1; 1 2 3 4 5
 Spin
 EJob
```

As here with keyword AMFI, when using command Coord to build a SEWARD input and we want to introduce other keywords, it is enough if we place them after the line corresponding to Coord. Observe that the nature of the triplet states obtained is in sequence one $n\pi^*$, two $\pi\pi^*$, and two $n\pi^*$. The RASSI output is somewhat complex to analyze, but it makes tables summarizing oscillator strengths and Einstein coefficients, if those are the magnitudes of interest. Notice that a table is first done with the spin-free states, while the final table include the spin-orbit coupled eigenstates (in the CASPT2 energy order here), in which each former triplet state has three components.

In many cases working with symmetry will help us to perform calculations in quantum chemistry. As it is a more complex and delicate problem we direct the reader to the examples section in this manual. However, we include here two inputs that can help the beginners. They are based on trans-1,3-butadiene, a molecule with a C_{2h} ground state. If we run the next input, the SEWARD and SCF outputs will help us to understand how orbitals are classified by symmetry, whereas reading the RASSCF output the structure of the active space and states will be clarified.

```
*CASSCF SA calculation on 1Ag excited states in tButadiene
*File: CASSCF.excited.tButadiene.1Ag
*
&SEWARD
  Title= t-Butadiene molecule
  Symmetry= Z XYZ
Basis set
C.STO-3G...
C1   -3.2886930 -1.1650250 0.0000000  Bohr
C2   -0.7508076 -1.1650250 0.0000000  Bohr
End of basis
Basis set
H.STO-3G...
H1   -4.3067080  0.6343050 0.0000000  Bohr
H2   -4.3067080 -2.9643550 0.0000000  Bohr
H3    0.2672040 -2.9643550 0.0000000  Bohr
End of basis
&SCF
&RASSCF
 LumOrb
 Title= tButadiene molecule (1Ag states). Symmetry order (ag bg bu au)
 Spin= 1; Symmetry= 1; Nactel= 4 0 0; Inactive= 7 0 6 0; Ras2= 0 2 0 2
 CiRoot= 4 4 1
&GRID_IT
 All
```

Using the next input will give information about states of a different symmetry. Just run it as a simple exercise.

```
*CASSCF SA calculation on 1Bu excited states in tButadiene
*File: CASSCF.excited.tButadiene.1Bu
*
&SEWARD
 Title= t-Butadiene molecule
 Symmetry= Z XYZ
Basis set
C.STO-3G...
C1   -3.2886930 -1.1650250 0.0000000  Bohr
```

```
C2   -0.7508076 -1.1650250 0.0000000  Bohr
End of basis
Basis set
H.STO-3G...
H1   -4.3067080  0.6343050 0.0000000  Bohr
H2   -4.3067080 -2.9643550 0.0000000  Bohr
H3    0.2672040 -2.9643550 0.0000000  Bohr
End of basis
&SCF
&RASSCF
 FileOrb= $Project.ScfOrb
 Title= tButadiene molecule (1Bu states). Symmetry order (ag bg bu au)
 Spin= 1; Symmetry= 1; Nactel= 4 0 0; Inactive= 7 0 6 0
 Ras2= 0 2 0 2
 CiRoot= 4 4 1
>COPY $Project.RasOrb $Project.1Ag.RasOrb
>COPY $Project.JobIph JOB001
&GRID_IT
 Name= $Project.1Ag.lus
 All
&RASSCF
 FileOrb= $Project.ScfOrb
 Title= tButadiene molecule (1Bu states). Symmetry order (ag bg bu au)
 Spin= 1; Symmetry= 3; Nactel= 4 0 0; Inactive= 7 0 6 0; Ras2= 0 2 0 2
 CiRoot= 2 2 1
>COPY $Project.RasOrb $Project.1Bu.RasOrb
>COPY $Project.JobIph JOB002
&GRID_IT
 Name= $Project.1Bu.lus
 All
&RASSI
 NrofJobIph= 2 4 2; 1 2 3 4; 1 2
```

Structure optimizations can be also performed at the CASSCF, RASSCF or CASPT2 levels. Here we shall optimize the second singlet state in the first (here the only) symmetry for acrolein at the SA-CASSCF level. It is strongly recommended to use the State-Average option and avoid single state CASSCF calculations for excited states. Those states are non-orthogonal with the ground state and are typically heavily contaminated. The usual set of input commands will be prepared, with few changes. In the RASSCF input two states will be simultaneously computed with equal weight (CiRoot 2 2 1), but, in order to get accurate gradients for a specific root (not an averaged one), we have to add Rlxroot and set it to two, which is, among the computed roots, that we want to optimize. The proper density matrix will be stored. The MCLR program optimizes, using a perturbative approach, the orbitals for the specific root (instead of using averaged orbitals), but the program is called automatically and no input is needed.

```
 *CASSCF excited state optimization in acrolein
 *File: CASSCF.excited_state_optimization.acrolein
 *
  &GATEWAY
Title= acrolein minimum optimization in excited state 2
Basis set
O.STO-3G...2s1p.
O1       1.608542      -0.142162       3.240198 Angstrom
End of basis
Basis set
C.STO-3G...2s1p.
C1      -0.207776       0.181327      -0.039908 Angstrom
C2       0.089162       0.020199       1.386933 Angstrom
C3       1.314188       0.048017       1.889302 Angstrom
End of basis
Basis set
```

```
H.STO-3G...1s.
H1       2.208371       0.215888       1.291927 Angstrom
H2      -0.746966      -0.173522       2.046958 Angstrom
H3      -1.234947       0.213968      -0.371097 Angstrom
H4       0.557285       0.525450      -0.720314 Angstrom
End of basis
>>> Do while
 &SEWARD
>>> If ( Iter = 1 ) <<<
 &SCF
Title= acrolein minimum optimization
>>> EndIf <<<
 &RASSCF
LumOrb
Title= acrolein
Spin= 1; nActEl= 4 0 0; Inactive= 13; Ras2= 4
CiRoot= 2 2 1
Rlxroot= 2
 &SLAPAF
>>> EndDo
```

In case of performing a CASPT2 optimization for an excited state, still the SA-CASSCF approach can be used to generate the reference wave function, but keyword Rlxroot and the use of the MCLR program are not necessary, because CASPT2 takes care of selecting the proper root (the last one).

A very useful tool recently included in Molcas is the possibility to compute minimum energy paths (MEP), representing steepest descendant minimum energy reaction paths which are built through a series of geometry optimizations, each requiring the minimization of the potential energy on a hyperspherical cross section of the PES centered on a given reference geometry and characterized by a predefined radius. One usually starts the calculation from a high energy reference geometry, which may correspond to the Franck-Condon (FC) structure on an excited-state PES or to a transition structure (TS). Once the first lower energy optimized structure is converged, this is taken as the new hypersphere center, and the procedure is iterated until the bottom of the energy surface is reached. Notice that in the TS case a pair of steepest descent paths, connecting the TS to the reactant and product structures (following the forward and reverse orientation of the direction defined by the transition vector) provides the minimum energy path (MEP) for the reaction. As mass-weighted coordinates are used by default, the MEP coordinate corresponds to the so-called Intrinsic Reaction Coordinates (IRC). We shall compute here the MEP from the FC structure of acrolein along the PES of the second root in energy at the CASSCF level. It is important to remember that the CASSCF order may not be accurate and the states may reverse orders at higher levels such as CASPT2.

```
*CASSCF excited state mep points in acrolein
*File: CASSCF.mep_excited_state.acrolein
*
 &GATEWAY
Title = acrolein mep calculation root 2
Basis set
O.STO-3G...2s1p.
 O1    1.367073     0.000000     3.083333 Angstrom
End of basis
Basis set
C.STO-3G...2s1p.
 C1    0.000000     0.000000     0.000000 Angstrom
 C2    0.000000     0.000000     1.350000 Angstrom
 C3    1.367073     0.000000     1.833333 Angstrom
End of basis
```

```
Basis set
H.STO-3G...1s.
 H1    2.051552     0.000000     0.986333 Angstrom
 H2   -0.684479     0.000000     2.197000 Angstrom
 H3   -1.026719     0.000000    -0.363000 Angstrom
 H4    0.513360     0.889165    -0.363000 Angstrom
End of basis
>>> EXPORT MOLCAS_MAXITER=300
>>> Do while
 &SEWARD
>>> If ( Iter = 1 ) <<<
 &SCF
>>> EndIf <<<
 &RASSCF
   Title="acrolein mep calculation root 2"; Spin=1
   nActEl=4 0 0; Inactive=13; Ras2=4; CiRoot=2 2 1; Rlxroot=2
 &SLAPAF
   MEP-search
   MEPStep=0.1
>>> EndDo
```

As observed, to prepare the input for the MEP is simple, just add the keyword MEP-SEARCH and specify a step size with MEPSTEP, and the remaining structure equals that of a geometry optimization. The calculations are time consuming, because each point of the MEP (four plus the initial one obtained here) is computed through a specific optimization. A file named *$Project.mep.molden* (read by MOLDEN) will be generated in $WorkDir containing only those points belonging to the MEP.

We shall now show how to perform geometry optimizations under nongeometrical restrictions, in particular, how to compute hypersurface crossings, which are key structures in the photophysics of molecules. We shall get those points as minimum energy crossing points in which the energy of the highest of the two states considered is minimized under the restriction that the energy difference with the lowest state should equal certain value (typically zero). Such point can be named a minimum energy crossing point (MECP). If a further restriction is imposed, like the distance to a specific geometry, and several MECP as computed at varying distances, it is possible to obtain a crossing seam of points where the energy between the two states is degenerated. Those degeneracy points are funnels with the highest probability for the energy to hop between the surfaces in internal conversion or intersystem crossing photophysical processes. There are different possibilities. A crossing between states of the same spin multiplicity and spatial symmetry is named a conical intersection. Elements like the nonadiabatic coupling terms are required to obtain them strictly, and they are not computed presently by MOLCAS. If the crossing occurs between states of the same spin multiplicity and different spatial symmetry or between states of different spin multiplicity, the crossing is an hyperplane and its only requirement is the energetic degeneracy and the proper energy minimization.

Here we include an example with the crossing between the lowest singlet (ground) and triplet states of acrolein. Notice that two different states are computed, first by using RASSCF to get the wave function and then ALASKA to get the gradients of the energy. Nothing new on that, just the information needed in any geometry optimizations. The GATEWAY input requires to add as constraint an energy difference between both states equal to zero. A specific instruction is required after calculating the first state. We have to copy the communication file *RUNFILE* (at that point contains the information about the first state) to *RUNFILE2* to provide later SLAPAF with proper information about both states:

```
*CASSCF singlet-triplet crossing in acrolein
```

```
*File: CASSCF.S-T_crossing.acrolein
*
 &GATEWAY
Title= Acrolein molecule
Basis set
O.sto-3g....
 O1            1.5686705444      -0.1354553340        3.1977912036  Angstrom
End of basis
Basis set
C.sto-3g....
 C1           -0.1641585340       0.2420235062       -0.0459895824  Angstrom
 C2            0.1137722023      -0.1389623714        1.3481527296  Angstrom
 C3            1.3218729238       0.1965728073        1.9959513294  Angstrom
End of basis
Basis set
H.sto-3g....
 H1            2.0526602523       0.7568282320        1.4351034056  Angstrom
 H2           -0.6138178851      -0.6941171027        1.9113821810  Angstrom
 H3           -0.8171509745       1.0643342316       -0.2648232855  Angstrom
 H4            0.1260134708      -0.4020589690       -0.8535699812  Angstrom
End of basis
Constraints
   a = Ediff
  Value
   a = 0.000
End of Constraints
>>> Do while
 &SEWARD
>>> IF ( ITER = 1 ) <<<
 &SCF
>>> ENDIF <<<
 &RASSCF
   LumOrb
   Spin= 1; Nactel= 4 0 0; Inactive= 13; Ras2= 4
   CiRoot= 1 1; 1
 &ALASKA
>>COPY $WorkDir/$Project.RunFile $WorkDir/RUNFILE2
 &RASSCF
   LumOrb
   Spin= 3; Nactel= 4 0 0; Inactive= 13; Ras2= 4
   CiRoot= 1 1; 1
 &ALASKA
 &SLAPAF
>>> EndDo
```

Solvent effects can be also applied to excited states, but first the reaction field in the ground (initial) state has to be computed. This is because solvation in electronic excited states is a non equilibrium situation in with the electronic polarization effects (fast part of the reaction field) have to treated apart (they supposedly change during the excitation process) from the orientational (slow part) effects. The slow fraction of the reaction field is maintained from the initial state and therefore a previous calculation is required. From the practical point of view the input is simple as illustrated in the next example. First, the proper reaction-field input is included in SEWARD, then a RASSCF and CASPT2 run of the ground state, with keyword RFPERT in CASPT2, and after that another SA-CASSCF calculation of five roots to get the wave function of the excited states. Keyword NONEQUILIBRIUM tells the program to extract the slow part of the reaction field from the previous calculation of the ground state (specifically from the *JOBOLD* file, which may be stored for other calculations) while the fast part is freshly computed. Also, as it is a SA-CASSCF calculation (if not, this is not required) keyword RFROOT is introduced to specify for which of the computed roots the reaction field is generated. We have selected here the fifth root because it has a very large dipole moment, which is also very different from the ground state dipole moment. If you compare the excitation energy obtained for the isolated and the solvated system, a the large red shift is obtained in the later.

```
*CASPT2 excited state in water for acrolein
*File: CASPT2.excited_solvent.acrolein
*
&GATEWAY
  Title= Acrolein molecule
  coord = acrolein.xyz; basis = STO-3G; group= c1
  RF-input
   PCM-model; solvent= water
  End of RF-input
&SEWARD
&RASSCF
  Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
  CiRoot= 1 1 1
&CASPT2
  Multistate= 1 1
  RFPert
&RASSCF
  Spin= 1; Nactel= 6 0 0; Inactive= 12; Ras2= 5
  CiRoot= 5 5 1
  RFRoot= 5
  NONEquilibrium
&CASPT2
  Multistate= 1 5
  RFPert
```

A number of simple examples as how to proceed with the most frequent quantum chemical problems computed with Molcas have been given above. Certainly there are many more possibilities in Molcas 8.2 such as calculation of 3D band systems in solids at a semiempirical level, obtaining valence-bond structures, the use of QM/MM methods in combination with a external MM code, the introduction of external homogeneous or non homogeneous perturbations, generation of atomic basis sets, application of different localization schemes, analysis of first order polarizabilities, calculation of vibrational intensities, analysis, generation, and fitting of potentials, computation of vibro-rotational spectra for diatomic molecules, introduction of relativistic effects, etc. All those aspects are explained in the manual and are much more specific. Next section 6 details the basic structure of the inputs, program by program, while easy examples can also be found. Later, another chapter includes a number of extremely detailed examples with more elaborated quantum chemical examples, in which also scientific comments are included. Examples include calculations on high symmetry molecules, geometry optimizations and Hessians, computing reaction paths, high quality wave functions, excited states, solvent models, and computation of relativistic effects.

Section 6

Program Based Tutorials

The MOLCAS 8.2 suite of Quantum Chemical programs is modular in design. The desired calculation is achieved by executing a list of MOLCAS program modules in succession, while potentially manipulating the program information files. If the information files from a previous calculation are saved, then a subsequent calculation need not recompute them. This is dependent on the correct information being preserved in the information files for subsequent calculations. Each module has keywords to specify the functions to be carried out, and many modules rely on the specification of keywords in previous modules.

The following sections describe the use of the MOLCAS modules and their inter-relationships. Each module is introduced in the approximate order for performing a typical calculation. A complete flowchart for the MOLCAS 8.2 suite of programs follows.

6.1 8.2 Flowchart

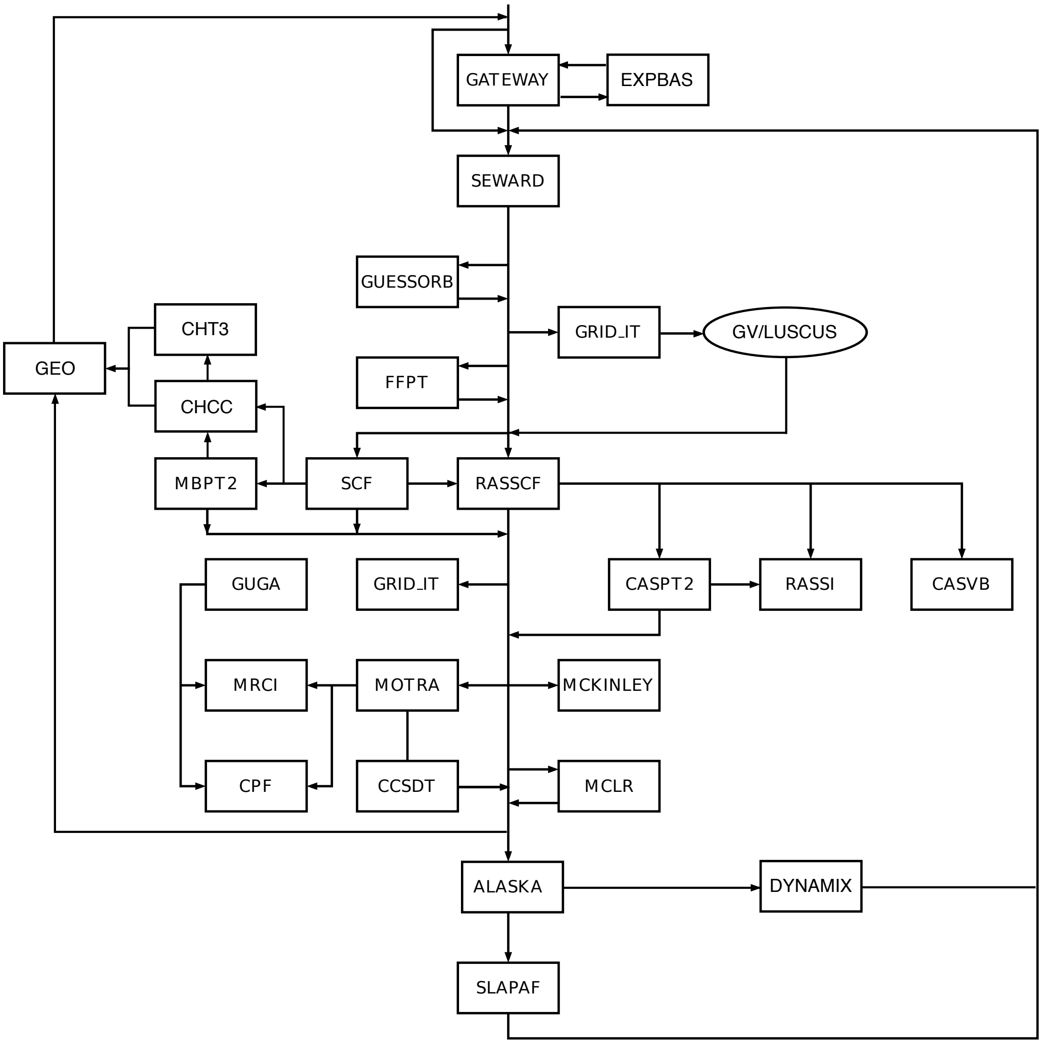


Figure 6.1: Flowchart for Module Dependencies in MOLCAS

6.2 Environment and EMIL Commands

The following are basic and most common commands for the MOLCAS environment variables and input language (EMIL):

Variable	*Purpose*
MOLCAS	MOLCAS home directory.
MOLCAS_MEM	Memory definition in Mb. Default is 2048Mb.
MOLCAS_PRINT	Set printing level: 2 for Normal, 3 for Verbose

MOLCAS_PROJECT Name used for the project/files.

MOLCAS_WORKDIR Scratch directory for intermediate files.

MOLCAS_NPROCS Use NPROCS parallel processes

Special commands in the input always start from symbol > (same as several repeated signs >>). Comment lines can be written using C/C++ syntax (//, or $/ * ... * /$).

Command	*Purpose*
>>Do While	Start of a loop in an input file for geometry optimization with conditional termination.
>>Foreach	Start of a loop in an input file over a number of items.
>>EndDo	End of a loop in an input file.
>>If (condition)	Start of If block.
>>EndIf	End of If block.
>>Label Mark	Setting the label ”Mark” in the input.
>>Goto Mark	Forward jump to the label ”Mark” skipping that part of the input.
>>SHELL command	Execute command once
>>EXEC command	Execute command for all processes
>>echo string	make an echo

Some simple examples of EMIL commands:

Say “Hello World” 5 times

```
>>export MOLCAS_MAXITER=5
>> Do While
>> echo 'Hello World'
&LOOP  // it needed to terminate loop
>> End Do
```

Do some arithmetics:

```
>>foreach XX in ( 1..10 )  // Note: All variables: XX, YY must be uppercased
>>eval YY=1/$XX
>>echo $YY
>>enddo
```

Print working directory:

```
>>shell pwd
```

Print working directory for each process:

```
>>export MOLCAS_NPROCS=2
>>exec pwd
```

6.3 GATEWAY - Definition of geometry, basis sets, and symmetry

The program GATEWAY handles the basic molecular parameters in the calculation. It generates data that are used in all subsequent calculations. These data are stored in the *RUNFILE*. GATEWAY is the first program to be executed, if the $WorkDir directory and the *RUNFILE* file has not already been generated by a previous calculation.

This tutorial is describes how to set up the basic Molcas input for the water molecule. For a more general description of the input options for GATEWAY, please refer to the Users Guide. The first line of the input is the program identifier &GATEWAY. Then follows the keyword used is TITLe which will also get printed in the GATEWAY section of the calculation output. The title line is also saved in the integral file and will appear in subsequent programs.

The GROUp keyword is followed by the generators for the C_{2v} point group, since the example deals with the water molecule. The specification of the C_{2v} point group given in Table 6.1 is not unique, but, in this tutorial, the generators have been input in an order that reproduces the ordering in the character tables. A complete list of symmetry generator input syntax is given in Table 6.1. The symmetry groups available are listed with the symmetry generators defining the group. The Molcas keywords required to specify the symmetry groups are also listed. The last column contains the symmetry elements generated by the symmetry generators.

```
 &GATEWAY
Title= Water in C2v symmetry - A Tutorial
Coord = water.xyz
Group =  XY Y
Basis Set = O.ANO-S-MB,H.ANO-S-MB
```

Table 6.1: Symmetries available in MOLCAS including generators, MOLCAS keywords and symmetry elements.

Group	Generators			Molcas			Elements							
	g_1	g_2	g_3	g_1	g_2	g_3	E	g_1	g_2	g_1g_2	g_3	g_1g_3	g_2g_3	$g_1g_2g_3$
C_1							E							
C_2	C_2			XY			E	C_2						
C_s	σ			X			E	σ						
C_i	i			XYZ			E	i						
C_{2v}	C_2	σ_v		XY	Y		E	C_2	σ_v	σ'_v				
C_{2h}	C_2	i		XY	XYZ		E	C_2	i	σ_h				
D_2	C_2^z	C_2^y		XY	XZ		E	C_2^z	C_2^y	C_2^x				
D_{2h}	C_2^z	C_2^y	i	XY	XZ	XYZ	E	C_2^z	C_2^y	C_2^x	i	σ^{xy}	σ^{xz}	σ^{yz}

To reduce the input, the unity operator E is always assumed. The twofold rotation about the z-axis, $C_2(z)$, and the reflection in the xz-plane, $\sigma_v(xz)$, are input as XY and Y respectively. The Molcas input can be viewed as symmetry operators that operate on the Cartesian elements specified. For example, the reflection in the xz-plane is specified by the input keyword Y which is the Cartesian element operated upon by the reflection.

The input produces the character table in the GATEWAY section of the output shown in Figure 6.3. Note that $\sigma_v(yz)$ was produced from the other two generators. The last column contains the basis functions of each irreducible symmetry representation. The totally symmetric a_1 irreducible representation has the z basis function listed which is unchanged by any of the symmetry operations.

```
           E   C2(z) s(xz) s(yz)
a1         1     1     1     1  z
b1         1    -1     1    -1  x, xz, Ry
a2         1     1    -1    -1  xy, Rz, I
b2         1    -1    -1     1  y, yz, Rx
```

The geometry of the molecule is defined using the keyword COORD. On the next line, the name of the xyz file that defines the geometrical parameters of the molecule (*water.xyz*) is given.

1. The first line of the *water.xyz* file contains the number of atoms.
2. The second line is used to indicate the units: Ångström or atomic units.

   The default is to use Ångström.
3. Then follows one line for each atom containing the name of each atom and its coordinates.

Basis sets are defined after the keyword BASIs sets. The oxygen and hydrogen basis set chosen, for this example, are the small Atomic Natural Orbitals (ANO) sets. There are three contractions of the basis included in the input, which can be toggled in or excluded with an asterisk, according to the desired calculation: minimal basis, double zeta basis with polarization, or triple zeta basis with polarization.

Figure 6.2: The geometry of the water molecule

```
 3

O        .000000        .000000      .000000
H       0.700000        .000000     0.700000
H      -0.700000        .000000     0.700000
```

6.3.1 GATEWAY Output

The GATEWAY output contains the symmetry character table, basis set information and input atomic centers. The basis set information lists the exponents and contraction coefficients as well as the type of Gaussian functions (Cartesian, spherical or contaminated) used.

The internuclear distances and valence bond angles (including dihedral angles) are displayed after the basis set information. Inertia and rigid-rotor analysis is also included in the output along with the timing information.

A section of the output that is useful for determining the input to the MOLCAS module SCF is the symmetry adapted basis functions which appears near the end of the GATEWAY portion of the output. This is covered in more detail in the SCF tutorial.

The most important file produced by the GATEWAY module is the ***RUNFILE*** which in our case is linked to ***water.RunFile***. This is the general MOLCAS communications file for transferring data between the various MOLCAS program modules. Many of the program modules add data to the ***RUNFILE*** which can be used in still other modules. A new ***RUNFILE*** is produced every time GATEWAY is run. It should finally be mentioned that for backwards compatibility one can run MOLCAS without invoking GATEWAY. The corresponding input and output will then be handled by the program SEWARD.

6.3.2 Basis Set Superposition Error (BSSE)

GATEWAY can operates with several coordinate files, which is convenient for computing BSSE corrections. BSSE followed by a number marks a XYZ file which should be treated as dummy atoms. The following example demonstrates this feature:

```
&GATEWAY
coord = ethanol.xyz
coord = water.xyz
bsse  = 1
basis = ANO-S-MB
NOMOVE
&SEWARD; &SCF
&GRID_IT
NAME = water
***************
&GATEWAY
coord = ethanol.xyz
coord = water.xyz
bsse  = 2
basis = ANO-S-MB
NOMOVE
&SEWARD; &SCF
&GRID_IT
NAME = ethanol
**************
&GATEWAY
coord = ethanol.xyz
coord = water.xyz
basis = ANO-S-MB
NOMOVE
&SEWARD; &SCF
&GRID_IT
NAME = akvavit
```

Note, that NOMOVE keyword prevents centering of the molecule, so the computed grids are identical. An alternative way to compute density difference is to modify coordinates, and change an element label to X.

6.3.3 GATEWAY Basic and Most Common Keywords

Keyword	*Meaning*
COORD	File name or inline number of atoms and XYZ coordinates
BASIS SET	Atom_label.Basis_label (for example ANO-L-VTZP)

Group	Full (find maximum), NoSym, or symmetry generators
SYMMetry	Symmetry generators: X, Y, Z, XY, XZ, YZ, XYZ (in native format)
RICD	On-the-fly auxiliary basis sets.

6.4 SEWARD — An Integral Generation Program

An *ab initio* calculation always requires integrals. In the Molcas suite of programs, this function is supplied by the SEWARD module. SEWARD computes the one- and two-electron integrals for the molecule and basis set specified in the input to the program GATEWAY, which should be run before SEWARD. SEWARD can also be used to perform some property expectation calculations on the isolated molecule. The module is also used as an input parser for the reaction field and numerical quadrature parameters.

We commence our tutorial by calculating the integrals for a water molecule. The input is given in Figure 6.4. Each Molcas module identifies input from a file by the name of the module. In the case of SEWARD, the program starts with the label &SEWARD, which is the first statement in the file shown below.

In normal cases no input is required for SEWARD, so the following input is optional. The first keyword used is TITLe. Only the first line of the title is printed in the output. The first title line is also saved in the integral file and appears in any subsequent programs that use the integrals calculated by SEWARD.

```
 &SEWARD
Title
 Water - A Tutorial. The integrals of water are calculated using C2v symmetry
```

In more complicated cases more input may be needed, to specify certain types of integrals, that use of Cholesky decomposition techniques (CHOLesky keyword), etc. We refer to the specific sections of the Users-Guide for more information. The output from a SEWARD calculation is small and contains in principle only a list of the different types of integrals that are computed.

The integrals produced by the SEWARD module are stored in two files in the working directory. They are ascribed the FORTRAN names *ONEINT* and *ORDINT* which are automatically symbolically linked by the Molcas script to the file names $Project.*OneInt* and $Project.*OrdInt*, respectively or more specifically, in our case, *water.OneInt* and *water.OrdInt*, respectively. The default name for each symbolical name is contained in the corresponding program files of the directory $MOLCAS/shell. The *ONEINT* file contains the one-electron integrals. The *ORDINT* contains the ordered and packed two-electron integrals. Both files are used by later Molcas program modules.

6.5 SCF — A Self-Consistent Field program and Kohn Sham DFT

The simplest *ab initio* calculations possible use the Hartree-Fock (HF) Self-Consistent Field (SCF) method with the program name SCF in the Molcas suite. It is possible to calculate the HF energy once we have calculated the integrals using the SEWARD module, although MOLCAS can perform a direct SCF calculation in which the two-electron integrals are not stored on disk. The MOLCAS implementation performs a closed-shell (all electrons are paired in orbitals) and open-shell (Unrestricted Hartree-Fock) calculation. It is not possible to perform an Restricted Open-shell Hartree-Fock (ROHF) calculation with the SCF. This is instead done using the program RASSCF. The SCF program can also be used to perform calculations using Kohn Sham Density Functional Theory (DFT).

The SCF input for a Hartree-Fock calculation of a water molecule is given in figure 6.3 which continues our calculations on the water molecule.

There are no compulsory keywords following the program name, &SCF. If no input is given the program will compute the SCF energy for a neutral molecule with the orbital occupations giving the lowest energy. Here, we have used the following input: the first is TITLE. As with the SEWARD program, the first line following the keyword is printed in the output.

No other keyword is required for a closed-shell calculation. The program will find the lowest-energy electron configuration compatible with the symmetry of the system and will distribute the orbitals accordingly. In complex cases the procedure may fail and produce a higher-lying configuration. It is possible to use the keyword OCCUPIED which specifies the number of occupied orbitals in each symmetry grouping listed in the GATEWAY output and given in Figure 6.4, forcing the method to converge to the specified configuration. The basis label and type give an impression of the possible molecular orbitals that will be obtained in the SCF calculation. For example, the first basis function in the a_1 irreducible representation is an s type on the oxygen indicating the oxygen $1s$ orbital. Note, also, that the fourth basis function is centered on the hydrogens, has an s type and is symmetric on both hydrogens as indicated by both hydrogens having a phase of 1, unlike the sixth basis function which has a phase of 1 on center 2 (input H1) and -1 on center 3 (generated H1). As an alternative you can use the keyword CHARGE with parameters 0 and 1 to indicate a neutral molecule and optimization procedure 1 that searches for the optimal occupation.

Figure 6.3: Sample input requesting the SCF module to calculate the ground Hartree-Fock energy for a neutral water molecule in C_{2v} symmetry.

```
 &SCF
Title= Water - A Tutorial. The SCF energy of water is calculated using C2v symmetry
End of Input
```

Figure 6.4: Symmetry adapted Basis Functions from a GATEWAY output.

```
          Irreducible representation : a1
          Basis function(s) of irrep: z

Basis Label        Type   Center Phase
  1   O1           1s        1     1
  2   O1           2s        1     1
  3   O1           2p0       1     1
  4   H1           1s        2     1      3     1

          Irreducible representation : b1
          Basis function(s) of irrep: x, xz, Ry
Basis Label        Type   Center Phase
  5   O1           2p1+      1     1
  6   H1           1s        2     1      3    -1

          Irreducible representation : b2
          Basis function(s) of irrep: y, yz, Rx

Basis Label        Type   Center Phase
  7   O1           2p1-      1     1
```

We have ten electrons to ascribe to five orbitals to describe a neutral water molecule in the ground state. Several techniques exist for correct allocation of electrons. As a test of the electron allocation, the energy obtained should be the same with and without symmetry. Water is a simple case, more so when using the minimal basis set. In this case, the fourth irreducible representation is not listed in the GATEWAY output as there are no basis functions in that representation.

To do a UHF calculation, the keyword UHF must be specified. To force a specific occupation for alpha and beta orbitals In this keyword OCCNumbers has to be used with two entries, one for alpha and beta occupied orbital. It is possible to use UHF together with keyword Charge or Aufbau, in this case you have to specify a keyword ZSPIN set to the difference between alpha and beta electrons.

If you want to do an UHF calculation for a closed shell system, for example, diatomic molecule with large interatomic distance, you have to specify keyword SCRAMBLE.

To do the Density Functional Theory calculations, keyword KSDFT followed in the next line by the name of the available functional as listed in the input section is compulsory. Presently following Functional Keywords are available: BLYP, B3LYP, B3LYP5, HFB, HFS, LDA, LDA5, LSDA, LSDA5, SVWN, SVWN5, TLYP, XPBE, MO6, MO6/HF, MO6/2X, MO6/L. The description of functional keywords and the functionals is defined in the section DFT Calculations 8.44.2.

The input for KSDFT is given as,

```
KSDFT= B3LYP5
```

In the above example B3LYP5 functional will be used in KSDFT calculations.

6.5.1 Running SCF

Performing the Hartree-Fock calculation introduces some important aspects of the transfer of data between the Molcas program modules. The SCF module uses the integral files computed by SEWARD. It produces a orbital file with the symbolic name *SCFORB* which contains all the MO information. This is then available for use in subsequent Molcas modules. The SCF module also adds information to the *RUNFILE*. Recall that the SEWARD module produces two integral files symbolically linked to *ONEINT* and *ORDINT* and actually called, in our case, *water.OneInt* and *water.OrdInt*, respectively (this is for non-Cholesky-type calculations only). Because the two integral files are present in the working directory when the SCF module is performed, Molcas automatically links them to the symbolic names.

If the integral files were not deleted in a previous calculation the SEWARD calculation need not be repeated. Furthermore, integral files need not be in the working directory if they are linked by the user to their respective symbolic names. Integral files, however, are often very large making it desirable to remove them after the calculation is complete. The linking of files to their symbolic names is useful in other case, such as input orbitals.

If nothing else is stated, the `SCF` program will use the guess orbitals produced by `SEWARD` as input orbitals with the internal name *`GUESSORB`*. If one wants to use any other input orbitals for the `SCF` program the option LUMOrb must be used. The corresponding file should be copied to the internal file *`INPORB`*. This could for example be an orbital file generated by an earlier SCF calculation, *`$Project.ScfOrb`*. Just copy or link the file as *`INPORB`*.

6.5.2 `SCF` Output

The `SCF` output includes the title from the input as well as the title from the `GATEWAY` input because we used the integrals generated by `SEWARD`. The output also contains the cartesian coordinates of the molecule and orbital specifications including the number of frozen, occupied and virtual (secondary) orbitals in each symmetry. This is followed by details regarding the `SCF` algorithm including convergence criteria and iteration limits. The energy convergence information includes the one-electron, two-electron, and total energies for each iteration. This is followed by the final results including the final energy and molecular orbitals for each symmetry.

The Density Functional Theory Program gives in addition to the above, details of grids used, convergence criteria, and name of the functional used. This is followed by integrated DFT energy which is the functional contribution to the total energy and the total energy including the correlation. This is followed results including the Kohn Sham orbitals for each symmetry.

The molecular orbital (MO) information lists the orbital energy, the electron occupation and the coefficients of the basis functions contributing to that MO. For a minimal basis set, the basis functions correspond directly to the atomic orbitals. Using larger basis sets means that a combination of the basis functions will be used for each atomic orbital and more so for the MOs. The MOs from the first symmetry species are given in Figure 6.5. The first MO has an energy of -20.5611 hartree and an occupation of 2.0. The major contribution is from the first basis function label '`O1 1s`' meaning an s type function centered on the oxygen atom. The orbital energy and the coefficient indicates that it is the MO based largely on the oxygen $1s$ atomic orbital.

Figure 6.5: Molecular orbitals from the first symmetry species of a calculation of water using C_{2v} symmetry and a minimal basis set.

```
  ORBITAL         1         2         3         4
  EneRGY     -20.5611   -1.3467    -.5957     .0000
  Occ. NO.     2.0000    2.0000    2.0000     .0000

1 O1   1s      1.0000    -.0131    -.0264    -.0797
2 O1   2s       .0011     .8608    -.4646    -.7760
3 O1   2p0      .0017     .1392     .7809    -.7749
4 H1   1s      -.0009     .2330     .4849    1.5386
```

The second MO has a major contribution from the second oxygen `1s` basis function indicating a mostly oxygen $2s$ construction. Note that it is the absolute value of the coefficient that determines it importance. The sign is important for determining the orthogonality of its orbitals and whether the atomic orbitals contributions with overlap constructively (bonding) or destructively (anti-bonding). The former occurs in this MO as indicated by the positive sign on the oxygen $2s$ and the hydrogen $1s$ orbitals, showing a bonding interaction between them. The latter occurs in the third MO, where the relative sign is reversed.

The third MO has an energy of -0.5957 hartree and major contributions from the second oxygen `1s` basis function, the oxygen `2p0` basis function and the hydrogen `1s` basis functions which are symmetrically situated on each hydrogen (see Figure 6.4). The mixing of the oxygen `2s` and `2p0` basis functions leads to a hybrid orbital that points away from the two hydrogens, to which it is weakly antibonding.

A similar analysis of the fourth orbital reveals that it is the strongly anti-bonding orbital partner to the third MO. The oxygen `2p0` basis function is negative which reverses the overlap characteristics.

The molecular orbital information is followed by a Mulliken charge analysis by input center and basis function. This provides a measure of the electronic charge of each atomic center.

Towards the end of the `SCF` section of the MOLCAS output various properties of the molecule are displayed. By default the first (dipole) and second cartesian moments and the quadrupoles are displayed.

6.5.3 SCF - Basic and Most Common Keywords

Keyword	*Meaning*
UHF	Unrestricted Hartee Fock or unrestricted DFT calculation
KSDFT	DFT calculations, with options: BLYP, B3LYP, B3LYP5, HFB, HFS, LDA, LDA5, LSDA, LSDA5, SVWN, SVWN5, TLYP, PBE, PBE0
CHARGE	Net charge of the system (default zero)
ZSPIN	Difference between α and β electrons
OCCUPIED	Specify the orbital occupations per irreps

6.6 MBPT2 — A Second-Order Many-Body PT RHF Program

The `MBPT2` program performs second-order Many Body Perturbation Theory calculations based on a RHF-type of wave function (MP2 method). The calculation is to some extent defined by the SCF calculation which must be performed before running the `MBPT2` program. Therefore, there is no difficulty related to the input file unless an analysis of the correlation energies of specific electron pairs or contribution from external orbitals wants to be performed. In this case keywords SFROzen and SDELeted have to be used as described in section 8.26 of the user's guide.

To run the program the *ORDINT* integral file(s) generated by the SEWARD program and the *RUNFILE* file generated by the SCF program are needed. The program can be otherwise run in a direct manner. Therefore the SEWARD program can be run with the option DIREct included in its input. Only the *ONEINT* will then be generated and used by the SCF module. The input file used to run an MBPT2 calculation on the ground state of the water molecule is displayed in figure 6.6. For large molecules it is also possible to use the Cholesky decomposition technique to speed up the calculations. This will be described in another section of the tutorials.

```
 &MBPT2
Title= MP2 of ground state of C2v Water
Frozen= 1 0 0 0
```

Figure 6.6: Sample input requested by the MBPT2 module to calculate the MP2 energy for the ground state of the water in C_{2v} symmetry.

The output of MBPT2 is self-explanatory.

6.7 RASSCF — A Multi Configurational Self-Consistent Field Program

One of the central codes in Molcas is the RASSCF program, which performs multiconfigurational SCF calculations. Both Complete Active Space (CASSCF) and Restricted Active Space (RASSCF) SCF calculations can be performed with the RASSCF program module [11]. An open shell Hartree-Fock calculation is not possible with the SCF but it can be performed using the RASSCF module. An input listing for a CASSCF calculation of water appears in Figure 6.7. RASSCF requires orbital information of the system which can be obtained in two ways. The LUMOrb indicates that the orbitals should be taken from a user defined orbital file, which is copied to the internal file INPORB. If this keyword is not given, the program will look for orbitals on the runfile in the preference order: *RASORB*, *SCFORB* and *GUESSORB*

Figure 6.7: Sample input requesting the RASSCF module to calculate the eight-electrons-in-six-orbitals CASSCF energy of the second excited triplet state in the second symmetry group of a water molecule in C_{2v} symmetry.

```
 &RASSCF
Title= The CASSCF energy of water is calculated using C2v symmetry. 2 3B2 state.
nActEl= 8 0 0
Inactive= 1 0 0 0; Ras2= 3 2 0 1
Symmetry= 2; Spin= 3
CIRoot= 1 2; 2
LumOrb
```

The TITLe performs the same function as in the previous Molcas modules. The keyword INACtive specifies the number of doubly occupied orbitals in each symmetry that will not be included in the electron excitations and thus remain doubly occupied throughout the calculation. A diagram of the complete orbital space available in the RASSCF module is given in Figure 6.8.

In our calculation, we have placed the oxygen 1*s* orbital in the inactive space using the INACTIVE keyword. The keyword FROZEN can be used, for example, on heavy atoms to reduce the Basis Set Superposition Error (BSSE). The corresponding orbitals will then not be optimized. The RAS2 keyword specifies the number of orbitals in each symmetry to be included in the electron excitations with all possible occupations allowable. Because the RAS1 and RAS3 spaces are zero (not specified in the input in Figure 6.7) the RASSCF calculation will produce a CASSCF wave function. The RAS2 space is chosen to use all the orbitals available in each symmetry (except the oxygen 1*s* orbital). The keyword NACTEL specifies the number of active electrons (8), maximum number of holes in the Ras1 space (0) and the maximum number of electrons in the Ras3 space (0). Using the keywords RAS1 and/or RAS3 to specify orbitals and specifying none zero numbers of holes/electrons will produce a RASSCF wave function.We are, therefore, performing an 8in6 CASSCF calculation of water.

Table 6.2: Examples of types of wave functions obtainable using the RAS1 and RAS3 spaces in the RASSCF module.

Description	Number of holes in RAS1 orbitals	RAS2 orbitals	Number of electrons in RAS3 orbitals
SD-CI	2	0	2
SDT-CI	3	0	3
SDTQ-CI	4	0	4
Multi Reference SD-CI	2	n	2
Multi Reference SD(T)-CI	3	n	2

There are a number of wave function types that can be performed by manipulating the RAS1 and RAS3 spaces. Table 6.2 lists a number of types obtainable. The first three are Configuration Interaction (CI) wave functions of increasing magnitude culminating with a Single, Double, Triples and Quadruples (SDTQ) CI. These can become multi reference if the number of RAS2 orbitals is non-zero. The last type provides some inclusion of the triples excitation by allowing three holes in the RAS1 orbitals but save computation cost by only allowing double excitations in the RAS3 orbitals.

The symmetry of the wave function is specified using the SYMMETRY keyword. It specifies the number of the symmetry subgroup in the calculation. We have chosen the second symmetry species, b_2, for this calculation. We have also chosen the triplet state using the keyword SPIN. The keyword CIROOT has been used to instruct RASSCF to find the second excited state in the given symmetry and spin. This is achieved by specifying the number of roots, 1, the dimension of the small CI matrix which must be as large as the highest required root and the number of the required second root. Only for averaged calculations CIROOT needs an additional line containing the weight of the selected roots (unless equal weights are used for all states).

As an alternative to giving inactive and active orbital input we can use the type index input on the *INPORB* and indicate there which type the different orbitals should belong to: frozen (f), inactive (i), RAS1 (1), RAS2 (2), RAS3 (3), secondary (s), or deleted (d). This approach is very useful when the input orbitals have been run through LUSCUS, which is used to select the different subspaces. LUSCUS will relabel to orbitals according to the users instructions

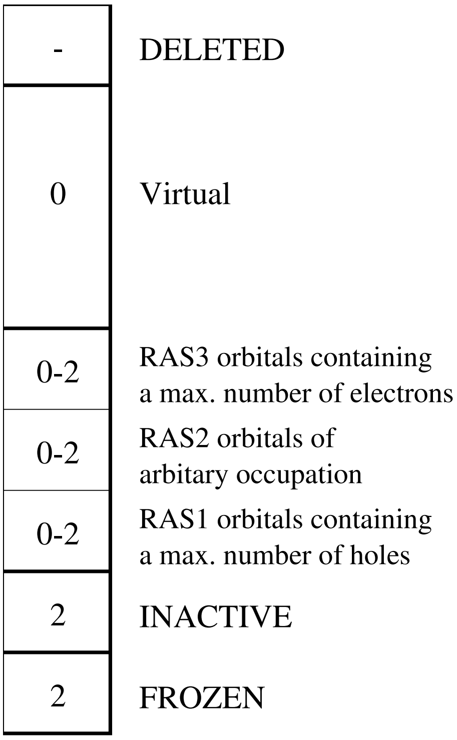


Figure 6.8: RASSCF orbital space including keywords and electron occupancy ranges.

and the corresponding orbital file ,*GvOrb* can be linked as the *INPORB* in the RASSCF program without any further input.

A level shift was included using the LEVSHIFT keyword to improve convergence of the calculation. In this case, the calculation does not converge without the use of the level shift. It is advisable to perform new calculations with a non-zero LEVSHIFT value (the default value is 0.5). Another possibility is to increase the maximum number of iterations for the macro and the super-CI Davidson procedures from the default values (200,100) using the keyword ITERATIONS.

Sometimes convergence problems might appear when the wave function is close to fulfill all the convergence criteria. An infrequent but possible divergence might appear in a calculation starting from orbitals of an already converged wave function, or in cases where the convergence thresholds have been decreased below the default values. Option TIGHT may be useful in those cases. It contains the thresholds criteria for the Davidson diagonalization procedure. In situations such as those described above it is recommended to decrease the first parameter of TIGHT to a value lower than the default, for instance 1.0d-06.

6.7.1 RASSCF Output

The RASSCF section of the MOLCAS output contains similar information to the SCF output. Naturally, the fact that we have requested an excited state is indicated in the output. In fact, both the lowest triplet state and the first excited state or second root are documented including energies. For both of these states the CI configurations with a coefficient greater than 0.05 are printed along with the partial electron distribution in the active space. Figure 6.9 shows the relevant output for the second root calculated. There are three configurations with a CI-coefficient larger than 0.05 and two with very much larger values. The number of the configuration is given in the first column and the CI-coefficient and weight are given in the last two columns. The electron occupation of the orbitals of the first symmetry for each configuration is given under the '`111`' using '`2`' for a fully occupied orbital and '`u`' for a singly occupied orbital containing an electron with an up spin. The down spin electrons are represented with a '`d`'. The occupation numbers of the active space for each symmetry is given below the contributing configurations. It is important to remember that the active orbitals are not ordered by any type of criterion within the active space.

Figure 6.9: RASSCF portion of output relating to CI configurations and electron occupation of natural orbitals.

```
printout of CI-coefficients larger than   .05 for root   2
energy=    -75.443990
conf/sym  111 22 4     Coeff  Weight
       3  22u u0 2    .64031  .40999
       4  22u 0u 2    .07674  .00589
      13  2u0 2u 2  -.75133  .56450
      14  2u0 u2 2    .06193  .00384
      19  udu 2u 2    .06489  .00421

Natural orbitals and occupation numbers for root  2
sym 1:   1.986957   1.416217    .437262
sym 2:   1.567238    .594658
sym 4:   1.997668
```

The molecular orbitals are displayed in a similar fashion to the `SCF` section of the output except that the energies of the active orbitals are not defined and therefore are displayed as zero and the electron occupancies are those calculated by the `RASSCF` module. In a state average calculation (more than one root calculated), the MOs will be the natural orbitals corresponding to the state averaged density matrix (called pseudo-natural orbitals) and the occupation numbers will be the corresponding eigenvalues. Natural orbital occupation numbers for each state are printed as shown in Figure 6.9, but the MOs specific to a given state are not shown in the output. They are, however, available in the *`JOBIPH`* file. A number of molecular properties are also computed for the requested electronic state in a similar fashion to the `SCF` module.

6.7.2 Storing and Reading `RASSCF` Orbitals and Wave Functions

Part of the information stored in the `RASSCF` output file, *`JOBIPH`*, for instance the molecular orbitals and occupation numbers can be also found in an editable file named ***`RASORB`***, which is automatically generated by `RASSCF`. In case more than one root is used the natural orbitals are also stored in files ***`RASORB.1`***, ***`RASORB.2`***, etc, up to ten. In such cases the file ***`RASORB`*** contains the averaged orbitals. If more roots are used the files can be generated using the OUTOrbitals keyword. The type of orbital produced can be either AVERaged, NATUral, CANOnical or SPIN (keywords) orbitals. The OUTOrbitals keyword, combined with the ORBOnly keyword, can be used to read the *`JOBIPH`* file and produce an orbital file, ***`RASORB`***, which can be read by a subsequent `RASSCF` calculation using the same input section. The formatted ***`RASORB`*** file is useful to operate on the orbitals in order to obtain appropriate trial orbitals for a subsequent RASSCF calculation. In particular the type index can be changed directly in the file if the RASSCF program has converged to a solution with wrong orbitals in the active space. The RASSCF program will, however, automatically place the orbital files from the calculation in the user's home directory under the name *$Project.RasOrb*, etc. In calculations with spin different from zero the program will also produce the spin orbital files *$Project.SpdOrb1*, etc for each state. These orbitals can be used by the program LUSCUS to produce spin densities.

6.7.3 RASSCF - Basic and Most Common Keywords

Keyword	*Meaning*
SYMMETRY	Symmetry of the wave function (according to GATEWAY) (1 to 8)
SPIN	Spin multiplicity
CHARGE	Molecular charge
NACTEL	Three numbers: Total number of active electrons, holes in Ras1, particles in Ras3
INACTIVE	By symmetry: doubly occupied orbitals
RAS1	By symmetry: Orbitals in space Ras1 (RASSCF)
RAS2	By symmetry: Orbitals in space Ras1 (CASSCF and RASSCF)
RAS3	By symmetry: Orbitals in space Ras1 (RASSCF)
CIROOT	Three numbers: number of CI roots, dimension of the CI matrix, relative weights (typically 1)
LUMORB/FILEORB	use definition of active space from Orbital file

6.8 CASPT2 — A Many Body Perturbation Program

Dynamic correlation energy of a molecular system can be calculated using the CASPT2 program module in MOLCAS. A CASPT2 calculation gives a second order perturbation estimate of the full CI energy using the CASSCF wave function of the system. The program can also perform Multi-State CASPT2 calculations (MS-CASPT2) in which different CASPT2 states are coupled using an effective Hamiltonian computed to second order in perturbation theory. This is necessary in cases where different CASSCF wave functions are strongly dependent on dynamical correlation effects. The wave function have to be obtained in a previous State-Average CASSCF calculation.

A sample input is given in Figure 6.10. The FROZEN keyword specifies the number of orbitals of each symmetry which will not be included in the correlation. We have chosen the RASSCF INACtive orbitals to be frozen for this calculation (the default is to freeze all core orbitals, so the input is strictly not needed). The remaining two keywords, CONVergence and MAXIter, are included with there default values. The MULTistate keyword is included for clarity even if not needed in this single state calculation. A single line follows indicating the number of simultaneously treated CASPT2 roots and the number of the roots in the previous SA-CASSCF calculation.

6.8.1 CASPT2 Output

In section 10.5.1 the meaning and significance of most of the features used and printed by the CASPT2 program are explained in the context of an actual example. We suggest a careful reading of that section because understanding the results of a CASPT2 calculation is important for the analysis of problems like intruder states, large coefficients, convergence, etc.

Figure 6.10: Sample input requesting the CASPT2 module to calculate the CASPT2 energy of a water molecule in C_{2v} symmetry with one frozen orbital.

```
 &CASPT2
Frozen= 1 0 0 0
Multistate= 1 1
MaxIter= 40
```

The output of the CASPT2 program begins with the title from the input as well as the title from the SEWARD input. It also contains the cartesian coordinates of the molecule and the CASSCF wave function and orbital specifications. This is followed by details about the type of Fock and H_0 operator used and, eventually, the value of the level-shift parameter employed. It is possible then to obtain, by input specifications, the quasi-canonical orbitals in which the wave function will be represented. The following CI vector and occupation number analysis will be performed using the quasi-canonical orbitals.

Two important sections follow. First a detailed report on small energy denominators, large components, and large energy contributions which will inform about the reliability of the calculation (see section 10.5.1) and finally the CASPT2 property section including the natural orbitals obtained as defined in the output and a number of approximated molecular properties.

If the MULTistate option is used, the program will perform one CASPT2 calculation for each one of the selected roots, and finally the complete effective Hamiltonian containing the selected states will be solved to obtain the final MS-CASPT2 energies and PM-CASSCF wave functions [12].

The CASPT2 module needs the integral files in $WorkDir and the *RUNFILE* file from the and the *JOBIPH* file from the RASSCF module. The orbitals are saved in the *PT2ORB* file. The new PM-CASSCF wave functions generated in a MS-CASPT2 calculation is saved in the *JOBMIX* file.

6.8.2 CASPT2 - Basic and Most Common Keywords

Keyword	*Meaning*
MULTISTATE	Multi-State CASPT2 calculation: number of roots and roots (Ex. 3 1 2 3)
IMAG	Value for the imaginary shift for the zero order Hamiltonian

6.9 RASSI — A RAS State Interaction Program

Program `RASSI` (RAS State Interaction) computes matrix elements of the Hamiltonian and other operators in a wave function basis, which consists of individually optimized CI expansions from the `RASSCF` program. Also, it solves the Schrödinger equation within the space of these wave functions. There are many possible applications for such type of calculations. The first important consideration to have into account is that `RASSI` computes the interaction among RASSCF states expanding the same set of configurations, that is, having the same active space size and number of electrons.

The `RASSI` program is routinely used to compute electronic transition moments, as it is shown in the Advanced Examples in the calculation of transition dipole moments for the excited states of the thiophene molecule using CASSCF-type wave functions. By default the program will compute the matrix elements and expectation values of all the operators for which `SEWARD` has computed the integrals and has stored them in the *`ONEINT`* file.

RASSCF (or CASSCF) individually optimized states are interacting and non-orthogonal. It is imperative when the states involved have different symmetry to transform the states to a common eigenstate basis in such a way that the wave function remains unchanged. The State Interaction calculation gives an unambiguous set of non-interacting and orthonormal eigenstates to the projected Schrödinger equation and also the overlaps between the original RASSCF wave functions and the eigenstates. The analysis of the original states in terms of RASSI eigenstates is very useful to identify spurious local minima and also to inspect the wave functions obtained in different single-root RASSCF calculations, which can be mixed and be of no help to compare the states.

Finally, the `RASSI` program can be applied in situations when there are two strongly interacting states and there are two very different MCSCF solutions. This is a typical situation in transition metal chemistry when there are many close states associated each one to a configuration of the transition metal atom. It is also the case when there are two close quasi-equivalent localized and delocalized solutions. `RASSI` can provide with a single set of orbitals able to represent, for instance, avoided crossings. `RASSI` will produce a number of files containing the natural orbitals for each one of the desired eigenstates to be used in subsequent calculations.

`RASSI` requires as input files the *`ONEINT`* and *`ORDINT`* integral files and the *`JOBIPH`* files from the `RASSCF` program containing the states which are going to be computed. The *`JOBIPH`* files have to be named consecutively as *`JOB001`*, *`JOB002`*, etc. The input for the `RASSI` module has to contain at least the definition of the number of states available in each of the input *`JOBIPH`* files. Figure 6.11 lists the input file for the `RASSI` program in a calculation including two *`JOBIPH`* files (2 in the first line), the first one including three roots (3 in the first line) and the second five roots (5 in the first line). Each one of the following lines lists the number of these states within each *JOBIPH* file. Also in the input, keyword NATOrb indicates that three files (named sequentially *NAT001*, *NAT002*, and *NAT003*) will be created for the three lowest eigenstates.

Figure 6.11: Sample input requesting the RASSI module to calculate the matrix elements and expectation values for eight interacting RASSCF states

```
 &RASSI
NROFjobiph= 2 3 5; 1 2 3; 1 2 3 4 5
NATOrb= 3
```

6.9.1 RASSI Output

The RASSI section of the Molcas output is basically divided in three parts. Initially, the program prints the information about the *JOBIPH* files and input file, optionally prints the wave functions, and checks that all the configuration spaces are the same in all the input states. In second place RASSI prints the expectation values of the one-electron operators, the Hamiltonian matrix, the overlap matrix, and the matrix elements of the one-electron operators, all for the basis of input RASSCF states. The third part starts with the eigenvectors and eigenvalues for the states computed in the new eigenbasis, as well as the overlap of the computed eigenstates with the input RASSCF states. After that, the expectation values and matrix elements of the one-electron operators are repeated on the basis of the new energy eigenstates. A final section informs about the occupation numbers of the natural orbitals computed by RASSI, if any.

In the Advanced Examples a detailed example of how to interpret the matrix elements output section for the thiophene molecule is displayed. The rest of the output is self-explanatory. It has to be remembered that to change the default origins for the one electron operators (the dipole moment operator uses the nuclear charge centroid and the higher order operators the center of the nuclear mass) keyword CENTer in GATEWAY must be used. Also, if multipoles higher than order two are required, the option MULTipole has to be used in GATEWAY.

The program RASSI can also be used to compute a spin-orbit Hamiltonian for the input CASSCF wave functions as defined above. The keyword AMFI has to be used in SEWARD to ensure that the corresponding integrals are available.

Figure 6.12: Sample input requesting the RASSI module to calculate and diagonalize the spin-orbit Hamiltonian the ground and triplet excited state in water.

```
 &RASSI
NROFjobiph= 2 1 1; 1; 1
Spinorbit
Ejob
```

The first *JOBMIX* file contains the wave function for the ground state and the second file the $^3B_2$ state discussed above. The keyword Ejob makes the RASSI program use the CASPT2 energies which have been written on the *JOBMIX* files in the diagonal of the spin-orbit Hamiltonian. The output of this calculation will give four spin-orbit states and the corresponding transition properties, which can for example be used to compute the radiative lifetime of the triplet state.

6.9.2 RASSI - Basic and Most Common Keywords

Keyword	*Meaning*
NROFJOB	Number of input files, number of roots, and roots for each file
EJOB/HDIAG	Read energies from input file / inline
SPIN	Compute spin-orbit matrix elements for spin properties

6.9.3 CASVB — A non-orthogonal MCSCF program

CASVB is a program for carrying out quite general types of non-orthogonal MCSCF calculations, offering, for example, all the advantages associated with working within a valence bond formalism.

Warning: as for any general MCSCF program, one may experience convergence problems, (*e.g.,* due to redundant parameters), and the non-orthogonal optimization of orbitals can furthermore give linear dependency problems. Several options in CASVB can help overcoming these difficulties.

This program can be used in two basic modes:

a) fully variational optimization

b) representation of CASSCF wavefunctions using overlap- (*relatively inexpensive*) or energy-based criteria.

CASVB executes the following logical steps: Setup of wavefunction information, starting guess generation, one, or several, optimization steps, various types of analysis of the converged solution.

6.9.4 CASVB input

CASVB attempts to define defaults for as many input quantities as possible, so that in the simplest case no input to the CASVB module is required. Sample input for a CASVB calculation on the lowest singlet state of CH_2:

```
 &GATEWAY
coord
3
ch2 molecule
C 0.000000  0.000000 0.000000
H 0.000000  0.892226 0.708554
H 0.000000 -0.892226 0.708554
group= x y; basis= sto-3g
 &SEWARD
 &SCF
 &RASSCF
nactel= 6 0 0; inactive= 1 0 0 0; ras2= 3 1 2 0
lumorb
 &CASVB
```

6.9.5 CASVB output

The amount of output in CASVB depends heavily on the setting of the PRINT levels. In case of problems with convergence behaviour it is recommended to increase these from their rather terse default values.

In the following the main features of the output are outlined, exemplified by the job in the input above. Initially, all relevant information from the previous RASSCF calculation is recovered from the *JOBIPH* interface file, after which the valence bond wavefunction information is summarized, as shown below. Since spatial configurations have not been specified explicitly in this example, a single covalent configuration is chosen as default. This gives 5 spin-adapted VB structures.

```
 Number of active electrons :   6
           active orbitals  :   6
           Total spin       : 0.0
           State symmetry   :   1

 Spatial VB configurations
 -------------------------
     Conf. =>   Orbitals
       1   =>    1  2  3  4  5  6

 Number of VB configurations :     1
           VB structures     :     5
           VB determinants   :    20
```

The output from the following optimization steps summarizes only the most relevant quantities and convergence information at the default print level. For the last optimization step, for example, The output below thus states that the VB wavefunction was found by maximizing the overlap with a previously optimized CASSCF wavefunction (output by the RASSCF program), and that the spin adaptation was done using the Yamanuchi-Kotani scheme. Convergence was reached in 7 iterations.

```
 -- Starting optimization - step  3 --------

 Overlap-based optimization (Svb).

 Optimization algorithm:            dFletch
 Maximum number of iterations:           50
 Spin basis:                         Kotani

 ------------------------------------------
 Optimization entering local region.
 Converged ... maximum update to coefficient:  0.59051924E-06
 Final Svb :    0.9978782695
 Number of iterations used:   7
```

Finally in the output below the converged solution is printed; orbital coefficients (in terms of the active CASSCF MOs) and structure coefficients. The overlap between orbitals are generally of interest, and, as also the structures are non-orthogonal, the structure weights in the total wavefunction. The total VB wavefunction is not symmetry-adapted explicitly (although one may ensure the correct symmetry by imposing constraints on orbitals and structure coefficients), so its components in the various irreducible representations can serve to check that it is physically plausible (a well behaved solution generally has just one non-vanishing component).

Next follows the one-electron density with natural-orbital analysis, again with quantities printed in the basis of the active CASSCF MOs.

```
 Orbital coefficients :
 ----------------------
          1           2           3           4           5           6
   1  0.43397359 -0.43397359 -0.79451779 -0.68987187 -0.79451780 -0.68987186
   2 -0.80889967  0.80889967 -0.05986171 -0.05516284 -0.05986171 -0.05516284
   3  0.00005587 -0.00005587  0.20401015 -0.20582094  0.20401016 -0.20582095
   4  0.39667145  0.39667145  0.00000000  0.00000000  0.00000000  0.00000000
   5 -0.00000001 -0.00000001 -0.53361427 -0.65931951  0.53361425  0.65931952
   6  0.00000000  0.00000000  0.19696124 -0.20968879 -0.19696124  0.20968879

 Overlap between orbitals :
 --------------------------
          1           2           3           4           5           6
   1  1.00000000 -0.68530352 -0.29636622 -0.25477647 -0.29636623 -0.25477647
   2 -0.68530352  1.00000000  0.29636622  0.25477647  0.29636623  0.25477646
   3 -0.29636622  0.29636622  1.00000000  0.81994979  0.35292419  0.19890631
   4 -0.25477647  0.25477647  0.81994979  1.00000000  0.19890634  0.04265679
   5 -0.29636623  0.29636623  0.35292419  0.19890634  1.00000000  0.81994978
   6 -0.25477647  0.25477646  0.19890631  0.04265679  0.81994978  1.00000000

 Structure coefficients :
 ------------------------
      0.00000000  0.00000001  0.09455957  0.00000000 -0.99551921

 Saving VB wavefunction to file VBWFN.

 Saving VB CI vector to file JOBIPH.

 Svb :          0.9978782695
 Evb :        -38.4265149062

 Chirgwin-Coulson weights of structures :
 ----------------------------------------
 VB spin+space (norm   1.00000000) :
      0.00000000  0.00000000 -0.00211737  0.00000000  1.00211737
 VB spin only  (norm   0.38213666) :
      0.00000000  0.00000000  0.00894151  0.00000000  0.99105849

 Symmetry contributions to total VB wavefunction :
 -------------------------------------------------
 Irreps 1 to 4 :  0.10000000E+01  0.15118834E-17  0.17653074E-17  0.49309519E-17

 Energies for components > 1d-10 :
 ---------------------------------
 Irreps 1 to 4 : -0.38426515E+02  0.00000000E+00  0.00000000E+00  0.00000000E+00

 One-electron density :
 ----------------------
          1           2           3           4           5           6
   1  1.98488829 -0.00021330  0.00011757  0.00000000  0.00000000  0.00000000
   2 -0.00021330  1.90209222 -0.00006927  0.00000000  0.00000000  0.00000000
   3  0.00011757 -0.00006927  0.02068155  0.00000000  0.00000000  0.00000000
   4  0.00000000  0.00000000  0.00000000  0.09447774  0.00000000  0.00000000
   5  0.00000000  0.00000000  0.00000000  0.00000000  1.97572540 -0.00030574
   6  0.00000000  0.00000000  0.00000000  0.00000000 -0.00030574  0.02213479

 Natural orbitals :
 ------------------
            1           2           3           4           5           6
   1 -0.99999668  0.00000000  0.00257629  0.00000000  0.00000000  0.00005985
   2  0.00257628  0.00000000  0.99999668  0.00000000  0.00000000 -0.00003681
   3 -0.00005995  0.00000000 -0.00003666  0.00000000 -0.00000001 -1.00000000
   4  0.00000000  0.00000000  0.00000000  1.00000000  0.00000001  0.00000000
   5  0.00000000  0.99999999  0.00000000  0.00000000  0.00015650  0.00000000
   6  0.00000000 -0.00015650  0.00000000 -0.00000001  0.99999999 -0.00000001

 Occupation numbers :
 --------------------
            1           2           3           4           5           6
   1  1.98488885  1.97572545  1.90209167  0.09447774  0.02213475  0.02068154
```

6.9.6 Viewing and plotting VB orbitals

In many cases it can be helpful to view the shape of the converged valence bond orbitals. MOLCAS therefore provides two facilities for doing this. For the Molden program, an interface file is generated at the end of each CASVB run (see also Section 9.1). Alternatively a CASVB run may be followed by RASSCF (Section 8.40) and GRID_IT (Section 8.19) with the VB specification, in order to generate necessary files for viewing with LUSCUS.

6.10 MOTRA — An Integral Transformation Program

Integrals saved by the SEWARD module are stored in the Atomic Orbital (AO) basis. Some programs have their own procedures to transform the integrals into the Molecular Orbital (MO) basis. The MOLCAS MOTRA module performs this task for Configuration Interaction (CI), Coupled- and Modified Coupled-Pair (CPF and MCPF, respectively) and Coupled-Cluster (CC) calculations.

The sample input below contains the MOTRA input information for our continuing water calculation. We firstly specify that the RASSCF module interface file will be the source of the orbitals using the keyword JOBIPH. The keyword FROZEN is used to specify the number of orbitals in each symmetry which will not be correlated in subsequent calculations. This can also be performed in the corresponding MRCI, CPF or CC programs but is more efficient to freeze them here. Virtual orbitals can be deleted using the DELETE keyword.

```
  &MOTRA
 JobIph
 Frozen= 1 0 0 0
```

6.10.1 MOTRA Output

The MOTRA section of the output is short and self explanatory. The integral files produced by SEWARD, *ONEINT* and *ORDINT*, are used as input by the MOTRA module which produces the transformed symbolic files *TRAONE* and *TRAINT*, respectively. In our case, the files are called *water.TraOne* and *water.TraInt*, respectively.

The MOTRA module also requires input orbitals. If the LUMOrb keyword is specified the orbitals are taken from the *INPORB* file which can be any formated orbital file such as *water.ScfOrb* or *water.RasOrb*. The JOBIph keyword causes the MOTRA module to read the required orbitals from the *JOBIPH* file.

6.10.2 MOTRA - Basic and Most Common Keywords

Keyword	*Meaning*
FROZEN	By symmetry: non-correlated orbitals (default: core)
RFPErt	Previous reaction field introduced as a perturbation
LUMORB	Input orbital file as ASCII (INPORB)
JOBIPH	Input orbital file as binary (JOBOLD)

6.11 GUGA — A Configuration Interaction Coupling Coefficients Program

Several of the Configuration Interaction (CI) modules in Molcas use the GUGA module to compute the CI coupling coefficients. We continue our water calculations using the input file shown in the input below. The TITLe keyword behaves in a similar fashion as described in previous modules. There are several compulsory keywords of the GUGA module. The number of electrons to be correlated is specified using the ELECtrons keyword. We are correlating the valence electrons. The spin state is specified using the SPIN keyword.

Sample input requesting the the GUGA module to calculate the coupling coefficients for neutral triplet water in C_{2v} symmetry with six electrons in the active space:

```
 &GUGA
Title= GUGA for C2v Water
Electrons= 8; Spin= 3
Inactive= 1 0 0 0; Active= 2 2 0 1
CIAll= 1
```

The keywords CIALl and REFErence are mutually exclusive. We specify CIALl which will calculate the energy using all possible references functions that can be constructed using the input set of occupation numbers of the active orbitals regardless of the spin coupling (all configurations used to build the corresponding CASSCF wave function). Specific selected references can be chosen using the REFErence keyword. Either the ACTIve or INACtive keyword should be used for a meaningful calculation. The default for both keywords is zero for all symmetries. These keywords function in a similar fashion to these in the RASSCF program module. The INACtive keyword specifies the orbitals that are fully occupied in each symmetry in all the reference functions and the ACTIve keyword specifies the orbitals that may have varying occupations in all references. The selection of INACtive orbitals in the input above is forcing the bonding *sp* hybrid orbital to remain fully occupied in all reference states.

6.11.1 `GUGA` Output

The `GUGA` section of the output lists the possible configurations in the active space. There are nine possible triplet configurations of six electrons in five orbitals. Apart from the various types of orbital in each symmetry the `GUGA` section of the output also gives the number of states that will coupled with various states. There are no input files for the `GUGA` module but the calculated coupling coefficients are stored in ***CIGUGA***.

6.12 MRCI — A Configuration Interaction Program

Multi Reference Single and Doubles Configuration Interaction (MR-SDCI) wave functions are produced by the `MRCI` program module in the Molcas codes. The SDCI keyword requests an ordinary Multi Reference Single and Doubles Configuration Interaction calculation. This is the default and is mutually exclusive with the ACPF keyword which requests an Average Coupled Pair Function calculation. The final keyword, ROOT, specifies the number of the CI root the calculation should compute. The second CI root is the first excited state and since the `GUGA` module has computed the coupling coefficients for a triplet state, the `MRCI` module will converge to the first excited triplet state.

6.12.1 `MRCI` Output

The `MRCI` section of the output lists the number of each type of orbital in each symmetry including pre-frozen orbitals that were frozen by the `GUGA` module. There is a list of the reference configurations with the inactive orbitals included. An empty orbital is listed as '`0`' and a doubly occupied as '`3`'. The spin of a singly occupied orbital by '`1`' (spin up) or '`2`' (spin down). The total number of configuration state functions (CSFs) is listed below the reference configurations.

Sample input requesting the the MRCI module to calculate the first excited MRCI energy for neutral triplet water in C_{2v} symmetry with six electrons in the active space:

```
 &MRCI
Title= MR-SDCI of 2nd CI root of C2v Water
SDCI; Root= 2
```

A listing of the possible CI roots is followed by the CI iteration and convergence information. The Davidson and ACPF corrections are included along with the important CSFs in the CI wave function. The molecular orbitals are listed near the end of the output.

There are four input files to the `MRCI` module; ***CIGUGA*** from `GUGA`, ***TRAONE*** and ***TRAINT*** from `MOTRA` and ***ONEINT*** from `SEWARD`. The orbitals are saved in ***CIORBnn*** where *nn* is the number of the CI root.

6.13 CPF — A Coupled-Pair Functional Program

The CPF program produces Single and Doubles Configuration Interaction (SDCI), Coupled-Pair Functional (CPF), Modified Coupled-Pair Functional (MCPF), and Averaged Coupled-Pair Functional (ACPF) wave functions (see CPF section of the user's guide) from one reference configuration. The difference between the MRCI and CPF codes is that the former can handle Configuration Interaction (CI) and Averaged Coupled-Pair Functional (ACPF) calculations with more than one reference configuration. For a closed-shell reference the wave function can be generated with the SCF program. In open-shell cases the RASSCF has to be used.

The TITLE keyword behaviors in a similar fashion to the other MOLCAS modules. The CPF keyword requests an Coupled-Pair Functional calculation. This is the default and is mutually exclusive with keywords MCPF, ACPF, and SDCI which request different type of calculations. The input below lists the input files for the GUGA and CPF programs to obtain the MCPF energy for the lowest triplet state of B_2 symmetry in the water molecule. The GUGA module computes the coupling coefficients for a triplet state of the appropriate symmetry and the CPF module will converge to the first excited triplet state. One orbital of the first symmetry has been frozen in this case (core orbital) in the MOTRA step.

6.13.1 CPF Output

The CPF section of the output lists the number of each type of orbital in each symmetry including pre-frozen orbitals that were frozen by the GUGA module. After some information concerning the total number of internal configurations used and storage data, it appears the single reference configuration in the MRCI format: an empty orbital is listed as '0' and a doubly occupied as '3'. The spin of a singly occupied orbital by '1' (spin up) or '2' (spin down). The molecular orbitals are listed near the end of the output.

Sample input requested by the GUGA and CPF modules to calculate the ACPF energy for the lowest B_1 triplet state of the water in C_{2v} symmetry:

```
 &GUGA
Title= H2O molecule. Triplet state.
Electrons= 8; Spin= 3
Inactive= 2 0 1 0; Active= 1 1 0 0
CiAll= 2
 &CPF
Title= MCPF of triplet state of C2v Water
MCPF
```

There are four input files to the CPF module; *CIGUGA* from GUGA, *TRAONE* and *TRAINT* from MOTRA and *ONEINT* from SEWARD. The orbitals are saved in *CPFORB*.

6.14 CCSDT — A Set of Coupled-Cluster Programs

The MOLCAS program CCSDT computes Coupled-Cluster Singles Doubles, CCSD, and Coupled-Cluster Singles Doubles and Non-iterative Triples Correction CCSD(T) wave functions for restricted single reference both closed- and open-shell systems.

In addition to the *ONEINT* and *ORDINT* integral files (in non-Cholesky calculations), the CCSDT code requires the *JOBIPH* file containing the reference wave function (remember that it is not possible to compute open-shell systems with the SCF program) and the transformed two-electron integrals produced by the MOTRA module and stored in the *TRAINT* file.

Previously to execute the CCSDT module, wave functions and integrals have to be prepared. First, a RASSCF calculation has to be run in such a way that the resulting wave function has one single reference. In closed-shell situations this means to include all the orbitals as inactive and set the number of active electrons to zero. Keyword OUTOrbitals followed by the specification CANOnical must be used in the RASSCF input to activate the construction of canonical orbitals and the calculation of the CI-vectors on the basis of the canonical orbitals. After that the MOTRA module has to be run to transform the two-electron integrals using the molecular orbitals provided by the RASSCF module. The files *JOBIPH* or *RASORB* from the RASSCF calculation can be used directly by MOTRA using the keywords JOBIph or LUMOrb in the MOTRA input. Frozen or deleted orbitals can be introduced in the transformation step by the proper options in the MOTRA input.

6.14.1 CCSDT Outputs

The section of the Molcas output corresponding to the CC program is self explanatory. The default output simply contains the wave function specifications from the previous RASSCF calculation, the orbital specifications, the diagonal Fock matrix elements and orbital energies, the technical description of the calculation, the iterations leading to the CCSD energy, and the five largest amplitudes of each type, which will help to evaluate the calculation. If triples excitations have been required the description of the employed method (from the three available) to compute perturbatively the triple excited contributions to the CC energy, the value of the correction, and the energy decomposition into spin parts will be available.

6.14.2 Example of a CCSD(T) calculation

Figure 6.13 contains the input files required by the SEWARD, SCF, RASSCF, MOTRA and CCSDT programs to compute the ground state of the HF^{+} cation. molecule, which is a doublet of Σ^{+} symmetry. A more detailed description of the different options included in the input of the programs can be found in the CCSDT section of the user's guide. This example describes how to calculate CCSD(T) energy for HF(+) cation. This cation can be safely represented by the single determinant as a reference function, so one can assume that CCSD(T) method will be suitable for its description.

The calculation can be divided into few steps:

1. Run SEWARD to generate AO integrals.
2. Calculate the HF molecule at the one electron level using SCF to prepare an estimate of MO for the RASSCF run.
3. Calculate HF(+) cation by subtracting one electron from the orbital with the first symmetry. There is only one electron in one active orbital so only one configuration is created. Hence, we obtain a simple single determinant ROHF reference.

4. Perform MO transformation exploiting `MOTRA` using MO coefficients from the `RASSCF` run.

5. Perform the Coupled Cluster calculation using `CCSDT` program. First, the data produced by the programs `RASSCF` and `MOTRA` need to be reorganized, then the CCSD calculation follows, with the chosen spin adaptation being T2 DDVV. Finally, the noniterative triple excitation contribution calculation is following, where the CCSD amplitudes are used.

This is an open shell case, so it is suitable to choose CCSD(T) method as it is defined by Watts *et al.* [13]. Since CCSD amplitudes produced by previous `CCSD` run are partly spin adapted and denominators are produced from the corresponding diagonal Fock matrix elements, final energy is sometimes referred as SA1 CCSD$(T)_d$ (see [14]).

Figure 6.13: Sample input containing the files required by the SEWARD, SCF, RASSCF, MOTRA, CCSORT, CCSD, and CCT3 programs to compute the ground state of the HF^+ cation.

```
 &SEWARD &END
Title= HF molecule
Symmetry
X Y
Basis set
F.ANO-S-VDZ
F      0.00000   0.00000   1.73300
End of basis
Basis set
H.ANO-S-VDZ
H      0.00000   0.00000   0.00000
End of basis
End of input
 &SCF
 &RASSCF
Title= HF(+) cation
OUTOrbitals= Canonical
Symmetry= 1; Spin= 2
nActEl= 1 0 0; Inactive= 2 1 1 0; Ras2= 1 0 0 0
LumOrb; OUTOrbitals= Canonical
 &MOTRA; JobIph; Frozen= 1 0 0 0
 &CCSDT
Iterations= 50; Shift= 0.2,0.2; Accuracy= 1.0d-7
Denominators= 2; Extrapolation= 5,4
Adaptation= 1; Triples= 3; T3Denominators= 0
```

`RASSCF` calculates the HF ionized state by removing one electron from the orbital in the first symmetry. Do not forget to use keyword CANONICAL. In the `CCSDT` run, the number of iterations is limited to 50. Denominators will be formed using orbital energies. (This corresponds to the chosen spin adaptation.) Orbitals will be shifted by 0.2 au, what will accelerate the convergence. However, final energy will not be affected by the chosen type of denominators and orbital shifts. Required accuracy is 1.0d-7 au. for the energy. T2 DDVV class of CCSD amplitudes will be spin adapted. To accelerate the convergence, DIIS procedure is exploited. It will start after 5th iteration and the last four iterations will be taken into account in each extrapolation step.

In the triples step the CCSD(T) procedure as defined by Watts *et al.* [13] will be performed. Corresponding denominators will be produced using diagonal Fock matrix elements.

6.14.3 CCSDT - Basic and Most Common Keywords

Keyword	*Meaning*
CCSD	Coupled-cluster singles and doubles method
CCT	CCSD plus a non iterative triples (T) calculation

6.15 ALASKA and SLAPAF: A Molecular Structure Optimization

One of the most powerful functions of *ab initio* calculations is geometry predictions. The minimum energy structure of a molecule for a given method and basis set is instructive especially when experiment is unable to determine the actual geometry. MOLCAS performs a geometry optimization with analytical gradients at the SCF or RASSCF level of calculation, and with numerical gradients at the CASPT2 level.

In order to perform geometry optimization an input file must contain a loop, which includes several calls: calculation of integrals (`SEWARD`), calculation of energy (`SCF`, `RASSCF`, `CASPT2`), calculation of gradients (`ALASKA`), and calculation of the new geometry (`SLAPAF`).

This is an example of such input

```
  &GATEWAY
   coord= file.xyz
   basis= ANO-S-MB
>> EXPORT MOLCAS_MAXITER=25
>> Do While  <<
  &SEWARD
  &SCF
  &SLAPAF
>> EndDo <<
```

The initial coordinates will be taken from xyz file file.xyz, and the geometry will be optimized at the SCF level in this case. After the wave function calculation, calculation of gradients is required, although code `ALASKA` is automatically called by MOLCAS. `SLAPAF` in this case required the calculation of an energy minimum (no input). Other options are transition states (TS), minimum energy paths (MEP-SEARCH), etc The loop will be terminated if the geometry converges, or maximum number of iterations (MaxIter) will be reached (the default value is 50).

There are several EMIL commands (see sect 7.3.2) , which can be used to control geometry optimization. For example, it is possible to execute some MOLCAS modules only once:

```
>> IF ( ITER = 1 )
* this part of the input will be executed only during the first iteration
>> ENDIF
```

Program `SLAPAF` is tailored to use analytical or numerical gradients produced by `ALASKA` to relax the geometry of a molecule towards an energy minimum (default, no input required then) or a transition state. The program is also used for finding inter state crossings (ISC), conical intersections (CI), to compute reaction paths, intrinsic reaction coordinate (IRC) paths, etc.

6.15.1 SLAPAF - Basic and Most Common Keywords

Keyword	*Meaning*
TS	Computing a transition state
FindTS	Computing a transition state with a constraint
MEP-search	Computing a steepest-descent minimum reaction path
ITER	Number of iterations
INTErnal	Definition of the internal coordinates

6.16 MCKINLEY — A Program for Integral Second Derivatives

`MCKINLEY` computes the analytic second derivatives of the one- and two-electron integrals with respect to the nuclear positions at the SCF and CASSCF level of theory. The differentiated integrals can be used by program `MCLR` to performs response calculations on single and multiconfigurational SCF wave functions. One of the basic uses of `MCKINLEY` and `MCLR` is to compute analytical hessians (vibrational frequencies, IR intensities, etc). Note that `MCKINLEY` for a normal frequency calculations will automatically start the `MCLR` module! For all other methods a numerical procedure is automatically invoked by `MCKINLEY` to compute the vibrational frequencies.

6.16.1 MCKINLEY - Basic and Most Common Keywords

Keyword	*Meaning*
PERTurbation	Suboptions Geometry (for geometry optimizations) or Hessian (full Hessian)

6.17 MCLR — A Program for Linear Response Calculations

MCLR computes response calculations on single and multiconfigurational SCF wave functions. One of the basic uses of MCKINLEY and MCLR is to compute analytical hessians (vibrational frequencies, IR intensities, etc). MCLR can also calculate the Lagrangian multipliers for a MCSCF state included in a state average optimization and construct the effective densities required for analytical gradients of such a state. The use of keyword RLXROOT in the RASSCF program is required. In both cases the explicit request of executing the MCLR module is not required and will be automatic. We postpone further discussion about MCLR to section 6.15.

It follows an example of how to optimize an excited state from a previous State-Average (SA) CASSCF calculation.

```
 &GATEWAY
Title= acrolein minimum optimization in excited state 2
Coord=$MOLCAS/Coord/Acrolein.xyz
*$
Basis= sto-3g
Group=NoSym
>>> Do while
 &SEWARD
 &RASSCF
Title= acrolein
Spin= 1; nActEl= 6 0 0; Inactive= 12; Ras2= 5
CiRoot= 3 3 1
Rlxroot= 2
 &SLAPAF
>>> EndDo
```

The root selected for optimization has been selected here with the keyword RLXROOT in RASSCF, but it is also possible to select it with keyword SALA in MCLR.

Now if follows an example as how to compute the analytical hessian for the lowest state of each symmetry in a CASSCF calculation (SCF, DFT, and RASSCF analytical hessians are also available).

```
 &GATEWAY
Title=p-benzoquinone anion. Casscf optimized geometry.
Coord = $MOLCAS/Coord/benzoquinone.xyz
Basis= sto-3g
Group= X Y Z
 &SEWARD
 &RASSCF
TITLE=p-benzoquinone anion. 2B3u state.
SYMMETRY=2; SPIN=2; NACTEL=9 0 0
INACTIVE=8  0  5  0  7  0  4  0
RAS2    =0  3  0  1  0  3  0  1
 &MCKINLEY; Perturbation=Hessian
```

The MCLR is automatically called after MCKINLEY and it is not needed in the input.

6.17.1 MCLR program - Basic and Most Common Keywords

Keyword	*Meaning*

SALA	Root to relax in geometry optimizations
ITER	Number of iterations

6.18 GENANO — A Program to Generate ANO Basis Sets

GENANO is a program for determining the contraction coefficients for generally contracted basis sets. They are determined by diagonalizing a density matrix, using the eigenvectors (natural orbitals) as the contraction coefficients, resulting in basis sets of the ANO (Atomic Natural Orbitals) type. The program can be used to generate any set of atomic or molecular basis functions. Only one or more wave functions (represented by formated orbital files) are needed to generate the average density matrix. These natural orbital files can be produced by any of the wave function generators, as it is described in section 8.17 of the user's guide. As an illustrative example, in the Advanced Examples there is an example of how to generate a set of molecular basis set describing Rydberg orbitals for the benzene molecule. The reader is referred to this example for more details.

The GENANO program requires several input files. First, one *ONEINT* file generated by the SEWARD module for each input wave function. The files must be linked as *ONE001*, *ONE002*, etc. If the wave functions correspond to the same system, the same *ONEINT* file must be linked with the corresponding names as many times as wave functions are going to be treated. Finally, the program needs one file for wave function containing the formated set of natural orbitals. The files must be linked as *NAT001*, *NAT002*, etc.

The input file for module GENANO contains basically three important keywords. CENTER defines the atom label for which the basis set is to be generated. The label must match the label it has in the SEWARD. SETS keyword indicates that the next line of input contains the number of sets to be used in the averaging procedure and WEIGHTS defines the relative weight of each one of the previous sets in the averaging procedure. Figure 6.14 lists the input file required by the GENANO program for making a basis set for the oxygen atom. Three natural orbital files are expected, containing the natural orbitals for the neutral atom, the cation, and the anion.

Figure 6.14: Sample input requesting the GENANO module to average three sets of natural orbitals on the oxygen atom.

```
  &GENANO
 Title= Oxygen atom basis set: O/O+/O-
 Center= O
 Sets= 3
 Weights= 0.50 0.25 0.25
```

As output files GENANO provides the file *ANO*, containing the contraction coefficient matrix organized such that each column correspond to one contracted basis function, and the file *FIG*, which contains a PostScript figure file of the obtained eigenvalues. The output of GENANO is self-explanatory.

6.19 FFPT — A Finite Field Perturbation Program

Many molecular properties of wave functions can be computed using the FFPT program module in MOLCAS. It adds the requested operator to the integrals computed by the SEWARD module. This must be done before the MOLCAS module calculating the required wave function is requested so the FFPT module is best run directly after the SEWARD module.

The TITLE keyword behaviors in a similar fashion to other MOLCAS modules. The sample input below contains the FFPT input requesting that the dipole moment operator be added to the integrals using the DIPOLE keyword. The size and direction is specified using the COMP keyword which accepts free format input. We can compute the dipole of the molecule by numerical determination of the gradient of the energy curve determined for several values of the dipole operator. From the second derivative we can obtain the polarizability component.

Sample input requesting the FFPT module to include a dipole moment operator in the integral file:

```
 &FFPT
Title= Finite Perturbation with a dipole in the x negative of strength 0.1 au
FFPT
Dipole
 Comp
 X -0.1
```

6.19.1 FFPT Output

The FFPT section of the output is short and self explanatory. The *ONEINT* file is updated with the requested operator.

6.20 VIBROT — A Program for Vibration-Rotation on Diatomic Molecules

The program VIBROT computes vibration-rotation spectra for diatomic molecules. As input it uses a potential curve computed pointwise by any of the wave function programs. It does not require other input file from any of the MOLCAS programs, just its standard input file.

In the Advances Examples the reader will find an overview of the input and output files required by VIBROT and the different uses of the program on the calculation of the electronic states of the C_2 molecule. The reader is referred to section 8.48 of the user's guide for a detailed description of the program.

6.21 SINGLE_ANISO — A Magnetism of Complexes Program

The program SINGLE_ANISO calculates nonperturbatively the temperature- and field- dependent magnetic properties (Van Vleck susceptibility tensor and function, molar magnetization vector and function) and the pseudospin Hamiltonians for Zeeman interaction (the g tensor and higher rank tensorial components) and the zero-field splitting (the D tensor and higher rank tensorial components) for arbitrary mononuclear complexes and fragments on the basis of ab initio spin-orbit calculations. SINGLE_ANISO requires as input file the *RUNFILE* containing all necessary *ab initio* information: spin orbit eigenstates, angular momentum matrix elements, the states been mixed by the spin-orbit coupling in RASSI, etc. Usually, the SINGLE_ANISO runs after RASSI.

For a proper spin-orbit calculation the relativistic basis sets should be used for the whole calcualtion. For SEWARD, the atomic mean-field (AMFI), Douglas-Kroll (DOUG) must be employed. To ensure the computation of angular momentum integrals the ANGMOM should be also used, specifying the origin of angular momentum integrals as the coordinates of the magnetic center of the molecule, i.e. the coordinates of the atom where the unpaired electrons mainly reside. For program RASSI the necessary keywords are: SPIN, since we need a spin-orbit coupling calculation, and MEES, to ensure the computation of angular momentum matrix elements in the basis of spin-free states (SFS).

In the cases where spin-orbit coupling has a minor effect on the low-lying energy spectrum (most of the isotropic cases: Cr^{3+}, Gd^{3+}, etc.) the pseudospin is usually the same as the ground spin. For these cases the SINGLE_ANISO may run without specifying any keywords in the input file.

```
&SINGLE_ANISO
```

In the cases when spin-orbit coupling play an important role in the low-lying energy spectrum, i.e. in the cases of e.g. octahedral Co^{2+}, most of the lanthanide complexes, the pseudospin differs strongly from the spin of the ground state. In these cases, the dimension of the pseudospin can be found by analysing the spin-orbit energy spectrum obtained at RASSI. The pseudospin is best defined as a group of spin-orbit states close in energy. Once specified, these eigenstates are further used by the SINGLE_ANISO to build proper pseudospin eigenfunctions. As an example of an input for SINGLE_ANISO requiring the computation of all magnetic properties (which is the default) and the computation of the g tensor for the ground Kramers doublet (i.e. pseudospin of a Kramers doublet is $\tilde{S}$=1/2).

```
&SINGLE_ANISO
 MLTP
 1
 2
```

SINGLE_ANISO has implemented pseudospins: $\tilde{S}$=1/2, $\tilde{S}$=1, ..., up to $\tilde{S}$=7/2. The user can also ask for more pseudospins at the same time:

```
&SINGLE_ANISO
 MLTP
 3
 2 4 2
```

For the above input example, the SINGLE_ANISO will compute the g tensor for the ground Kramers doublet (spin-orbit states 1 and 2), the g tensor, ZFS tensor and coefficients of higher rank ITO for the pseudospin $\tilde{S}$=3/2 (spin orbit functions 3-6), and the g tensor for the third excited Kramers doublet (spin orbit functions 7 and 8).

6.21.1 SINGLE_ANISO Output

The SINGLE_ANISO section of the Molcas output is divided in four parts. In the first part, the g tensor and higher rank Zeeman tensors are computed. They are followed by D tensor and higher rank ZFS tensors. The program also computes the angular moments in the direction of the main magnetic axes.

In the second part, the paramaters of the crystal field acting on the ground atomic multiplet of lanthanides are calculated.

In the third part, the powder magnetic susceptibility is printed, followed by the magnetic susceptibility tensors with and without intermolecular interaction included.

In the fourth part, magnetization vectors (if required) are printed, and then the powder molar magnetization calculated for the TMAG temperature.

The keywords TINT and HINT control the temperature and field intervals for computation of magnetic susceptibility and molar magnetization respectively. Computation of the magnetic properties at the experimental temperature and field points with the estimation of the standard deviation from experiment is also possible via TEXP, defining the experimental temperature and measured magnetic susceptibility and HEXP, defining the experimental field and averaged molar magnetization.

```
&SINGLE_ANISO
TITLE
g tensor and magnetic susceptibility
TYPE
4
MLTP
2
3 3
TINT
0.0 100 101 0.001
```

The above input requires computation of the parameters of two pseudospins $\tilde{S}$=1: the ground (spin-orbit functions 1-3) and first excited (spin-orbit functions 4-6) and the magnetic susceptibility in 101 steps equally distributed in the temperature domain 0.0-100.0 K.

6.21.2 SINGLE_ANISO - Basic and Most Common Keywords

Keyword	*Meaning*
MLTP	Specifies the number and dimension of the pseudospins Hamiltonians
TMAG	Sets the temperature for the computation of molar magnetization
MVEC	Number and radial coordinates of directions for which the magnetization vector will be computed

6.22 GRID_IT: A Program for Orbital Visualization

GRID_IT is an interface program for calculations of molecular orbitals and density in a set of cartesian grid points. Calculated grid can be visualized by separate program LUSCUS in the form of isosurfaces.

GRID_IT generates the regular grid and calculates amplitudes of molecular orbitals in this net. Keywords Sparse,Dense, Npoints specify the density of the grid. And keywords ORange (occupation range), ERange (energy range), Select allow to select some specific orbitals to draw.

As default GRID_IT will use grid net with intermediate quality, and choose orbitals near HOMO-LUMO region. Note, that using keyword All - to calculate grids for all orbitals or Dense - to calculate grid with very high quality you can produce a very huge output file.

GRID_IT requires the communication file *RUNFILE*, processed by GATEWAY and any formated *INPORB* file: *SCFORB*, *RASORB*, *PT2ORB*, generated by program SCF, RASSCF, or CASPT2, respectively. The output file *M2MSI* contains the graphical information.

Normally you do not need to specify any keywords for GRID_IT: the selection of grid size, as well as the selection of orbitals done automatically.

An input example for GRID_IT is:

```
 &GRID_IT
Dense
* compute orbitals from 20 to 23 form symmetry 1 and orbital 4 from symmetry 2
SELECT
1:20-23,2:4
```

GRID_IT can be run in a sequence of other computational codes (if you need to run GRID_IT several times, you have to rename grid file by using EMIL command, or by using keyword NAME)

```
&GATEWAY
 ...
&SEWARD
&SCF
&GRID_IT
NAME=scf
&RASSCF
&GRID_IT
NAME=ras
```

or, you can run GRID_IT separately, when the calculation has finished.

```
&GATEWAY
&GRID_IT
FILEORB=/home/joe/project/water/water.ScfOrb
```

This is quite important to understand that the timing for GRID_IT, and the size of generated grid file depends dramatically on the targeting problem. To get a printer quality pictures you have to use Dense grid, but in order to see the shape of orbitals - low quality grids are much more preferable.

The following table illustrates this dependence:

C_{24} molecule, 14 orbitals.

Keywords	Time (sec)	filesize	picture quality
Dense, ASCII	188	473 Mb	best
Dense	117	328 Mb	best
Dense, Pack	117	41 Mb	below average
Default (no keywords)	3	9 Mb	average
Pack	3	1.4 Mb	average
Sparse	1.3	3 Mb	poor
Sparse, Pack	1.3	620 Kb	poor

6.22.1 GRID_IT - Basic and Most Common Keywords

Keyword	*Meaning*
ASCII	Generate the ***grid*** file in ASCII (e.g. to transfer to another computer), can be only used in combitation with NoLUSCUS
ALL	Generate all orbitals
SELECT	Select orbitals to compute

6.23 Writing MOLDEN input

By default the `GUESSORB`, `SCF`, `MBPT2`, `RASSCF`, `SLAPAF`, `LOCALISATION`, and `MCLR` modules generate input in Molden format. The `SCF`, `MBPT2`, `RASSCF`, and `LOCALISATION` modules generate input for molecular orbital analysis, `SLAPAF` for geometry optimization analysis, minimum energy paths, Saddle optimization paths and IRC TS analysis, and the `MCLR` module generates input for analysis of harmonic frequencies. Molden files can be visualized by `GV` or by `MOLDEN` (http://www.caos.kun.nl/˜schaft/molden/molden.html).

The generic name of the input file and the actual name are different for the nodes as a reflection on the data generated by each module. Hence, the actual names (generic name) for the Molden files in each module are

- GUESSORB module: $Project.guessorb.molden (MD_GSS)
- SCF module: $Project.scf.molden (MD_SCF)
- MBPT2 module: $Project.mp2.molden (MD_MP2)
- RASSCF module: $Project.rasscf.molden (MD_CAS) for the state-averaged natural orbitals, and $Project.rasscf.x.molden (MD_CAS.x) for the state-specific natural spin orbitals, where x is the index of a CI root.

- SLAPAF module: $Project.geo.molden (MD_GEO) for geometry optimizations, $Project.mep.molden (MD_MEP) for minimum energy paths, $Project.irc.molden (MD_IRC) for IRC analysis of a TS, and $Project.saddle.molden (MD_SADDLE) for Saddle method TS optimizations.
- LOCALISATION module: $Project.local.molden (MD_LOC)
- MCLR module: $Project.freq.molden (MD_FREQ)

6.24 Most frequent error messages found in MOLCAS

Due to the large number of systems where the Molcas package is executed and the large number of options included in each of the programs it is not possible to compile here all the possible sources of errors and error messages occurring in the calculations. The Molcas codes contain specific error message data basis where the source of the error and the possible solution is suggested. Unfortunately it is almost impossible to cover all the possibilities. Here the user will find a compendium of the more usual errors showing up in Molcas and the corresponding error messages.

Many of the error messages the user is going to obtain are specific for the operative system or architecture being used. The most serious ones are in most of cases related with compiler problems, operative system incompatibilities, etc. Therefore the meaning of this errors must be checked in the proper manuals or with the computer experts, and if they are characteristic only of Molcas, with Molcas authors. The most common, however, are simple mistakes related to lack of execution or reading permission of the shell scripts, Molcas executable modules, etc.

In the following the most usual errors found in Molcas are listed.

- The shell is unable to find the command molcas. The message in this case is, for instance:

  ```
  molcas:  not found
  ```

  The solution is to add into the PATH the location of molcas driver script.

- If the Molcas environment is not properly installed the first message showing up in the default error file is:

  ```
  ***
  *** Error: Could not find molcas driver shell
  *** Currently MOLCAS=
  ```

  Typing a command molcas, you can check which molcas installation will be used. Check the value of the variable MOLCAS, and define it in order to point to the proper location of molcas installation.

- Environment is not defined

  An attempt to run an executable without molcas driver scripts gives an error:

  ```
  Usage: molcas module_name input
  ```

- A call for a program can find problems like the three following ones:

  ```
  Program NNNN is not defined
  ```

  An error means that requested module is missing or the package is not installed.

- When the input file required for a Molcas program is not available, the program will not start at all and no output will be printed, except in the default error file where the following error message will appear:

  ```
   Input file specified for run subcommand not found : seward
  ```

- All the codes communicate via file *RUNFILE*, if for a some reason the file is missing or corrupted, you will get an error

  ```
  ***    Record not found in runfile
  ```

  The simple solution - restart seward to generate proper *RUNFILE*

- All the codes need integral files generated by SEWARD in files *ONEINT* and *ORDINT*. Even the direct codes need the one-electron integrals stored in *ONEINT*. The most common problem is then that a program fails to read one of this files because SEWARD has not been executed or because the files are read in the wrong address. Some of the error messages found in those cases are listed here.

  In the SCF module, the first message will appear when the one-electron integral file is missing and the second when the two-electron integral file is missing:

  ```
  Two-electron integral file was not found!
   Try keyword DIRECT in SEWARD.
  ```

- Molcas use dynamical allocation of memory for temporary arrays. An error message 'Insufficient memory' means that requested value is too small - you have to specify MOLCAS_MEM variable and restart your calculation.

- if user ask to allocate (via MOLCAS_MEM) an amount of memory, which is large than possible on this computer, the following error message will be printed.

  ```
  MA error: MA_init: could not allocate 2097152152 bytes
  The initialization of the memory manager failed ( iRc=  1 ).
  ```

- An improper input (e.g. the code expects to read more numbers, than user specified in input file) will terminate the code with errorcode 112.

- Input/Output (I/O) problems are common, normally due to insufficient disk space to store the two-electron integral files or some of the intermediate files used by the programs. The error message would depend on the operative system used. An example for the SCF is shown below:

  ```
   ***************************************************************************
   ***************************************************************************
   ***                                                                     ***
   ***                                                                     ***
   ***    Location: AixRd                                                  ***
   ***    File: ORDINT                                                     ***
   ***                                                                     ***
   ***                                                                     ***
  ```

```
***    Premature abort while reading buffer from disk:                     ***
***    Condition: rc != LenBuf                                             ***
***    Actual   :                0!=          262144                       ***
***                                                                        ***
***                                                                        ***
*******************************************************************************
*******************************************************************************
```

The error indicates that the file is corrupted, or there is a bug in the code.

- Sometimes you might experience the following problem with GEO/HYPER run:

```
 Quaternion tested
 mat. size =     4x    1
  -0.500000000
   0.500000000
  -0.500000000
   0.000000000
###############################################################################
###############################################################################
###                                                                         ###
###                                                                         ###
###    Location: CheckQuater                                                ###
###                                                                         ###
###                                                                         ###
###                                                                         ###
###    Quaternion does not represent a rotation                             ###
###                                                                         ###
###                                                                         ###
###############################################################################
###############################################################################
```

The error indicates that you need to rearrange the Cartesian coordinates (atoms) of one or another fragment.

6.25 Tools for selection of the active space

Selecting an active space is sometimes easy. For a small molecule, an active space for the ground and the lowest valence excited states is usually the valence orbitals, i.e. orbitals composed of atomic orbitals belonging to the usual 'valence shells' (there are some exceptions to this rule). Problems arise for medium or large molecules, for higher excited states, and for molecules including transition, lanthanide or actinide elements. A good wish list of orbitals will give a CASSCF/CASPT2 calculation that demand unrealistically large computer resources and time. Compromises must be made. Any smaller selection of active orbitals can in general affect your results, and the selection should be based on the specific calculations: see 6.26 for advise.

The following three tools may be help in the process:

localisation is a program that can take a (subrange of) orbitals from an orbital file, and produce a new orbital file where these orbitals have been transformed to become localized, while spanning the same space as the original ones.

expbas can take an orbital file using a smaller basis set, and 'expand' it into a new orbital file using a larger basis.

LUSCUS (is of course also described elsewhere) is the orbital viewer.

It is of course best to have a good perception of the electronic structure of the molecule, including all states of interest for the calculation. If it is a larger system, where lots of ligands can be assumed not to partake in non-dynamic correlation, it is a good idea to run some simple exploratory calculations with a much smaller model system. Check the literature for calculations on similar systems or model systems.

First of all, you need to know how many orbitals (in each symmetry) that should be active. Their precise identity is also good to know, in order to have a good set of starting orbitals, but we come to that later. **Necessary** active orbitals are: Any shells that may be open in any of the states or structures studied. Breaking a bond generally produces a correlated bond orbital and a correlating antibonding orbital, that must both be active (Since it is the **number** of orbitals we are dealing with as yet, you may as well think of the two radical orbitals that are produced by completely breaking the bond). You probably want to include one orbital for each aromatic carbon. **Valuable correlated** active orbitals are: Oxygen lone pair, CC π bonds. **Valuable correlating** active orbitals are: the antibonding π^* CC orbitals, and one additional set of correlating d orbitals for most transition elements (sometimes called the 'double d-shell effect').

The valuable correlated orbitals can be used as Ras-1 orbitals, and correlating ones can be used as Ras-3 orbitals, if the active space becomes too large for a casscf calculation.

Assuming we can decide on the number of active orbitals, the next task is to prepare starting orbitals that enables CASSCF to converge, by energy optimization, to the actual starting orbitals for your calculation. Use a very small basis set to begin with: This will usually be one of the minimal bases, e.g. ANO-S-MB. This is not just to save time: the small basis and the large energy spacings make it much easier to get well-defined correlating orbitals.

Performing the actual casscf (or rasscf) calculation may give you the active space you want: Viewing the orbitals by `LUSCUS` may confirm this, but very often the orbitals are too mixed up (compared to ones mental picture of what constitutes the best orbitals). Using localisation program solves this problem. In order to localise without mixing up orbitals from different subspaces may require to produce the new orbital file through several runs of the program; however, for the present perpose, it may be best not to have so very strict restrictions, for example: Allow mixing among a few high inactive and the most occupied orbitals; and also among the weakly occupied and some virtual orbitals.

Running the localisation program, and viewing the localised orbitals, is a great help since directly in `LUSCUS` one can redefine orbitals as being inactive, or ras3 , or whatever, to produce a new orbital file. The resulting annotated localised orbitals can be used in a new run.

Once a plausible active space has been found, use the expbas tool to obtain starting orbitals using, e.g. ANO-VDZP basis, or whatever is to be used in the bulk of the production run.

It is also a good idea to, at this point, 'waste' a few resources on a single-point calculation for a few more states than you are really interested in, and maybe look at properties, etc. There may be experimental spectra to compare with.

And please have a look at the following section, 'Some practical HINTS' 6.26.

6.26 Some practical HINTS

This section contains a collection of practical hints and advices how to use MOLCAS in solving quantum chemistry problems.

6.26.1 GATEWAY/SEWARD program:

- Try the Cholesky approximation (or RI)! It saves disk space and possibly time.
- Think about basis set. ANO-like basis sets have many advantages, but they are "marginal".
- Try to avoid inline basis sets, use the library.
- Remember that the quality of basis set should match quality of computational method.
- Use ANO-RCC even for atoms in the 2nd row.
- Be extremely careful when computing anions. Remember that special situations requires special basis sets.
- Use minimal or small basis set for understanding chemical problem. You always can use expbas later..

6.26.2 SCF program

- HF orbitals are in many cases good starting orbitals, but quite often GuessOrb is a better choice.
- Very large basis set together with HF can lead to linear dependences.

- Remember! Hartree-Fock method allows multiple solutions (even for trivial systems)
- Be reasonable selecting convergence thresholds
- UHF convergence is much poor comparing to RHF.
- Sometime you have to use Scramble keyword to break the symmetry.

- DFT convergence can't be better that HF convergence. Think about starting orbitals for DFT.
- Remember that DFT is a powerful method but it is still single configurational method. Don't use it beyond it's limits.
- Choose your favorite functional, and stay with your decision
- Note that MOLCAS is not the best DFT code available

6.26.3 RASSCF program:

- MCSCF are multi-solution methods that heavily depend on the starting orbitals and level of calculation (roots).
- On convergence ALWAYS (ALWAYS, ALWAYS, etc) check the orbitals (LUSCUS, molden, CMOcorr, etc). MCSCF methods will lead to different solutions for active spaces of different nature. Use your chemical intuition and let the calculation guide you.
- Analyze carefully the CI coefficients and natural occupation numbers together with the orbitals (average orbitals are fine in general for that) in order to understand the nature of the states .

- You get average orbitals, and orbitals for individual roots, which you can visualized by `LUSCUS` or molden etc, contain the natural orbitals of the different roots.
- Try increasing the number of SA-CASSCF roots to locate more excited states. They can be low-lying solutions at the CASPT2 level. In high symmetry cases you may also need to consider roots that have high energy at the initial steps and can become lower roots in the converged calculation.
- It is NOT advisable to play games with weights for the different roots. Roots with equal weights make your calculation more clear and reproducible.

- MOLCAS can handle only D_2h subgroups. Molecules with other symmetry (C_{3v}, D_{4d}, T_d, O_h) have a problem. Especially if you use approximations, like CD.
- Work in a symmetry point group that allows degenerate states to belong to the same irreducible representation (e.g. C_2 for linear molecules). Try C_1 too.
- Working in a too high symmetry might prevent you from obtaining less symmetric lowest-lying localized solutions (e.g. Ni^{2+})

- Start with clean symmetric orbitals (GUESSORB). Sometime (for example for a radical), an orbital for positively charged system can be more symmetric.
- use if needed, CLEANUP and SUPSYM, or for linear molecules: the LINEAR keyword.

- use it! RASSCF is a simple way to increase an active space.
- Balance RAS1/3 and RAS2 subspaces. Try to change orbitals between these subspaces.
- Removing RAS2 space completely is not a good idea.
- Note that RAS calculations have a slower convergence, and demand more resources.

- Increase LEVShift parameter in cases of slow or difficult convergence.
- Sometimes RASSCF is very sensitive when is close to convergence. Try restarting the calculation from the previous JOBIPH file
- Try to restart from orbitals (or JOBIPH) instead of starting from scratch.

6.26.4 Selection of active spaces:

- Always compare calculations with the same active space size (and nature if possible).
- Ask yourself first which is your goal. The selection of the active space depends on that.
- If you made a selection once, try to reuse orbitals! Especially for a set of calculations with different geometries
- In ground state calculations many orbitals can have an occupation close to 2 and 0 and might rotate with others in the inactive (secondary) space. It might be wise to skip them
- For low-lying excited states and few roots you might leave inactive quite a number of orbitals. Check with RASSCF for instance.

- SCF orbital energies sometimes help to choose the orbitals by using the energy order criterion, but you must learn to see the problems (like lone pair orbitals having too low energies at the SCF level).
- You typically will need correlating orbitals, that is, if you have a π orbital you need a π^*, the same for σ,σ^*, but not for lone pairs.
- CASSCF/RASSCF geometry optimizations are the worst case. If you miss orbitals you might end up in a totally wrong geometry (e.g. breaking a bond usually requires the bonding and antibonding orbitals in the space).
- Organic (1st row atoms) molecules usually require open shell orbitals, π, π^*, and lone pairs. If 2nd row atoms are added (S, P, Si, etc) s orbitals enter in action (s bonds are longer). CH bonds can often be left be inactive.

- Rydberg states require additional diffuse basis sets and specific orbitals in the active space. Use basis sets centered in the charge center of the positive ion (consult the manual).

- Transition metal chemistry (1st row) sometimes requires a double *d* shell description in the active space.

- Lanthanides have a quite inert $4f$ shell that must be active together with 5d, 6s (6p). Actinides $5f$, $6d$, $7s$.

- **use expbas!** start from minimal basis set, decide the active space, and expand the basis to "normal". With small basis set you can clearly identify orbitals.

- **use localization!** Especially for virtual orbitals.

- **expand active space by adding RAS1/3** - give the system a freedom, and see how it reacts.

6.26.5 CASPT2 program:

- The new IPEA = 0.25 zeroth Hamiltonian is the default. It particularly improves open shell cases. But there are some cases where IPEA=0 gives better correlation with experiment.

- Energy differences between different states or situations are only reliable between calculations with the same active space size and similar reference weights in CASPT2.

- An intruder state (low reference weight in the CASPT2 state) might be informing you that your active space lacks an important orbital. Check the list of large perturbative contributions (small denominators combined with large RHS values; check the output) and also the occupation number of the CASPT2 orbitals.

- For weakly interacting intruder states cases try the IMAGINARY level shift parameter. Don't use the level shift to reach agreement with experiment!

- For heavy valence-Rydberg mixing cases or for closely degenerated CASPT2 states, use MS-CASPT2.

- If the MS-CASPT2 description differs a lot from the CASPT2 one, try to check the calculation by increasing the active space (introducing angular correlation if possible) until the result is converged. The "true" solution is typically between both cases (CASPT2 and MS-CASPT2). If you are suspicious about the MS-CASPT2 result, better keep the CASPT2 one. It has worked out generally well so far.

6.26.6 RASSI program:

- Remember that the program shows first the interaction among the input states and later this description might change. ALWAYS check the changing order of states.

- For spin-orbit coupling calculations don't forget to include the CASPT2 energies as input (EJOB or HDIAG keywords) because the results depend on the energy gap. In other cases having the CASPT2 energies as input will help you to get the right oscillator strength and Einstein coefficient in the final table.
- If you have degenerate states be sure that the CASPT2 energies are degenerate. If they are not (which is common) average the energies for the degenerate set (the two components of E symmetry for example).
- Remember that the spin-orbit coupled results (e.g. TDM) depend on the number of interacting singlet and triplet states included in RASSI.

6.26.7 Geometry optimization

- Not all methods have analytical derivatives.
- Default thresholds in slapaf are typically too tight. Do not waste computer time!
- Use constrained optimization
- For minima on flat hypersurfaces, such in loosely bound fragments, or in slow convergence cases you might have to decrease the CUTOFF threshold in ALASKA
- Be careful with the bond angle definition if you are close to a linear bond. You may have to switch to the LAngle definition
- Don't forget that CASSCF does not include dynamical correlation. In some cases you better change to DFT or numerical CASPT2 optimizations or, if this is not feasible, may be preferable to run RASSCF optimizations
- Poor active spaces may lead you to symmetry broken wrong solutions (e.g. a C_s minimum for water below the true C_{2v} one)
- Poor geometry convergence might be reduced or at least controlled by reducing the initial trust radius with the MAXSTEP keyword or/and by doing the optimization in Cartesian coordinates (CARTESIAN)
- In order to obtain localized solutions it might be a good idea to feed the program with a slightly distorted geometry that helps the method to reach the non symmetric solutions. Other possibilities are to use an electric field, to add a charge far from the system or use a solvent cavity. In all cases you break symmetry and allow less symmetric situations.
- Linearly interpolated internal coordinates geometries may be a good starting point to locate a transition state. Use also the useful FindTS command. Sometimes can be wise to compute a MEP from the TS to prove that it is relevant for the studied reaction path. Try also the new Saddle approach!
- When locating a CASSCF surface crossing (MECP) ALWAYS compute CASPT2 energies at that point. The gap between the states can be large at that level. In severe cases you might have to make a scan with CASPT2 to find a better region for the crossing.

- Remember that (so far) MOLCAS does not search for true conical intersections (CIs) but minimum energy crossing points (MECP) because it lacks NACMEs. Note however that typically computed minimum energy CIs (MECIs) may not be photochemically relevant if they are not easily accessible. Barriers have to be computed. Use MEPs!!

- Numerical hessians and optimizations may lead you to bad solutions when different electronic states are too close. As you move your calculation from the equilibrium geometry some of the points may belong to other state and corrupt your result. This might be the case for numerical CASPT2 crossing search. Use then MS-CASPT2 search.

- Remember that SA-RASSCF analytical gradients and SA-CASSCF analytical hessians are not implemented.

- Be careful with the change of roots and nature along a geometry optimization or MEP. For example, you start with the state in root 3 (at the CASSCF level) and reach a region of crossing root 3 and root 2. You may need to change to root 2 for your state. Not an easy solution (so far).

6.26.8 Solvent effects

- Some effects of the solvent are very specific, such as hydrogen bonds, and require to include explicit solvent molecules. Try adding a first solvent shell (optimized with molecular mechanics for instance) and then a cavity, for instance with PCM.

- Too small cavity sizes can lead you to unphysical solutions, even if they seem to match experiment.

- Remember using NonEquilibrium (final state) and RFRoot (SA-CASSCF) when required

- QM/MM is a much powerful strategy, but it requires experience and knowledge of the field

Part IV

User's Guide

Section 7

The Molcas environment

This section describes how to use the Molcas program system. The reader is assumed to be familiar with the workings of the operating system, and only issues that are Molcas specific will be covered.

7.1 Overview

Molcas contains a set of ab initio quantum chemical programs. These programs are essentially separate entities, but they are tied together by a shell. The exchange of information between the programs is through files. The shell is designed to allow ease of use with a minimum amount of specifications in a 'run of the mill' case. The shell is flexible and allows the user to perform any calculation possible within the limitations of the various codes supplied with Molcas.

To make a calculation using Molcas you have to decide on which programs you need to use, prepare input for these, and construct a command procedure file to run the various programs. This command procedure file is submitted for batch execution. The following two subsections describe the programs available and the files used in Molcas.

7.1.1 Programs in the system

Below is a list of the available programs given. The programs are tied together with a shell and the inter-program information is passed through files. These files are also specified in this list to indicate the program module interdependencies.

Program	*Purpose*
`ALASKA`	This program computes the first derivatives of the one- and two-electron integrals with respect to the nuclear positions. The derivatives are not stored on files, but contracted immediately with the one- and two-electron densities to form the molecular gradients.

`CASPT2` This program computes the second order Many Body Perturbation Theory correction to CASSCF or RASSCF wave function.

`CASVB` This program performs various types of valence bond calculations. It may be called directly (for VB interpretation of CASSCF wave functions), or within the RASSCF program (for fully variational VB calculations). In the former case it requires the information in the *JOBIPH* file generated by the `RASSCF` program, possibly also the integral files *ONEINT* and *ORDINT*.

`CCSDT` This program performs the iterative ROHF CCSD procedure, optionally followed by the (T) calculation contribution. It requires the *JOBIPH* file produced by `RASSCF`, and *TRAONE* and *TRAINT* files produced by `MOTRA`.

`CHCC` This program performs Closed-Shell Coupled-Clusters Singles and Doubles calculations based exclusively on the Cholesky (or RI) decomposed 2-electron integrals.

`CMOCORR` This program compares the orbital spaces of two orbitals files.

`CPF` This program produces a CPF, MCPF or ACPF wave function from a single reference configuration.

`DYNAMIX` This program allows to do molecular dynamics simulations using the velocity Verlet algorithm. It has also the capability to detect nonadiabatic transition using a surface hopping algorithm.

`ESPF` The ElectroStatic Potential Fitted (ESPF) method adds some one-electron operators to the one-electron hamiltonian in order to compute the interaction between the charge distribution and any kind of external electrostatic potential, field, and field derivatives.

`EXPBAS` This program takes one orbital file generated with a smaller basis set (ANO) and expands it to a larger basis set.

`GATEWAY` This program collects all information about computed system, including geometry, basis sets and symmetry, and stores the data for a future use.

`GENANO` This program is used to construct ANO type basis sets.

`GRID_IT` This program calculates densities and molecular orbitals in a set of cartesian grid points, and produce a file for visualisation of MO's and densities.

`FFPT` This program applies perturbations to the one-electron Hamiltonian for finite field perturbation calculations.

`GUGA` This program generates the coupling coefficients required by the `MRCI` and `CPF` programs.

`LOCALISATION` This program generates localised occupied orbitals according to one of the following procedures: Pipek-Mezey, Boys, Edmiston-Ruedenberg, or Cholesky. Orthonormal, linearly independent, local virtual orbitals may also be generated from projected atomic orbitals (Cholesky PAOs).

`LOPROP` This program computes molecular properties based on the one-electron density or transition-density and one-electron integrals like charges, dipole moments and polarizabilities

`MBPT2` This program computes the second order Many Body Perturbation Theory correction to an SCF wave function.

`MCKINLEY` This program calculates the second and first order derivatives of integrals that are used for calculating second order derivatives of the energies with perturbation dependent basis sets.

`MCLR` This program calculates the response of the wave function and related second order properties.

`MOTRA` This program transforms one- and two-electron integrals from AO basis to MO basis. The integrals that are transformed are the one-electron Hamiltonian and the two-electron repulsion integrals.

`MRCI` This program produces a Multi Reference CI wave function from an arbitrary set of reference configurations. Alternatively the program can produce an Averaged CPF wave function.

`MULA` This program calculates intensities of vibrational transitions between electronic states.

`NEMO` A set of computational modules for calculation of interaction energies between molecules.

`RASSCF` This program generates CASSCF, RASSCF and GASSCF type wave functions.

`RASSI` This program computes the interaction between several RASSCF wave functions. A spin-orbit Hamiltonian can be used.

`SCF` This program generates Closed Shell SCF or Unrestricted SCF, and Kohn-Sham DFT wave functions.

`SEWARD` This program generates one- and two-electron integrals needed by other programs. If requested the two-electron integrals are computed in the form of Cholesky decomposed vectors.

`SINGLE_ANISO` This program allows the non-perturbative calculation of effective spin (pseudospin) Hamiltonians and static magnetic properties of mononuclear complexes and fragments completely ab initio, including the spin-orbit interaction.

SLAPAF	This program is a general purpose facility for geometry optimization, transition state search, MEP, conical intersections, intersystem crossings, etc. using analytical or numerical gradients produced by ALASKA.
VIBROT	This program computes the vibrational-rotational spectrum of a diatomic molecule. Spectroscopic constants are computed. The program can also compute transition moments, life times , etc for excited state potentials.

7.1.2 Files in the system

The following is a list of the most common files in Molcas that are used to exchange information between program modules. The names given in this list are the FORTRAN file names, defined in the source code. Actual file names are constructed from so called prgm tables, specific for each individual module. (More information about PRGM files can be found in the Molcas Programming Guide).

File	*Contents*
RUNFILE	This file contains general information of the calculation. All programs read from it and write to it. GATEWAY creates a new *RUNFILE* corresponding to a new calculation.
ONEINT	This file contains the one-electron integrals generated by the program SEWARD.
ORDINT	This file contains the ordered two-electron integrals generated by the program SEWARD.
RYSRW	Data base for the fast direct evaluation of roots and weights of the Rys polynomials. This file is a part of the program system and should not be manipulated by the user.
ABDATA	Data base for the evaluation of roots and weights of high order Rys polynomial. This file is a part of the program system and should not be manipulated by the user.
CHVEC	This file contains the Cholesky vectors representing the two-electron integrals as generated by the program SEWARD.
CHORST	This file contains information about each of the Cholesky vectors generated by the program SEWARD.
CHRED	This file contains information about the storage mode of the Cholesky vectors generated by the program SEWARD.
CHOR2F	File containing the mapping between Cholesky vector storage and the canonical orbital ordering.

TRAINT This file contains the transformed two-electron integrals generated by the program MOTRA.

TRAONE This file contains the transformed one-electron integrals generated by the program MOTRA.

INPORB A generic name for an orbital file. Different programs uses and/or generates *INPORB* files with a specific name:

GSSORB	generated by the program GUESSORB.
SCFORB	generated by the program SCF.
RASORB	generated by the program RASSCF.
CIORB	generated by the program MRCI
CPFORB	generated by the program CPF.
SIORB	generated by the program RASSI.
PT2ORB	generated by the program CASPT2.

JOBIPH This file contains the RASSCF wave function information generated by the RASSCF program.

JOBOLD This file contains the RASSCF wave function information generated by the RASSCF program in the file *JOBIPH*, and is used as input for a subsequent RASSCF calculation.

JOBMIX This file contains the multi-state CASPT2 wave function information generated by the CASPT2 program, and is used as input for a subsequent RASSI calculation.

GRID This file contains binary or ASCII data generated by GRID_IT program for visualization of density or molecular orbitals.

7.2 Commands and environment variables

This section will describe the usage of Molcas in an UNIX environment.

Production jobs using Molcas in an UNIX environment can be performed as batch jobs. This requires the creation of a shell script that contains a few simple commands. Further you need to create input for each program module that you intend to use. This section describes the necessary steps you have to take in order to make a successful job using Molcas. Input examples for a typical Molcas run can be found in *doc/samples/problem_based_tutorials/* directory. Also you can use some input examples in *Test/input* subdirectory.

7.2.1 Commands

There is a command supplied with the Molcas package, named molcas, that the user issue to perform a given task. A sequence of such commands will perform the calculation requested by the user.

Command	*Purpose*
molcas	This command tells which MOLCAS installation will be used, and gives some help about usage of MOLCAS command
molcas *input-file*	This command executes a command in the MOLCAS system.
molcas help PRGM	This command gives the list of available keywords for program PRGM.
molcas help PRGM KEYWORD	This command gives description of a KEYWORD.
molcas help ENVIRONMENT	This command gives a list of MOLCAS specific environment variables.
molcas help BASIS ELEMENT	This command gives a list of basis sets available for an ELEMENT.

The following is an example of running MOLCAS by using a single input file:

```
molcas $Project.input
```

An alternative way of running MOLCAS as a sequence of separate calls:

```
molcas $Project.seward.input    # Execute seward
molcas $Project.scf.input       # Execute scf
```

By default, the output will go directly to the screen. If can be redirected by using flag -f, e.g. molcas -f water.inp will store the output in *water.log* and *water.err* files.

The default behavior of MOLCAS execution can be altered by setting environment variables.

7.2.2 Project name and working directory

When running a project, MOLCAS uses the variable PROJECT giving a project name, and a scratch directory defined by the variable WORKDIR. This serves the purpose of maintaining structure of the files and facilitating automatic file mapping.

There are several ways to set up these variables. By default, the name of the Project constructed from the name of the input file, by removing the last suffix, e.g. for example for an input name *Water.SCF.input* the PROJECT name will be WATER.SCF. Alternatively, user can set environment variable PROJECT, or MOLCAS_PROJECT.

Scratch directory can be set by environment variable MOLCAS_WORKDIR. If it is set to value "PWD", current directory will be used. Otherwise, it can be set to a directory name. In this case scratch area will be located in a subdirectory $MOLCAS_WORKDIR/$Project. It is also possible to overwrite the value of scratch area, by setting environment variable WORKDIR.

- Project=...; export Project
- WorkDir=...; export WorkDir

MOLCAS modules communicates between each other via files, located in the WORKDIR. The description of internal filenames and file mapping can be found at Appendix.

7.2.3 Input

When you have decided which program modules you need to use to perform your calculation, you need to construct input for each of these. There is no particular structure enforced on the input files, but it is recommended that you follow:

- *$Project.*"*prgm-name*"*.input*

which is the name of the input files assumed in the sample shell script.

7.2.4 Preparing a job

When you prepare a job for batch processing, you have to create a shell script. It is recommended that you use the sample shell script supplied with MOLCAS as a starting point when building your own shell script. The following steps are taken in the shell script:

1. Define and export the MOLCAS variables
   - Project (or use *MOLCAS_PROJECT*)
   - WorkDir (or *MOLCAS_WORKDIR*)
2. Issue a sequence of MOLCAS commands.
3. Remove the scratch directory and all files in it.

The following is an example of a shell script.

```
Project=HF; export Project                        # Define the project id
WorkDir=/temp/$LOGNAME/$Project.$RANDOM; export WorkDir  # Define scratch directory
molcas $Project.input                             # Run molcas with input file, which
                                                  #   contains inputs for several modules
rm -r $WorkDir                                    # Clean up
```

The file ***$ThisDir/$Project.input*** contains the ordered sequence of MOLCAS inputs and the EMIL interpreter will call the appropriate programs. See section 7.3 for an explanation of the additional tools available in the EMIL interpreter.

The following is an example of a shell script to be submitted for batch execution.

```
Project=HF; export Project                        # Define the project id
WorkDir=/temp/$LOGNAME/$Project.$RANDOM; export WorkDir  # Define scratch directory
molcas $Project.seward.input                      # Execute seward
molcas $Project.scf.input                         # Execute scf
rm -r $WorkDir                                    # Clean up
```

An alternative way to control teh usage of the WorkDir is to use flags in molcas command:

Variable	*Purpose*
-new	clean WorkDir before the usage

-clean	clean WorkDir after the usage

Note, that if you configured your working environment by using setuprc script, the only command you have to place into the shell script is:

```
molcas $Project.input
```

7.2.5 System variables

Molcas contains a set of system variables that the user can set to modify the default behaviour of Molcas. Two of them (Project and WorkDir) must be set in order to make Molcas work at all. There are defaults for these but you are advised not to use the defaults.

There are several ways of using Molcas environment variables:

- These variables can be exported in your shell script

  ```
  export MOLCAS_MEM=512
  molcas input
  ```

- These variables can be included into Molcas input:

  ```
  * begin of the input file
  >>> export MOLCAS_MEM=512
    . . .
  ```

- variables can be included directly into molcas command in the form:

  ```
  molcas MOLCAS_MEM=512 input
  ```

The simplest way to set up default environment for Molcas is to use script ***setuprc***, which can be run as command molcas setuprc. This interactive script creates a resource file ***molcasrc***, located either in $*MOLCAS* or $*HOME*/*.Molcas* directory. The priority of these settings is: user defined settings (e.g. in molcas command), user resource file, Molcas resource file.

Two flags in Molcas command are related to resource files:

Variable	*Purpose*
-env	Display current Molcas environment e.g. molcas -env input will print information about environment variables, used during execution of the input file.
-ign	Ignore resource files e.g. molcas -ign input will process input file without settings, which are stored in $*MOLCAS*/*molcasrc* and in $*HOME*/*molcasrc* files.

The most important environment variables, used in Molcas:

Variable	*Purpose*

Project — This variable can be set in order to overwrite the default name of the project you are running. The default (and recommended) value of the project name is the name of the input file (without the file extension).

WorkDir — This variable can be used to specify directly the directory where all files that Molcas creates are placed. See MOLCAS_WORKDIR for more options.

CurrDir — This variable corresponds to the location of the input, and it is used as a default location for all output files, generated by Molcasmodules.

MOLCAS — This variable indicates the location of Molcas. The default version of Molcas to be used is specified at file *.Molcas/molcas*, located at user HOME directory.

MOLCAS_NPROCS — This variable should be used to run Molcascode in parallel. It defines the number of computational units (cores or nodes) which will be used.

MOLCAS_MEM — This environment variable controls the size (soft limit) of the work array utilized in the programs that offer dynamic memory. It is specified in Megabytes, i.e.
MOLCAS_MEM=256; export MOLCAS_MEM
will assign 256Mb for the working arrays. It is also possible to use Gb (Tb) to specify memory in Gb or Tb.

- MOLCAS_MEM is undefined — The default amount of memory (2048Mb), will be allocated for the work arrays.
- MOLCAS_MEM is defined but nonzero — This amount of memory will be allocated.

See also MOLCAS_MAXMEM.

The complete list of Molcas-related environment variables:

Variable	*Purpose*
MOLCAS_COLOR	By default molcas uses markup characters in the output. To overwrite, set the key to NO.
MOLCAS_NPROCS	See above
MOLCAS_DEBUGGER	This variable can be set to the name of debugger (or another code) which will be used on top of molcas executables. The option is useful for tracing an error in the code
MOLCAS_DISK	The value of this variable is used to split large files into a set of smaller datasets, as many as are needed (max. 20 subsets). It is specified in Megabytes, for instance, MOLCAS_DISK=1000; export MOLCAS_DISK, and the following rules apply:

- MOLCAS_DISK is undefined — The program modules will ignore this option and the file size limit will be defined by your hardware (2 GBytes for 32-bit machines).
- MOLCAS_DISK=0 (zero) — The programs will assume a file size limit of 2 GBytes (200GBytes on 64-bit machines).
- MOLCAS_DISK is defined but nonzero — The files will be limited to this value (approximately) in size.

MOLCAS_ECHO_INPUT An environment variable to control echoing of the input. To suppress print level, set MOLCAS_ECHO_INPUT to 'NO'.

MOLCAS_FIM Activates the Files In Memory I/O layer. See section 3.1.5 for more details. *Note that this setting is available only in MOLCAS compiled without Global Arrays.*

MOLCAS_GRID `SEWARD`, `SCF`, `RASSCF` can compute (screen quality) grid automatically. If set to AUTO, only the last program in the input will compute grid. Yes - force the code always compute grids (it might take time) No - disable the automatic generation of grids.

MOLCAS_KEEP_WORKDIR If set to NO MOLCAS will remove scratch area after a calculation. This setting can be overwritten by running molcas with flag -clean.

MOLCAS_LICENSE An environment which specifies the directory with MOLCAS license file *`license.dat`*. The default value of this variable is directory *`.Molcas/`* in user home directory..

MOLCAS_LINK An environment variable to control information about linking of files. By default (MOLCAS_LINK is not set) only essential information about linking will be printed. To increase/decrease the print level, set MOLCAS_LINK to 'Yes'/'No'.

MOLCAS_MAXITER An environment variable to control maximum number of iterations in DO WHILE loop.

MOLCAS_MAXMEM An environment variable to set up a hard limit for allocated memory (in Mb). If is not specified, then it takes value of MOLCAS_MEM. Otherwise, the (MOLCAS_MAXMEM-MOLCAS_MEM) amount of RAM will be primarily used for keeping files in memory (FiM), or allocating Distributed Global Arrays. *Note that this setting is available only in MOLCAS compiled without Global Arrays.*

MOLCAS_MEM See above.

MOLCAS_MOLDEN If MOLCAS_MOLDEN set to 'ON' a Molden style input file will be generated regardless of the number of orbitals.

MOLCAS_NEW_WORKDIR If set to YES MOLCAS will never reuse files in scratch area. This setting can be overwritten by running molcas with flag -old: molcas -old input

MOLCAS_OUTPUT This variable can alter the default directory for extra output files, such as orbitals files, molden files, etc. If set, Molcas will save output files to the specified directory. The directory name can be set in the form of absolute PATH, or relative PATH (related to the submit directory). A special value 'WORKDIR' will keep all output files in WorkDir. A special value 'NAME' will create a subdirectory with a name of Project. If the variable is not set, all output files will be copied or moved to the current directory. Default value can be forced by MOLCAS_OUTPUT=PWD.

MOLCAS_PRINT MOLCAS_PRINT variable controls the level of output. The value could be numerical or mnemonic: SILENT (0), TERSE (1), NORMAL (2), VERBOSE (3), DEBUG (4) and INSANE (5).

MOLCAS_PROJECT If set to value NAME, Molcas will use the prefix of the input file as a project name. Otherwise, it set a project name for the calculation. If set to the value NAMEPID, the Project name still will be constructed from the name of input file, however, the name of scratch area will be random.

MOLCAS_PROPERTIES If MOLCAS_PROPERTIES is set to 'LONG' properties with the individual MO contributions will be listed.

MOLCAS_REDUCE_PRT If set to NO, print level in DO WHILE loop is not reduced.

MOLCAS_REDUCE_NG_PRT If set to NO, print level in `NUMERICAL_GRADIENT` loop is not reduced.

MOLCAS_SAVE This variable can alter the default filenames for output files. If not set (default), all files will overwrite old files. If set to 'INCR' all output files will get an incremental filenames. If set to 'ORIG' - an existent file will be copied with an extension '.orig'

MOLCAS_TIME If set, switch on timing information for each module

MOLCAS_TIMELIM Set up a timelimit for each module (in minutes). By default, the maximum execution time is set to unlimited. *Note that this setting is available only in MOLCAS compiled without Global Arrays.*

MOLCAS_TRAP If MOLCAS_TRAP set to 'OFF' Molcas modules will continue to be executed, even if a non-zero return code was produced.

MOLCAS_WORKDIR A parent directory for all scratch areas. It can be set to an absolute PATH (recommended), to a relative PATH, or to a special value PWD (to use current directory for scratch files)

User can customize his installation by adding MOLCAS environment variable into ***`molcasrc`*** file.

Another way of customizing Molcas is to use prologue and epilogue scripts. If user created a file ***`prologue`*** in ***`$HOME/.Molcas`*** directory it will be executed (as ./prologue) before Molcas calculation starts. ***`epilogue`*** in ***`$HOME/.Molcas`*** directory will be executed at the end of calculation. Files ***module.prologue*** and ***module.epilogue*** contains commands executing before and after each executable molcas module. These files may use internal Molcas variables, such as $\$Project$, $\$WorkDir$, $\$MOLCAS_MEM$ etc. Note that prologue/epilogue scripts should be executable. For debug purposes, the location of prologue and epilogie files can be set by $\$MOLCAS_LOGUE_DIR$ variable.

Example:

```
prologue:
 echo Calculation of $Project input will start at ‘date‘
module.prologue:
 echo Running module $MOLCAS_CURRENT_PROGRAM at $WorkDir
```

7.3 General input structure. EMIL commands

This is a general guide to the input structure of the programs in the Molcas program system. All programs conform to the same conventions except where explicitly stated otherwise.

The programs are driven by keywords, which are either used without further information, or followed by additional specifications on the line(s) following the keyword, and is normally numeric in nature. *All numerical inputs are read in free format, note that in general* Molcas *will not be able to process lines longer than 120 characters.* The keywords can be given in mixed case (both upper and lower case are allowed). In the input stream you can insert comment lines anywhere, except between a keyword and the following additional specifications, with a comment line identified by an asterisk (*) in the first position on the line.

Most codes look at the first 4 characters of the keyword and ignores the rest. The entries in the lists of keywords below follow the standard that the significant characters are in upper case and larger than the nonsignificant characters. This do not imply that the keywords have to be typed in upper case; they can be typed freely in mixed case.

All inputs begin with a name of the program followed by the keywords:

```
  &PROGRAM
 * here follows the keywords
```

where PROGRAM is the name of the Molcas module. The input listing is finished when a new program name, precede by the symbol &, is found (or the end of file or an EMIL command).

The following is an example of a list of keywords common to most of the programs:

Keyword	*Meaning*
TITLe	This keyword starts the reading of title line. The following line is treated as title line.

The programs only decode the first four characters of a keyword (except otherwise specifically indicated). For clarity it is however recommended to write the full keyword name. The keywords can be typed freely in upper, lower or mixed case.

An example for an input file used to run the SCF program follows:

```
 &SCF
Title
 Water molecule. Experimental equilibrium geometry
* The symmetries are: a1, b2, b1 and a2.
Occupied
3 1 1 0
* The ivo keyword prepares virtual orbitals for MCSCF.
Ivo
```

Interpretation of MOLCAS input is performed by MOLCAS.EXE. The internal language used by MOLCAS.EXE is EMIL (Extended Molcas Input Language). It includes two different types of input commands:

- Sections with MOLCAS input.
- EMIL commands (a line started with > character)

7.3.1 Molcas input

EMIL allows to write MOLCAS input in a more compact way: user can omit &*END*, as well as a compulsory (in previous versions of MOLCAS) keyword END OF INPUT. As soon as a new module (or EMIL command) is requested in a user input, the input for the module is terminated.

Also, it is possible to separate lines by ; sign, or by = sign (to create a pair *keyword* = *value*). In some rare occasions signs ; and = are used in the input for a MOLCAS module. In order to keep these symbols unchanged, user can mark a part of an input, containing these symbols, by EMIL commands >> *verbatim* and >> *endverbatim*.

It means that the input:

```
 &SCF &END
CHARGE
 1
End of input
 &ALASKA &END
End of input
 &SLAPAF &END
End of input
```

could be written as:

```
 &SCF; CHARGE=1
 &ALASKA;  &SLAPAF
```

User can comment parts of input, by using ∗ at the beginning of line, or use C-style comments (/ ∗ ... ∗ /) to comment several lines.

In a rare occasion user might want to execute a UNIX command from the input. It is important to understand that not all UNIX commands can be understood and interpreted by EMIL. Also, EMIL should know where to execute a command – only at the master node, or for all parallel tasks. In the past, EMIL supports the usage of commands started from an exclamation mark, or with command UNIX. To avoid confusions, the serial execution of a command is now related to SHELL, and the parallel execution to EXEC.

The difference between SHELL and EXEC should be obvious from the following example. If Molcas compiled in serial, both commands are the same. But for parallel molcas:

```
>>export MOLCAS_NPROCS=2
>>shell pwd
```

will print only the directory for the master process.

In contrary,

```
>>export MOLCAS_NPROCS=2
>>exec pwd
```

will print two directories: one for the master and one to the slave.

7.3.2 EMIL commands

EMIL commands can be written in a short form:

```
> KEY [VALUE]
```

or in a nice form:

```
>>>>>>>>>>  KEY  [VALUE]  <<<<<<<<<
```

EMIL commands are not case sensitive, but the variables used in commands must be written in upper case. Also, it is important to place spaces in between elements (words) in the commands.

Here is a list of EMIL commands:

Command	*Purpose*
>> EXPORT A=B	a command to set environment variable A to value B
>> EXIT	a command to terminate execution. An optional value for this command is the return code (default value is 0)
>> INCLUDE file	a command to include a file into the input A compulsory value for this command is the filename.
>> FILE file	A compulsory value for this command is the filename. A command to inline a file in the input file. The file will be extracted into WorkDir before the start of the calculation. The end of file should be marked as EOF command. Note that the file is only created in the master process WorkDir, if the slaves need access to it, you'll need to use the COPY command (see below). All files specified with FILE are created at the beginning of the calculation, regardless of their placement in the input.

>> EOF	A command to close inlined file.
>> SHELL	a command to execute a unix command in serial.
>> EXEC	a command to execute a unix command in parallel.
>> LINK	a command to make a link between two files, located in WorkDir. The command is similar to ’!ln -s FILE1 FILE2’ but in parallel environment it is executed in all WorkDirs. The command assumes that FILE1 does exist, and FILE2 does not at the moment. >>LINK -FORCE allows to link a file which does not exist. User should avoid the usage of LINK commands in the input.
>> COPY	a command to make a copy. The command is similar to ’!cp -f /path/to/FILE1 FILE2’ but can be used also in a parallel environment, in which case it will take the source file and distribute to the work directories of all processes. The destination must be located in the work directory. Note that EMIL command does not allow to use masks in the command. If FILE1 does not exist, the command returns an error code.
>> CLONE	a command to make a clone copy of a file, doing a local copy on all slaves if parallel. It is mostly used internally, e.g. to distribute an input file to all WorkDirs.
>> COLLECT	A command to copy one file to another, collecting files on slaves and put them on the master if parallel. It is mostly used internally, e.g. to collect output files.
>> SAVE	A command to copy one file to another, only on the master if parallel
>> RM	a command to delete a file. The command is similar to ’!rm FILE’ but can be used also in parallel environment. Note that EMIL command does not allow to use masks in the command. An attempt to remove non existent file leads to an error. It is possible to use -FORCE flag to allow deleting of non-existent file.
>> EVAL A=B	evaluate a numerical value

Keywords to organize loops in input, and execute modules conditionally:

Command	*Purpose*
>> DO WHILE	a command to start a loop. The loop should be terminated by SLAPAF or LOOP module, followed by ENDDO command
>> DO GEO	a command to start a special loop for geometry optimization with constrained internal coordinates. The loop should be terminated by ENDDO command. (See documentation for `GEO` for more details.)

>> FOREACH A in (B, C, D)	a command to loop when the value of A is in the comma or space separated list. The list also can be written in the format "From .. To". Note that variable in the loop must be uppercased.
>> ENDDO	a command to finish the loop. If last module (before ENDDO command) returns 1 - the loop will be executed again (if number of iterations is less than MAXITER). If the return code is equal to 0 the loop will be terminated.
>> IF (ITER = N)	- a command to make conditional execution of modules/commands on iteration N (N possibly could be a space separated list)
>> IF (ITER NE N)	- a command to skip execution of modules/commands on iteration N
>> IF (ITER != N)	- same as above
>> IF ($VAR = N)	- a command to make conditional execution if $VAR value equals to N (if statement terminated by ENDIF command)
>> IF ($VAR = N) GOTO JUMP	- a command to make conditional goto to a label JUMP
>> IF (-FILE file)	- test for existence of a file
>> LABEL JUMP	- a command to define a label. Note! Only forward jumps are allowed.
>> ENDIF	terminate IF block. Note nested if's are not allowed.

EMIL interpreter automatically stops calculation if a module returns a returncode higher than 0 or 1. To force the interpretor to continue calculation even if a returncode equal to 16 (which is a return code for non-convergent calculation) one should set environment variable MOLCAS_TRAP='OFF'.

SLAPAF returns a special return code in the case of converged (non converged) geometry. So, to organize a structure calculation one should place the call to SLAPAF as a last statement of loop block. The summary of geometry optimization convergence located in a file *$Project.structure*. The programs following a geometry optimization will automatically assume the optimized geometry and wave function. Any new SEWARD calculation after an optimization (minimum or transition state) will disregard the input coordinates and will take the geometry optimized at previous step.

It is also possible to use a special dummy program LOOP to organize infinite loops, or loops terminated by the counter (set by MOLCAS_MAXITER)

Keyword SET is obsolete and should be changed to EXPORT.

Verbatim input.

If an input for a module must contain special symbols, such as ; or =, user can mark a corresponding part of the input by EMIL command *VERBATIM*

Command	*Purpose*

>> VERBATIM << start verbatim input

>> END VERBATIM << finish verbatim input

Below are different input examples.

The first example shows the procedure to perform first a CASSCF geometry optimization of the water molecule, then a numerical hessian calculation on the optimized geometry, and later to make a CASPT2 calculation on the optimized geometry and wave function. Observe that the position of the SLAPAF inputs controls the data required for the optimizations.

```
*
*    Start Structure calculation
*
>>EXPORT MOLCAS_MAXITER=50
 &GATEWAY
coord
$MOLCAS/Coord/Water.xyz
BASIS = ANO-S
>>>>>>>>>>>>> Do while <<<<<<<<<<<<
 &SEWARD
>>>>>>>> IF ( ITER = 1 ) <<<<<<<<<<
 &SCF
>>>>>>>  ENDIF <<<<<<<<<<<<<<<<<<<<
 &RASSCF
Title
 H2O ANO(321/21).
Nactel   = 6  0  0
Spin     = 1
Inactive = 1  0  0  0
Ras2     = 3  1  0  2
 &ALASKA; &SLAPAF
>>>>>>>>>>>>>  ENDDO  <<<<<<<<<<<<<<
 &CASPT2
Maxit = 20
Lroot = 1
 &GRID_IT
```

Another example demonstrate a possibility to use loops. SCF module will be called twice - first time with BLYP functional, second time with B3LYP functional.

```
*-----------------------------------------------------
 &GATEWAY
coord
$MOLCAS/Coord/C2H6.xyz
basis
ANO-S-VDZ
group
y xz
*-----------------------------------------------------
 &SEWARD
Title
Ethane DFT test job
*-----------------------------------------------------
>>foreach DFT in (BLYP, B3LYP )
 &SCF ; KSDFT = $DFT
>>enddo
*-----------------------------------------------------
```

The next examples calculates HF energy for the several structures:

```
* modify coordinates in place
```

```
>>foreach DIST in (1.0, 2.0, 20.0)
 &GATEWAY
Coord
2
hydrogen molecule
H 0 0 0
H $DIST 0 0
BASIS= ANO-S-MB
GROUP= C1
 &SEWARD
 &SCF
UHF
SCRAMBLE=0.3
>>enddo
* incremental change of coordinates
>>export DIST=1.0
>>foreach L in ( 1 .. 3 )
>>eval DIST=$DIST+0.1
 &GATEWAY
Coord
2
hydrogen molecule
H 0 0 0
H $DIST 0 0
BASIS= ANO-S-MB
GROUP= C1
 &SEWARD
 &SCF
>>enddo
* different coordinate files
>> FILE H2001.xyz
2
H  0.300000000  0.000000000  0.000000000
H -0.300000000  0.000000000  0.000000000
>> FILE H2002.xyz
2
H  0.350000000  0.000000000  0.000000000
H -0.350000000  0.000000000  0.000000000
>> FILE H2003.xyz
2
H  0.400000000  0.000000000  0.000000000
H -0.400000000  0.000000000  0.000000000
>>foreach COO in ( 000, 001, 002)
 &GATEWAY
Coord = H2$COO.xyz
BASIS= ANO-S-MB
GROUP= C1
 &SEWARD
 &SCF
>>enddo
```

7.3.3 Use of shell parameters in input

The MOLCAS package allows the user to specify parts or variables in the the input file with shell variables, which subsequently are dynamically defined during execution time. **Note:** the shell variable names must be in upper case. Find below a simple example where a part of the H_2 potential curve is computed. First, the script used to run the calculation:

```
#! /bin/sh
#
Home=‘pwd‘ ;                     export Home
Project=H2 ;                     export Project
WorkDir=/tmp/$Project ;          export WorkDir
```

```
#
# Create workdir and cd to it
#
rm -fr $WorkDir
mkdir $WorkDir
#
# Loop over distances
#
for R in 0.5 0.6 0.7 0.8 0.9 1.0
do
   export R
   molcas $Home/$Project.input > $Home/$Project-$R-log 2> $Home/$Project-$R-err
done
#
# Cleanup WorkDir
#
rm -fr $WorkDir
```

In this sh shell script we have arranged the call to the MOLCAS package inside a loop over the various values of the distances. This value is held by the variable $R which is exported every iterations. Below is the input file used, note that the third cartesian coordinate is the variable $R.

```
 &SEWARD
Symmetry
 x y z
Basis set
H.sto-3g....
H   0.000   0.000   $R
End of basis
End of input
 &SCF
```

7.3.4 Customization of molcas input

EMIL interpretor supports templates (aliases) for a group of program calls or/and keywords. The definition of these templates can be located in file *alias* located at MOLCAS root directory, or at *.Molcas/* directory. The definition should be written in the following format: $@name\{sequenceofEMILcommands\}$. In order to use the alias, the input should contain $@name$.

For example, user can define

```
@DFTgeometry
>> DO WHILE
 &SEWARD
 &SCF; KSDFT=B3LYP;
 &SLAPAF
>>ENDDO
```

and so, an input for geometry optimization can be written in the following form:

```
&GATEWAY; Coord=Water.xyz; Basis = ANO-L-MB;
@DFTgeometry
```

It is also possible to use parameters. In the alias file, possible parameters have names: $1,$2, etc. up to 5 parameters. In the user input an alias should be followed by parenthesis with comma separated list of values.

Modifying the previous example:

```
@DFTgeometry
>> DO WHILE
 &SEWARD
 &SCF; CHARGE=$1; KSDFT=$2;
 &SLAPAF
>>ENDDO
```

Input file now looks like:

```
@DFTgeometry(0,B3LYP)
```

Section 8

Programs

8.1 ALASKA

This module is automatically invoked by the SLAPAF module. This is the preferred mode of operation! In connection with numerical gradients it will ensure that the rotational and translational invariance is fully utilized in order to reduce the number of used displacements.

The ALASKA module compute analytic or for numerical gradients requests the execution of an alternative module. The ALASKA module figures out the method automatically. Analytic methods are implemented for the HF, MBPT2, KS-DFT, and RASSCF and SA-CASSCF method. Numerical methods are implemented for SCF, KS-DFT, RASSCF, MBPT2, CCSDT, the CASPT2 and MS-CASPT2 methods, including the use of the Cholesky decomposition for the methods were that has been implemented.

Both analytic and numerical procedures are parallelized.

For SA-CASSCF gradient the ALASKA module will automatically start up the MCLR module if required.

8.1.1 Analytic gradients

Gradients of the energy with respect to nuclear coordinates can be computed for any type of wave function as long as an effective first order density matrix, an effective Fock matrix, and an effective second order density matrix is provided. The term effective is related to that these matrices in the case of non-variational parameters in the wave function (e.q. CI, MP2, CASPT2, etc.) are modified to include contributions from the associated Lagrange multipliers. The gradient expression apart from these modifications is the same for any wave function type. ALASKA is the gradient program, which will generate the necessary integral derivatives and combine them with the matrices mentioned in the text above.

8.1.2 Description

ALASKA is written such that gradients can be computed for any kind of basis function that SEWARD will accept.

ALASKA is able to compute the following integral derivatives:

- overlap integrals,
- kinetic energy integrals,
- nuclear attraction integrals (point charges or finite nuclei),
- electron repulsion integrals,
- external electric field integrals,
- ECP and PP integrals,
- reaction field integrals,
- and Pauli repulsion integrals.

ALASKA employs two different integration schemes to generate the one- and two-electron integral derivatives. The nuclear attraction and electron repulsion integrals are evaluated by a modified Rys-Gauss quadrature [15]. All other integral derivatives are evaluated with the Hermite-Gauss quadrature. The same restriction of the basis sets applies as to SEWARD. None of the integral derivatives are written to disk but rather combined immediately with the corresponding matrix from the wave function.

At present the following limitations are built into ALASKA:

Max number of unique basis functions:	2000
Max number of symmetry independent centers:	500
Highest angular momentum:	15
Highest symmetry point group:	D_{2h}

8.1.3 Numerical gradients

The module is parallelized over the displacements, which in case of large jobs gives a linear speed up compared to a serial execution, although in order to obtain this it is important to choose the number of nodes such that the number of contributing perturbations is a multiple of the number of nodes. For a given molecule the number of perturbations equals the number of atoms times 6 (a perturbation with plus and minus delta for each of the three axes). Symmetry can of course reduce this number.

8.1.4 Dependencies

ALASKA depends on the density and Fock matrices generated by SCF or RASSCF. In addition it needs the basis set specification defined in SEWARD. The dependencies of the numerical part of the module is the union of the dependencies of the SEWARD, SCF, RASSCF, MBPT2, MOTRA, CCSDT, and CASPT2 modules. All these dependencies, however, are totally transparent to the user.

8.1.5 Files

Input files

Apart from the standard input unit ALASKA will use the following input files: *RYSRW*, *ABDATA*, *ONEINT*, *RUNFILE* (for more information see 7.1.2).

The files of the SEWARD, SCF, RASSCF, MBPT2, MOTRA, CCSDT, and CASPT2 modules are needed for the numerical procedure.

Output files

In addition to the standard output unit ALASKA will generate the following files.

File	*Contents*
RUNFILE	The *runfile* is updated with information needed by the SLAPAF relaxation program. ALASKA will write the molecular Cartesian gradients on this file.
ALASKA.INPUT	File with the latest input processed by ALASKA.

8.1.6 Input

Below follows a description of the input to ALASKA. Note that input options are related to the analytic gradient procedure if not otherwise noted!

In addition to the keywords and the comment lines the input may contain blank lines. The input is always preceded by the program name:

```
&ALASKA
```

Optional keywords for analytical gradients

Keyword	*Meaning*
TEST	With this keyword the program will process only the input. It is a debugging aid to help you check your input.
NAC	Requests a calculation of the nonadiabatic coupling vector between the two specified roots in a SA-CASSCF calculation.
NOCSF	In a NAC calculation, neglects the so-called CSF contribution. Note that this contribution is responsible for the translational and rotational non-invariance, and it has been suggested that not including it may give more physical results in dynamics simulations [16].
ONEONLY	Compute only the nuclear repulsion and one-electron integrals contribution to the gradient. The default is to compute all contributions to the molecular gradient.

CUTOff Threshold for ignoring contributions to the molecular gradient follows on the next line. The default is 1.0d-7. The prescreening is based on the 2nd order density matrix and the radial overlap contribution to the integral derivatives.

OFEMbedding Performs a Orbital-Free Embedding gradient calculation, available only in combination with Cholesky or RI integral representation. The runfile of the environment subsystem renamed AUXRFIL is required. An example of input for the keyword OFEM is the following:

```
OFEMbedding
 ldtf/pbe
dFMD
 1.0
```

(see the OPTIONAL keyword :kword:'DFMD' below). The keyword OFEM requires the specification of two functionals in the form fun1/fun2, where fun1 is the functional used for the Kinetic Energy (available functionals: Thomas-Fermi, with acronym LDTF, and the NDSD functional), and where fun2 is the xc-functional (LDA, LDA5, PBE and BLYP available at the moment).

DFMD In combination with OFEM, specifies the fraction of correlation potential to be added to the OFE potential (zero for KSDFT and one for HF).

POLD The gradient is printed in the old format. Note: by default gradient is not printed any longer.

PNEW The gradient is printed in the new human-readable format.

VERBose The output will be a bit more verbose.

SHOW gradient contributions The gradient contributions will be printed.

Optional keywords for numerical gradients

Keyword *Meaning*

NUMErical Forces the use of numerical gradients even if analytical ones are implemented. The default is to use analytical gradients whenever possible.

ROOT Specifies which root to optimize the geometry for, if there is more than one root to choose from. In a RASSCF optimization, the default is to optimize for the same root as is relaxed. In a MS-CASPT2 calculation, the default is to optimize for root 1. It can be used to override the default root in an analytical calculation too.

DELTa For use with numerical gradients only! The displacement for a given center is chosen as the distance to the nearest neighbor, scaled by a factor. This factor can be set through the DELTa keyword. The default value is 0.01.

KEEPOldGradient When computing numerical gradients with constraints, the gradient of the constrained degrees of freedom is normally set to zero. If this keyword is specified, the existing value of the gradient (probably computed analytically with a different method) is maintained instead. This is used, in combination with NGEXclude in GATEWAY (or "phantom" constraints), to set up composite gradients [17].

The following is an example of an input which will work for almost all practical cases. Note that it is very rarely that you need to run this program explicitly. It is usually controlled by the program SLAPAF.

```
&Alaska
```

8.2 AVERD

8.2.1 Description

AVERD computes average densities and corresponding natural molecular orbitals. Given a set of density matrices in the same basis set, an average density matrix is constructed and diagonalized to give average natural orbitals. These orbitals have non-integer occupation numbers, although they usually are fairly well clustered in one strongly occupied part and one weakly occupied part. From basic mathematical properties of natural orbitals, a truncated set of the orbitals constructed this way constitutes the most compact basis of one-electron functions of that given size. In other words, the smallest set of functions to span the average space within a certain accuracy has been obtained. AVERD is in essence very similar to GENANO.

8.2.2 Dependencies

AVERD needs a set of input densities. Any of the programs, which generate a density must preceed.

8.2.3 Files

Below is a list of the files that are used/created by AVERD.

Input files

File	*Contents*
RUNFILE	File for communication of auxiliary information generated by the program SEWARD.
ONEINT	File with one-electron integrals generated by SEWARD.

*RUN****	A set of RunFiles on which a density matrix is stored. This density matrix is read and added to the average. If ORBITALS is given, these files are not needed.
*NAT****	A set of orbitals in the format generated by `SCF` or `RASSCF`. The orbitals are used to generate a density matrix, which is added to the average. Only required if ORBITALS is given.

Output files

File	*Contents*
AVEORB	The average orbitals generated by `AVERD`. They are stored in the same way as SCF-orbitals and can be used as INPORB.

8.2.4 Input

Keyword	*Meaning*
TITLE	Title of the calculation.
WSET	Followed by two rows. On the first the number of input orbitals are given, N. Second row contains N numbers each giving a weight for the k^{th} input density matrix to the average density. The weights are normalized by `AVERD`, hence only the ratio of the numbers have any significance. **This keyword is mandatory**.
PRINT	Print level. 1 is default. Higher than 3 is not recommended for the average user.
ORBITALS	This keyword signifies that the densities should be created from average orbitals in the files NAT***, not directly from the density matrices on the files RUN***. The default is to use the density matrices on RUN***.
OCCUPATION	Followed by one number. The number of average orbitals with an occupation higher than this number is reported for each symmetry. Since the occupation is the guide for how to truncate the orbitals in subsequent applications, this is an easy way to get hold of that number. The default is 10^{-5}.

Input example

In this example, two density matrices are averaged and their average orbitals are computed.

```
 &Seward
Basis set
O.ano-s.Pierloot.10s6p3d.7s4p2d.
O  0.0000  0.0000  0.3000
End of Basis set
Basis set
H.ano-s.Pierloot.7s3p.4s1p.
H1 0.0000 -1.4300 -0.8070
H2 0.0000  1.4300 -0.8070
End of Basis Set
 &FfPt
Dipo
z 0.005
End of Input
 &Scf
Occupation
5
>>COPY $Project.RunFile RUN001
 &FfPt
Dipo
z -0.005
 &Scf
Occupation
5
End of Input
>>COPY $Project.RunFile RUN002
 &Averd &End
Wset
2
1.0 1.0
Occupation
1d-6
```

8.3 CASPT2

Second order multiconfigurational perturbation theory is used in the program CASPT2 [18, 19] to compute the (dynamic) correlation energy. The reference state is usually of the CAS type, but the program has been extended to also accept RAS reference states[20, 21]. The first step is therefore a RASSCF calculation and the CASPT2 calculation gives a second order estimate of the difference between the RASSCF and the full CI energy. For calculations using a true RAS reference, benchmark calculations were reported by Sauri et al. [21]. For CASSCF references, the CASPT2 method has been tested in a large number of applications [22, 23]. Here follows a brief summary of results.

Bond distances are normally obtained with an accuracy of better that 0.01 Å for bonds between first and second row atoms. With the standard Fock matrix formulation, bond energies are normally underestimated with between 2 and 5 kcal/mol for each bond formed. This is due to a systematic error in the method[24]. In every process where the number of paired electrons is changed, an error of this size will occur for each electron pair. For example, the singlet-triplet energy difference in the methylene radical (CH_2) is overestimated with about 3 kcal/mol [19]. Heats of reactions for isogyric reactions are predicted with an accuracy of ±2 kcal/mol. These results have been obtained with saturated basis sets and all valence electrons active. The use of smaller basis sets and other types of active spaces may, of course, affect the error.

These systematic errors have recently been considerably reduced by the introduction of a modified zeroth order Hamiltonian [25]. The method introduces a shift (the IPEA shift) that modifies the energies of active orbitals such that they become closer to ionization energies when excited from and closer to electron affinities when excited out of. The approach has been tested for 49 diatomic molecules, reducing the mean error in D_0 from 0.2 to 0.1 eV. For the triply bonded molecules N_2, P_2, and As_2 it was reduced from 0.45 eV to less than 0.15 eV. Similar improvements were obtained for excitation and ionization energies. The IPEA modified H_0 (with a shift parameter of 0.25) is default in Molcas from version 6.4.

An alternative to IPEA is to use the options, called 'g_1', 'g_2', and 'g_3'(See Ref. [26]), that stabilizes the energies of the active orbitals. The remaining error is no longer systematic, and is generally reduced. For example, the error in the singlet-triplet separation of CH_2 is reduced to 1 kcal/mol [26]. This option is, however, not recommended any longer because it has been replaced by the IPEA Hamiltonian.

The CASPT2 method can be used in any case where a valid reference function can be obtained with the CASSCF method. There is thus no restriction in the number of open shells or the spin coupling of the electrons. Excited states can be treated at the same level as ground states. Actually one of the major successes with the method has been in the calculation of excitation energies. A large number of applications have been performed for conjugated organic molecules. Both Rydberg and valence excited states can be treated and the error in computed excitation energies is normally in the range 0.0–0.2 eV. Similar results have been obtained for ligand field and charge-transfer excitations in transition metal compounds. From Molcas-6 it is possible to use the CASPT2 method in conjunction with the Douglas-Kroll-Hess relativistic Hamiltonian, which has made possible calculations on heavy element compounds such a third row transition metal compounds and actinides with accurate results.

The CASPT2 method can also be used in combination with the `FFPT` program to compute dynamic correlation contributions to properties with good results in most cases. Numerical gradients are available with the `SLAPAF` module.

The CASPT2 method is based on second order perturbation theory. To be successful, the perturbation should be small. A correct selection of the active space in the preceding CASSCF calculation is therefore of utmost importance. All near-degeneracy effects leading to configurations with large weights must be included at this stage of the calculation. If this is not done, the first order wave function will contain large coefficients. When this occurs, the `CASPT2` program issues a warning. If the energy contribution from such a configuration is large, the results is not to be trusted and a new selection of the active space should be made.

Especially in calculations on excited states, intruder states may occur in the first order wave function. Warnings are then issued by the program that an energy denominator is small or negative. Such intruder states often arise from Rydberg orbitals, which have not been included in the active space. Even if this sometimes leads to large first order CI coefficients, the contribution to the second order energy is usually very small, since the interaction with the intruding Rydberg state is small. It might then be safe to neglect the warning. A safer procedure is to include the Rydberg orbital into the active space. It can sometimes be deleted from the MO space.

Calculations on compounds with heavy atoms (transition metals, actinides, etc) may yield many virtual orbitals with low energies. The interaction energies for excitations to states where these orbitals are occupied are often very small and the low denominators can then be removed by a suitable level shift (see below). But it is always safer to include such orbitals in the active space.

Two keywords have been introduced to deal with this fairly common situation, for excited states, that weakly coupled intruders cause spurious singularities, 'spikes' in e.g. a potential curve. The two keywords SHIFT and IMAGINARY SHIFT (mutually exclusive) will introduce a shift in the energy denominators, thus avoiding singularities, and will also correct the energy for the use of this shift. The net effect is that the energy is almost unaffected except in the vicinity of the weak singularity, which is removed. The SHIFT keyword adds a real shift, and the use of this procedure is well tested [27, 28]. The IMAGINARY SHIFT adds an imaginary quantity, and then uses the real value of the resulting second-order energy [29]. This offers some advantage, in particular for weak intruder states.

In some cases, where one can expect strong interaction between different CASSCF wave functions, it is advisable to use the Multi-State (MS) CASPT2 method [12]. A second order effective Hamiltonian is constructed for a number of CASSCF wave functions obtained in a state-average calculation. This introduces interaction matrix elements at second order between the different CASSCF states. The effective Hamiltonian is diagonalized to obtain the final second order energies. The program also produces a file, *JOBMIX*, with the new effective zeroth order wave functions, which are linear combinations of the original CASSCF states. This method has been used successfully to separate artificially mixed valence and Rydberg states and for transition metal compounds with low lying excited states of the same symmetry as the ground state.

It is clear from the discussion above that it is not a 'black box' procedure to perform CASPT2 calculations on excited states. It is often necessary to iterate the procedure with modifications of the active space and the selection of roots in the CASSCF calculation until a stable result is obtained. Normally, the CASSCF calculations are performed as average calculations over the number of electronic states of interest, or a larger number of states. It is imperative that the result is checked before the CASPT2 calculations are performed. The solutions should contain the interesting states. If all of them are not there, the number of roots in the CASSCF calculation has to be increased. Suppose for example, that four states of a given symmetry are required. Two of them are valence excited states and two are Rydberg states. A CASSCF calculation is performed as an average over four roots. Inspection of the solution shows only one valence excited state, the other three are Rydberg states. After several trials it turns out that the second valence excited state occurs as root number seven in the CASSCF calculation. The reason for such a behavior is, of course, the very different dynamic correlation energies of the valence excited states as compared to the Rydberg states. It is important that the AO basis set is chosen to contain a good representation of the Rydberg orbitals, in order to separate them from the valence excited states. For more details on how to perform calculations on excited states we refer to the literature [27, 28] and section 10.5 of the examples manual.

The first order wave function is obtained in the CASPT2 program as an iterative solution to a large set of linear equations. The size of the equation system is approximately $n^2 * m^2/2$ where n is the sum of inactive and active orbitals and m is the sum of active and secondary orbitals. Symmetry will reduce the size with approximately a factor g_{sym}, the number of irreps of the point group.

CASPT2 produces a set of molecular orbitals that can be used as start orbitals for other programs or further calculations. A minimal CASSCF and CASPT2 gives orbitals and occupation numbers which can be used to design a proper larger calculation. By default, the orbitals are natural orbitals obtained from the density matrix of the (normalized) wave function through first order. However, the active/active block of that density matrix is not computed exactly. An approximation has been designed in such a way that the trace is correct, and the natural occupation numbers of active orbitals are between zero and two. Due to the approximation, any properties computed using these orbitals are inexact and can be used only qualitatively. An exact first order density matrix can be computed but this is more time-consuming. It is controlled by the keyword DENSity. Use this keyword to compute properties like dipole moments, etc. The most secure accurate way to do that is. however, to use finite field perturbation theory (FFPT).

8.3.1 Dependencies

The CASPT2 program needs the *JOBIPH* file from a RASSCF calculation, and in addition one- and two-electron integrals and some auxiliary files from SEWARD.

8.3.2 Files

Input files

CASPT2 will use the following input files: *ONEINT*, *ORDINT*,*RUNFILE*, *JOBIPH* (for more information see 7.1.2).

Output files

File	*Contents*
PT2ORB	Molecular orbitals.

8.3.3 Input

This section describes the input to the CASPT2 program, starting with its name:

```
&CASPT2
```

Keywords

Keyword	*Meaning*
TITLe	This keyword is followed by one title line.

MULTistate

Enter number of root states, and a list of which CI vector from the CASSCF calculation to use for each state, for example "`2 1 2`" would specify the first and second root. Also used for single-state calculations, when the root state is not the ground state, for example "`1 2`" would specify the second root. The special value "`all`" can be used if all the states included in the CASSCF orbital optimization (keyword CIRoot in `RASSCF`) are desired.

IPEAshift

This shift corrects the energies of the active orbitals and is specified in atomic units. It will be weighted by a function of the diagonal density matrix element D_{pp}. This option is used to modify the standard definition of the zeroth order Hamiltonian (H_0), which includes an IPEA shift of 0.25 [25]. The modification of H_0 has been introduced (Nov 2005) to reduce the systematic error which leads to a relative overestimation of the correlation energy for open shell system. It also reduces the intruder problems. Default is to use an IPEA shift of 0.25.

IMAGinary

Add an imaginary shift to the external part of the zero order Hamiltonian. The correlation energy computed is the real part of the resulting complex perturbation energy. Also, a corrected value, obtained by Hylleraas' variational formula, is computed. See Ref. [29]. As with the real shift, this option is used to eliminate intruder problems.

SHIFt

Add a shift to the external part of the zero order Hamiltonian. See Refs. [29, 27, 23]. In addition to the conventionally computed second order energy value, another energy obtained by Hylleraas' variational formula is computed. This energy is then very close to the unshifted energy, except close to singularities due to intruders. This option should only be used to eliminate intruder state problems.

AFREeze

This keyword is used to select atoms for defining the correlation orbital space for the CASPT2 calculation. Assume that you have a large molecule where the activity takes place in a limited region (the active site). It could be a metal atom with its surrounding ligands. You can then use this option to reduce the size of the CASPT2 calculation by freezing and deleting orbitals that have only a small population in the active site. An example: The cobalt imido complex Co^{III}(nacnac)(NPh) has 43 atoms. The active site was cobalt and the surrounding ligand atoms. Using the AFRE option reduces the time for the CASPT2 calculation from 3 hrs to 3 min with a loss of accuracy in relative energies for 24 electronic states of less than 0.1 eV. The first line after the keyword contains the number of selected atoms then the selection thresholds (the recommended value is 0.1 or less). An additional line gives the names of the atoms as defined in the Seward input. Here is a sample input for the cobalt complex mentioned above.

```
AFREeze
 6 0.10 0.00
 Co N1 N2 C5 C6 C7
```

This input means that inactive orbitals with less than 0.1 of the density on the active sites will be frozen, while no virtual orbitals will be deleted.

LOVCaspt2

“Freeze-and-Delete” type of CASPT2, available only in connection with Cholesky or RI. Needs (pseudo)canonical orbitals from RASSCF. An example of input for the keyword LOVC is the following:

```
LovCASPT2
 0.3
DoMP2  (or DoEnv)
```

In this case, both occupied and virtual orbitals (localized by the program) are divided in two groups: those mainly located on the region determined (automatically) by the spatial extent of the active orbitals (“active site”), and the remaining ones, which are obviously “outside” this region. The value of the threshold (between 0 and 1) is used to perform this selection (in the example, 30% of the gross Mulliken population of a given orbital on the active site). By default, the CASPT2 calculation is performed only for the correlating orbitals associated with the active site. The keyword DoMP2 is optional and forces the program to perform also an MP2 calculation on the “frozen region”. Alternatively, one can specify the keyword VirAll in order to use all virtual orbitals as correlating space for the occupied orbitals of the active site. A third possibility is to use the keyword DoEnv to compute the energy of the environment as total MP2 energy minus the MP2 energy of the active site.

FNOCaspt2

Performs a Frozen Natural Orbital (FNO) CASPT2 calculation, available only in combination with Cholesky or RI integral representation. Needs (pseudo)canonical orbitals from RASSCF. An example of input for the keyword FNOC is the following:

```
FNOCaspt2
0.4
DoMP2
```

The keyword FNOC has one compulsory argument (real number in]0,1]) specifying the fraction of virtual orbitals (in each irrep) to be retained in the FNO-CASPT2 calculation. The keyword DoMP2 is optional and used to compute the (estimated) correction for the truncation error.

FOCKtype

Use an alternative Fock matrix. The default Fock matrix is described in [18, 19] and the other original CASPT2 references. The three different modifications named G1, G2 and G3 are described in [26]. Note: from 6.4 it is not recommended to use this keyword but stay with the IPEA modified H_0, which is default.

FROZen

This keyword is used to specify the number of frozen orbitals, i.e. the orbitals that are not correlated in the calculation. The next line contain the number of frozen orbitals per symmetry. The default is to freeze the max of those that were frozen in the `RASSCF` calculation and the deep core orbitals. The frozen orbitals are always the first ones in each symmetry.

DELEted
This keyword is used to specify the number of deleted orbitals, i.e. the orbitals that are not used as correlating orbitals in the calculation. The next line contain the number deleted orbitals per symmetry. The default is to delete those that were deleted in the RASSCF calculation. The deleted orbitals are always the last ones in each symmetry.

DENSity
Computes the full density matrix from the first order wave function, rather than approximated as is the (faster) default option. Used to compute CASPT2 properties, such as dipole moments, etc.

RFPErt
This keyword makes the program add reaction field effects to the energy calculation. This is done by adding the reaction field effects to the one-electron Hamiltonian as a constant perturbation, i.e. the reaction field effect is not treated self consistently. The perturbation is extracted from RUNOLD, if that file is not present if defaults to RUNFILE.

RLXRoot
Specifies which root to be relaxed in a geometry optimization of a multi state CASPT2 wave function. Defaults to the highest root or root defined by the same keyword in the RASSCF module.

THREsholds
On next line, enter two thresholds: for removal of zero-norm components in the first-order perturbed wave function, and for removal of near linear dependencies in the first-order perturbed wave function. Default values are 1.0d-10 and 1.0d-08 respectively.

MAXIter
On next line, enter the maximum allowed number of iterations in a procedure for solving a system of linear equations using a conjugate gradient method. Default is 20. A gradient norm is reported. This gradient is a residual error from the CASPT2 equation solution and should be small, else the number of iterations must be increased.

CONVergence
On next line, enter the convergence threshold for the procedure described above. The iterative procedure is repeated until the norm of the residual (RNORM) is less than this convergence threshold. Default is 1.0d-06.

NOMIx
Normally, a Multi-State CASPT2 calculation produces new jobiph file named *JOBMIX*. It has the same CASSCF wave functions as the original ones, except that those CI vectors that was used in the Multi-State CASPT2 calculation have been mixed, using the eigenvectors of the effective Hamiltonian matrix as transformation coefficients. Keyword NOMIX prevents creation of this *JOBMIX* file.

NOMUlt
This keyword removes the multi-state part of the calculation and only runs a series of independent CASPT2 calculations for the roots specified by the MULTistate keyword. Useful when many roots are required, but multi-state is not needed, or desired. Note that a

`JOBMIX` file is produced anyway, but the vectors will not be mixed, and the energies will be single-state CASPT2 energies.

ONLY This keyword requires the MULTistate keyword, and is followed by an integer specifying one of the roots. In a Multistate calculation, it requests to compute the energy of only the specified root. However, the effective Hamiltonian coupling terms bewteen this root and all the others included in the Multistate treatement will be computed and printed out. This output will be used in a subsequent calculation, in conjuction with the EFFE keyword.

EFFE This keyword requires the MULTistate keyword. It is followed by a matrix of real numbers, specifying the effective Hamiltonian couplings, as provided in a previous calculation using the ONLY keyword. In a Multistate calculation over, e.g., 3 states, 3 separate calculations with the ONLY keyword will be performed, possibly on separate computing nodes, so as to speed up the overall process. The three couplings vectors will be given to the EFFE keyword in matrix form, i.e. the first column is made by the couplings of the first computed root, etc. The program will then quickly compute the Multistate energies.

NOORbitals In calculations with very many orbitals, use this keyword to skip the printing of the MO orbitals.

PROPerties Normally, a CASPT2 calculation does not produce any density matrix, natural orbitals or properties in order to save time and memory (especially for large calculations). Keyword PROP activates these calculations, at the expense of (some) extra time and memory (especially if used together with the DENS keyword).

NOTRansform This keyword specifies that the wave function should not be transformed to use quasi-canonical orbitals, even if `CASPT2` does not know if this was done or not and by default would do such a transformation. Effectively, the Fock matrix is replaced by a diagonal approximation in the input orbital system.

TRANsform This keyword specifies that the wave function should be transformed to use pseudo-canonical orbitals, even if this was specified as option to the CASSCF calculation and should be unnecessary. (Default is: to transform when necessary, and not else.)

OFEMbedding Adds an Orbital-Free Embedding potential to the hamiltonian. Available only in combination with Cholesky or RI integral representation. No arguments required. The runfile of the environment subsystem (AUXRFIL) must be available.

GHOStdelete Excludes from PT2 treatment orbitals localized on ghost atoms. A threshold for this selection must be specified.

OUTPut Use this keyword, followed by any of the words BRIEF, DEFAULT, or LONG, to control the extent of orbital listing. BRIEF gives a very short orbital listing, DEFAULT a normal output, and LONG a detailed listing.

PRWF — This keyword is used to specify the threshold for printing the CI coefficients, default is 0.05.

PRSD — This keyword is used to request that not only CSFs are printed with the CI coefficients, but also the determinant expansion.

The given default values for the keywords Convergence and Thresholds normally give a second order energy which is correct in eight decimal places.

Input example

```
 &CASPT2
Title
 The water molecule
Density matrix
```

The CASPT2 energy and density matrix is computed for the water molecule with the O(1s) orbital frozen. The standard IPEA-H_0 is used.

8.4 CASVB

This program can be used in two basic modes:

a) variational optimization of quite general types of nonorthogonal MCSCF or modern valence bond wavefunctions

b) representation of CASSCF wavefunctions in modern valence form, using overlap- (*relatively inexpensive*) or energy-based criteria.

For generating representations of CASSCF wavefunctions, the program is invoked by the command CASVB. For variational optimization of wavefunctions it is normally invoked inside RASSCF by the sub-command VB (see 8.40.4).

Bibliography: see [30, 31, 32, 33].

8.4.1 Dependencies

The CASVB program needs the *JOBIPH* file from a RASSCF calculation, and in addition also the *ONEINT* and *ORDINT* files from SEWARD.

8.4.2 Files

Input files

CASVB will use the following input files: *ONEINT*, *ORDINT*, *RUNFILE*, *JOBIPH*, (for more information see 7.1.2), and *VBWFN* with valence bond wavefunction information (orbital and structure coefficients).

Output files

File	*Contents*
JOBIPH	On exit, the RASSCF interface file is overwritten with the CASVB wavefunction.
VBWFN	Valence bond wavefunction information (orbital and structure coefficients).

8.4.3 Input

This section describes the input to the CASVB program. The input for each module is preceded by its name like:

```
&CASVB
```

Keywords

Optional keywords

Keyword	*Meaning*
END OF INPUT	This marks the end of the input to the program.

Optional keywords to define the CASSCF wavefunction. Not generally required because values stored in the job interface file or used by the RASSCF program will normally be appropriate.

Keyword	*Meaning*
FROZEN	Specifies frozen orbitals, as in the RASSCF program.
INACTIVE	Specifies inactive orbitals, as in the RASSCF program.
NACTEL	Specifies the number of active electrons, as in the RASSCF program.
RAS2	Specifies RAS2 orbitals, as in the RASSCF program.
SPIN	Specifies the total spin, as in the RASSCF program.
SYMMETRY	Specifies the CASSCF wavefunction symmetry, as in the RASSCF program.

Optional keywords to define the VB wavefunction

Keyword	*Meaning*

CON

The spatial VB configurations are defined in terms of the active orbitals, and may be specified using one or more `CON` keywords:

CON
n_1 n_2 n_3 n_4 . . .

The configurations can be specified by occupation numbers, so that n_i is the occupation of the ith valence bond orbital. Alternatively a list of $Nact$ orbital numbers (in any order) may be provided – the program determines which definition applies. The two specifications `1 0 1 2` and `1 3 4 4` are thus equivalent.

Input configurations are reordered by `CASVB`, so that configurations have non-decreasing double occupancies. Configurations that are inconsistent with the value for the total spin are ignored.

If no configurations are specified the single 'covalent' configuration $\phi_1\phi_2\cdots\phi_{Nact}$ is assumed.

COUPle

COUPLE
key

key may be chosen from `KOTANI` (default), `RUMER`, `PROJECT` or `LTRUMER`, specifying the scheme for constructing the spin eigenfunctions used in the definition of valence bond structures. `PROJECT` refers to spin functions generated using a spin projection operator, `LTRUMER` to Rumer functions with the so-called "leading term" phase convention.

WAVE

WAVE
N S_1 S_2 . . .

This keyword can be used to specify explicitly the number of electrons and spin(s) to be used with a configuration list. If N is less than the present number of active electrons, the input wavefunction fragment is assumed to form part of a direct product. Otherwise, the spins specified may be greater than or equal to the SPIN value specified as input to the `RASSCF` program. Defaults, for both N and S, are the values used by `RASSCF`.

Optional keywords for the recovery and/or storage of orbitals and vectors

Keyword — *Meaning*

STARt

START
key-1=*filename*-1
key-2=*filename*-2
. . .

Specifies input files for VB wavefunction (*key-i* = VB), CASSCF CI vector (*key-i* = CI) and/or CASSCF molecular orbitals (*key-i* = MO). By default, the required information is taken from the file *`JOBOLD`*.

SAVE	SAVE *key*-1=*filename*-1 *key*-2=*filename*-2 . . . Specifies output files for VB wavefunction (*key*-i = VB) and/or the VB CI vector (*key*-i = VBCI). By default, the VB CI vector is written to the file JOBIPH.

Optional keywords to override the starting guess

Keyword	*Meaning*
GUESs	GUESS *key-1 . . .* *key-2 . . .* `ENDGUESs` The GUESS keyword initiates the input of a guess for the valence bond orbitals and/or structure coefficients. *key-i* can be either ORB or STRUC. These keywords modify the guess provided by the program. It is thus possible to modify individual orbitals in a previous solution so as to construct the starting guess. The ENDGUESs keyword terminates the guess input. ORB $i\ c_1\ c_2\ \ldots c_{mact}$ Specifies a starting guess for valence bond orbital number i. The guess is specified in terms of the *mact* active MOs defining the CASSCF wavefunction. STRUC $c_1\ c_2\ \ldots c_{NVB}$ Specifies a starting guess for the NVB structure coefficients. If this keyword is not provided, the perfect-pairing mode of spin coupling is assumed for the spatial configuration having the least number of doubly occupied orbitals. Note that the definition of structures depends on the value of `COUPLE`. Doubly occupied orbitals occur first in all configurations, and the spin eigenfunctions are based on the singly occupied orbitals being in ascending order.
ORBPerm	ORBPERM $i_1\ \ldots i_{mact}$ Permutes the orbitals in the valence bond wavefunction and changes their phases according to $\phi'_j = \mathrm{sign}(i_j)\phi_{\mathrm{abs}(i_j)}$. The guess may be further modified using the GUESS keyword. Additionally, the structure coefficients will be transformed according to the given permutation (note that the configuration list must be closed under the orbital permutation for this to be possible).

Optional keywords for optimization control

Keyword	*Meaning*
CRIT	CRIT *method* Specifies the criterion for the optimization. *method* can be OVERLAP or ENERGY (OVERLAP is default). The former maximizes the normalized overlap with the CASSCF wavefunction: $\max\left(\frac{\langle\Psi_{CAS}\|\Psi_{VB}\rangle}{(\langle\Psi_{VB}\|\Psi_{VB}\rangle)^{1/2}}\right)$ and the latter simply minimizes the energy: $\min\left(\frac{\langle\Psi_{VB}\|\hat{H}\|\Psi_{VB}\rangle}{\langle\Psi_{VB}\|\Psi_{VB}\rangle}\right).$
MAXIter	MAXITER N_{iter} Specifies the maximum number of iterations in the second-order optimizations. Default is N_{iter}=50.
(NO)CASProj	(NO)CASPROJ With this keyword the structure coefficients are picked from the transformed CASSCF CI vector, leaving only the orbital variational parameters. For further details see the bibliography. This option may be useful to aid convergence.
SADDle	SADDLE *n* Defines optimization onto an n^{th}-order saddle point. See also T. Thorsteinsson and D. L. Cooper, Int. J. Quant. Chem. **70**, 637–50 (1998).
(NO)INIT	(NO)INIT Requests a sequence of preliminary optimizations which aim to minimize the computational cost while maximizing the likelihood of stable convergence. This feature is the default if no wavefunction guess is available and no OPTIM keyword specified in the input.
METHod	METHOD *key* Selects the optimization algorithm to be used. *key* can be one of: FLETCHER, TRIM, TRUSTOPT, DAVIDSON, STEEP, VB2CAS, AUGHESS, AUG2, CHECK, DFLETCH, NONE, or SUPER. Recommended are the direct procedures DFLETCH or AUGHESS. For general saddle-point optimization TRIM is used. Linear (CI only) optimization problems use DAVIDSON. NONE suspends optimization, while CHECK carries out a finite-difference check of the gradient and Hessian. The default algorithm chosen by `CASVB` will be usually be adequate.

TUNE

TUNE
. . .

Enables the input of individual parameters to be used in the optimization procedure (*e.g.* for controlling step-size selection and convergence testing). Details of the values used are output if *print*(3)≥3 is specified. For expert use only.

OPTIm

More than one optimization may be performed in the same `CASVB` run, by the use of OPTIM keywords:

OPTIM
[. . .
ENDOPTIM]

The subcommands may be any optimization declarations defined in this section, as well as any symmetry or constraints specifications. Commands given as arguments to OPTIM will apply only to this optimization step, whereas commands specified outside will act as default definitions for all subsequent `OPTIM` specifications.

The OPTIM keyword need not be specified if only one optimization step is required,

When only a machine-generated guess is available, `CASVB` will attempt to define a sequence of optimization steps that aims to maximize the likelihood of successful convergence (while minimizing CPU usage). To override this behaviour, simply specify one or more OPTIM keywords. The ENDOPTIm keyword marks the end of the specifications of an optimization step.

ALTErn

A loop over two or more optimization steps may be specified using:

ALTERN
Niter
. . .
ENDALTERN

The program will repeat the specified optimization steps until either all optimizations have converged, or the maximum iteration count, *Niter*, has been reached. The ENDALTErn keyword marks the end of the specification of an ALTERN loop.

Optional keywords for definitions of molecular symmetry and any constraints on the VB wavefunction

Keyword *Meaning*

SYMElm

Various issues associated with symmetry-adapting valence bond wavefunctions are considered, for example, in: T. Thorsteinsson, D. L. Cooper, J. Gerratt and M. Raimondi, Theor. Chim. Acta **95**, 131 (1997).

SYMELM
label sign

Initiates the definition of a symmetry operation referred to by *label* (any three characters). *sign* can be + or −; it specifies whether the total wavefunction is symmetric or antisymmetric under this operation, respectively. A value for *sign* is not always necessary but, if provided, constraints will be put on the structure coefficients to ensure that the wavefunction has the correct overall symmetry (note that the configuration list must be closed under the orbital permutation induced by *label* for this to be possible). The default for *label* is the identity.

The operator is defined in terms of its action on the active MOs as specified by one or more of the keywords IRREPS, COEFFS, or TRANS. Any other keyword, including optional use of the ENDSYMELM keyword, will terminate the definition of this symmetry operator.

IRREPS
$i_1\ i_2\ \ldots$

The list $i_1\ i_2\ \ldots$ specifies which irreducible representations (as defined in the CASSCF wavefunction) are antisymmetric with respect to the *label* operation. If an irreducible representation is not otherwise specified it is assumed to be symmetric under the symmetry operation.

COEFFS
$i_1\ i_2\ \ldots$

The list $i_1\ i_2\ \ldots$ specifies which individual CASSCF MOs are antisymmetric with respect to the *label* operation. If an MO is not otherwise specified, it is assumed to be symmetric under the symmetry operation. This specification may be useful if, for example, the molecule possesses symmetry higher than that exploited in the CASSCF calculation.

TRANS
$n_{dim}\ i_1\ \ldots i_{n_{dim}}\ c_{11}\ c_{12}\ \ldots c_{n_{dim}n_{dim}}$

Specifies a general n$_{dim}$ × n_{dim} transformation involving the MOs $i_1,\ \ldots i_{n_{dim}}$, specified by the *c* coefficients. This may be useful for systems with a two- or three-dimensional irreducible representation, or if localized orbitals define the CASSCF wavefunction. Note that the specified transformation must always be orthogonal.

ORBREL In general, for a VB wavefunction to be symmetry-pure, the orbitals must form a representation (not necessarily irreducible) of the symmetry group. Relations between orbitals under the symmetry operations defined by `SYMELM` may be specified according to:

ORBREL
$i_1\ i_2$ *label1 label2 …*

Orbital i_1 is related to orbital i_2 by the sequence of operations defined by the *label* specifications (defined previously using `SYMELM`). The operators operate right to left. Note that i_1 and i_2 may coincide. Only the minimum number of relations required to define all the orbitals should be provided; an error exit will occur if redundant `ORBREL` specifications are found.

(NO)SYMProj

As an alternative to incorporating constraints, one may also ensure correct symmetry of the wavefunction by use of a projection operator:

(NO)SYMPROJ
[$irrep_1$ $irrep_2$ ...]

The effect of this keyword is to set to zero the coefficients in unwanted irreducible representations. For this purpose, the symmetry group defined for the CASSCF wavefunction is used (always a subgroup of D_{2h}). The list of irreps in the command specifies which components of the wavefunction should be kept. If no irreducible representations are given, the current wavefunction symmetry is assumed. In a state-averaged calculation, all irreps are retained for which a non-zero weight has been specified in the wavefunction definition. The SYMPROJ keyword may also be used in combination with constraints.

FIXOrb

FIXORB
i_1 i_2 ...

This command freezes the orbitals specified in the list i_1 i_2 ... to that of the starting guess. Alternatively the special keywords ALL or NONE may be used. These orbitals are eliminated from the optimization procedure, but will still be normalized and symmetry-adapted according to any ORBREL keywords given.

FIXStruc

FIXSTRUC
i_1 i_2 ...

Freezes the coefficients for structures i_1, i_2,... . Alternatively the special keywords ALL or NONE may be used. The structures are eliminated from the optimization procedure, but may still be affected by normalization or any symmetry keywords present.

DELStruc

DELSTRUC
i_1 i_2,...

Deletes the specified structures from the wavefunction. The special keywords ALL or NONE may be used. This specification should be compatible with the other structure constraints present, as defined by SYMELM and ORBREL.

ORTHcon

`ORTHCON`
key-1 ...
key-2 ...
...

The `ORTHCON` keyword initiates the input of orthogonality constraints between pairs/groups of valence bond orbitals. The sub-keywords *key-i* can be any of `ORTH`, `PAIRS`, `GROUP`, `STRONG` or `FULL`. Orthogonality constraints should be used with discretion. Note that orthogonality constraints for an orbital generated from another by symmetry operations (using the `ORBREL` keyword) cannot in general be satisfied. The `ENDORTHcon` keyword can be used to terminate the input of orthogonality constraints.

`ORTH` i_1 i_2 ...

Specifies a list of orbitals to be orthogonalized. All overlaps between pairs of orbitals in the list are set to zero.

`PAIRS` i_1 i_2 ...

Specifies a simple list of orthogonalization pairs. Orbital i_1 is made orthogonal to i_2, i_3 to i_4, etc.

`GROUP` *label* i_1 i_2 ...

Defines an orbital group to be used with the `ORTH` or `PAIRS` keyword. The group is referred to by *label* which can be any three characters beginning with a letter a–z. Labels defining different groups can be used together or in combination with orbital numbers in `ORTH` or `PAIRS`. i_1 i_2 ... specifies the list of orbitals in the group. Thus the combination `GROUP` AAA 1 2 `GROUP` BBB 3 4 `ORTH` AAA BBB will orthogonalize the pairs of orbitals 1-3, 1-4, 2-3 and 2-4.

`STRONG`

This keyword is short-hand for strong orthogonality. The only allowed non-zero overlaps are between pairs of orbitals ($2n-1$, $2n$).

`FULL`

This keyword is short-hand for full orthogonality and is mainly useful for testing purposes.

Optional keywords for wavefunction analysis

Keyword — *Meaning*

CIWEights

For further details regarding the calculation of weights in `CASVB`, see T. Thorsteinsson and D. L. Cooper, J. Math. Chem. **23**, 105-26 (1998).

`CIWEIGHTS`
key1 key2 ... [$N_{\rm conf}$]

Prints weights of the CASSCF wavefunction transformed to the basis of nonorthogonal VB structures. For the *key* options see `VBWEIGHTS` below. Note that the evaluation of inverse overlap weights involves an extensive computational overhead for large active spaces. Weights are given for the total CASSCF wavefunction, as well as the orthogonal complement to Ψ_{VB}. The default for the number of configurations requested, $N_{\rm conf}$, is 10. If $N_{\rm conf}=-1$ all configurations are included.

REPOrt

`REPORT`
[...
`ENDREPORT`]

Outputs orbital/structure coefficients and derived information. The `ENDREPOrt` keyword can be used to mark the end of the specification of a report step.

(NO)SCORr

`(NO)SCORR`

With this option, expectation values of the spin operators $(\hat{s}_\mu + \hat{s}_\nu)^2$ are evaluated for all pairs of μ and ν. Default is `NOSCORR`. The procedure is described by: G. Raos, J. Gerratt, D. L. Cooper and M. Raimondi, Chem. Phys. **186**, 233–250 (1994); ibid, 251–273 (1994); D. L. Cooper, R. Ponec, T. Thorsteinsson and G. Raos, Int. J. Quant. Chem. **57**, 501–518 (1996).

This analysis is currently only implemented for spin-coupled wavefunctions.

VBWEights

For further details regarding the calculation of weights in `CASVB`, see T. Thorsteinsson and D. L. Cooper, J. Math. Chem. **23**, 105-26 (1998).

`VBWEIGHTS`
key1 key2 ...

Calculates and outputs weights of the structures in the valence bond wavefunction Ψ_{VB}. *key* specifies the definition of nonorthogonal weights to be used, and can be one of:

`CHIRGWIN` Evaluates Chirgwin-Coulson weights (see: B. H. Chirgwin and C. A. Coulson, Proc. Roy. Soc. Lond. **A201**, 196 (1950)).

`LOWDIN` Performs a symmetric orthogonalization of the structures and outputs the subsequent weights.

`INVERSE` Outputs “inverse overlap populations” as in G. A. Gallup and J. M. Norbeck, Chem. Phys. Lett. **21**, 495–500 (1973).

`ALL` All of the above.

`NONE` Suspends calculation of structure weights.

The commands `LOWDIN` and `INVERSE` require the overlap matrix between valence bond structures, so that some additional computational overhead is involved.

Optional keywords for further general options

Keyword *Meaning*

PREC

`PREC`
iprec iwidth

Adjusts the precision for printed quantities. In most cases, *iprec* simply refers to the number of significant digits after the decimal point. Default is *iprec*=+8. *iwidth* specifics the maximum width of printed output, used when determining the format for printing arrays.

PRINt

`PRINT`

i_1 i_2 ...

Each number specifies the level of output required at various stages of the execution, according to the following convention:

- -1 No output except serious, or fatal, error messages.
- 0 Minimal output.
- 1 Standard level of output.
- 2 Extra output.

The areas for which output can be controlled are:

- i_1 Print of input parameters, wavefunction definitions, etc.
- i_2 Print of information associated with symmetry constraints.
- i_3 General convergence progress.
- i_4 Progress of the 2nd-order optimization procedure.
- i_5 Print of converged solution and analysis.
- i_6 Progress of variational optimization.
- i_7 File usage.

For all, the default output level is +1. If $i_5 \geq 2$ VB orbitals will be printed in the AO basis (provided that the definition of MOs is available).

SHSTruc

Prints overlap and Hamiltonian matrices between VB structures.

STATs

`STATS`

Prints timing and usage statistics.

Input example

```
&seward
symmetry
x y
basis set
c.sto-3g....
c 0 0 -0.190085345
end of basis
basis set
h.sto-3g....
h 0 1.645045225 1.132564974
end of basis
&scf
occupied
3 0 1 0
&rasscf
inactive
1 0 0 0
ras2
3 1 2 0
nactel
6 0 0
lumorb
&casvb
```

Viewing and plotting VB orbitals

In many cases it can be helpful to view the shape of the converged valence bond orbitals, and Molcas therefore provides two facilities for doing this. For the Molden program, an interface file is generated at the end of each CASVB run (see also Section 9.1). Alternatively a CASVB run may be followed by RASSCF to get orbitals (Section 8.40) and GRID_IT with the VB specification (Section 8.19), in order to generate a three-dimensional grid, for viewing, for example, with LUSCUS program.

8.5 CCSDT

CCSDT performs the iterative single determinant CCSD procedure for open shell systems and the noniterative triple contribution calculation to the CCSD energy. For further details the reader is referred to the sections 6.14 and 10.4 of the tutorials and examples manual.

8.5.1 Dependencies

CCSDT requires a previous run of the RASSCF program to produce orbital energies, Fock matrix elements, wave function specification, and some other parameters stored in file *JOBIPH*. The RASSCF program should be run with the options that produce canonical output orbitals, which is not default. CCSDT also requires transformed integrals produced by MOTRA and stored in the files *TRAONE* and *TRAINT*.

It is well known that the CCSD procedure brings the spin contamination into the final wave function $|\Psi\rangle$ even in the case where the reference function $|\Phi\rangle$ is the proper spin eigenfunction. The way how to reduce the spin contamination and mainly the number of independent amplitudes is to introduce the spin adaptation. Besides the standard nonadapted (spinorbital) CCSD procedure this program allows to use different levels of spin adaptation of CCSD amplitudes (the recommended citations are Refs. [34, 35]):

- DDVV T2 adaptation.
  This is the most simple and most universal scheme, in which only the dominant part of T2 amplitudes, namely those where both electrons are excited from *doubly occupied (inactive)* to *virtual (secondary)* orbitals, are adapted. The remaining types of amplitudes are left unadapted, i.e. in the spinorbital form. This alternative is an excellent approximation to the full adaptation and can be used for any multiplet.
- Full T1 and T2 adaptation (only for doublet states yet).
  In this case full spin adaptation of all types of amplitudes is performed. In the present implementation this version is limited to systems with the single unpaired electrons, i.e. to the doublet states only.

Besides these two possibilities there are also available some additional partial ones (see keyword ADAPTATION in Section 8.5.3). These adaptations are suitable only for some specific purposes. More details on spin adaptation in the CCSD step can be found in Refs. [35, 34, 36]. The current implementation of the spin adaptation saves no computer time. A more efficient version is under development.
The noniterative triples calculation can follow these approaches:

- CCSD + T(CCSD) - according to Urban et al. [37]
- CCSD(T) - according to Raghavachari et al. [38]
- CCSD(T) - according e.g. to Watts et al. [13]

Actual implementation and careful analysis and discussion of these methods is described in Ref. [14], which is a recommended reference for this program.
The first alternative represents the simplest noniterative T3 treatment and contains only pure $\langle T3|WT2\rangle$ term. Second possibility represents the well known extension to the first one by the $\langle T3|WT1\rangle$ term (W is the two electron perturbation). For closed shell systems this is the most popular and most frequently used noniterative triples method. For single determinant open shell systems, described by the ROHF reference function standard (Raghavachari et. al.) method needs to be extended by the additional fourth order energy term, namely $\langle T3|UT2\rangle$ (U is the off-diagonal part of the Fock operator).
In contrast to the iterative CCSD procedure, noniterative approaches are not invariant with respect to the partitioning of the Hamiltonian. Hence, we obtain different results using orbital energies, Fock matrix elements or some other quantities in the denominator. According to our experiences [14], diagonal Fock matrix elements in the denominator represent the best choice. Using of other alternatives requires some experience. Since the triple excitation contribution procedure works strictly within the restricted formalism, resulting noniterative triples contributions depend also on the choice of the reference function. However, differences between this approach (with the reference function produced by a single determinant RASSCF procedure and the diagonal Fock matrix elements considered in the denominator) and the corresponding invariant treatment (with the semicanonical orbitals) are found to be chemically negligible.
For noniterative T3 contribution both non-adapted (spin-orbital) and spin-adapted CCSD amplitudes can be used. For more details, see Ref. [14].

8.5.2 Files

Input files

CCSDT will use the following input files: *TRAONE*, *TRAINT*,*RUNFILE*, *JOBIPH*, (for more information see 7.1.2).

Output files

File	*Contents*
RSTART	file with CC amplitudes and CC energy. The name of the file can be changed using keyword RESTART. It contains restart information, like T1aa,T1bb,T2aaaa,T2bbbb,T2abab, CC energy and the number of iterations.

T3hfxyy	These files contain integrals of $\langle ia\|bc\rangle$ type where x represents the symmetry and yy the value of the given index i. The number of these files is equal to the number of α occupied orbitals (*inactive + active.*)

8.5.3 Input

The input for each module is preceded by its name like:

```
&CCSDT
```

Keyword	*Meaning*
TITLe	This keyword should be followed by precisely one title line. It should not begin with a blank (else it will not be printed!) This keyword is *optional.*
CCSD	This keyword specifies that only CCSD calculation will follow and the integrals will be prepared for the CCSD procedure only. This keyword is *optional.* (Default=OFF)
CCT	This keyword specifies that after CCSD calculation also noniterative T3 step will follow. For such calculations this key must be switched on. The integrals for the triple contribution calculation will then be prepared. This keyword is *optional.* (Default=ON)
ADAPtation	The parameter on the following line defines the type of spin adaptations of CCSD amplitudes. 0 - no spin adaptation - full spinorbital formalism 1 - T2 DDVV spin adaptation 2 - T2 DDVV + T1 DV spin adaptation (only recommended for specific purposes, since the adaptation of T1 included incompletely) 3 - full T2 and T1 spin adaptation (in current implementations limited to doublets only) 4 - full T2 adaptation without SDVS coupling (for doublets only) This keyword is *optional.* (Default=0)
DENOminators	The parameter on the following line specifies the type of denominators that will be used in the CCSD procedure. 0 - diagonal Fock matrix elements (different for α and β spins) 1 - spin averaged diagonal Fock matrix elements - $\frac{f_{\alpha\alpha}+f_{\beta\beta}}{2}$ 2 - orbital energies In some cases alternatives 1 and 2 are identical. For nonadapted CCSD calculations the resulting CCSD energy is invariant with respect to the selection of denominators. However, convergence may be affected. In the present implementation a symmetric denominators (i.e. the

input 1 or 2) should be used for spin adapted CCSD calculations. This keyword is *optional.* (Default=0)

SHIFts

Following line contains *socc* and *svirt* levelshift values for occupied and virtual orbitals respectively. Typical values are in the range 0.0 - 0.5 (in *a.u.*)
$dp(occ) = dp(occ) - socc$
$dp(virt) = dp(virt) + svirt$
For spin adaptations 3 and 4 only inactive (D) and active (V) orbitals will be shifted, due to the character of the adaptation scheme. For other cases all orbitals are shifted.
This keyword is *optional.* (Defaults: $socc = 0.0, svirt = 0.0$)

TRIPles

The parameter on the following line specifies the type of noniterative triples procedure. There are three different types of perturbative triples available (see Section 8.5).
0 - CCSD approach (no triples step)
1 - CCSD+T(CCSD) according to Urban et. al [37]
2 - CCSD(T) according to Raghavachari et. al. [38]
3 - CCSD(T) according e.g. to Watts et. al. [13]
This keyword is *optional.* (Default=3)

T3DEnominators

The parameter on the following line specifies the type of denominators that will be used in noniterative triples procedure.
0 - diagonal Fock matrix elements (different for α and β spins)
1 - spin averaged diagonal Fock matrix elements - $\frac{f_{\alpha\alpha}+f_{\beta\beta}}{2}$
2 - orbital energies
In some cases alternatives 1 and 2 are identical. This keyword is *optional.* (Default=0)

T3SHifts

The following line contains *socc* and *svirt* levelshift values for occupied and virtual orbitals respectively. Typical values are in the range 0.0 - 0.5 (in *a.u.*)
$dp(occ) = dp(occ) - socc$
$dp(virt) = dp(virt) + svirt$
In contrast to the iterative CCSD procedure, in noniterative T3 step results are not invariant with respect to the denominator shifting. It is extremely dangerous to use any other than 0.0 0.0 shifts here, since resulting T3 energy may have no physical meaning. This keyword may be useful only in estimating some trends in resulting energy, however, using of default values is strongly recommended.
This keyword is *optional.* (Defaults: $socc = 0.0, svirt = 0.0$)

ITERations

This keyword is followed on the next line by the maximum number of iterations in the CCSD procedure. In the case of the RESTART run this is the number of last allowed iteration, since counting of iterations in RESTART run starts from the value taken from the ***RSTART*** file. This keyword is *optional.* (Default=30)

ACCUracy

The real value on the following line defines the convergence criterion on CCSD energy. This keyword is *optional.* (Default=1.0d-7)

END of input
This keyword indicates that there is no more input to be read. This keyword is *optional.*

EXTRapolation
This keyword switches on the DIIS extrapolation. This keyword is followed by two additional parameters on the next line *n1* and *n2.*
n1 - specifies the first iteration, in which DIIS extrapolation procedure will start for the first time. This value must not be less then *n2*, recommended value is 5-7.
n2 - specifies the size of the DIIS procedure, i.e. the number of previous CCSD steps which will be used for new prediction. In the present implementation *n2* is limited to 2-4.
This keyword is *optional.* (Default=OFF)

PRINt
The parameter on the next line specifies the level of output printing
0 - minimal level of printing
1 - medium level of printing
2 - full output printing (useful for debugging purposes)
This keyword is *optional.* (Default=0)

LOAD
This keyword is followed by the line which specifies the name of the CCSD amplitudes and energy file. The default name is ***RSTART***, but it can be changed in CCSD step using RESTART keyword. This keyword is *optional.* (Default=***RSTART***)

RESTart
This keyword defines the restart conditions and modifies the name of the file, in which restart information (CC amplitudes, CC energy and the number of iterations) is saved. On the following two lines there are control key *nn* and the name of restart information storing file *name.*
nn - restart status key
0 - restart informations will be not saved
1 - restart informations will be saved after each iteration in *name.*
2 - restart run. CC amplitudes and energy will be taken from *name* file and the CCSD procedure will continue with these values as an estimate.
name - specifies the restart information storing key. The name is limited to 6 characters.
This keyword is *optional.* (Defaults: *nn*=1, *name*=RSTART)

IOKEy
This keyword specifies the input-output file handling.
1 - Internal Fortran file handling
2 - Molcas DA file handling
The default (1) is recommended in majority of cases, since when calculating relatively large systems with low symmetry, the size of some intermediate files produced may become large, what could cause some troubles on 32-bit machines (2 GB file size limit).

MACHinetyp
This keyword specifies which type of matrix multiplication is preferred on a given machine. The following line contains two parameters *nn,limit.*
nn =1 standard multiplication $A \times B$ is preferred

nn =2 transposed multiplication $A^T \times B$ is preferred
Parameter *limit* specifies the limit for using $A^T \times B$ multiplication, when *nn*=2. (It has no meaning for *nn*=1.)
If *size(A)/size(B)* $\geq$ *limit* - standard multiplication is performed,
size(A)/size(B) $<$ *limit* - transposed multiplication is performed.
(*size(A,B)* - number of elements in matrix A,B). Recommended value for *limit* is 2-3.
Using of transposed matrix (*nn*=2) multiplication may bring some computer time reduction only in special cases, however, it requires some additional work space. Default is optimal for absolute majority of cases.
This keyword is *optional.* (Default=1).

Note, that CCSD and CCT keywords are mutually exclusive.

8.5.4 How to run closed shell calculations using ROHF CC codes

First of all it should be noted here, that it is not advantageous to run closed shell calculations using ROHF CC codes, since in the present implementation it will require the same number of arithmetical operations and the core and disk space like corresponding open shell calculations.

Since ROHF CC codes are connected to the output of RASSCF code (through the *JOBIPH* file), it is necessary to run closed shell Hartree-Fock using the RASSCF program. This can be done by setting the number of active orbitals and electrons to zero (also by including only doubly occupied orbitals into the active space; this has no advantage but increases the computational effort). to guarantee the single reference character of the wave function.

The CC program will recognize the closed shell case automatically and will reorganize all integrals in a required form. For more information the reader is referred to the tutorials and examples manual.

Below is an input file for HF$^+$ CCSD(T) calculation.

```
 &CCSDT
Title
 HF(+) CCSD(T) input example
CCT
Triples
3
```

8.6 CHCC

The CHCC is a Closed-Shell Coupled-Clusters Singles and Doubles program based exclusively on the Cholesky (or RI) decomposed 2-electron integrals aimed towards calculation of large systems on highly parallel architectures. Use of point-group symmetry is not implemented. Main advantage compared to the CCSDT module in MOLCAS is in its more efficient parallelization and dramatically lowered memory (and eventually disk) requirements.

8.6.1 Dependencies

CHCC requires a previous run of the RHF SCF program to produce molecular orbitals and orbital energies stored in *RUNFILE*. The SCF program (as well as SEWARD) must be run in Cholesky/RI mode.

The algorithm used for almost complete elimination of the CHCC limits in calculated system size due to the computer memory bottleneck relies on blocking of the virtual orbitals. Number of blocks (further also referred to as the "large" segmentation, LARGE), N′, should be as small as possible, because increasing of the segmentation brings in more CPU and I/O overhead. Furthermore, blocking can be "fine tuned" by, so called, "small" segmentation (SMALL), N″, which affects only the (typically) most demanding O^2V^4 scaling terms. The "large" segmentation can range from 1 to 32, "small" segmentation from 1 to 8, but their product, i.e. "large x small" must be no more than 64.

Selected blocking also determines the number of "independent" parallel tasks that must be executed in each iteration of the CCSD equations. In other words, particular segmentation predetermines the optimal number of computational nodes (i.e., if the best possible parallelization is desired). If the requested "large" segmentation is N′, then N'^2 terms scaling as O^3V^3 and 1/2 N'^2 terms scaling as O^2V^4 result. Depending on which of these terms dominated in the calculations (O^3V^3 is more demanding for systems with large number of occupied orbitals and rather small basis set, while O^2V^4 dominated for relatively large basis sets, i.e. large number of virtual orbitals), number of these task should be divisible by the number of computational nodes for optimal performance. To make it simple, as a rule of thumb, $N'^2/2$ should be divisible by the number of nodes, since the O^3V^3 are typically twice less expensive then the O^2V^4 step. Otherwise, any reasonable (i.e. the number of tasks is larger than the number of computational nodes, obviously) combination is allowed.

8.6.2 Files

Input files

CHCC will use the following input files: *CHVEC*, *CHRED*, *CHORST*, *RUNFILE*, and *CHOR2F* (for more information see 7.1.2).

Output files

File	*Contents*
L0xxxx, L1xxxx, L2xxxx	MO-transformed Cholesky vectors
T2xxxx	T2 (ij, a′b′) excitation amplitudes
RstFil	Communication file containing T1 amplitudes, restart informations, etc.

8.6.3 Input

The input for each module is preceded by its name like:

```
&CHCC
```

Optional keywords

Keyword	*Meaning*
TITLe	This keyword is followed by one title line.
FROZen	Integer on the following line specifies number of inactive occupied orbitals in the CCSD calculation. (Default=0)
DELEted	Integer on the following line specifies number of inactive virtual orbitals in the CCSD calculation. (Default=0)
LARGe	Integer on the following line specifies the main segmentation of the virtual orbitals. Value must be between 1 (no segmentation) and 32. Product of Large and Small segmentation must be lower than 64. (Default=1)
SMALl	Integer on the following line specifies the auxiliary segmentation of the virtual orbitals. Value must be between 1 (no segmentation) and 8. Product of Large and Small segmentation must be lower than 64. Small segmentation doesn't generate extra parallel tasks. (Default=1)
CHSEgmentation	Integer on the following line specifies the block size of the auxiliary (Cholesky/RI) index. Value must lower than the minimal dimension of the auxiliary index on each computational node. (Default=100)
MHKEy	Integer on the following line specifies if library BLAS (MHKEy=1) or hard-coded fortran vector-vector, matrix-vector and matrix-matrix manipulation is used. (Default=1)
NOGEnerate	This keyword specifies that the pre-CCSD steps (regeneration of integrals from the Cholesky/RI vectors, etc.) are skipped. (Default=OFF)
ONTHefly	This keyword specifies that all integral types scaling steeper then O^2V^2 are generated "on-the-fly" from the Cholesky/RI vectors. Use of this keyword leads to dramatically savings of the disk resources, but leads to significant arithmetic overhead. Keywords "ONTHefly" and "PRECalculate" are mutually exclusive. (Default=OFF)
PRECalculate	This keyword specifies that all integral are precalculated before the CCSD iterative procedure starts. Use of this keyword leads to significant consumption of the disk space, especially is single-processor runs. (Default=ON)

NODIstribute This keyword (in combination with the ”PRECalculate” keyword) specifies that all integral are stored on each computational node. In case of all integrals being stored on each node, extra permutation symmetry can be applied, thus leading to significant savings of the disk space. However, in case of massively parallel runs (i.e. more than ≈8 nodes), savings from keeping only subset of integrals required on particular node are more significant than savings due to permutational symmetry. (Default=OFF)

JOINlkey The parameter on the following line specifies, which algorithm is used for precalculation and of the integrals in parallel run. In parallel runs, SEWARD produces AO Cholesky/RI vectors segmented in auxiliary index over parallel nodes. Depending on the network bandwidth and computational power of each node, different algorithms can lead to optimal performance. Following options are available:
0 - None: no cumulation of Cholesky/RI vectors is needed (debug only).
1 - Minimal: Cholesky/RI vectors are cumulated prior to integral precalculation. Low network bandwidth is required.
2 - Medium: O^2V^2 integrals are generated from local Cholesky/RI vectors and cumulated along with the Cholesky/RI vectors afterwards. Other integrals are calculated from cumulated intermediates.
3 - Full: All integrals are generated from local Cholesky/RI vectors and cumulated afterwards. High network bandwidth is required.
(Default=2)

MAXIterations Integer on the following line specifies maximum number of CCSD iteration (Default=40)

RESTart This keyword specifies that CCSD calculation is restarted from previous run. This keyword is currently under development, thus disabled. (Default=OFF)

THREshold Double precision floating point number on the following line specifies the convergence threshold for the CCSD correlation energy. (Default=1.0d-6)

PRINtkey The integer on the following line specifies the print level in output
1 - Minimal
2 - Minimal + timings of each step of the CCSD iterations
10 - Debug
(Default=1)

END of input This keyword indicates that there is no more input to be read.

```
 &CHCC &END
Title
Benzene dimer
Frozen
12
Deleted
0
```

```
Large
4
Small
2
CHSEgment
100
Precalculate
Join
2
Maxiter
50
Threshold
1.0d-6
Print
2
End of Input
```

8.7 CHT3

CHT3 is a Closed-Shell Coupled-Clusters perturbative triples program based exclusively on the Cholesky (or RI) decomposed 2-electron integrals aimed towards calculation of large systems on highly parallel architectures. Use of point-group symmetry is not implemented. Main advantage compared to the CCSDT module is MOLCAS is in its more efficient parallelization and dramatically lowered memory (and eventually disk) requirements.

8.7.1 Dependencies

CHT3 requires previous run of the CHCC Cholesky/RI based CCSD program to produce T1 and T2 excitation amplitudes stored in *T2xxxx* and *RstFil* files. The CHCC program (as well as SEWARD and SCF) must be run in Cholesky/RI mode.

The algorithm used for almost complete elimination of the CHT3 limits in calculated system size due to the computer memory bottleneck relies on blocking of the virtual orbitals. Size of blocks is, unlike in CHCC program, determined automatically for optimal performance.

8.7.2 Files

Input files

File	*Contents*
RUNFILE	File for communication of auxiliary information.
L0xxxx, L1xxxx, L2xxxx	MO-transformed Cholesky vectors
T2xxxx	T2 (ij,a'b') excitation amplitudes
RstFil	Communication file containing T1 amplitudes, restart informations, etc.

Intermediate files

All the intermediate files are created, used and removed automatically, unless you yourself create a link or a file with the specified name.

File	*Contents*
KMATAA, KMATBA, LMATAA, LMATBA	Temporary integral files

Output files

File	*Contents*
None	

8.7.3 Input

The input for each module is preceded by its name like:

```
&CHT3
```

Keyword	*Meaning*
TITLe	This keyword starts the reading of title lines, with the number of title lines limited to 10. Reading the input as title lines is stopped as soon as the input parser detects one of the other keywords, however only ten lines will be accepted. This keyword is *optional.*
FROZen	Integer on the following line specifies number of inactive occupied orbitals in the (T) calculation. This keyword is *optional.* (Default=0)
DELEted	Integer on the following line specifies number of inactive virtual orbitals in the (T) calculation. This keyword is *optional.* (Default=0)
LARGe	Integer on the following line specifies the main segmentation of the virtual orbitals used in previous CCSD run. This keyword is *compulsory.* (Default=None)
MHKEy	Integer on the following line specifies if library BLAS (MHKEy=1) or hard-coded fortran vector-vector, matrix-vector and matrix-matrix manipulation is used. This keyword is *optional.* (Default=1)

NOGEnerate — This keyword specifies that the pre-(T) steps (generation of integrals from the Cholesky/RI vectors, etc.) are skipped. This keyword can be used for restarting the (T) calculation if the required integrals were already generated. This keyword is *optional.* (Default=OFF)

NOTRiples — This keyword specifies that the post integral preparation steps, i.e. the real calculation of (T) contribution will not be done. Job can be restarted from this point using the NOGEnerate keyword. This keyword is *optional.* (Default=OFF)

ALOOp — Two integers on the following line specify first and last triplet of virtual orbitals blocks to be calculated in the first (”A loop”) of the two parts of the (T) calculation. Using this keyword enables user to split the (T) calculation into separate jobs. Information about the total number of triplets in the ”A loop” can be found in the output of the ”preparation” step of the (T) program. Values -1, -1 mean, that the whole ”A loop” is either executed or skipped, depending on the parameters of the BLOOp keyword. This keyword is *optional.* (Default=-1,-1)

BLOOp — Two integers on the following line specify first and last triplet of virtual orbital block to be calculated in the second (”B loop”) of two parts of the (T) calculation. Using this keyword enables user to split the (T) calculation into separate jobs. Information about the total number of triplets in the ”B loop” can be found in the output of the ”preparation” step of the (T) program. Values -1, -1 mean, that the whole ”B loop” is either executed or skipped, depending on the values of the ALOOp keyword. This keyword is *optional.* (Default=-1,-1)

PRINtkey — The integer on the following line specifies the print level in output 1 - Minimal
2 - Minimal + timings of each (T) step
10 - Debug
This keyword is *optional.* (Default=1)

```
 &CHT3
Title  = Benzene dimer
Frozen = 12
Large  = 4
ALOOp  = 20 120
BLoop  = 1 250
Print  = 2
```

8.8 CMOCORR

8.8.1 Description

The CMOCORR is a small utility that is used to compare orbital spaces for two orbital vector files. This is useful for checking that a calculation has maintained the orbital spaces intended by the user.

8.8.2 Dependencies

The CMOCORR program requires two orbitals files as input generated by any of the modules that produces orbitals.

8.8.3 Files

Input files

Two orbitals files with the names *CMOREF* and *CMOCHK* are needed by the program, and it is the responsability of the user to make the proper links to these files, no links are done automatically.

Output files

There are no output files.

8.8.4 Input

Below follows a description of the input to CMOCORR The input for each module is preceded by its name like:

```
&CMOCORR
```

Argument(s) to a keyword, either individual or composed by several entries, can be placed in a separated line or in the same line separated by a semicolon. If in the same line, the first argument requires an equal sign after the name of the keyword. Note that all character in a keyword is necessary, not only the first four.

Keyword	*Meaning*
DoMetric	Compare the metric of the two files. If the files correspond to different geometries the metric will be different.
DoSpaces	Compare the orbitals spaces of the two files. This keyword implies DoMetric.
DoOrbitals	Compare the orbitals one by one in the two files. This keyword implies DoMetric and DoSpaces.
sortcmo	Sort the orbitals according to the type index. This might be necessary if one of the files are created by LUSCUS for example.
Thresholds	This keyword is followed by two parameters, t_1 and t_2, the first specifying at what overlap to report that orbitals from the two files have a small overlap. In addition, orbitals in the reference file with best match is located. The second parameter is similar, but no search for matching orbitals is done. The defaults are $t_1 = 0.6$ and $t_2 = 0.8$.
End of input	This keyword terminates the reading of the input.

Input examples

First we have the bare minimum of input. This will only check that the files have the same buber of orbitals and symmetries.

```
 &CMOCORR
```

The next example is almost as simple, and all checks are perfomed.

```
 &CMOCORR
DoOrbitals   -- check everything
```

8.9 CPF

The CPF program generates SDCI, CPF[39], MCPF[40] or ACPF[41], wavefunctions from **one** reference configuration.

The CPF program is a modification to a CPF program written by P. E. M. Siegbahn and M. Blomberg (Institute of Physics, Stockholm University, Sweden).

The program is based on the Direct CI method[42], with the coupling coefficients generated by the Graphical Unitary Group Approach[43]–[44] (See program description for GUGA). CPF generates natural orbitals that can be fed into the property program to evaluate certain one electron properties. Also, the natural orbitals can be used for Iterative Natural Orbital calculations.

Orbital subspaces

The orbital space is divided into the following subspaces: Frozen, Inactive, Active, Secondary, and Deleted orbitals. Within each symmetry type, they follow this order. Their meaning is the same as explained in the GUGA and MOTRA sections, except that, in this case, there is only a single reference configuration. Therefore, the active orbitals in this case are usually only open shells, if any. Since explicit handling of orbitals is taken care of at the integral transformation step, program MOTRA, orbital spaces are not specified in the input, except when orbitals are frozen or deleted by the CPF program, rather than by MOTRA (which should normally be avoided).

8.9.1 Dependencies

The CPF program needs the coupling coefficients generated by the program GUGA and the transformed one and two electron integrals from the program MOTRA .

8.9.2 Files

Input files

The CPF program need the coupling coefficients generated by GUGA and the transformed integrals from MOTRA.

CPF will use the following input files: *ONEINT*, *RUNFILE*, *CIGUGA*, *TRAINT*, *TRAONE* (for more information see 7.1.2). and *CPFVECT* (for restarted calculations).

Output files

CPF generates an two output files:

File	*Contents*
CPFORB	The natural orbitals from the CPF functional.
CPFVECT	The CI expansion coefficients. These may be used for restarting an unconverged calculation.

8.9.3 Input

8.9.4 CPF

This section describes the input to the CPF program in the MOLCAS program system. The input for each module is preceded by its name like:

```
&CPF
```

Optional keywords

Keyword	*Meaning*
TITLE	Followed by a title line
SDCI	Specifies that a SDCI calculation is to be performed. No additional input is required. Only one of the choices SDCI, CPF, MCPF or ACPF should be chosen.
CPF	Specifies that a CPF calculation is to be performed. Only one of the choices SDCI, CPF, MCPF or ACPF should be chosen.
MCPF	Specifies that a Modified CPF calculation is to be performed. This option is in fact the default choice. Only one of the choices SDCI, CPF, MCPF or ACPF should be chosen.
ACPF	Specifies that an Average CPF calculation is to be performed. Only one of the choices SDCI, CPF, MCPF or ACPF should be chosen.
RESTART	Restart the calculation from a previous calculation.
THRPR	Threshold for printout of the wavefunction. All configurations with a coefficient greater than this threshold are printed in the final printout. The default is 0.05.

ECONvergence Energy convergence threshold. The update procedure is repeated until the energy difference between the last two iterations is less than this threshold. The default is 1.0e-8.

PRINt Print level of the program. Default is 5.

MAXIterations Maximum number of iterations in the update procedure. Default 20. The maximum value of this parameter is 75.

FROZen Specify the number of orbitals to be frozen in **addition** to the orbitals frozen in the integral transformation. Default is 0 in all symmetries.

DELEted Specify the number of orbitals to be deleted in **addition** to the orbitals deleted in the integral transformation. Default is 0 in all symmetries.

LOW Specifies that this is a low spin case, i.e. the spin is less than the maximum possible with the number of open shells in the calculation. See Refs. [39, 40]. This requires special considerations.

MAXPulay Maximum number of iterations in the initial stage. After that, DIIS extrapolation will be used. Default is 6.

LEVShift Levelshift in the update procedure. Default is 0.3.

Input example

```
 &CPF
Title
 Water molecule. 1S frozen in transformation.
MCPF
```

8.10 DYNAMIX

The DYNAMIX program performs molecular dynamics (MD) simulations in Molcas. Here the nuclei are moved according to the classical Newton's equations which are solved numerically using the velocity Verlet algorithm[45]. The algorithm requires coordinates, velocities and forces as input. DYNAMIX can be used with any electronic structure method in Molcas. Also environmental effects can be taken into account in the MD simulation: the solvent can be considered implicitly using the reaction field keyword in GATEWAY or explicitly in hybrid QM/MM calculation which requires the ESPF program.

When multiple electronic states are involved in a MD simulation, a trajectory surface hopping (TSH) algorithm allows non-adiabatic transitions between different states. This TSH algorithm evaluates the change of the wavefunction along the trajectory and induces a hop if certain criteria a met (for further details read the RASSI section). In the current implementation the surface hopping algorithm can be used only with state averaged CASSCF wavefunction. However, an extension for CASPT2 and other methods are in preparation.

The Tully algorithm is available in a separate module SURFACEHOP.

8.10.1 Dependencies

The coordinates and the forces are required by the DYNAMIX program. DYNAMIX reads the initial coordinates from the *RUNFILE* and updates them in each iteration. In addition DYNAMIX depends on the ALASKA program, since it generates forces.

8.10.2 Files

Input files

File	*Contents*
velocity.xyz	Contains the initial velocities of the MD simulation.

Output files

File	*Contents*
RUNFILE	Trajectory information such as current time, velocities, etc. are stored in this file.
md.xyz	The coordinates for each step of the MD trajectory are saved here.
md.energies	The potential, kinetic and total energies are written to this file. In case of multiple electronic states, the energies of all roots are saved.

8.10.3 Input

This section describes the input syntax of DYNAMIX in the MOLCAS program package. In general a MD simulation requires a FOREACH loop which contains several programs to compute the energy and ALASKA for subsequent gradient computation. The input of the DYNAMIX begins with the program name, and is followed by the only compulsory keyword VELV which specifies the velocity Verlet algorithm:

```
&DYNAMIX
 VELV
```

General keywords

Keyword	*Meaning*
VELVerlet	This keyword specifies the velocity Verlet algorithm [45] to solve Newton's equations of motion. It's the only compulsory keyword in the program.
DTime	Defines the δt which is the time step in the MD simulation and which is used for the integration of Newton's equations of motion. The program expects the time to be given in floating point format and in atomic unit of time (1 a.u. of time = $2.42 \cdot 10^{-17}$ s). (Default = 10).
VELOcities	Specifies how the initial velocities are generated. This keyword is followed by an integer on the next line. The internal unit of the velocities is [Bohr·(a.u. of time)$^{-1}$]. 0 - Zero velocities. (Default) 1 - The velocities are read from the file *$Project.velocity.xyz* in *$WorkDir*. This file contains velocities in the xyz format given in the same order as the atoms in coordinate file. The unit of the velocities is [Bohr·(a.u. of time)$^{-1}$]. 2 - This option allows to read in mass-weighted velocities from the file *$Project.velocity.xyz* in [Bohr·$\sqrt{a.m.u.}$·(a.u. of time)$^{-1}$]. 3 - This option takes random velocities from a Maxwell-Boltzmann distribution, at a given temperature, assuming that every component of the velocity can be considered as an independent gaussian random variable.
THERmostat	Regulates the control of the temperature by scaling the velocities. The option is an integer given on the next line. 0 - No velocity scaling. (Default) 1 - The velocities are scaled in order to keep the total energy constant. 2 - The velocities are scaled according to the Nosé-Hoover chain of thermostats algorithm, used to perform molecular symulation at constant temperature, resulting in statistics belonging to the canonical ensemble (NVT).
TEMPerature	Defines the numerical value of the temperature, which is used together with the Nosé-Hoover chain of thermostats to perform molecular dynamics at constant temperature. (Default = 298.15K)
HOP	Enables the trajectory surface hopping algorithm if the integer given in the next line is bigger than 0. The integer also specifies how many non-adiabatic transitions are allowed between electronic states.

RESTART This keyword allows to restart the trajectory at a given time. The time is given on the next line in atomic units.

H5RESTART This keyword allows to restart a trajectory calculation from an HDF5 file. The name of the restart file is given on the next line.

Input examples

The following example shows the input for an excited state CASSCF molecular dynamics simulation of a methaniminium cation using the DYNAMIX program. The FOREACH loop allows 1000 steps with 10 a.u. of time step size which leads to a total duration of 242 fs. In the RASSCF program the second root is selected for gradient calculation using the keyword MDRLXR. This input assumes that the a *JOBIPH* file with orbitals is already given. In each iteration the *JOBIPH* is updated to achieve a fast convergence of the CASSCF wavefunction. A Nosé-Hoover chain of thermostats, enabled with THERmo= 2, is used to reproduce dynamics at constant temperature, where the initial velocities are taken from a Maxwell-Boltzmann distribution at 300 K.

```
&GATEWAY
 COORD
 6
 Angstrom
 C  0.00031448  0.00000000  0.04334060
 N  0.00062994  0.00000000  1.32317716
 H  0.92882820  0.00000000 -0.49115611
 H -0.92846597  0.00000000 -0.49069213
 H -0.85725321  0.00000000  1.86103989
 H  0.85877656  0.00000000  1.86062860
 BASIS= 3-21G
 GROUP= nosym
>> FOREACH ITER in (1 .. 1000)
&SEWARD
>> IF ( $ITER = 1 )
&RASSCF
 LUMORB
 FileOrb= $Project.GssOrb
 Symmetry= 1
 Spin= 1
 nActEl= 2 0 0
 Inactive= 7
 RAS2= 2
 CIroot= 3 3 1
>> COPY $Project.JobIph $Project.JobOld
>> ENDIF
&RASSCF
 JOBIPH; CIRESTART
 Symmetry= 1
 Spin= 1
 nActEl= 2 0 0
 Inactive= 7
 RAS2= 2
 CIroot= 3 3 1
 MDRLXR= 2
>> COPY $Project.JobIph $Project.JobOld
&ALASKA
&DYNAMIX
 VELVer
 DT= 10.0
 VELO= 3
```

```
  THER= 2
  TEMP=300
  HOP= 1
 >> END DO
```

8.11 EMBQ

8.11.1 Description

The EMBQ program of the Molcas program system computes the geometrical positions and values of point charges, which reproduce the electrostatic potential in a finite volume. These point charges can be used in the embedded cluster calculations to mimic the electrostatic potential of an infinite lattice.

In embedded cluster and quantum mechanics/molecular mechanics (QM/MM) methods, a QM description of a part of the system is combined with an empirical description of its surroundings. The QM region is said to be embedded into the potential produced by its surrounding. The EMBQ program implements a method, which allows one to generate the electrostatic embedding potential for however complex crystalline lattice. To this end (i) a lattice unit cell is complemented with point charges which zero out all multipole moments of the unit cell up to any predefined M-tupole and (ii) a finite nano-cluster is constructed from these redefined unit cells. As the size of this nano-cluster increases, the electrostatic potential in its inner region converges to that calculated using the Ewald's procedure.

The details of this method and examples are provided in the literature [46, 47].

8.11.2 Files

Input files

File	*Contents*
EMBQ.INP	This file contains keywords and control parameters for the EMBQ program including information about the crystal lattice cell and atoms of the QM cluster. No point group symmetry is assumed.

Intermediate files

All the intermediate files are created, used and removed automatically.

Output files

In all output files coordinates are given in Ångstroms and the values of charges – in atomic units.

File	*Contents*
EMBQ_cell.xyz	Lattice cell: coordinates and ionic charges of the lattice cell atoms. No point group symmetry is assumed. Format: XYZ.
EMBQ_cell+Q.xyz	Modified cell: coordinates and charges of the lattice cell atoms and complementary point charges generated by EMBQ. Format: XYZ. Note that the lattice atoms and complementary charges may coincide.
EMBQ_ncQ.xyz	Coordinates and charges of all species of the nano-cluster constructed using the modified unit cell. Format: XYZ.
EMBQ_ncQ.dat	Coordinates and charges of all species of the nano-cluster constructed using the modified unit cell. Format: 4 columns containing Cartesian coordinates and the value of the charge.
EMBQ_ncQ-QM.xyz	Coordinates and charges of all species of the nano-cluster *without* atoms of the QM cluster (if specified). Format: XYZ.
EMBQ_ncQ-QM.dat	Coordinates and charges of all species of the nano-cluster *without* atoms of the QM cluster (if specified). Format: 4 columns containing Cartesian coordinates and the value of the charge.
EMBQ_elpot.dat	Coordinates and charges of the nano-cluster, distance from the centre of the nano-cluster to each centre, centre number, on-site electrostatic potential and components of the field.

8.11.3 Input

Description of the input to EMBQ is below. The keywords are always significant to four characters, but in order to make the input more transparent, it is recommended to use the full keywords. The EMBQ program section of the MOLCAS input is bracketed by a preceding dummy namelist reference

```
 &EMBQ

End of Input
```

Argument(s) to a keyword are always supplied on the next line of the input file, except explicitly stated otherwise.

Compulsory keywords

Keyword	*Meaning*
ELMOMENT	Keyword, followed by a single integer, which specifies the largest electric multipole to be eliminated. A non-negative integer has to be supplied as argument. Default value is 0.

UCVEctors	Specifies parameters of the crystal cell (in Å). Three lines, containing three real numbers each, have to be supplied: 1^{st} line – components of the cell vector $\mathbf{a}_1$; 2^{nd} line – components of the cell vector $\mathbf{a}_2$; 3^{rd} line – components of the cell vector $\mathbf{a}_3$.
UCV1	Keyword, followed by three real numbers. Specifies components of the crystallographic cell vector $\mathbf{a}_1$ (in Å). Can be used as an alternative to keyword UCVEctors. Should be used together with keywords UCV2 and UCV3. There is no default value.
UCV2	Keyword, followed by three real numbers. Specifies components of the crystallographic cell vector $\mathbf{a}_2$ (in Å). Can be used as an alternative to keyword UCVEctors. Should be used together with keywords UCV1 and UCV3. There is no default value.
UCV3	Keyword, followed by three real numbers. Specifies components of the crystallographic cell vector $\mathbf{a}_3$ (in Å). Can be used as an alternative to keyword UCVEctors. Should be used together with keywords UCV1 and UCV2. There is no default value.
UCATоms	Keyword, followed by a single integer equal to the number of atoms in the lattice cell and a list of the corresponding coordinates (in Å) and ionic charges (in atomic units).

Optional keywords

Keyword	*Meaning*
TETRahedra	Keyword, followed by a single line containing eight integers equal to either 0 or 1. Specifies which tetrahedra are used to complement the crystal cell with `EMBQ` charges. Each integer correspond to a single tetrahedron associated with a single corner of the crystal cell. The tetrahedron is used if the corresponding parameter equals to 1 and not used if it equals to 0. Default: use all eight tetrahedra. Orientation of the tetrahedra are determined by their axes, which are either parallel (+) or anti-parallel (–) to the cell vectors $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$ as shown in the table below.

Tetrahedron	Orientation		
	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$
1	+	+	+
2	−	+	+
3	+	−	+
4	+	+	−
5	−	−	+
6	−	+	−
7	+	−	−
8	−	−	−

SHIFt — Keyword, followed by a single real number. Specifies the shift (in fractional coordinates) of the tetrahedra from the cell corners outwards. Default value is zero.

NANOcluster — Keyword followed by two input lines. The first line contains a single integer number (n) which specifies the shape of a nano-cluster generated using the modified cells. Possible values of n:
1 – to generate a *cubic* nano-cluster
2 – to generate a *block* nano-cluster
3 – to generate a *spherical* nano-cluster.
The size of the nano-cluster is defined in the following line. The number of input parameters depends on the shape of the nano-cluster. For a *cube*, provide one integer k to generate a nano-cluster of $(2k{+}1)^3$ unit cells. For a *block*, provide six integers k_1, k_2, m_1, m_2, n_1, n_2 to generate a nano-cluster of

$$(k_2 - k_1 + 1) \times (m_2 - m_1 + 1) \times (n_2 - n_1 + 1)$$

unit cells. For a *sphere*, provide one real number to generate a nano-cluster of radius R (in Å).
This keyword can be used instead of keywords NCCube, NCBLock, and NCSPhere. Note that only one nano-cluster will be generated. Default: the nano-cluster is not generated.

NCCUbe — Keyword, followed by a single integer number k. Specifies the shape and size of the nanocluster constructed from the modified unit cells. The nanocluster is generated as a block of

$$(2k + 1) \times (2k + 1) \times (2k + 1)$$

cells along the lattice vectors $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$, respectively. Default: the nanocluster is not generated.

NCBLock — Keyword followed by six integers: k_1 k_2 m_1 m_2 n_1 n_2. Specifies the shape and size of the nanocluster constructed from the modified unit cells. The nanocluster is generated as a block of

$$(k_2 - k_1 + 1) \times (m_2 - m_1 + 1) \times (n_2 - n_1 + 1)$$

cells along the lattice vectors $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$, respectively. Default: the nanocluster is not generated.

NCSPhere — Keyword followed by a single real number. Specifies the shape and radius (in Å) of the nano-cluster constructed from the modified unit cells. Default: the nanocluster is not generated.

PRINt — Keyword, followed by a single integer number. Specifies the general print level:
0 – minimal print out;
1 – intermediate print out;
2 – full print out.
Default: use the global MOLCASprint level.

CALCulate — Keyword, followed by a single integer number. Requests calculation of the electrostatic potential and field at all centres of the nano-cluster. Possible values are:

0 – calculate neither the potential nor components of the field vector (default);

1 – calculate the potential only;

2 – calculate the potential and components of the field vector.

QMCLuster — Keyword, followed by a single integer equal to the number of atoms in the QM cluster and a list of the corresponding Cartesian coordinates (in Å). Specifies geometrical structure of the QM cluster. Default: number of the QM cluster atoms is zero.

Limitations

The largest electric moment ELMOment is limited to 10.
Number of atoms in UCATOms is limited to 1000.
Number of atoms in QMCLuster is limited to 1000.
Tetrahedra in TETRahedra are oriented so as three of their edges are parallel to the cell vectors.
The value of SHIFt is the same for all tetrahedra.

Input example

```
 &EMBQ &END
Elmoment
4                              Largest moment to eliminate
Tetrahedra
1 1 1 1 0 0 0 0                Use the tetrahedra (if 1) or not (if 0)
Shift
0.5                            Shift the tetrahedra from the corner sites outward by this value
Nanocluster
3                              Shape of the nano-cluster (1 -- cube, 2 -- cuboid, 3 -- sphere)
30.0                           Size of the nano-cluster. Here, radius of the sphere (in Å).
Print
2                              Printing level
Calculate
2                              Calculate electrostatic potential and its derivatives.
UCvectors
4.593730 0.000000 0.000000     Unit cell vector a1 (in Å)
0.000000 4.593730 0.000000     Unit cell vector a2 (in Å)
0.000000 0.000000 2.958120     Unit cell vector a3 (in Å)
UCatoms
6                              Number of atoms in the cell
0.000000000   0.000000000   0.000000000   4.0
2.296865000   2.296865000   1.479060000   4.0
1.402465769   1.402465769   0.000000000  -2.0
3.699330769   0.894399231   1.479060000  -2.0
3.191264231   3.191264231   0.000000000  -2.0
0.894399231   3.699330769   1.479060000  -2.0
QMatoms
4                              Number of atoms in the QM cluster
1.402465769   1.402465769   0.000000000
2.296865000   2.296865000  -1.479060000
2.296865000   2.296865000   1.479060000
```

```
 0.000000000   0.000000000   0.000000000
 End of Input
```

8.12 ESPF (+ QM/MM INTERFACE)

8.12.1 Description

The ElectroStatic Potential Fitted (ESPF) method adds contributions to the one-electron Hamiltonian for computing the interaction between the charge distribution in Molcas and *any* external electrostatic potential, field, and field derivatives. The approximate interaction energy is expressed as:

$$\Delta E^{\mathrm{ESPF}} = \left(\sum_a \langle \Psi \left| Q^a \right| \Psi \rangle + Z_a \right) V^a \tag{8.1}$$

with Q^a a multipole-like operator whose matrix elements are fitted to the electron potential integrals (determined on a grid surrounding the QM atoms) and V^a the *external* electrostatic potential (and derivatives) at nucleus a. Both energy and gradient computations are available. A call to `ESPF` right after `SEWARD` is required to carry out such calculations.

NOTE: Always run `SEWARD` + `ESPF`. If not, very strange results may happen due to interactions counted twice or more!

NOTE: Symmetry is ignored since the external potential usually breaks the one given in `GATEWAY`.

If no external potential is given, the `ESPF` module can be used to compute atomic point charges fitted to the electrostatic potential produced by the nuclei and electrons.

8.12.2 ESPF and QM/MM

Whereas the ESPF method can be used standalone, it has been developed for hybrid quantum mechanics/molecular mechanics (QM/MM) computations, in which an extended molecular system is divided into two subsystems: the ’active’ center described by any QM method and its surroundings qualitatively treated with an empirical atomic forcefield. The current implementation can be used with either a modified version of the `TINKER` program or with the `GROMACS` program as MM code.

Using Molcas together with Tinker. In order to obtain the modified `TINKER` code, you must run the ”molcas get_tinker” command.

The current patched version of `TINKER`* is **6.3.3**.

IMPORTANT: The environment variable TINKER must point to the directory in which the `TINKER` executable binaries are stored (usually in $MOLCAS/tinker/bin).

The most convenient way to define (i) the QM and MM subsystems and (ii) which atoms are to be known by Molcas (all the QM ones and some MM ones, see below) requires to

*http://dasher.wustl.edu/tinker/

simply add the keyword TINKER in GATEWAY. This way, GATEWAY will ask TINKER to pass it all information needed.

Alternatively, an *old* input style can be used in GATEWAY, using the following syntax:

```
Basis set
X..... / MM
  name  x  y  z  Angstrom
Charge
  q
End of basis
```

where q gives its MM point charge value, name is its name and x, y, z are its coordinates given in au or in Å if the ANGSTROM keyword is given. This way, a MM atom is simply a QM atom without basis functions and with a non-integer atomic charge. Usually only MM atoms needed to define some constrained link atoms (see below) positions are included, however this is not mandatory.

Using Molcas together with Gromacs. The interface to GROMACS differs from the TINKER interface in that the MM code is not run as a separate program but included in MOLCAS as a library. In this way, the communication between the QM and MM codes is handled by simple function calls instead of using data files. The interface is automatically installed along with MOLCAS provided that the GROMACS library (currently a development version[†]) is available at configuration time[‡]. Instructions how to install the GROMACS library can be found at the official web site[§]. Make sure that the installation is done in double precision since this is the precision used by MOLCAS. Also make sure to source the GROMACS GMXR script in your shell startup file, otherwise the MOLCAS configuration procedure will not be able to detect the relevant library path.

The recommended (and the only verified) approach of using the MOLCAS/GROMACS interface is to define the full QM+MM system in the GROMACS input. The system definition can then be imported into MOLCAS by adding the keyword GROMACS in GATEWAY (see section 8.16 for details). For efficiency reasons, the MOLCAS part of the interface separates the MM subsystem into two different atom types: *inner* MM atoms and *outer* MM atoms. These are completely equivalent as far as interactions are concerned. However, whereas the coordinates of the inner MM atoms are stored and updated using MOLCAS standard mechanism, the outer MM atoms are handled using a mechanism specifically designed with large systems in mind. The division into inner and outer MM atoms can be controlled with options to the GROMACS keyword in GATEWAY (see section 8.16).

Please note that the MOLCAS/GROMACS interface is still under development and is currently provided not intended for production runs.

The QM/MM method. The Hamiltonian of the full QM/MM system is divided into three terms:

$$H = H_{\text{QM}} + H_{\text{MM}} + H_{\text{QM/MM}} \tag{8.2}$$

The first one describes the QM part as it would be *in vacuo*, the second one describes the surroundings using a classical MM forcefield and the last one deals with the interactions

[†] http://repo.or.cz/w/gromacs.git/shortlog/refs/heads/qmmm

[‡] Configurations done with CMake need to include the flag -DGROMACS=ON

[§] http://www.gromacs.org

between the QM and the MM subsystems. In its usual formulation, the last term is (for q point charges interacting with N nuclei and n electrons):

$$H_{\mathrm{QM/MM}} = \sum_{a=1}^{q}\sum_{b=1}^{N}\frac{Q_a Z_b}{R_{ab}} - \sum_{a=1}^{q}\sum_{i=1}^{n}\frac{Q_a}{r_{ai}} + \sum_{a=1}^{q}\sum_{b=1}^{N}E_{ab}^{\mathrm{vdW}} + E^{\mathrm{bonded}} \tag{8.3}$$

The first two terms deal with the electrostatic interactions between the QM charge distribution and the MM electrostatic potential. In MOLCAS the ESPF method is used for this purpose. A short-range van der Waals term is added (van der Waals parameters are assigned to all the atoms - both QM and MM). If the frontier between the two subsystems involves a bond, some empirical bonded terms may also be used. For the sake of simplicity, the standard MM parameters are kept unchanged for the MM atoms but should be modified (or calculated) for the QM atoms (e.g. it may be necessary to fit the QM van der Waals parameters).

The usual forcefields use the ”1-4 condition” to separate the bonded interactions (stretching, bending, torsion) from the non-bonded ones (electrostatic and vdw). This means that the non-bonded potentials are applied only if atoms are separated by 3 bonds or more. As for the QM/MM interactions, this procedure is kept with the exception that all the QM atoms experience the electrostatic potential generated by *all* the MM point charges (the QM/MM frontier case is considered later).

NOTE: Starting with MOLCAS 8, all MM point charges interact with the QM charge distribution using the ESPF method (at variance with previous MOLCAS versions in which the few MM atoms defined in `GATEWAY` were interacting directly with the QM electrons and nuclei).

Link atoms. When no bonds are involved between the QM and the MM parts, the QM/MM frontier definition is obvious and only the electrostatic and vdw interactions are taken into account. However, if one or several chemical bonds exist, the definition of a smooth but realistic frontier is needed. Several schemes, more or less sophisticated, have been proposed. In the current implementation, only the most basic one, the link atom (LA) approach is included. In the LA approach, each QM/MM bond that should be cut is saturated with a monovalent atom — most often a hydrogen atom — on the QM side. The position of a link atom is often restrained: frozen distance from the corresponding QM frontier atom and always on the segment defined by the two frontier atoms (Morokuma's method, selected by the LAMOROKUMA keyword).

From the macromolecular point of view, link atoms do not exist, i.e. they should not interact with the MM part. However, this leads to severe overpolarization of the frontier, due to unbalanced interactions. Hence interactions between the link atoms and the MM potential is kept. To remove problems that may arise from too strong interactions between a link atom and the closest MM point charges, these point charges may be spread in the MM neighborhood. For instance, in a protein, this procedure is mainly justified if the MM frontier atom is an α carbon (Amber or Charmm-type forcefields usually set these point charges close to zero).

Geometry optimization – microiterations. In a QM/MM geometry optimization job, a MOLCAS step costs as hundreds of `TINKER` or `GROMACS` steps. Thus it is very convenient to use the microiteration technique, that is, converging the MM subsystem geometry every

MOLCAS step. In the case of TINKER, this is requested in the TINKER keyword file, whereas if GROMACS is used, it is requested directly in ESPF. In order to improve the optimization convergence, an improved QM/MM Hessian can be built in SLAPAF using the RHIDDEN keyword (note that adding the keyword CARTESIAN may help too).

8.12.3 Dependencies

The ESPF program depends on SEWARD for modifying the core Hamiltonian matrix and on ALASKA for computing the extra contributions to the gradient.

8.12.4 Files

ESPF will use the following input files: *RYSRW*, *ABDATA*, *RUNFILE*, *ONEINT* (for more information see 7.1.2). In addition, ESPF uses *ESPFINP* (the ESPF input file) and *SEWARINP* (the Seward input file).

Please note that the external potential can be given within a file, separated from the ESPF input file.

In calculations using the MOLCAS/GROMACS interface, ESPF will additionally need access to the GROMACS tpr file.

Intermediate files

All the intermediate files are related to the use of ESPF together TINKER. The files allow for communication between the ESPF program and the MM code. MOLCAS uses one file to pass the QM atoms coordinates and ESPF-derived point charges to TINKER. TINKER uses the same file to pass the external potential, the MM-only energy and gradient components to MOLCAS.

File	*Contents*
TINKER.LOG	The log file of the Tinker run.
$Project.xyz	The coordinate file for TINKER.
$Project.key	The keyword file for TINKER.
$Project.qmmm	The communication file between MOLCAS and TINKER.

Output files

File	*Contents*
ONEINT	One-electron integral file generated by the SEWARD program.

RUNFILE	Communication file for subsequent programs.
ESPF.DATA	Ascii file containing some specific informations needed for subsequent calls to the ESPF module.
GMX.LOG	Logfile for the GROMACS library routines.

8.12.5 Input

Below follows a description of the input to ESPF.

In addition to the keywords and the comment lines the input may contain blank lines. The input for each module is preceded by its name like:

```
&ESPF
```

Compulsory keywords

Keyword *Meaning*

EXTErnal

Specify how the external potential is given. This keyword is compulsory in the first run of ESPF. On the next line, one integer or a text string must be given:

- One integer n is given. If n is 0, the next lines give the numbering, the values for the external potential, the field and field gradients for each atom. If n is greater than 0, the n next lines specify the sources of the external potential, each line gives three cartesian coordinates, one point charge, and (optionally) three dipole components. If Å is used as the length unit, the ANGSTROM keyword must be given right after n.
- The NONE word means that no external potential is given. Accordingly, the ESPF module will compute the atomic point charges (and optionally dipoles) deriving from the electrostatic potential due to all electrons and nuclei.
- The word is TINKER, which means that the current job is a QM/MM job using the Molcas/TINKER interface. Accordingly the external potential will be computed directly by TINKER. Note that TINKER requires at least two input files, ending with .xyz (coordinates) and .key (keywords). These files must share the name of the current Molcas project. Optionally, you can add the MULLIKEN or LOPROP keyword after TINKER: it indicates what kind of charges are passed to TINKER. These charges may be used during the MM microiterations. If no keyword is given, the ESPF multipoles are selected.
- The word is GROMACS, which means that the current job is a QM/MM job using the Molcas/GROMACS interface, with the external potential computed by GROMACS. The binary input file read by GROMACS, the so-called tpr file, must be named as

'topol.tpr' and must be manually copied to the working directory. As above, a second keyword on the same line can be used to select the type of multipoles sent to the MM code. Default is to use the ESPF multipoles.

- Any other word. The following characters up to the next space are taken as a file name and the rest of the line is ignored. Instead, the full input (including the first line) is read from the specified file and must follow the syntax specified above.

Optional keywords

Keyword	*Meaning*
TITLE	Title of the job.
MULTipoleorder	Multipolar order of the ESPF operators. For TINKER, allowed values are 0 (charge-like) or 1 (charge- and dipole-like). For GROMACS, only 0 is allowed. Default value is 0.
GRID	Modify the grid specifications. The grid is made of points belonging to molecular surfaces defined according to the van der Waals radii of each quantum atom. Two schemes are available. The first one is the GEPOL procedure, as implemented into the PCM SCRF method. The other one is called PNT and is the default. On the next line, first select the method with the GEPOL or PNT option. On the same line, one integer number and one real number are given if PNT is selected. The first one gives the maximum number of shells around the van der Waals surface of the quantum atoms. The second one gives the distance between the shells. Note that all points within the van der Waals envelope are discarded to avoid the penetration effects. Default values are 4 shells separated by 1 Å. Alternatively, if GEPOL is selected, the same line must contain 1 integer indicating the number of surfaces to be computed (must be ¡ 6).
SHOW	Requires the printing of the ESPF.DATA file.
LAMOrokuma	Activate the Morokuma scheme for scaling the link atom positions in a QM/MM calculation. Note that in the case of TINKER, the scaling factor is currently hard-coded and is determined from the radii of the atoms involved in the QM/MM frontier bond. This differs from the GROMACS interface in which this factor must be provided by the user through the LINKATOMS keyword in GATEWAY.
MMITerations	Maximum number of microiterations used to optimize the outer MM atoms in a Molcas/GROMACS run. The default is 0, which disables microiterations and leaves the outer MM atoms frozen. For the TINKER interface, microiterations are requested in the TINKER keyword file.

MMConvergence Convergence threshold for the MM microiterations (GROMACS only). The optimization of the (outer) MM atoms will stop when the maximum force component is smaller than this number, in atomic units. The default is 0.001 atomic units (50 kJ/mol/nm).

8.12.6 Examples

ESPF example

This is a typical input for the calculation of the energy and the gradient of a glycine molecule feeling the external potential of 209 TIP3P water molecules.

```
  &Gateway
Basis set
C.sto-3g.....
  C1   1.11820     0.72542    -2.75821 Angstrom
  C2   1.20948     0.66728    -1.25125 Angstrom
End of basis
Basis set
O.sto-3g.....
  O1   2.19794     1.10343    -0.67629 Angstrom
End of basis
Basis set
H.sto-3g.....
  H1   2.02325     1.18861    -3.14886 Angstrom
  H2   0.25129     1.31794    -3.04374 Angstrom
  H3   1.02458    -0.28460    -3.15222 Angstrom
End of basis
Basis set
N.sto-3g.....
  N1   0.17609     0.12714    -0.61129 Angstrom
End of basis
Basis set
C.sto-3g.....
  C3   0.09389    -0.01123     0.84259 Angstrom
  C4  -1.21244    -0.67109     1.28727 Angstrom
End of basis
Basis set
O.sto-3g.....
  O2  -2.06502    -1.02710     0.48964 Angstrom
End of basis
Basis set
H.sto-3g.....
  H4  -0.61006    -0.21446    -1.14521 Angstrom
  H5   0.92981    -0.61562     1.19497 Angstrom
  H6   0.16338     0.97444     1.30285 Angstrom
End of basis
Basis set
N.sto-3g.....
  N2  -1.41884    -0.85884     2.57374 Angstrom
End of basis
Basis set
H.sto-3g.....
  H7  -0.73630    -0.57661     3.25250 Angstrom
  H8  -2.28943    -1.29548     2.82140 Angstrom
End of basis
 &seward
 &espf
MultipoleOrder = 0
External = 0
1  -0.048 -0.002 -0.006 -0.001  0.007 -0.009  0.002 -0.001  0.001 -0.001
```

```
 2  -0.047 -0.002  0.001 -0.002  0.003  0.000 -0.004  0.000 -0.001  0.000
 3  -0.053  0.004  0.000 -0.011  0.002  0.002 -0.004  0.002  0.003 -0.007
 4  -0.046  0.011 -0.009 -0.001  0.006 -0.005 -0.001  0.003  0.003 -0.004
 5  -0.042 -0.016 -0.011 -0.006  0.005 -0.007  0.003 -0.004 -0.001 -0.005
 6  -0.050  0.000  0.008  0.001  0.006 -0.006  0.000 -0.002  0.000 -0.001
 7  -0.039 -0.008  0.001  0.000  0.001 -0.002  0.001 -0.001 -0.001 -0.001
 8  -0.032 -0.007 -0.002  0.004  0.002 -0.003  0.001 -0.002  0.002 -0.001
 9  -0.011 -0.009  0.004  0.001  0.002  0.000 -0.002 -0.001  0.001  0.001
10   0.000 -0.011  0.003  0.004  0.001  0.002 -0.003  0.001 -0.001  0.001
11  -0.028 -0.008  0.004 -0.001 -0.001 -0.002  0.002 -0.001  0.001 -0.002
12  -0.026  0.003 -0.008  0.014  0.002 -0.001 -0.001 -0.008  0.006 -0.009
13  -0.037 -0.008 -0.003  0.004 -0.007  0.007  0.000  0.001  0.007 -0.001
14  -0.016 -0.007  0.007 -0.008  0.003  0.003 -0.006  0.000  0.002  0.002
15  -0.025  0.003  0.012 -0.007  0.003 -0.001 -0.002 -0.006  0.005  0.009
16  -0.010 -0.011  0.000 -0.014  0.001  0.007 -0.008  0.001  0.000 -0.001
 &scf
Charge = 0
 &alaska
```

Molcas/Tinker example

A typical start for a QM/MM calculation with the Molcas/TINKER interface is given in the following input. It is quite general since all the information related to the QM and MM subsystem definitions are already included in the TINKER key file.

```
> EXPORT TINKER=$MOLCAS/tinker/bin_qmmm
> COPY $PATH_TO/$Project.xyz $WorkDir/$Project.xyz
> COPY $PATH_TO/$Project.key $WorkDir/$Project.key
 &Gateway
Tinker
Basis = STO-3G
Group = Nosym
 &Seward
 &Espf
External = Tinker
LAMorok
```

This can be used, e.g. with the following TINKER files. In this example, the asparate anion is cut into two pieces, the QM subsystem contains the end of the side-chain until the β carbon atom. There is a link atom between the QM β and MM α carbon atoms.

QMMM.xyz

```
16  ASP
 1 N3   -0.040452    0.189961    0.173219   448    2    6    14   15
 2 CT   -0.011045   -0.060807    1.622395   449    1    3     7   11
 3 C     1.446535   -0.110535    2.028518   450    2    4     5
 4 O     1.902105    0.960982    2.409042   452    3
 5 O     2.137861   -0.898168    1.387158   452    3
 6 H     0.559257   -0.496270   -0.262338   451    1
 7 CT   -0.789906   -1.336520    1.982558   216    2    8    12   13
 8 C    -2.256402   -1.184505    1.571038   218    7    9    10
 9 O2   -2.460769   -0.949098    0.356151   219    8
10 O2   -3.120135   -1.188969    2.465678   219    8
11 H1   -0.478878    0.773493    2.145163   453    2
12 HC   -0.356094   -2.194944    1.466324   217    7
```

```
13 HC    -0.720511   -1.505463    3.058628   217     7
14 H     -0.996208    0.061130   -0.151911   451     1
15 H      0.304306    1.116522   -0.018698   451     1
16 HLA   -0.283317   -0.506767    1.748300  2999     2     7
```

QMMM.key

```
* Change $PATH_TO_TINKER
parameters $PATH_TO_TINKER/params/amber99.prm
QMMM 8
QM -8 10 7 12 13
MM 2
LA 16
* Add the atom type for the LA
atom   2999    99    HLA     "Hydrogen Link Atom"        1      1.008     0
charge -2  0.0
charge -11 0.0
QMMM-MICROITERATION ON
```

Molcas/Gromacs example

To be provided soon.

8.13 EXPBAS

This module is used to manipulate orbital files. By default it is used to expand the orbital file to a larger basis set. This feature is available only for generally contracted basis sets (ANO or CC). The original orbital file ***INPORB*** is needed. By default, the Runfile, ***RUNFIL1***, from the small calculation is also needed. One runs also the GATEWAY module with the larger basis set to obtain the corresponding runfile, ***RUNFIL2***. The program will produce the file ***EXPORB***, which can be used as the input orbital file for calculations using the larger basis set. The DESYMMETRIZE keyword is used to transform orbitals with symmetry to C1 symmetry. If one wants to desymmetrize the orbitals without expanding the basis set, the keyword NOEXPAND must be used. When the DESYMMETRIZE keyword is used, in combination with NOEXPAND, only the ***INPORB*** file is needed. One runs the GATEWAY with the same basis set of the original orbital file to obtain the corresponding runfile. The program will produce the file ***DESORB***.

8.13.1 Dependencies

The EXPBAS requires one InpOrb file and two Runfiles produced by other programs. If the keyword NOEXPAND is used, only the ***INPORB*** file is needed.

8.13.2 Files

Input files

INPORB, an orbital file for the original basis set.

Two more files are needed only if the basis set expansion is to be performed: *RUNFIL1*, the runfile for the small basis set. *RUNFIL2*, the runfile for the larger basis set.

Output files

EXPBAS generates the files: *EXPORB* which contains orbitals, orbital energies, occupation numbers, and type indices for the larger basis set. *DESORB* which contains orbitals, orbital energies, occupation numbers, and type indices for the desymmetrized basis set.

8.13.3 Input

```
&EXPBAS
```

Input example

```
>>COPY name.InpOrb INPORB
>>COPY name.RunFil1 RUNFIL1
>>COPY name.Runfil2 RUNFIL2
 &EXPBAS
```

Optional general keywords

Keyword	*Meaning*
NOEXPAND	Deactivates the basis set expansion.
DESYMMETRIZE	Activates the basis set desymmetrization.
FILEORB	The next line specifies the filename containing the original orbitals that will be expanded to the larger basis set. By default a file named *INPORB* will be used.

Comments. For the basis set expansion procedure, this is the strategy:

1. Do a calculation (SCF, DFT, CASSCF, etc) with a small basis set. Save the produced orbital file and the runfile as: name.Inporb and name.Runfil1.
2. Run a GATEWAY to create the second runfile, which is then saved as name.Runfil2
3. Run EXPBAS as indicated above. The file name.ExpOrb is produced. This file can be used as input for large basis set calculations. Note that the inactive and active orbitals will have the same place in the list as with the smaller basis set, so they can be easily identified.

8.14 FALCON

8.14.1 Description

FALCON calculates total energy of the large system based on the fragment approach. Total energy of the whole system is calculated from total energies of fragments as follows,

$$E^{whole} = \sum C_i^{fragment} E_i^{fragment},$$

where $E_i^{fragment}$ is the total energy of fragment i, and $C_i^{fragment}$ is its coefficient.

In addition to the total energy, FALCON can calculate orbitals of the whole system. Fock matrix and overlap matrix of the whole system are calculated from ones of fragments using following equations,

$$F^{whole} = \sum C_i^{fragment} \mathbf{F}_i^{fragment},$$

and

$$S^{whole} = \sum C_i^{fragment} \mathbf{S}_i^{fragment},$$

where $\mathbf{F}_i$ and $\mathbf{S}_i$ are the Fock matrix and overlap matrix, respectively, of fragment i.

Then

$$\mathbf{FC} = \mathbf{SC}\varepsilon$$

is solved to obtain the orbitals, $\mathbf{C}$, and orbitals energies, ε.

8.14.2 Input

Below follows a description of the input to FALCON.

The input for each module is preceded by its name like:

```
&FALCON
```

Argument(s) to a keyword, either individual or composed by several entries, can be placed in a separated line or in the same line separated by a semicolon. If in the same line, the first argument requires an equal sign after the name of the keyword.

Keywords

Keyword	*Meaning*
TITLe	One-line title.
FRAGment	Takes one, two or three argument(s). The first value (integer) defines the fragment number, the second value (real) determines coefficient, and the third value (integer) is the fragment number that is equivalent to this fragment when translational symmetry is used. A default for the second value is 1.0 where the first and third values have no default. Other keyword(s) specific to this fragment must follow this keyword.

OPERator — A real value following this keyword represents a coefficient, $C_i^{fragment}$, of fragment i (current fragment), where i is a value specified by FRAGMENT keyword. This keyword is equivalent with the second value of keyword, FRAGMENT.

EQUIvalence — An integer, j, following this keyword declares that current fragment is translationally equivalent with fragment j, and information provided for fragment j are tranfered to current fragment. This keyword is equivalent with the third value of keyword, FRAGMENT.

TRANslate — Three real numbers following this keyword specifies the translational vector by which the current fragment is translated to give new coordinate. A unit of either bohr or angstrom can follow. The default unit is angstrom. This keyword takes effect only when the equivalent fragment is specified.

RUNFile — Following this keyword specifies the name of RunFile file for the corresponding fragment.

ONEInt — Following this keyword specifies the name of OneInt file for the corresponding fragment.

NFRAgment — An integer following this keyword specifies the number of fragments. If this keyword is not given, the largest fragment number given by FRAGMENT keyword is set to be the number of fragment.

NIRRep — An integer following this keyword specifies the number of irreducible representation of point group symmetry.

OCCUpation — A list of integer(s) following this keyword specifies the number of occupied orbitals in each symmetry representation in the unfragmented system.

DISTance — A real number following this keyword specifies the distance of two atoms that are equivalent to each other, followed by a unit that is eather angstrom or bohr. Default is angstrom.

NEAR — A real number following this keyword specifies the distance of two atoms within which atoms are considered to be too close each other. An unit that is eather angstrom or bohr can follow. Default is angstrom.

PRINt — An integer following this keyword specifies the format of orbital print out.

ORBEne — A real number follwing this keyword stands for the threshold for orbital print out. The orbitals with orbital energy below this value are print out.

ORBOcc — A real number follwing this keyword stands for the threshold for orbital print out. The orbitals with occupation number above this value are print out.

Input examples

Below shows an example of input file for the three fragment system of which energy, E^{whole}, is written as

$$E^{whole} = E_1^{fragment} + E_2^{fragment} - E_3^{fragment},$$

by fragment energies, $E_1^{fragment}$, $E_2^{fragment}$, and $E_3^{fragment}$.

```
&FALCON
Fragment=1,  1.0
Fragment=2,  1.0
Fragment=3, -1.0
```

which can be simplified as,

```
&FALCON
Fragment=3, -1.0
```

The next example is a two fragment system in which fragment 1 and fragment 2 are equivalent except for their positons. When their difference in position is described by a vector, (1.0, 1.0, -1.0), a translational symmetry can be used and the input becomes as follows,

```
&FALCON
Fragment=2, 1.0, 1
Translate=1.0, 1.0, -1.0
```

If the total energy of the whole system is given by the sum of total energies of three fragment,

$$E^{whole} = E_1^{fragment} + E_2^{fragment} + E_3^{fragment},$$

input is simplly as follows,

```
&FALCON
nFragment=3
```

8.15 FFPT

The program FFPT prepares the one-electron integral file generated by SEWARD for subsequent finite-field perturbation calculations. To do so, the core Hamiltonian matrix is always reconstructed from the nuclear attraction and kinetic energy integrals. The perturbation matrix is then added to the core Hamiltonian matrix where the external perturbation and its strength is specified by input. Any suitable combination of the perturbations is allowed. Following some examples

1. **Dipole moment operator:** This option corresponds to a homogeneous external field perturbation and can be used to calculate dipole moments and dipole polarizabilities.

2. **Quadrupole and higher electric moment operators:** This option corresponds to a non homogeneous external field perturbation and can be used to calculate quadrupole moments and quadrupole polarizabilities, etc.

3. **Relativistic corrections:** This option is used to calculate perturbational relativistic corrections (sum of the mass-velocity and the one-electron Darwin contact term) to the total energy. Note that care must be taken to avoid variational collapse, i.e. the perturbation correction should be small.

For a complete list of one-electron integrals which can be evaluated by the program SEWARD check out the section 8.44.1 and, especially, the subsection 10.8

Note, the perturbation matrices consist of the electronic contributions, only. The quadrupole, electric field gradient and higher electric moment perturbation matrices are given as the traceless tensors.

8.15.1 Dependencies

In order to complete successfully, the program FFPT needs the one-electron integral file. The latter must include all types of integrals needed to construct the perturbed one-electron Hamiltonian.

8.15.2 Files

Input files

The program FFPT needs *ONEINT* (for more information see 7.1.2).

Output files

The program FFPT creates/updates file *ONEINT* on output:

8.15.3 Input

The input to the FFPT program begins with the program name:

```
&FFPT
```

General keywords

The following keywords are known to the FFPT utility:

Keyword	*Meaning*
TITLE	Followed by a title line
DIPO	Add the dipole moment perturbation operator. By default, the dipole moment integrals are always computed with respect to the center of nuclear charge. The keyword is followed by up to three additional input lines. Each line consists of two entries, the component of the dipole operator and the perturbation length. The component is specified by a single letter (X, Y or Z).

QUAD Add the quadrupole moment perturbation operator. The keyword is followed by at least one additional input line and may be complemented by as many additional lines as needed. Each line consists of two entries, the component of the operator and the perturbation strength. The component is specified by a pair of letters (XX, XY, XZ, YY, YZ or ZZ). By default, the quadrupole moment integrals are calculated with respect to the center of mass. For any other selection the origin of the perturbation operator also needs to be specified by entering a line starting with the string ORIG followed by the coordinates.

OCTU Add the octupole moment perturbation operator. The keyword is followed by at least one additional input line and may be complemented by as many additional lines as needed. Each line consists of two entries, the component of the operator and the perturbation strength. The component is specified by a triple of letters (XXX, XXY, XXZ, XYY, XYZ, XZZ, YYY, YYZ, YZZ, or ZZZ). By default, the octupole moment integrals are calculated with respect to the center of mass. For any other selection the origin of the perturbation operator also needs to be specified by entering a line starting with the string ORIG followed by the coordinates.

EFLD Add the electric field perturbation operator. The keyword is followed by at least two additional input lines and may be complemented by as many additional lines as needed. Each line consists of two entries, the component of the operator and the perturbation strength. The component is specified by a single letter (X, Y or Z). In addition, the origin of the perturbation operator also needs to be specified by entering a line starting with the string ORIG followed by the coordinates.

EFGR Add the electric field gradient perturbation operator. The keyword is followed by at least one additional input line and may be complemented by as many additional lines as needed. Each line consists of two entries, the component of the operator and the perturbation strength. The component is specified by a pair of letters (XX, XY, XZ, YY, YZ or ZZ). In addition, the origin of the perturbation operator also needs to be specified by entering a line starting with the string ORIG followed by the coordinates.

RELA Add the relativistic correction (mass-velocity and one-electron Darwin contact term). The command is followed by one additional line of input specifying the perturbation strength.

GLBL This command marks the beginning of a more general perturbation description which is not included as a subcommand of the FFPT command. This card is followed by as many additional input lines as needed and is terminated if the next input line starts with a command. Each input line contains only one perturbation description and three data fields which are: Label, component and perturbation strength. The label consists of a character string of length 8 and names the one- electron integrals produced by SEWARD. The component of an operator is given as an integer. The last parameter denotes the strength of a perturbation operator and is given as a real number. For a list of the available one-electron integral labels refer to section 8.44.

For example to add Pauli repulsion integrals for reaction field calculations the input would look like:

```
 &FFPT
GLBL
 'Well   1' 1 1.000
 'Well   2' 1 1.000
 'Well   3' 1 1.000
```

SELEctive With the same localization scheme as used in LOPROP, the perturbation from FFPT is localized in an orthogonal basis. Then the user can specify on which basis functions the perturbation should act. For example, the input

```
 &FFPT
DIPO
X 0.005
SELECTIVE
2
.true. 1 26
.false. 67 82
.true.
0.5
```

leads to that the perturbation only acts on densities with (1) both basis function indexes in the set $\{1, \cdots, 26\}$ or (2) one index in the set $\{1, \cdots, 26\}$ while the other is in the set $\{67, \cdots, 82\}$, and in this case the perturbation should be multiplied by 0.5.; all other densities are unaffected by the perturbation. We call the former type of subset an atom domain and the latter a bond domain. Generally, the input structure is this: First line specifies how many subsets, N, that will be defined. Then follow N lines starting with a logical flag telling if the subset is an atom domain with the starting and ending basis function indexes thereafter. $N-1$ lines follow where the bond domain is defined in the following way:

```
Do i=2,nSets
  Read(*,*)(Bonds(i,j),j=1,i-1)
Enddo
```

Finally a scalar is given which scales the defined bond domains.

The LoProp-functions will almost coincide with the original input AO-basis, although the localization will modify the meaning slightly, hence it is not possible to exactly localize the perturbation to a group of atoms; LOPROP is a way to come close to perfect localization. FFPT calls LOPROP internally and no call to LOPROP has to specified by the user.

CUMUlative Adds the perturbation to the current H0, enabling many consecutive FFPT calls. Without this keyword, the perturbation always starts from the unperturbed H0.

Input example

The following input will prepare the one-electron integral file generated by SEWARD for subsequent finite-field perturbation calculations by adding a linear electric field in z-direction.

```
 &FFPT
DIPO
Z 0.001
```

Response properties are obtained by numerical differentiation of the total energy with respect to the field parameter. For definitions of the response properties the interested reader is referred to the paper of A.D. Buckingham in *Adv. Chem. Phys.*, Vol 12, p 107 (1967). According to the definition of the dipole moment, it is obtained as the first derivative of the energy with respect to the field strength. Similarly, the dipole polarizability is given by the second derivative of the energy with respect to the field strength.

8.16 GATEWAY

The Gateway module collects information about molecular system (geometry, basis sets, symmetry) to be used for future calculations.

Gateway module is a subset of SEWARD. All keywords for this module can also appear as an input for SEWARD, however, for clearity the information about molecular system can be placed as an input for this module. Note, that gateway module does not compute any integral, and so must be followed by run of SEWARD module.

GATEWAY destroys the communication file *RUNFILE*, if it is used in a combination with geometry optimization it should run outside the optimization loop.

8.16.1 Input

This sections will describe the various possible input blocks in GATEWAY. These control

- the molecular structure (coordinates, symmetry and basis sets),
- explicit auxiliary basis sets in terms of CD basis sets (aCD and acCD) or external auxiliary basis sets,
- parameters for reaction field calculations, i.e. parameters for the Kirkwood model or the PCM model and options for Pauli repulsion integral and external field integrals,
- options for finite nuclear charge distribution models in association with relativistic calculations, and
- the option to use the Saddle method to locate transitions state geometries.

The GATEWAY input section always starts with the program reference:

```
&GATEWAY
```

General keywords

Keyword	*Meaning*
TITLE	The keyword followed by a title.
BASDIR	The keyword allows to set up an extra location for basis set files. The value can be either an absolute path (started from /) or relative to submit directory, e.g. BASDIR=. In order to use a local copy of a basis set file with name FOO - place this file into directory specified in BASDIR
BASLIB	The keyword followed by the absolute path to the basis set library directory. The default is the $MOLCAS/basis_library directory. Note that this directory must also be host to local copies of the .tbl files.
RTRN	Max number of atoms for which bond lengths, angles and dihedral angles are listed, and the radius defining the maximum length of a bond follows on the next line. The latter is used as a threshold when printing out angles and dihedral angles. The length can be followed by BOHR or ANGSTROM which indicates the unit in which the length was specified, the default is BOHR. The default values are 15 and 3.0 au.

Molecular structure: coordinates, symmetry and basis sets

There are three different ways to specify the molecular structure, symmetry and the basis sets in GATEWAY:

- the so-called native input (old MOLCAS standard),
- XYZ input and

Note that only XYZ input for GATEWAY is supported by Graphical User interface. GATEWAY makes a decision about the type of the input based on keywords. If COORD is used, it assumes that the input is in XYZ format.

The three different modes will be described below.

Native input If the geometry is specified in a native MOLCAS format, only symmetry inequivalent atoms should be specified. The default units are atomic units. By default, symmetry is not used in the calculation.

Keyword	*Meaning*

SYMMetry

Symmetry specification follows on next line. There may be up to three different point group generators specified on that line. The generators of a point group is the minimal set of symmetry operators which is needed to generate all symmetry operators of a specific point group. A generator is in the input represented as a sequence of up to three of the characters x, y, and z. The order within a given sequence is arbitrary and the generators can be given in any sequence. Observe that the order of the irreps is defined by the order of the generators as (E, g_1, g_2, g_1g_2, g_3, g_1g_3, g_2g_3, $g_1g_2g_3$)! Note that E is always assumed and should never be specified.

Below is listed the possible generators.

- **x** — Reflection in the yz-plane.
- **y** — Reflection in the xz-plane.
- **z** — Reflection in the xy-plane.
- **xy** — Twofold rotation around the z-axis.
- **xz** — Twofold rotation around the y-axis.
- **yz** — Twofold rotation around the x-axis.
- **xyz** — Inversion through the origin.

The default is no symmetry.

BASIs Set

This notes the start of a basis set definition. The next line always contains a basis set label. The basis set definition is alway terminated with the "End of Basis" keyword. For the definitions of basis set labels see the subsequent sections. Below follows a description of the options associated with the basis set definition.

- **Label** [/ **option**] - The label is a specification of a specific basis set, e.g. C.ANO...4s3p2d., which is an ANO basis set. If no option is specified `GATEWAY` will look for the basis set in the default basis directory. If an option is specified it could either be the name of an alternative basis directory or the wording “Inline” which defines that the basis set will follow in the current input file. For the format of the **Inline** option see the section ‘Basis set format’. Observe that the label is arbitrary for this option and will not be decoded. The **Label** card is mandatory.
- **Name x, y, z (Angstrom or Bohr)** - This card specifies an arbitrary (see next sentence!) name for a symmetry distinct center and its Cartesian coordinates. Observe, that the name "DBAS" is restricted to assign the center of the diffuse basis functions required to model the continuum orbitals in R-matrix calculations. The label is truncated to four characters. Observe that this label must be unique to each center. The coordinate unit can be specified as an option. The default unit is Bohr. There should at least be one card of this type in a basis set definition.

- **Charge** - The real entry on the subsequent line defines the charge associated with this basis set. This will override the default which is defined in the basis set library. The option can be used to put in ghost orbitals as well as to augment the basis sets of the library. The **Charge** card is optional.
- **Spherical** [option] - Specifying which shells will be in real spherical Gaussians. Valid options are ”all” or a list of the shell characters separated by a blank. The shell characters are s, p, d, f, etc. All shells after p are by default in real spherical Gaussians, except for the d-functions in the 6-31G family of basis sets which are in Cartesian. The **Spherical** card is optional. The s and p shells and the d-functions of the 6-31G family of basis sets are by default in Cartesian Gaussians.
- **Cartesian** [option] - Specifying which shells will be in a Cartesian Gaussian representation. For syntax consult the corresponding **Spherical** keyword.
- **Contaminant** [option] - Specifying for which shells the contaminant will be kept. The contaminants are functions of lower rank which are generated when a Cartesian shell is transformed to a spherical representation (e.g. $r^2 = x^2 + y^2 + z^2$ for d-shells, p contaminants for f-shells, s and d contaminants for g-shells, etc). Valid options are the same as for the **Spherical** keyword. The default is no contaminant in any shell. The **Contaminant** card is optional.
- **Muon** [option] - Specifying that the basis set is muonic.
- **End of Basis set** - Marks the end of the basis set specification. This card is mandatory.

Example of an input in native MOLCAS format:

```
 &GATEWAY
Title
formaldehyde
SYMMETRY
X Y
Basis set
H.STO-3G....
H1           0.000000    0.924258   -1.100293 /Angstrom
End of basis
Basis set
C.STO-3G....
C3           0.000000    0.000000   -0.519589 /Angstrom
End of basis
Basis set
O.STO-3G....
O            0.000000    0.000000    0.664765 /Angstrom
End of basis
End of input
```

Z-matrix and XYZ input Some times it is more convenient to set up information about coordinates in a standard form of Z-matrix or Cartesian coordinates. In this case, the basis set for the atoms should be specified after the XBAS keyword. After that either ZMAT or

XYZ should appear to specify the coordinates. Note that coordinates in these formats use ångström as units.

Keyword *Meaning*

XBAS

A keyword to specify the basis for atoms. The specification is very similar to the native format: ATOM.BasisSet. Each new eatom is written at a new line. The end of the keyword is marked by an 'End of basis' line.

If all atoms have the same basis, e.g. ANO-S-VDZ, it is possible to use this name without element name. In this case there is no need to specify 'End of basis'.

Example:

```
XBAS=STO-3G
```

or

```
XBAS
C.STO-3G
H.STO-3G
End of basis
```

ZMAT

Alternative format to give coordinates in terms of bond lengths, bond angles, and dihedral angles. Each line of a Z-matrix gives the internal coordinates for one of the atoms within the molecule with the following syntax:
Name I bond-length J bond-angle K dihedral-angle
Name is the label (atomic symbol + string) for a symmetry distinct center L;
I bond-length distance of L from atom I;
J bond-angle planar angle between atoms L-I-J;
K dihedral-angle dihedral angle between atoms L-I-J-K.
Note that the first atom only requires the **Name** and defines the origin of Cartesian axis. The second atom requires **Name I bond-length** and it will define the Z axis. The third atom requires **Name I bond-length J bond-angle** and defines the XZ plane (and implicitly, the Y axis).
Only numerical values must be used (no variable names) and ångstrms and degrees are assumed as units.

Two types of special atoms are allowed: ***dummy*** **X** atoms and ***ghost*** **Z** atoms. The former will appear in the calculations, they have a nuclear charge of 0 and have not electrons and Basis Set. They will also appear in the definition of internal coordinates in `SLAPAF`. The latter are used only within the Z-Matrix definition of the geometry but they will appear in the final Z-matrix section in `SLAPAF`. Both special atoms can be used to define the Cartesian axis and the symmetry elements.

End of ZMAT or a blank line mark the end of the section.
Here is an example for (S)-1-chloroethanol (C_1 symmetry):

```
XBAS
H.ANO-L...2s1p.
C.ANO-L...3s2p1d.
O.ANO-L...3s2p1d.
Cl.ECP.Huzinaga.7s7p1d.1s2p1d.7e-NR-AIMP.
End of basis
ZMAT
C1
O2 1   1.40000
C3 1   1.45000   2   109.471
H4 1   1.08900   2   109.471     3   120.000
Cl5 1   1.75000   2   109.471     3  -120.000
H6 2   0.94700   1   109.471     3   180.000
H7 3   1.08900   1   109.471     2   180.000
H8 3   1.08900   1   109.471     7   120.000
H9 3   1.08900   1   109.471     7   240.000
End of z-matrix
```

In geometry optimizations, SLAPAF will regenerate the coordinates as Z-matrix in the section with the summary concerning each iteration. This will be possible only if *ghost* atoms are used within the first three atoms or if they are not used at all.
Both BASIs and ZMAT keywords can be used at the same time. Here is an example for a complex between methanol and water (C_s symmetry):

```
Symmetry
 Y
XBAS
H.ANO-L...1s.
C.ANO-L...2s1p.
O.ANO-L...2s1p.
End of basis
ZMAT
C1
O2  1 1.3350
H3  1 1.0890  2 109.471
H4  1 1.0890  2 109.471  3 -120.
H6  2 1.0890  1 109.471  3  180.
End of z-matrix
Basis set
O.ANO-L...2s1p.
 O    -2.828427     0.000000  2.335000  / Angstrom
End of basis
Basis set
H.ANO-L...1s.
 H    -2.748759     0.819593  2.808729  / Angstrom
End of basis
```

In this case SLAPAF will not regenerate the Z-matrix.

XYZ

The keyword is followed by XYZ formatted file (a reference to a file), or file, inlined into the input.

Example:

```
XBAS=STO-3G
XYZ=$CurrDir/Water.xyz
```

or

```
XBAS=STO-3G
XYZ
1
 note Angstrom units!
C 0 0 0
```

Currently, the XYZ keyword does not operate with symmetry, and the calculation is always performed without symmetry.

Advanced XYZ input If the geometry is specified in XYZ format, all atoms should be specified. The default units are Ångstroms. By default, maximum possible symmetry is used.

'Molcas XYZ' file format is an extension of plain XYZ format.

First line of this file contains the number of atoms.

Second line (a comment line) can contain 'a.u.' or 'bohr' to use atomic units, instead of default Ångstroms. Also this line can contain keyword TRANS, followed by 3 numbers, and/or ROTATE, followed by 9 numbers (in this case coordinates will be Translated by specified vector, and/or Rotated), and SCALE (or SCALEX, SCALEY, SCALEZ) followed by a scale factor.

Remaining lines are used to specify Element and cartesian coordinates.

Element name might be optionally followed by a Number (e.g. $H7$), a Label (separated by _ sign: e.g. H_INNER), or Basis Set (separated by . , e.g. $H.STO-3G$)

Keyword *Meaning*

COORD The keyword followed on the next line by the name of an HDF5 file (created by any module), or the name of an XYZ file, or inline coordinates in XYZ format. If the file is located in the same directory, where MOLCAS job was submitted there is no need to specify the PATH to this file. The keyword may appear several times. In this case all coordinate files will be concatenated, and considered as individual fragments.

BASIS The keyword can be used to specify global basis set for all atoms, or for a group of atoms. The keyword followed by a label of basis set, or by comma separated list of basis sets for individual atoms.

Note! The basis set definition in XYZ mode does not allow to use inline basis set.

Example:

```
COORD
4
C           0.00000 0.00000 0.00000
H           1.00000 0.00000 0.00000
H           0.00000 1.00000 0.00000
H           0.00000 0.00000 1.00000
```

```
BASIS
STO-3G, H.6-31G*
```

In this example, the C atom (in the origin) will have the basis set STO-3G and the H atoms 6-31G*.

If keyword BASIS never appears in the input, the default basis, ANO-S-MB, will be used.

GROUP

The keyword can be used to specify the symmetry of the molecule.

The keyword must be followed by one of:

- FULL (default) - use maximum possible subgroup of D_{2h}
- NOSYM (same as E, or C1)
- space separated list of generators: e.g. X XY (for more details see SYMMETRY keyword)

Limitations: in the current implementation atom labels, and basis sets are ignored during symmetry recognition.

If XYZ input has been used in `GATEWAY`, a file with native Molcas input will be produced and stored in working directory under the name ***`findsym.std`***.

Note that choosing XYZ input you are expecting that the coordinates might be transformed. It can be shown by the following example:

```
&gateway
coord
3

O 0 0 0
H 1.0000001 0 0
H 0 1 0.0000001
*nomove
*group=c1
```

The geometry of the molecule is slightly distorted, but within a threshold it is C_{2v}. Thus by default (keywords NOMOVE and GROUP are not active), the coordinates will be transformed to maintain the highest possible symmetry. If keyword NOMOVE is active, the molecule is not allowed to rotate, and a mirror plane XY is the only symmetry element. Since the third hydrogen atom is slightly out of this plane, it will be corrected. Only activation of the keyword GROUP=C1 will ensure that the geometry is unchanged.

Advanced keywords:

Keyword	*Meaning*
SYMTHRESHOLD	followed by a real number - threshold for symmetry recognition (default is 0.1)

MOVE allow to translate and rotate molecule in order to find highest possible symmetry. (this is a default for all groups, except of C_1)

NOMOVE do not allow to transform coordinates while searching for highest group (default for C_1 group)

BSSE followed by an integer. Indicates which XYZ-file that should be treated like ghost atoms.

VART Specifies that the energy should not be considered invariant to translations. Translational variance is detected automatically, but sometimes it may be useful to enforce it.

VARR Specifies that the energy should not be considered invariant to rotations. Rotational variance is detected automatically, but sometimes it may be useful to enforce it.

NUMErical Forces the calculation of numerical gradients even when analytic gradients are available.

SHAKe Randomly modifies the initial coordinates of the atoms, maintaining the input (or computed) symmetry. This can be useful to avoid a geometry optimization converging to a higher-symmetry saddle point. The maximum displacement in the axes x, y and z is read from the following real number. This number can be followed by Bohr or Angstrom, which indicates the unit in which the displacement is specified, the default is Bohr.

```
Example:
 &GATEWAY
COORD
water.xyz
BASIS
STO-3G
```

or, in short EMIL notation:

```
 &GATEWAY
COORD=water.xyz; BASIS=STO-3G
```

Coordinate file may contain variables, as demonstrated in an example:

```
>>FILE H2.input
2
scale $DD
H 0.35 0 0
H -0.35 0 0
>>EOF
>> FOREACH DD IN ( 0.9 1.0 1.1 )
&GATEWAY
COORD=$WorkDir/H2.input
BASIS=STO-3G
&SEWARD
&SCF
>>ENDDO
```

The atom name in XYZ file can contain an orbitrary label (to simplify assigning of different basis sets). To indicate the label, use _: e.g. C_SMALL. The same label should be defined in the basis section: $BASIS = C\_SMALL.ANO - S - MB$. The basis set label can be also added into the name of an element:

```
COORD
1
O.ANO-S-VDZP 0 0 0
```

XYZ file can also contain information about point charges. There are three possibilities to setup atomic charges in XYZ file. The main option is to use Q as an element name, and in this case the forth number, the charge, should be specified. Another possibility is to use element names ended with minus sign. In this case, a formal charge for the element will be used. E.g. $H-, Li-, Na-, K-$ defines +1 charge located in the corresponding location. $Mg-, Ca-$ - defines charge +2, $Al-$ − +3, $C-, Si-$ +4, for anions, $F-, Cl-, Br-, I-$ defines −1, $O-, S-$ - −2. Finally, a label at the comment line of XYZ file - CLUSTER followed by an integer number can specify how many atoms are 'real', so the rest will be treated as charges with default values for this element.

Constraints

In case of optimizations with constraints these are defined in the GATEWAY input. For a complete description of this keyword see the section 8.46.4.

Keyword	*Meaning*
CONSTRAINTS	This marks the start of the definition of the constraints which the optimization is subject to. This section is always ended by the keyword END OF CONSTRAINTS. This option can be used in conjunction with any definition of the internal coordinates.
NGEXCLUDE	This marks the start of the definition of additional restrictions for numerical differentiation. This section is always ended by the keyword END OF NGEXCLUDE. The syntax of this section is like that of normal constraints, and the degrees of freedom specified here will be excluded from numerical differentiation (like phantom constraints). If a line containing only "Invert" is included inside the section, the definition is reversed and only these degrees of freedom are differentiated. NGEXCLUDE is intended for use with the KEEPOLDGRADIENT keyword in ALASKA, and can be combined with CONSTRAINTS, which will further reduce the numerical differentiation subspace [17]. Note that the value assigned to the constraints in this section is unused, but a "Value" block must still be included.

Explicit auxiliary basis sets

The so-called Resolution of Identity (RI) technique (also called Density Fitting, DF) is implemented in the MOLCAS package. This option involves the use of an auxiliary basis set in the effective computation of the 2-electron integrals. MOLCAS incorporates both the use of conventionally computed, externally provided, auxiliary basis sets (RIJ, RIJK, and RIC types), and on-the-fly generated auxiliary basis sets. The latter are atomic CD (aCD) or the atomic compact CD (acCD) basis sets, based on the Cholesky decomposition method. The externally provided auxiliary basis sets are very compact, since they are tailored for special wave function methods. However, they are not provided for all available valence basis sets. The aCD or acCD RI auxiliary basis sets are a more general option and provides auxiliary basis sets for any wave function model and valence basis set.

Keyword	*Meaning*
RIJ	Use the RI-J basis in the density fitting (DF) approach to treat the two-electron integrals. Note that the valence basis set must have a supporting auxiliary basis set for this to work.
RIJK	Use the RI-JK auxiliary basis in the density fitting (DF) approach to treat the two-electron integrals. Note that the valence basis set must have a supporting auxiliary basis set for this to work.
RIC	Use the RI-C auxiliary basis in the density fitting (DF) approach to treat the two-electron integrals. Note that the valence basis set must have a supporting auxiliary basis set for this to work.
RICD	Use the aCD or acCD approach [7] to treat the two-electron integrals. This procedure will use an on-the-fly generated auxiliary basis set.
CDTHRESHOLD	Threshold for on-the-fly generation of aCD or acCD auxiliary basis sets for RI calculations (default value 1.0d-4).
SHAC	Skip high angular combinations à la Turbomole when creating on-the-fly basis sets (default off).
KHAC	Keep high angular combinations when creating on-the-fly basis sets (default on).
ACD BASIS	Generate an atomic CD (aCD) auxiliary basis sets (default off).
ACCD BASIS	Generate an atomic compact CD (acCD) auxiliary basis sets (default on).

Reaction field calculations

The effect of the solvent on the quantum chemical calculations has been introduced in MOLCAS through the reaction field created by the surrounding environment, represented by a polarizable dielectric continuum outside the boundaries of a cavity containing the solute molecule. MOLCAS supports Self Consistent Reaction Field (SCRF) and Multi Configurational Self Consistent Reaction Field (MCSCRF) calculations within the framework of the `SCF` and the `RASSCF` programs. The reaction field, computed in a self-consistent fashion, can be later added as a constant perturbation for the remaining programs, as for example CASPT2.

The purpose of this facility is to incorporate the effect of the environment (a solvent or a solid matrix) on the studied molecule. The utility itself it is not a program, but requires an additional input which has to be provided to the GATEWAY program. Two methods are available for SCRF calculations: one is based on the Kirkwood model, the other is the so called Polarizable Continuum Model (PCM). The reaction field is computed as the response of a dielectric medium polarized by the solute molecule: the solute is placed in a "cavity" surrounded by the dielectric. In Kirkwood model the cavity is always spherical, whereas in PCM the cavity is modeled on the actual solute shape.

The possible set of parameters controlled by input are:

- the Kirkwood model,
- the PCM model, and
- one-electron integrals representing Pauli repulsion and external fields.

First a brief presentation of the Kirkwood and the PCM models.

The Kirkwood Model The Kirkwood model is an expansion of the so-called Onsager model where the surrounding will be characterized by its dielectric permitivity and a radius describing a spherical cavity, indicating where the dielectric medium starts. (Note that all atoms in the studied molecule must be inside the spherical cavity.) The Pauli repulsion due to the medium can be introduced by use of the spherical well integrals which are generated by SEWARD. The charge distribution of the molecule will introduce an electric field acting on the dielectric medium. This reaction field will interact with the charge distribution of the molecule. This interaction will manifest itself as a perturbation to the one-electron Hamiltonian. The perturbation will be automatically computed in a direct fashion (no multipole integrals are stored on disk) and added to the one-electron Hamiltonian. Due to the direct way in which this contribution is computed rather high terms in the multipole expansion of the charge can be afforded.

The Polarizable Continuum Model, PCM The PCM has been developed in order to describe the solvent reaction field in a more realistic way, basically through the use of cavities of general shape, modeled on the solute. The cavity is built as the envelope of spheres centered on solute atoms or atomic groups (usually, hydrogen atoms are included in the same sphere as the heavy atoms they are bonded to). The reaction field is described by means of apparent charges (solvation charges) spread on the cavity surface, designed to reproduced the electrostatic potential due to the polarized dielectric inside the cavity. Such charges are used both to compute solute-solvent interactions (modifying the total energy of the solute), and to perturb the molecular Hamiltonian through a suitable operator (thus distorcing the solute wave-function, and affecting all the electronic properties). The PCM operator contains both one- and two-electron terms: it is computed using atomic integrals already present in the program, through a "geometry matrix" connecting different points lying on the cavity surface. It can be shown that with this approach the SCF and RASSCF variational procedures lead to the free energy of the given molecule in solution: this is the thermodynamic meaning of the SCF or CI energy provided by the program. More precisely, this is the solute-solvent electrostatic contribution to the free energy (of course, other terms depending on solute atomic motions, like vibrational and rotational free energies, should be included separately); it can be used to get a good approximation of the solvation free energy, by subtracting the SCF or CI energy computed in vacuo, and also to compute directly energy surfaces and reaction paths in solution. On the other hand, the solute wave-function perturbed by the reaction field can be used to compute any electronic property in solution.

Also other quantities can be computed, namely the cavitation free energy (due the the work spent to create the cavity in the dielectric) and the dispersion-repulsion free energy: these terms affect only the total free energy of the molecule, and not its electronic distribution. They are collectively referred to as non-electrostatic contributions.

Note that two other keywords are defined for the `RASSCF` program: they refer to the CI root selected for the calculation of the reaction field (RFROOT), and to the possibility to perform a non-equilibrium calculation (NONEQ) when vertical electronic transitions are studied in solution. These keywords are referenced in the `RASSCF` section. To include the reaction field perturbation in a `SCF`, `RASSCF`, `CASPT2` or `RASSI` calculation, another keyword must be specified (RFPERT), as explained in the respective program sections.

Complete and detailed examples of how to add a reaction field, through the Kirkwood or the PCM model, into quantum chemical calculations in Molcas is presented in section 10.6 of the examples manual. The user is encouraged to read that section for further details.

Input for the Kirkwood and PCM models

Files The reaction field calculations will store the information in the following files, which will be used by the following programs

File	*Contents*
ONEINT	One-electron integral file used to store the Pauli repulsion integrals
RUNFILE	Communications file. The last computed self-consistent reaction field (SCF or RASSCF) will be stored here to be used by following programs
GV.off	Input file for the external program "geomview" (see Tutorial section "Solvent models"), for the visualization of PCM cavities

Input Below follows a description of the input to the reaction field utility in the `GATEWAY` program. The `RASSCF` program has its own keywords to compute reaction fields for excited states.

Compulsory keywords

Keyword	*Meaning*

RF-Input	Activate reaction field options.
END Of RF-Input	This marks the end of the input to the reaction field utility.

Optional keywords for the Kirkwood Model

Keyword	*Meaning*
REACtion Field	This command is exclusive to the Kirkwood model. It indicates the beginning of the specification of the reaction field parameters. The subsequent line will contain the dielectric constant of the medium, the radius of the cavity in Bohrs (the cavity is always centered around the origin), and the angular quantum number of the highest multipole moment used in the expansion of the change distribution of the molecule (only charge is specified as 0, charge and dipole moments as 1, etc.). The input specified below specifies that a dielectric permitivity of 80.0 is used, that the cavity radius is 14.00 a.u., and that the expansion of the charge distribution is truncated after l=4, i.e. hexadecapole moments are the last moments included in the expansion. Optionally a fourth argument can be added giving the value of the dielectric constant of the fast component of the solvent (default value 1.0).

Sample input for the reaction field part (Kirkwood model)

```
RF-Input
Reaction field
80.0 14.00 4
End Of RF-Input
```

Sample input for a complete reaction field calculation using the Kirkwood model. The SCF computes the reaction field in a self consistent manner while the MRCI program adds the effect as a constant perturbation.

```
 &GATEWAY
Title =  HF molecule
Symmetry
X Y
Basis set
F.ANO-S...3S2P.
F      0.00000   0.00000   1.73300
End of basis
Basis set
H.ANO-S...2S.
H      0.00000   0.00000   0.00000
End of basis
Well integrals
 4
 1.0 5.0  6.75
 1.0 3.5  7.75
 1.0 2.0  9.75
 1.0 1.4 11.75
RF-Input
Reaction field
 80.0 4.75 4
```

```
End of RF-Input
 &SEWARD
 &SCF
Occupied =  3 1 1 0
 &MOTRA
LumOrb
Frozen   =  1 0 0 0
RFPert
 &GUGA
Electrons =     8
Spin      =     1
Inactive  =     2    1    1    0
Active    =     0    0    0    0
CiAll     =     1
 &MRCI
SDCI
```

Optional keywords for the PCM Model

Keyword *Meaning*

PCM-model If no other keywords are specified, the program will execute a standard PCM calculation with water as solvent. The solvent reaction field will be included in all the programs (`SCF`, `RASSCF`, `CASPT2`, etc) invoked after `SEWARD`: note that in some cases additional keywords are required in the corresponding program sections. Some PCM parameters can be changed through the following keywords.

SOLVent Used to indicate which solvent is to be simulated. The name of the requested solvent must be written in the line below this keyword. Find implemented solvents in the PCM model below this section.

DIELectric constant Defines a different dielectric constant for the selected solvent; useful to describe the system at temperatures other that 298 K, or to mimic solvent mixtures. The value is read in the line below the keyword. An optional second value might be added on the same line which defines a different value for the infinite frequency dielectric constant for the selected solvent (this is used in non-equilibrium calculations; by default it is defined for each solvent at 298 K).

CONDuctor version It requires a PCM calculation where the solvent is represented as a polarized conductor: this is an approximation to the dielectric model which works very well for polar solvents (i.e. dielectric constant greater than about 5), and it has some computational advantages being based on simpler equations. It can be useful in cases when the dielectric model shows some convergence problems.

AAREa It is used to define the average area (in Å^2) of the small elements on the cavity surface where solvation charges are placed; when larger elements are chosen, less charges are defined, what speeds up the calculation but risks to worsen the results. The default value is 0.4 Å^2 (i.e. 60 charges on a sphere of radius 2 Å). The value is read in the line below the keyword.

R-MIn
: It sets the minimum radius (in Å) of the spheres that the program adds to the atomic spheres in order to smooth the cavity surface (default 0.2 Å). For large solute, if the programs complains that too many sphere are being created, or if computational times become too high, it can be useful to enlarge this value (for example to 1 or 1.5 Å), thus reducing the number of added spheres. The value is read in the line below the keyword.

PAULing
: It invokes the use of Pauling's radii to build the solute cavity: in this case, hydrogens get their own sphere (radius 1.2 Å).

SPHEre radius
: It is used to provide sphere radii from input: for each sphere given explicitly by the user, the keyword "Sphere radius" is required, followed by a line containing two numbers: an integer indicating the atom where the sphere has to be centered, and a real indicating its radius (in Å). For example, "Sphere radius" followed by "3 1.5" indicates that a sphere of radius 1.5 Å is placed around atom #3; "Sphere radius" followed by "4 2.0" indicates that another sphere of radius 2 Å is placed around atom #4 and so on.

Solvents implemented in the PCM model are

Name	Dielectric constant	Name	Dielectric constant	Name	Dielectric constant
water	78.39	dichloroethane	10.36	toluene	2.38
dimethylsulfoxide	46.70	quinoline	9.03	benzene	2.25
nitromethane	38.20	methylenchloride	8.93	carbontetrachloride	2.23
acetonitrile	36.64	tetrahydrofuran	7.58	cyclohexane	2.02
methanol	32.63	aniline	6.89	heptane	1.92
ethanol	24.55	chlorobenzene	5.62	xenon	1.71
acetone	20.70	chloroform	4.90	krypton	1.52
isoquinoline	10.43	ethylether	4.34	argon	1.43

Sample input for the reaction field part (PCM model): the solvent is water, a surface element average area of 0.2 Å^2 is requested.

```
RF-input
PCM-model
Solvent
water
AAre
0.2
End of RF-input
```

Sample input for a standard PCM calculation in water. The SCF and RASSCF programs compute the reaction field self consistently and add its contribution to the Hamiltonian. The RASSCF is repeated twice: first the ground state is determined, then a non-equilibrium calculation on the first excited state is performed.

```
 &GATEWAY
Coord
4
formaldehyde
O 0.000000 0.000000 -1.241209
C 0.000000 0.000000 0.000000
H 0.000000 0.949585 0.584974
H 0.000000 -0.949585 0.584974
Basis = STO-3G
Group = C1
RF-input
PCM-model
solvent = water
End of RF-input
 &SEWARD ;  &SCF
 &RASSCF
nActEl   = 4 0 0
Symmetry = 1
Inactive = 6
Ras2     = 3
CiRoot
1 1
1
LumOrb
 &RASSCF
nActEl   = 4 0 0
Symmetry = 1
Inactive = 6
Ras2     = 3
CiRoot
2 2
1 2
0 1
JOBIPH
NonEq
RFRoot  = 2
```

Again the user is recommended to read section 10.6 of the examples manual for further details.

Keywords associated to one-electron integrals

Keyword	*Meaning*
FNMC	Request that the so-called Finite Nuclear Mass Correction exclude, by the Born–Oppenheimer approximation, be added to the one-electron Hamiltonian.
WELL INTEGRALS	Request computation of Pauli repulsion integrals for dielectric cavity reaction field calculations. The first line specifies the total number of primitive well integrals in the repulsion integral. Then follows a number of lines, one for each well integral, specifying the coefficient of the well integral in the linear combination of the well integrals which defines the repulsion integral, the exponent of the well integral, and the distance of the center of the Gaussian from the origin. In total three entries on each line. All entries in atomic units. If zero

or a negative number is specified for the number of well integrals a standard set of 3 integrals with their position adjusted for the radius of the cavity will be used. If the distance of the center of the Gaussian from the origin is negative displacements relative to the cavity radius is assumed.

XFIEld integrals Request the presence of an external electric field represented by a number of partial charges and dipoles. Optionally, polarisabilities may be specified whose induced dipoles are determined self-consistently during the SCF iteration. The first line may contain, apart from the first integer [nXF] (number of centers), up to four additional integers. The second integer [nOrd] specifies the maximum multipole order, or -1 signifying no permanent multipoles. Default is 1 (charges and dipoles). The third integer [p] specifies the type of external polarisabilities: 0 (default) no polarisabilities, 1 (isotropic), or 2 (anisotropic). The fourth integer [nFrag] specifies the number of fragments one multipole may contribute to (relevant only if polarisabilities are present). The default is 0, meaning that each permanent multipole is only excluded in the calculation of the field at its own polarisability, 1 means that one gives a fragment number to each multipole and that the static multipoles do not contribute to the polarising field within the same fragment, whereas 2 can be used in more complex situations, e.g. polymers, allowing you to specify a second fragment number so that junction atoms does not contribute to either of the neighbouring fragments. Finally, the fifth and last integer [nRead] (relevant only if Langevin dipoles are used) may be 0 or 1 (where 0 is default), specifying wheather an element number (e.g. 8 for oxygen) should be read for each multipole. In that case the default radius for that element is used to determine which Langevin grid points should be annihilated. A negative element number signifies that a particular radius should be used for that multipole, in thousands of a Bohr (-1400 meaning 1.4 Bohr). Then follows nXF lines, one for each center. On each line is first nFrag+nRead (which may equal 0) integers, specifying the fragments that the multipole should not contribute to (the first fragment is taken as the fragment that the polarisability belongs to) and the element number. Then follows the three coordinates of the center, followed by the multipoles and polarisabilities. The number of multipole entries is 0 for nOrd=-1, 1 for nOrd=0, 4 for nOrd=1, and 10 for nOrd=2. The number of polarisability entries are 0 for p=0, 1 for p=1, and 6 for p=2. The order of quadrupole moment and anisotropic polarisability entries is xx, xy, xz, yy, yz, zz. If default is used, i.e. only specifying the number of centers on the first line, each of these lines will contain 7 entries (coordinates, charge, and dipole vector). All entries are in atomic units, if not otherwise requested by the Angstrom keyword that must be placed between nXF and nOrd. All these data can be stored in a separate file whose name must be passed as an argument of the XField keyword.

ANGM Supplement ***ONEINT*** for transition angular momentum calculations.

Entry which specifies the angular momentum origin (in au).

OMQI

Supplement *ONEINT* for transition orbital magnetic quadrupole calculations. Entry which specifies the orbital magnetic quadrupole origin (in au).

AMPR

Request the computation of angular momentum product integrals. The keyword is followed by values which specifies the angular momentum origin (in au).

DSHD

Requests the computation of diamagnetic shielding integrals. The first entry specifies the gauge origin. Then follows an integer specifying the number of points at which the diamagnetic shielding will be computed. If this entry is zero, the diamagnetic shielding will be computed at each nucleus. If nonzero, then the coordinates (in au) for each origin has to be supplied, one entry for each origin.

EPOT

An integer follows which represents the number of points for which the electric potential will be computed. If this number is zero, the electric field acting on each nucleus will be computed. If nonzero, then the coordinates (in au) for each point have to be supplied, one entry for each point. This keyword is mutually exclusive with EFLD and FLDG.

EFLD

An integer follows which represents the number of points for which the electric potential and electric field will be computed. If this number is zero, the electric field acting on each nucleus will be computed. If nonzero, then the coordinates (in au) for each point have to be supplied, one entry for each point. This keyword is mutually exclusive with EPOT and FLDG.

FLDG

An integer required which represents the number of points for which the electric potential, electric field and electric field gradient will be computed. If this number is zero, the electric field gradient acting on each nucleus will be computed. If nonzero, then either the coordinates (in au) for each point or labels for each atom center have to be supplied, one entry for each point. In case a label is supplied it must match one of those given previous in the input during specification of the coordinates of the atom centers. Using a label instead of a coordinate can e.g. be useful in something like a geometry optimization where the coordinate isn't known when the input is written. This keyword is mutually exclusive with EPOT and EFLD.

EMPC

Use point charges specified by the keyword XField when calculating the Orbital-Free Embedding potential.

RF-Input

Specification of reaction field parameters, consult the reaction field section of this manual.

Keywords associated with nuclear charge distribution models

Input parameters associated with different models of the nuclear charge distribution. The default is to use a point charge representation.

Keyword	*Meaning*
FINIte	Request a finite center representation of the nuclei by a single exponent s-type Gaussian.
MGAUSsian	Request a finite center representation of the nuclei by a modified Gaussian.

The Saddle method for transition state optimization

The Saddle method [48] is a method to locate transition states (TS). The method, in practice, can be viewed as a series of constrained optimization along the reaction path, which connects two starting structure (could be the reactants and products of a reaction), to locate the region of the TS and a subsequent unconstrained optimization to locate the TS. The only data needed for the procedure are the energies and coordinates of the two structures. **Note** that this option will overwrite the coordinates which have already been specified with the normal input of the molecular geometry. However, this does not make that input section redundant and should always be included.

Keyword	*Meaning*
RP-Coordinates	This activates the Saddle method for TS geometry optimization. The line is followed by an integer specifying the number of symmetry unique coordinates to be specified. This is followed by two sets of input - one line with the energy and then the Cartesian coordinates in bohr - for each of the two starting structures of the Saddle method. Note that the order of the coordinates must always match the order specified with the conventional input of the coordinates of the molecular system. Alternatively, two lines with the filenames containing the coordinates of reactants and products, respectively, (in XYZ format) can be given.
NOALign	By default, the two starting structures are aligned to minimize the root mean square distance (RMSD) between them, in particular, the first structure is moved and the second structure remains fixed. If this keyword is given, the starting structures are used as given.
ALIGn only	The two starting structures are aligned, but nothing more is done. An input block for SEWARD is still needed, but no integrals are computed.

WEIGhts Relative weights of each atom to use for the alignment and for the calculations of the "distance" between structures. The possibilities are:

MASS. This is the default. Each atom is given a weight proportional to its mass. Equivalent to mass-weighted coordinates.

EQUAL. All atoms have an equal weight.

HEAVY. Only heavy atoms are considered, with equal weights. Hydrogens are given zero weight.

A list of N numbers can also be provided, and they will be used as weights for the N symmetry-unique atoms. For example:

```
WEIGht
0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0
```

will align only atoms 7–12 out of 16.

Note that, in any case, weights of 0 are likely to cause problems with constraints, and they will be increased automatically.

SADDle Step size reduction for each macro iteration of the saddle method. The value is given in weighted coordinates, divided by the square root of the total weight (see the WEIGHTS keyword). Default value is 0.1 au.

Geometry optimization using constrained internal coordinates

These keyword are used together with the GEO to optimize the relative position of two or more rigid fragments. The starting geometry can either be defined by supplying an xyz-file for each fragment using the keyword COORD or by placing a file named *$Project.zmt* in a directory named *$Project.GEO* . The z-matrix should be in the following format:

```
H
O     0.982011 0                                 1
H     0.982013 0   104.959565 0                  2   1
H     1.933697 1   107.655494 1   114.496053 1   2   3   1
O     0.988177 0   173.057942 1   -56.200750 1   4   2   3
H     0.979890 0   104.714572 0   179.879745 1   5   4   2
```

where the three columns of real numbers are internal coordinates, and the last three columns of integers indicate which other atoms that are used to define the coordinate. The type of coordinates from left to right are bond distances, bond angles and dihedral angels, both for the coordinates and the link. The column of integers just to the right of each coordinate indicate if this coordinate should be optimized or not (1 = optimize, 0 = do not optimize).

There are also two utility-keywords used to create a z-matrix or to write out a constraint-definition for SLAPAF and keywords to rotate and translate fragments. (See documentation for GEO for more details)

Keyword	*Meaning*

HYPER

This keyword is used to specify that a geometry optimization with constrained internal coordinates shall be performed later, a z-matrix and a set of displaced geometries are therefore constructed. The keyword should be followed by three real numbers defining the maximum displacement for each coordinate type. The order from left to right is bond distances, bond angles and dihedral angles. To use default values for the parameters the mutually exclusive keyword GEO should be entered instead.

GEO

This keyword is used to specify that a geometry optimization with constrained internal coordinates shall be performed later, a z-matrix and a set of displaced geometries are therefore constructed. Default values of 0.15 Å, 2.5 degrees, and 2.5 degrees are used for the maximum displacement of bond distances, bond angles and dihedral angles respectively. To enter other values for the parameters the mutually exclusive keyword HYPER should be used.

OPTH

This keyword is used to define the specific details of the optimization algorithm used for the geometry optimization in constrained internal coordinates. This keyword should be followed by two to three lines of parameter. The first line should contain an integer indicating optimization type (1 = steepest descent, 2 = a mix of steepest descent and Newton's method, and 3 = Newton's method). The second line should contain a real number defining a step factor. This number is multiplied with the gradient to obtain the step length. For optimization type 2 a third line containing a real number that defines a gradient limit should be entered. This limit determines how large the gradient must be for the steepest descent algorithm to be used. When the gradient is smaller than this limit Newton's method is used instead.

OLDZ

This keyword is used both to start a new calculation from a user-defined z-matrix and to restart calculations. When using the keyword for a new calculation a directory `$Project.GEO` must exist and contain a file called `$Project.zmt` with a z-matrix in the format defined above. The directory must not contain any files with the suffix `.info` when performing a fresh calculation since these files contain restart information.

ZONLY

This keyword is used to construct a z-matrix from a set of xyz-files (fragments) and store it in the directory `$Project.GEO`. The optimization parameters of the resulting z-matrix are set so that only coordinates linking fragments are set to 1 (= optimize coordinate).

ZCONS

This keyword is used to define constraints from a set of xyz-files (fragments) on a form that could be supplied to the SLAPAF in order to keep the fragments rigid. The resulting constraints-file is named `$Project.cns` and stored in the directory `$Project.GEO`. The atom-numbers in this constraint-file will not match those of your original xyz-file and should not be used together with this. Instead a new xyz-file named `cons.xyz` is created and placed into the directory

$Project.GEO, this has the proper numbering to use together with the constraints.

ORIGIN — This keyword is used to translate and rotate a set of xyz-files. The keyword must be entered before the xyz-files is entered with COORD. The keyword should be followed by one line with the number of fragments and then one line for each fragment that should be translated. This row should contain 13 numbers. One integer defining which fragment that should be moved, (the fragments are numbered based on order of appearance in the input-file from top to bottom), 3 real numbers defining a translation (x, y, z) and 9 numbers defining a rotation (row1, row2, row3 of rotation matrix). The keyword ORIGIN is mutually exclusive with the keyword FRGM which is an alternative way to express the same rotations and translations.

FRGM — This keyword is used together with the keywords ROT and TRANS to define rotation and translation of a specific fragment. FRGM defines an active fragment (each xyz-file is considered a fragment, the files are numbered based on order of appearance in the input from top to bottom). The keyword must be entered before the xyz-file it is supposed to modify is entered with COORD. Each occurence of FRGM should be followed by either one of or both of the keywords ROT and TRANS to define rotation and translation. This keyword is mutually exclusive with the keyword ORGIN

ROT — This keyword should be followed by nine real numbers defining the rotation for the fragment defined by the preceeding FRGM. The numbers should be the nine elements of a rotation matrix listed with one full row at the time.

TRANS — This keyword should be followed by three real numbers defining the translation for the fragment defined by the preceeding FRGM. The numbers should be the x, y and z coordinates of the translation in that order.

Example of an input:

```
 &GATEWAY
Title
Water Dimer
frgm=2
trans=3.0 0.0 0.0
Coord=water_monomer.xyz
Coord=water_monomer.xyz
Group=c1
basis=cc-pVTZ
hyper
0.2 3.0 3.0
opth
3
15.0d0
```

In this example a water dimer is constructed from a single monomer by translating it 3.0 Åwith the keyword trans. An optimization in constrained internal coordinates using newtons method with a step-factor of 15.0d0 are prepared for. For more details on these optimization see the manual entry for the module GEO.

QM/MM calculations with Molcas/Gromacs

The following keywords apply to QM/MM calculations performed with the MOLCAS/GROMACS interface (see section 8.12 for more details).

Keyword	*Meaning*
GROMACS	Requests that the definition of the full QM+MM system should be imported from GROMACS. The keyword should be followed by one of the options SIMPLE or CASTMM on the next line. In the case of SIMPLE, all MM atoms defined in the GROMACS input will be treated as *outer* MM atoms in MOLCAS. This means, for example, that in a geometry optimization, their positions will be updated using microiterations rather than the conventional optimization scheme. Conversely, CASTMM requests that certain MM atoms should be treated as *inner* MM atoms in MOLCAS. Their positions will be updated with the same scheme as used for the QM atoms. The CASTMM option should be followed by two additional input lines, the first one containing the number of MM atoms to convert from outer to inner type, and the second containing a list of those atoms (using their corresponding GROMACS indices).
LINKATOMS	Defines link atoms for use with the Morokuma updating scheme. The desired number of link atoms should be given as an integer on the next line. This should be followed by additional input lines, one for each link atom to be defined. Each definition should be of the form ILA, IQM, IMM, SCALE, where ILA, IQM and IMM are the GROMACS indices of the link atom and the corresponding QM and MM frontier atoms, respectively. SCALE is the scaling factor to be used in the Morokuma scheme. Note that each link atom must be defined as a QM atom in the GROMACS input. In addition, the frontier MM atom must be an inner MM atom specified as discussed above.

8.17 GENANO

GENANO is a program for determining the contraction coefficients for generally contracted basis sets [49]. They are determined by diagonalizing a density matrix, using the eigenvectors (natural orbitals) as the contraction coefficients, resulting in basis sets of the ANO (Atomic Natural Orbitals) type [50].

Some elementary theory: We can do a spectral resolution of a density matrix D

$$D = \sum_k \eta_k c_k c_k^\dagger \tag{8.4}$$

where η_k is the k'th eigenvalue (occupation value) and c_k is the k'th eigenvector (natural orbital). The occupation number for a natural orbital is a measure of how much this orbital contributes to the total one-electron density. A natural choice is to disregard the natural orbitals with small occupation numbers and use those with large occupation numbers to form contracted basis functions as

$$\varphi_k = \sum_i c_{ki}\chi_i \tag{8.5}$$

where χ_i is the i'th primitive basis function.

As a generalization to this approach we can average over density matrices from several wave functions, resulting in basis sets of the density matrix averaged ANO type, see for example [51, 52, 53, 54]. We can view the averaging of density matrices as a sequence of rank-1 updates in the same way as in equation 8.4. We have more update vectors than the rank of the matrix, but this does not really change anything. The important observation is that all η's are positive and no information is lost in the averaging.

The general guideline for which wave functions to include is based on what you want to be able to describe. All wave functions you want an accurate description of should be included in the averaging.

As an example, let us consider the oxygen atom. We want to be able to describe the atom by itself accurately, thus a wave function for the atom is needed, usually at the CI level. In molecular systems, oxygen usually has a negative charge, thus including O^- is almost mandatory. A basis set derived from these two wave function is well balanced for the majority of systems containing oxygen. A logical conclusion would be that you need to include a few *molecular* wave functions of systems containing oxygen, but in practice this is not necessary. This is due to the fact that the degrees of freedom describing the orbital shape distortion when forming bonds are virtually identical to the lowest correlating orbitals. On the other hand, a few molecular species have oxygen with positive charge, thus it may be appropriate to include O^+ in the basis set.

A wide range of specialized basis sets can also be generated, for example a molecular basis set describing Rydberg orbitals, see the example in the “Tutorials and Examples” part, section 10.5.1. There is a possibility to create rydberg orbitals automatically by using the keyword RYDBERG. Here all unoccupied orbitals with negative orbital energies will be used with the associated occupation numbers

$$\eta_k = e^{6.9\times(\epsilon_k/\epsilon_0-1)} \tag{8.6}$$

where ϵ_k is the orbital energy of orbital k and ϵ_0 is the lowest orbital energy of all virtual orbitals. In order to use this option you need to use the `SCF` or `RASSCF` program to compute the orbitals for a cationic system.

You need one or more wave functions, represented by formatted orbital files, to generate the average density matrix. These natural orbital files can be produced by any of the wave function generators `SCF`, `RASSCF`, `MRCI` or `CPF`. You could also use `MBPT2` or `CASPT2`. This approach has been used in the generation of the ANO-RCC basis sets. Your specific requirements dictate the choice of wave function generator, but `MRCI` would be most commonly used.

You are not restricted to atomic calculations but can mix molecular and atomic calculations freely. The restrictions are that the name of the center, for which you are constructing a basis set, must be the same in all wave functions. The center may not be "degenerate", i.e. it may not generate other centers through symmetry operations. See the description of SEWARD on page 363 for a more extensive discussion. For example for O_2 you cannot use D_{2h} symmetry since this would involve one center that is mirrored into the other. Another restriction is, of course, that you must use the same primitive set in all calculations.

8.17.1 Dependencies

GENANO needs one or more wave functions in the form of natural orbitals. Thus you need to run one or more of SCF, RASSCF, MRCI or CPF. You could also use, for example, MBPT2 or CASPT2 but this is in general not recommended. GENANO also needs the one electron file *ONEINT* and the *RUNFILE* generated by SEWARD.

8.17.2 Files

Below is a list of the files that GENANO reads/writes. Files *ONEnnn*, *RUNnnn* and *NATnnn* must be supplied to the program. Files *ANO* and *FIG* are generated. File *PROJ* is an optional input file.

Input files

File	*Contents*
RUNnnn	This file contains miscellaneous information for the nnn'th wave function, generated by the program SEWARD. One file per wave function must be supplied, *RUN001*, *RUN002*, ...
ONEnnn	This file contains the one-electron integrals corresponding to the nnn'th wave function, generated by the program SEWARD. One file per wave function must be supplied, *ONE001*, *ONE002*, ...
NATnnn	This file contains the natural orbitals corresponding to the nnn'th wave function, generated by the appropriate wave function generating program. One file per wave function must be supplied, *NAT001*, *NAT002*, ...
PROJ	This file contains orbitals used for projection of the densities. Needs to be available if the keyword PROJECT is specified. It is compatible in format with the file *ANO*, and can thus be the the file *ANO* from a previous run of GENANO.

Output files

File	*Contents*
FIG	This file contains a PostScript figure file of eigenvalues.
ANO	This file contains the contraction coefficient matrix organized such that each column correspond to one contracted basis function.

8.17.3 Input

The input file must contain the line

```
&GENANO
```

right before the actual input starts. Below is a list of the available keywords. Please note that you can not abbreviate any keyword.

Keyword	*Meaning*
TITLE	This keyword starts the reading of title lines, with no limit on the number of title lines. Reading the input as title lines is stopped as soon an the input parser detects one of the other keywords. This keyword is *optional.*
SETS	This keyword indicates that the next line of input contains the number of sets to be used in the averaging procedure. This keyword must precede WEIGHTS if both are supplied. This keyword is *optional*, with one set as the default.
CENTER	This keyword is followed, on the next line, by the atom label for which the basis set is to be generated. The label must match the label you supplied to SEWARD. In previous versions of GENANO this label had to be in uppercase, but this restriction is now lifted and the case does not matter. This keyword is *compulsory.*
ROWWISE	This keyword makes GENANO produce the contraction coefficients rowwise instead of column-wise as is the default. This keyword is *optional.*
WEIGHTS	This keyword must be subsequent to keyword SETS if both are supplied. This keyword is *optional*, with equal weight on each of the sets as default.
PROJECT	This keyword states that you want to project out certain degrees of freedom from the density matrix. This can be useful for generating, for example, node less valence orbitals to be used with ECP's. If this keyword is specified, you must supply the file *PROJ* obtained as file *ANO* from a previous GENANO calculation, for instance. This keyword is *optional.*

LIFTDEGENERACY This keyword will modify the occupation numbers read from the orbitals files. The purpose is to lift th degeneracy of core orbitals to avoid rotations. The occupation numbers are changed according to $\eta' = \eta \times (1 + 10^{-3}/n)$ where n is the sequence number of the orbital in its irreducible representation. This keyword is *optional.*

RYDBERG This keyword enables automatic generation of rydberg orbitals. With this keyword all occupied orbitals will get occupation number zero while the virtual orbitals will get a small occupation number decreasing with orbital number. Useful with a calculation on an cation where the virtual orbitals are near perfect rydberg orbitals. Note that you must use orbitals from the SCF or RASSCF program. This keyword is *optional.*

NOTHRESHOLD This keyword is used to specify the threshold for keeping NO's (natural orbitals). Orbitals with occupation numbers less than the threshold are discarded. The threshold is read from the line following the keyword. Default value is 1.0d-8.

Below is a simple input example, where we construct an ANO basis set for the carbon atom. Two wave functions are used, the SCF wave function and the SDCI wave function for the ground state of the atom.

```
 &SEWARD
Title
 Carbon atom
Symmetry
x y z
Expert
Basis set
C..... / inline
  6.0 2
   10   10
5240.6353 782.20479 178.35083 50.815942 16.823562 6.1757760 2.4180490
.51190000 .15659000 .05480600
1. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 1. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 1. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 1. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 1. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 1. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 1. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 1. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 1. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 1.
    6    6
18.841800 4.1592400 1.2067100 .38554000 .12194000 .04267900
1. 0. 0. 0. 0. 0.
0. 1. 0. 0. 0. 0.
0. 0. 1. 0. 0. 0.
0. 0. 0. 1. 0. 0.
0. 0. 0. 0. 1. 0.
0. 0. 0. 0. 0. 1.
    3    3
1.2838000 .34400000 .09220000
1. 0. 0.
0. 1. 0.
0. 0. 1.
C  0.000000  0.000000  0.000000
```

```
End of basis
 &SCF
Occupied =  2 0 0 0 0 0 0 0
 &RASSCF
Symmetry =  4
Spin     =  3
nActEl   =  2 0 0
Frozen   =  0 0 0 0 0 0 0 0
Inactive =  2 0 0 0 0 0 0 0
Ras2     =  0 1 1 0 0 0 0 0
LevShft  =  0.00
LumOrb
Thrs     =  0.1d-8 0.1d-4 0.1d-4
 &MOTRA
LumOrb
Frozen   =  1 0 0 0 0 0 0 0
 &GUGA
Electrons =  4
Spin      =  3
Inactive  =  1 0 0 0 0 0 0 0
Active    =  0 1 1 0 0 0 0 0
CiAll     =  4
 &MRCI
SDCI
>>COPY $Project.RunFile RUN001
>>COPY $Project.RunFile RUN002
>>COPY $Project.OneInt  ONE001
>>COPY $Project.OneInt  ONE002
>>COPY $Project.RasOrb  NAT001
>>COPY $Project.CiOrb   NAT002
 &GENANO
Title
 Carbon atom
Project
sets
 2
Center
C
Weights
 0.5 0.5
>>RM ONE001
>>RM ONE002
>>RM NAT001
>>RM NAT002
```

8.18 GEO

8.18.1 Description

The GEO module handles geometry optimization in constrained internal coordinates[55]. The module is called automatically when the emil command > DO GEO is used and should never be called explicitly by the user. All input relevant for these type of geometry optimization should be supplied in GATEWAY and all the relevant keywords are described in detail in the manual section for GATEWAY.

The purpose of the GEO module is to perform geometry optimization of several molecular fragments by keeping the fragments rigid and only optimize their relative position. The xyz-file of each fragment is supplied to the program by a separate COORD input in GATEWAY. Internal coordinates for the whole complex is then constructed and stored into the file *$project.zmt* of the directory *$GeoDir* which is a communication directory set up and used by the GEO module. The internal coordinates are chosen so that only a maximum of six coordinates link each fragment. All coordinates within fragments are frozen and the optimization is only carried out for these linking coordinates. The geometry optimization is performed using a numerical gradient (and hessian if needed) and could in principle be used together with any energy relaxation method that is implemented into molcas. (Currently it only works with SCF, MP2, RASSCF, CASPT2, CHCC, CHT3, and CCSDT but it is very simply to extend it to other modules.)

The module is intended to use in cases where one knows the geometry of each fragment and do not expect it to change much during the optimization.

Note that it is often advantageous to run a GEO job in parallel. The number of processors (cores) required can be easily evaluated by the following formula[55]:

$$N_{procs} = 2 + \frac{N_{var} \cdot (N_{var} + 3)}{2}, \tag{8.7}$$

where N_{var} is the total number of active coordinates. In the simplest case of the N_{frag} rigid polyatomic fragments, when only relative positions of fragments are going to be optimized, the N_{var} parameter reads:

$$N_{var} = 6 \cdot N_{frag} \tag{8.8}$$

In particular, for constrained two-fragment geometry optimization one needs 29 processes to run GEO job in a fully parallel manner.

Creating the z-matrix

The z-matrix is created by choosing the first atom in the xyz-file and put that in top of the z-matrix file (the example is a methane dimer:

```
H
```

The next atom is chosen as the closest atom not already in the z-matrix and in the same fragment as the first. The distance to the closest neighbor within the z-matrix is calculated and written into the zmatrix:

```
C
H   1.104408  0                           1
```

The number zero means "do not optimize" and is appended to all coordinates within fragments and the 1 at the right hand side keeps track of which atom that were the closest neighbour. The next molecule is again chosen as the one not in the z-matrix but in the fragment. The distance between the third atom and its closest neighbour and the angle between the third atom, the closest atom and the second closest atom with the second closest in the middle is added.

```
C
H     1.104408 0                              1
H     1.104438 0   109.278819 0               1   2
```

For the fourth atom a dihedral angle is added and after that all remaining atoms only have coordinates related to three neighbors. When all the molecule in the fragment has entered the z-matrix the closest atom in another fragment is taken as the next atom. Neighbors to atoms in that fragment is chosen from the same fragment if possible and then from the previous fragment. When two atoms from the second fragment has been added the z-matrix looks like this:

```
C
H    1.104408 0                                   1
H    1.104438 0   109.278819 0                    1   2
H    1.104456 0   109.279688 0  -119.533195 0    1   2   3
H    1.104831 0   109.663377 0   120.232913 0    1   2   3
H    3.550214 1    75.534230 1  -133.506752 1    3   4   1
C    1.104404 0    78.859588 1  -146.087839 1    7   3   3
```

Note that coordinates linking fragments have a 1 after and will be optimized and that the last carbon has one neighbor within its own fragment and two within the previous fragment.

There are two special exceptions to the rules described above. Firstly, if it is possible neighboring atoms are chosen from non-hydrogen atoms. (When linking an atom to neighbors, not when choosing the next line in the z-matrix.) Secondly, dihedral angles are rejected if the angles between the three first or the three last atoms are smaller than 3 degrees since very small angles makes the dihedral ill-defined.

8.18.2 Dependencies

The GEO program requires that GATEWAY have been run with either the keyword GEO or HYPER to setup internal communication files.

8.18.3 Files

Input files

Apart from the standard file GEO will use the following input files.

File	*Contents*
RUNFILE	File for communication of auxiliary information.

GEO will also use internal communication files in the directory *$GeoDir* described in more detail in the next section.

Geo communication files

When the emil command > DO GEO or the keywords zonl or zcon are used a directory *$GeoDir* is created (by default in the input-directory: *$CurrDir/$Project.Geo*). This directory is used to store files related to geometry optimization and z-matrix generation.

File	*Contents*
$project.zmt	The file with the z-matrix as described above.

general.info A file used for storing general info about the geometry optimization, it is human readable with labels. The file is automatically setup by the program.

example:

```
Number of iterations:            3
Number of atoms:                 8
Number of internal coordinates:  6
Internal Coordinates:
    2.159252   99.560213  123.714490   99.612396 -179.885031 -123.791319
Displacement parameters:
    0.150000    2.500000    2.500000    2.500000    2.500000    2.500000
Coordinate types:
badadd
```

Most of the lines are self-explanatory, coordinate-type is one character for each internal coordinate to optimize with b=bond, a=angle and d=dihedral, displacement parameters is the coordinates defined by hyper.

disp????.info A file that contains all displaced coordinates. A new instance of the file is created automatically for each geometry step.

```
example:
disp0001.xyz 2.16485  99.52171 123.71483  99.57425 -179.88493 -123.79124
disp0002.xyz 2.17548 101.22524 124.91941 100.50733 -179.12307 -123.14735
disp0003.xyz 2.16485 101.22952 124.92244 100.50968 -179.12116 -123.14573
disp0004.xyz 2.17145  98.46360 125.21123 100.73337 -178.93851 -122.99136
```

$project.disp????.energy A one line file that simply state the current energy for the displacement. If several energies are calculated this will first contain a scf energy and is then updated with the mp2 value when that calculation is finished. The file is written by a small addition to the add-info files and currently collects energies from SCF, MP2, RASSCF, CASPT2, CHCC, CHT3, and CCSDT. If it should be used with new energy relaxation methods this must be added manually.

$project.final????.xyz An xyz-file with coordinates for all fragments after XXXX geometry optimizations. This is just for the benefit of the user and should probably be replaced with the same output as created by slapaf, opt.xyz-file for optimized geometry and molden-file for history. Some internal history would still be needed for building more advanced geometry optimization algorithms and convergence criteria though.

$project.geo.molden A molden file with information about the geometry optimization. The file could be browsed in molden or LUSCUS. The last energy-value is set to zero since the file cannot be created after the last energy calculation and need to be inserted by the user.

8.18.4 input

The general input structure of a geo-calculation looks like this:

```
&Gateway
```

```
[keywords to modify fragment position (frgm,origin)]
coord=fragment1.xyz
...
coord=fragmentN.xyz
[regular GATEWAY keywords + keywords to modify GEO]
>> DO GEO
 &Seward
[energy relaxation methods with any of their keywords]
>> END DO
```

Both the keywords used to translate and rotate xyz-files (FRGM and ORIGIN) and the keyword to modify the behaviour of the optimization (HYPER, and OLDZ) is described in more details in the GATEWAY section of the manual.

Here is an example input to calculate the relative orientation of two methane molecules:

```
>> export GeoDir=$CurrDir/$Project.GEO
&Gateway
Coord=$MOLCAS/Coord/Methane1.xyz
Coord=$MOLCAS/Coord/Methane2.xyz
Group=c1
basis=aug-cc-pVDZ
hyper
0.15 2.5 2.5
>> Do Geo
&Seward
CHOLESKY HIGH
&SCF
&MP2
>> End Do
```

In order to run the job above in a task-farm parallel mode, one just needs to slightly modify the original input:

```
*activating an TaskFarm's interface in GEO/HYPER
>> export MOLCAS_TASK_INPUT=NEW
*specifying number of cores/cpus available for parallel execution
>> export MOLCAS_NPROCS=2
>> export GeoDir=$CurrDir/$Project.GEO
&Gateway
Coord=$MOLCAS/Coord/Methane1.xyz
Coord=$MOLCAS/Coord/Methane2.xyz
Group=c1
basis=aug-cc-pVDZ
hyper
0.15 2.5 2.5
>> Do Geo
&Seward
CHOLESKY HIGH
&SCF
&MP2
>> End Do
*setting the TaskMode's execution model to MPP
*each energy/displacement is computed in paralel
*by using 2 cores
>>> EXPORT MOLCAS_TASKMODE=1
*actual execution of TaskFarm via EMIL
>>> UNIX SERIAL $MOLCAS/sbin/taskfarm
```

8.19 GRID_IT

8.19.1 Description

GRID_IT is an interface program for calculations of molecular orbitals and density in a set of Cartesian grid points. The calculated grid can be visualized by LUSCUS program.

8.19.2 Dependencies

The GRID_IT program requires the communication file *RUNFILE*, produced by GATEWAY and an orbital file *INPORB*: *SCFORB*, *RASORB*, *PT2ORB*, generated by program SCF(calculated with RHF or UHF hamiltonian), RASSCF, or CASPT2, respectively.

8.19.3 Files

Below is a list of the files that are used/created by the program GRID_IT.

Input files

File	*Contents*
RUNFILE	File for communication of auxiliary information generated by the program GATEWAY.
INPORB	*SCFORB* or another orbitals file (*RASORB*, *CIORB*, *CPFORB*, *SIORB*, *PT2ORB*) containing calculated orbitals. If used after SCF run, the information about one-electron energies is also retrieved.

Output files

File	*Contents*
LUS	Output file in LUSCUS format, with default extension *.lus* — the only file needed for drawing program. In the case of UHF calculation, GRID_IT produces two files: *a.lus* and *b.lus* with grids for alpha and beta electrons. LUSCUS code could be used to combine grid files for visualization of total or spin density.
GRID	Obsolete format of the grid file, can be converted to Cube files using the GRID2CUBE tool. In order to generate a file with data written in grid format, keyword NOLUSCUS should be used.

8.19.4 Input

Normally, it is no reason to change any default setting of calculated grid - the choice of appropriate grid size, net frequency, as well as choice of MO can be done automatically.

If user did not specified the selection of orbitals, GRID_IT makes a decision based on information in the *InpOrb* file. For example, if *InpOrb* contains data after SCF calculation, a set of orbitals around HOMO-LUMO gap will be used. If *InpOrb* contains active orbitals, they will be used as a default set.

Below follows a description of the input to GRID_IT. The keywords are always significant to four characters, but in order to make the input more transparent, it is recommended to use the full keywords. The GRID_IT program section of the MOLCAS input starts with the standard reference to the code:

```
&GRID_IT
```

Argument(s) to a keyword are always supplied on the next line of the input file, except when explicitly stated otherwise.

Optional general keywords

Keyword	*Meaning*
TITLE	One line following this one is regarded as title.
NAME	One line following this one is used for generation of grid filename in the form: “Project.Name.lus”.
FILE	On the following line user can specify the filename, which will be used instead of INPORB (default). For example: FILEORB=$CURRDIR/$PROJECT.SCFORB.
SPARSE	Set up sparse Cartesian net with 1 grid point per a.u. Note that quality of the grid can be poor. Default (without SPARSE or DENSE) is 3 points per a.u.
DENSE	Set up net with 10 grid points per a.u. Note that using this option without choice of orbitals to draw you can produce very large output file.
GAP	Keyword, followed by real equals to distance between the atomic nuclei in the molecule and the border of grid. Default value is 4.0 a.u.
ORBITAL	Direct specification of orbitals to show. Next line set up number of calculated grids. And at next line(s) pairs of integers - symmetry and orbital within this symmetry is given.
SELECT	Direct specification of orbitals to show. Follows by one line in the format: symmetry:first_orbital-last_orbital (Ex: 1:2-7 2:5-8)

MULLiken Compute Mulliken charges separately for each occupied MO specified in the GRID_IT input. ”LONG print” is an optional argument for more detailed printout.

NoSort Do not sort orbitals by occupation numbers and orbital energies

ORANge Followed by 2 numbers, to limit the interval of orbitals by occupation numbers.

ERANge Followed by 2 numbers, to limit the interval of orbitals by one-electron energies

ALL Calculate grids for all molecular orbitals. Using this keyword you can produce a huge output file!

TOTAl Request to calculate a grid for the (correct) total density computed from contributions of all orbitals, instead of (default) just from the orbitals chosen by user.

VB This keyword enables plotting of the orbitals from the latest CASVB orbitals. Note that the appropriate *RASORB* orbitals must be available in the *INPORB* file.

ATOM Calculate density at the position of atoms.

ASCII Obsolete keyword for ASCII format of output file. This option can only be used in combination with NOLUSCUS. This keyword is useful if a calculation of the grid file and visualization should be done on computers with different architectures.

NPOInts Keyword, followed by 3 integers equal to number of grid points in x, y, z directions. Using for non-automatic choice of grid network.

GRID Keyword to set manually coordinates of a grid. Followed by number of Cartesian coordinates, and on next lines - x y z coordinates of a grid (in a.u.)

GORI Keyword to set manually the parallelepiped spanning a grid. Followed by four lines of three columns each. The first line defines the x y z location of the origin, the next three lines are three linearly independent vectors that span the parallelepiped of the grid. This keyword requires NPOINTS to build up the lattice of gridpoints.

NOLUSCUS Produce data file in obsolete format (which can be read by old (before 2015) versions of GV).

XFIELD Use GRID_IT it in a special mode (to produce the grid for non-cartesian points).

Input example

An example for high quality picture, containing selected orbitals (from symmetry 1 and 4):

```
 &GRID_IT
Dense
Select
1:10-20,4:3-7
```

An example for screen quality picture, containing all orbitals:

```
 &GRID_IT
SPARSE
ALL
```

An example for selection orbitals with partial occupation

```
 &GRID_IT
ORange = 0.01 1.99
```

Program GRID_IT also can do simple arithmetic operations with produces luscus files. In this case the input is different from the described above.

User have to specify two compusory keywords (LUS1 and LUS2) with teh file name as a parameter, and an operation PLUS or MINUS (the latter is default). It is also possible to specify a name of the output file by using keyword NAME.

An example. The first two runs generate two luscus files: $Project.STO.lus and $Project.ANO.lus , which later are transformed into the third file: $Project.Difference.lus

```
&GATAWAY
coord=$MOLCAS/Coord/Benzene.xyz
Basis=STO-3G
&SEWARD
&SCF
&GRID_IT
ALL
NAME=STO
&GATAWAY
coord=$MOLCAS/Coord/Benzene.xyz
Basis=ANO-S-VDZP
&SEWARD
&SCF
&GRID_IT
ALL
NAME=ANO
&GRID_IT
LUS1=$CurrDir/STO.lus
LUS2=$CurrDir/ANO.lus
MINUS
NAME=Difference
```

8.20 GUESSORB

8.20.1 Description

The GUESSORB program generates a start guess for orbitals. The file *GSSORB* is created containing these orbitals. They are also put to the runfile and their presence is automatically detected by the programs SCF and RASSCF if needed.

8.20.2 Dependencies

The GUESSORB program requires that the one electron file *ONEINT* as well as the communication file *RUNFILE* exist. These are generated by the program SEWARD

8.20.3 Files

Below is a list of the files that are used/created by the program GUESSORB.

Input files

GUESSORB will use the following input files: *ONEINT*, *RUNFILE* (for more information see 7.1.2).

Output files

File	*Contents*
GSSORB	GUESSORB orbital output file. Contains a start guess for orbitals.
RUNFILE	Communication file for subsequent programs.
MD_GSS	Molden input file for molecular orbital analysis.

8.20.4 Input

Below follows a description of the input to GUESSORB.

Keywords

Keyword	*Meaning*
PRMO	This keyword will make GUESSORB print the orbitals that are generated. On the next line an integer is to be specified that control how much output you get, see below. On the same line you can optionally specify a floating point number that control how many orbitals are printed. Only orbitals with orbital energy less than this number will be printed, default is 5.0au. 1 — Only occupation numbers and orbital energies are printed. 2 — As for 1 but with an additional sorted list of orbital energies. 3 — As for 2 but with orbitals printed in compact format.

4 — As for 3 but orbitals are printed in full format.

PRPOPULATION This keyword will print a Mulliken population analysis based on the assumptions guessorb make with regards to populating orbitals.

STHR This keyword controls how many orbitals will be deleted. On the next line you specify a threshold that have the default 1×10^{-5}. The overlap matrix is diagonalized and only eigenvectors with eigenvalues larger that this threshold will be used, the other will be deleted. This removes near liner dependence.

TTHR This keyword controls how many orbitals will be deleted. On the next line you specify a threshold that have the default 1×10^{6}. The kinetic energy matrix is diagonalized in the space of virtual orbitals and only orbitals with energies below this threshold is used, the other will be deleted. This removes degrees of freedom describing core correlation.

GAPTHR This keyword controls how guessorb attempt to populate the orbitals. On the next line a threshold is specified that have the default 0.01. Using this threshold guessorb will make a closed shell configuration if it can find a HOMO/LUMO gap that is larger than the specified threshold. If that can not be done, guessorb will place a number of orbitals in an active space in such a way that the gap between the three spaces (inactive, active and secondary) will be larger than the threshold.

END OF INPUT

Input examples

In this example GUESSORB is used to produce a Mulliken population based on assumptions that are made for population of orbitals.

```
 &GUESSORB
PrPopulation
```

In this example GUESSORB is used to construct an active space for RASSCF if there are near degeneracies around the fermi level. The orbital file that is produced can be fed directly into RASSCF without specifying the active space.

```
 &GUESSORB
GapThr =  0.5
 &RASSCF
LumOrb
```

8.21 GUGACI

The GUGACI program generates Multi Reference SDCI wavefunctions. The program is based on the Direct CI method[42], and with distict row table generated by GUGADRT (See program description for GUGADRT). The hole-particle symmetry based on GUGA is used in GUGACI[56]–[57]. If requested, GUGACI computes matrix elements of those one-electron properties for which it can find integrals in the *ONEINT* file. It also generates natural orbitals that can be fed into the property program to evaluate certain one electron properties. The natural orbitals are also useful for Iterated Natural Orbital (INO) calculations.

The GUGACI code is written by Yubin Wang, Zhenyi Wen, Zhenting Gan, Bingbing Suo and Yibo Lei (Institute of Modern Physics, Northwest University, China).

The program can calculate several eigenvectors simultaneously.

Orbital subspaces

The orbital space is divided into the following subspaces: Frozen, Inactive, Active, Secondary, and Deleted orbitals. Within each symmetry type, they follow this order.

- **Frozen:** Frozen orbitals are always doubly occupied, i.e., they are not correlated. Orbitals should be frozen already in the integral transformation step, program MOTRA, and need not be specified in the input to the MRCI program. If it's specified, it will be ignored.

- **Inactive:** Inactive orbitals are doubly occupied in all reference configurations, but excitations out of this orbital space are allowed in the final CI wavefunction, i.e., they are correlated but have two electrons in all reference configurations.

- **Active:** Active orbitals are those which may have different occupation in different reference configurations.

- **Secondary:** This subspace is empty in all reference configurations, but may be populated with up to two electrons in the excited configurations. This subspace is not explicitly specified, but consists of the orbitals which are left over when other spaces are accounted for.

- **Deleted:** This orbital subspace does not participate in the CI wavefunction at all. Typically the 3s,4p,. . . components of 3d,4f. . ., or orbitals that essentially describe core correlation, are deleted. Similar to freezing, deleting should be done in MOTRA, which is more efficient, and do not need to be specified in the MRCI program.

Since ordinarily the frozen and deleted orbitals were handled by MOTRA and the subdivision into inactive and active orbitals were defined in GUGADRT, program GUGACI will neglect them.

8.21.1 Dependencies

The program needs the distict row table generated by the program GUGADRT and transformed one- and two-electron integrals generated by the program MOTRA.

8.21.2 Files

Input files

File	*Contents*
GUGADRT	Distict row table from `GUGADRT`.
*TRAINT**	Transformed two-electron integrals from `MOTRA`.
TRAONE	Transformed one-electron integrals from `MOTRA`.
ONEINT	One-electron property integrals from `SEWARD`.
MRCIVECT	Used for input only in restart case.

Output files

File	*Contents*
CIORBnn	One or more sets of natural orbitals, one for each CI root, where nn stands for 01,02, etc.
CIVECT	CI vector, for later restart.

Note that these file names are the FORTRAN file names used by the program, so they have to be mapped to the actual file names. This is usually done automatically in the MOLCAS system. However, in the case of several different numbered files

Local files

File	*Contents*
FTxxF01	MRCI produces a few scratch files that are not needed by any other program in MOLCAS. Presently, these are xx=14, 15, 16, 21, 25, 26, 27, and 30. The files are opened, used, closed and removed automatically. See source code for further information.

8.21.3 Input

This section describes the input to the `GUGACI` program in the MOLCAS program system, with the program name:

```
&MRCI
```

Keywords

The first four characters are decoded and the rest are ignored.

Keyword	*Meaning*
TITLe	The lines following this keyword are treated as title lines, until another keyword is encountered. A maximum of ten lines is allowed.
NRROots	Specifies the number of CI roots (states) to be simultaneously optimized. The default is 1. The value is read from the next line.
RESTart	Restart the calculation from a previous calculation. No additional input is required. The *MRCIVECT* file is required for restarted calculations.
THRPrint	Threshold for printout of the wavefunction. All configurations with a coefficient greater than this threshold are printed. The default is 0.05. The value is read from the line following the keyword.
CONvergence	Energy convergence threshold. The result is converged when the energy of all roots has been lowered less than this threshold in the last iteration. The default is 1.0d-8. The value is read from the line following the keyword.
PRINt	Print level of the program. Default is 5. The value is read from the line following the keyword.
MAXIterations	Maximum number of iterations. Default 20. The value is read from the line following the keyword. The maximum possible value is 200.
PRORbitals	Threshold for printing natural orbitals. Only orbitals with occupation number larger than this threshold appears in the printed output. The value is read from the line following the keyword. Default is 1.0d-5.
CPROperty	Request to calculate properties. Property integrals should be saved in file *ONEINT*.

Input example

```
 &GUGACI
Title
 Water molecule. 1S frozen in transformation.
Nrroots
 1
```

8.22 GUGADRT

The GUGADRT program generates distict row table (DRT) used in the GUGACI in Direct CI calculations[42] . Only DRT in active space are generated because the hole-particle symmetry is used in GUGACI[56]–[57]. These DRT are used to evaluated the coupling coefficients by the Graphical Unitary Group Approach[43]–[44], for wavefunctions with at most two electrons excited from a set of reference configurations. The reference configurations can be specified as a list, where the occupation numbers are given for each active orbital (see below) in each reference configuration, or as a Full CI within the space defined by the active orbitals. In the GUGADRT and GUGACI the orbitals are classified as follows: Frozen, Inactive, Active, Secondary, and Deleted orbitals. Within each symmetry type, they follow this order. For the GUGADRT program, only the active orbitals are relevant.

- **Inactive:** Inactive orbitals are doubly occupied in all reference configurations, but excitations out of this orbital space are allowed in the final CI wavefunction, i.e., they are correlated but have two electrons in all *reference* configurations. Since only single and double excitations are allowed, there can be no more than two holes in the active orbitals.
- **Active:** Active orbitals are those which may have different occupation in different reference configurations.

8.22.1 Dependencies

8.22.2 Files

Input files

File	*Contents*
TRAONE	Transformed one-electron integrals from MOTRA. Orbital information such as frozen, deleted orbitals will be read from this file.

Output files

File	*Contents*
GUGADRT	This file contains the DRT that is needed in subsequent CI calculations.

8.22.3 Input

This section describes the input to the GUGADRT program in the MOLCAS program system, with the program name:

```
&GUGADRT
```

The first four characters of the keywords are decoded and the rest are ignored.

Keywords.

Formally, there are no compulsory keyword. Obviously, some input must be given for a meaningful calculation.

Keyword	*Meaning*
TITLe	The lines following this keyword are treated as title lines, until another keyword is encountered.
SPIN	The spin degeneracy number, i.e. 2S+1. The value is read from the line following the keyword, in free format. The default value is 1, meaning a singlet wave function.
ELECtrons	The number of electrons to be correlated in the CI calculation. The value is read from the line following the keyword, in free format. Note that this number should include the nr of electrons in inactive orbitals. An alternative input specification is NACTEL. Default: Twice nr of inactive orbitals.
NACTel	The number of electrons in active orbitals in the reference configurations. The value is read from the line following the keyword, in free format. Note that this number includes only the of electrons in active orbitals. An alternative input specification is ELECTRONS. Default: Zero.
INACtive	The number of inactive orbitals, i.e. orbitals that have occupation numbers of 2 in all reference configurations. Specified for each of the symmetries. The values are read from the line following the keyword, in free format.
ACTIve	The number of active orbitals, i.e. orbitals that have varying occupation numbers in the reference configurations. Specified for each of the symmetries. The values are read from the line following the keyword, in free format. At least one of the INACTIVE or ACTIVE keywords must be present for a meaningful calculation. If one of them is left out, the default is 0 in all symmetries.
REFErence	Specify selected reference configurations. The additional input that is required usually spans more than one line. The first line after the keyword contains the number of reference configurations, and

the total number of active orbitals, and these two numbers are read by free format. Thereafter the input has one line per reference configuration, specifying the occupation number for each of the active orbitals, read by 80I1 format. Note that REFERENCE and CIALL are mutually exclusive.

SYMMETRY

Specify the selected symmetry type (the irrep) of the wave function as a number between 1 and 8 (see SEWARD). Default is 1, which always denote the totally symmetric irrep.

CIALL

Use a Full CI within the subspace of the active orbitals as reference configurations. The symmetry of the wavefunction must be specified. The value is read from the line following the keyword, in free format. Note that CIALL and REFERENCE are mutually exclusive. One of these two alternatives must be chosen for a meaningful calculation.

PRINT

Printlevel of the program. Default printlevel (0) produces very little output. Printlevel 5 gives some information that may be of interest. The value is read from the line following the keyword, in free format.

Input example

```
 &GUGADRT
Title     =  CH2 molecule.
Electrons =  8
Spin      =  1
Inactive  =  1    0    0    0
Active    =  2    2    2    0
Symmetry  =  1
Ciall
```

8.23 GUGA

The GUGA program generates coupling coefficients used in the MRCI and the CPF programs in Direct CI calculations[42]. These coupling coefficients are evaluated by the Graphical Unitary Group Approach[43]–[44], for wavefunctions with at most two electrons excited from a set of reference configurations. The program was written by P. E. M. Siegbahn, Institute of Physics, Stockholm University, Sweden. Only the MRCI program can use several reference configurations. The reference configurations can be specified as a list, where the occupation numbers are given for each active orbital (see below) in each reference configuration, or as a Full CI within the space defined by the active orbitals. In the GUGA, MRCI and CPF programs, the orbitals are classified as follows: Frozen, Inactive, Active, Secondary, and Deleted orbitals. Within each symmetry type, they follow this order. For the GUGA program, only the inactive and active orbitals are relevant.

- **Inactive:** Inactive orbitals are doubly occupied in all reference configurations, but excitations out of this orbital space are allowed in the final CI wavefunction, i.e., they are correlated but have two electrons in all *reference* configurations. Since only single and double excitations are allowed, there can be no more than two holes in the active orbitals. Using keyword NoCorr (See input description) a subset of the inactive orbitals can be selected, and at most a single hole is then allowed in the selected set. This allows the core-polarization part of core-valence correlation, while preventing large but usually inaccurate double-excitation core correlation.

- **Active:** Active orbitals are those which may have different occupation in different reference configurations. Using keyword OneOcc (See input description) a restriction may be imposed on some selection of active orbitals, so that the selected orbitals are always singly occupied. This may be useful for transition metal compounds or for deep inner holes.

8.23.1 Dependencies

The GUGA program does not depend on any other program for its execution.

8.23.2 Files

Input files

The GUGA program does not need any input files apart from the file of input keywords.

Output files

File	*Contents*
CIGUGA	This file contains the coupling coefficients that are needed in subsequent CI calculations. For information about how these coefficients are structured you are referred to the source code[44]. The theoretical background for the coefficient can be found in Refs [43]–[44] and references therein.

8.23.3 Input

This section describes the input to the GUGA program in the MOLCAS program system, with the program name:

```
&GUGA
```

Keywords.

Formally, there are no compulsory keyword. Obviously, some input must be given for a meaningful calculation.

Keyword	*Meaning*
TITLe	The line following this keyword is treated as title line
SPIN	The spin degeneracy number, i.e. 2S+1. The value is read from the line following the keyword, in free format. The default value is 1, meaning a singlet wave function.
ELECtrons	The number of electrons to be correlated in the CI of CPF calculation. The value is read from the line following the keyword, in free format. Note that this number should include the nr of electrons in inactive orbitals. An alternative input specification is NACTEL. Default: Twice nr of inactive orbitals.
NACTel	The number of electrons in active orbitals in the reference configurations. The value is read from the line following the keyword, in free format. Note that this number includes only the of electrons in active orbitals. An alternative input specification is ELECTRONS. Default: Zero.
INACtive	The number of inactive orbitals, i.e. orbitals that have occupation numbers of 2 in all reference configurations. Specified for each of the symmetries. The values are read from the line following the keyword, in free format.
ACTIve	The number of active orbitals, i.e. orbitals that have varying occupation numbers in the reference configurations. Specified for each of the symmetries. The values are read from the line following the keyword, in free format. At least one of the Inactive or Active keywords must be present for a meaningful calculation. If one of them is left out, the default is 0 in all symmetries.
ONEOcc	Specify a number of active orbitals per symmetry that are required to have occupation number one in all configurations. These orbitals are the first active orbitals. The input is read from the line after the keyword, in free format.
NOCOrr	Specify the number of inactive orbitals per symmetry out of which at most one electron (total) is excited. These orbitals are the first inactive orbitals. The input is read from the line after the keyword, in free format.
REFErence	Specify selected reference configurations. The additional input that is required usually spans more than one line. The first line after the keyword contains the number of reference configurations, and the total number of active orbitals, and these two numbers are read by free format. Thereafter the input has one line per reference configuration, specifying the occupation number for each of the active orbitals, read by 80I1 format. Note that Reference and CIall are mutually exclusive.

CIALL Use a Full CI within the subspace of the active orbitals as reference configurations. The symmetry of the wavefunction must be specified. The value is read from the line following the keyword, in free format. Note that CIALL and REFERENCE are mutually exclusive. One of these two alternatives must be chosen for a meaningful calculation.

FIRST Perform a first order calculation, i.e. only single excitations from the reference space. No additional input is required.

NONINTERACTING SPACE By default, those double excitations from inactive to virtual orbitals are excluded, where the inactive and virtual electrons would couple to a resulting triplet. With the NonInteracting Space option, such 'non-interacting' configurations are included as well.

PRINT Printlevel of the program. Default printlevel (0) produces very little output. Printlevel 5 gives some information that may be of interest. The value is read from the line following the keyword, in free format.

Input example

```
 &GUGA
Title
 Water molecule. 2OH correlated.
Electrons =     4
Spin      =     1
Active    =     2    2    0    0
Interacting space
Reference
    3    4
  2020  ;  0220  ;  2002
```

8.24 LOCALISATION

8.24.1 Description

The LOCALISATION program of the MOLCAS program system generates localised occupied orbitals according to one of the following procedures: Pipek-Mezey[58], Boys[59, 60], Edmiston-Ruedenberg[61], or Cholesky[62]. Orthonormal, linearly independent, local orbitals may also be generated from projected atomic orbitals (Cholesky PAOs)[62].

Orbital localisation makes use of the fact that a Hartree-Fock wave function is invariant under unitary transformations of the occupied orbitals,

$$\tilde{C}_{\mu i} = \sum_{j} C_{\mu j} U_{ji}, \tag{8.9}$$

where $\mathbf{U}$ is unitary (i.e. orthogonal for real orbitals). The same is true for the inactive or active orbitals in a CASSCF wave function. Whereas the Pipek-Mezey[58], Boys[59, 60], and Edmiston-Ruedenberg[61] procedures define $\mathbf{U}$ through an iterative maximization of a localisation functional, the Cholesky orbitals are simply defined through the Cholesky decomposition of the one-electron density, i.e.

$$\sum_i \tilde{C}_{\mu i}\tilde{C}_{\nu i} = P_{\mu\nu} = \sum_i C_{\mu i}C_{\mu i}. \tag{8.10}$$

Cholesky orbitals are thus not optimum localised orbitals by any of the Pipek-Mezey, Boys, or Edmiston-Ruedenberg measures, but rather inherit locality from the density matrix, see [62] for details.

Although these localisation schemes are mostly meant for localising occupied orbitals (except for PAOs which are defined for the virtual orbitals), the LOCALISATION program will attempt to localise any set of orbitals that the user specifies. This means that it is possible to mix occupied and virtual orbitals and thereby break the Hartree-Fock invariance. The default settings, however, do not break the invariance.

For Pipek-Mezey, Boys, and Edmiston-Ruedenberg localisations, iterative optimizations are carried out. We use the η-steps of Subotnik *et al.*[63] for Edmiston-Ruedenberg, whereas the traditional Jacobi sweeps (consecutive two-by-two orbital rotations)[58, 63] are employed for the Pipek-Mezey and Boys schemes.

8.24.2 Dependencies

The LOCALISATION program requires the one-electron integral file *ONEINT* and the communications file *RUNFILE*, which contains, among other data, the basis set specifications processed by GATEWAY and SEWARD. In addition, the Edmiston-Ruedenberg procedure requires the presence of Cholesky decomposed two-electron integrals produced by SEWARD.

8.24.3 Files

Below is a list of the files that are used/created by the program LOCALISATION.

Input files

LOCALISATION will use the following input files: *ONEINT*, *RUNFILE*, *INPORB*. For Edmiston-Ruedenberg localisation, it also needs *CHVEC*, *CHRED* and *CHORST* files (for more information see 7.1.2).

Output files

File	*Contents*
LOCORB	Localised orthonormal orbital output file. Note that *LOCORB* contains all orbitals (localised as well as non-localised according to the input specification).

DPAORB	Linearly dependent nonorthonormal projected atomic orbital output file (only produced for PAO runs).
IPAORB	Linearly independent nonorthonormal projected atomic orbital output file (only produced for PAO runs).
RUNFILE	Communication file for subsequent programs.
MD_LOC	Molden input file for molecular orbital analysis.

8.24.4 Input

Below follows a description of the input to LOCALISATION. The LOCALISATION program section of the MOLCAS input is bracketed by a preceding program reference

```
&LOCALISATION
```

Optional general keywords

Keyword	*Meaning*
FILEORB	The next line specifies the filename containing the input orbitals that will be localised. By default a file named *INPORB* will be used.
NORBITALS	The following line specifies the number of orbitals to localise in each irreducible representation. The default is to localise all occupied orbitals as specified in the *INPORB* input file, except for PAO runs where all the virtual orbitals are treated by default.
NFROZEN	The following line specifies the number of orbitals to freeze in each irreducible representation. The default is not to freeze any orbitals, except for the localisations of the virtual space (see keywords PAO and VIRTUAL) where the default is to freeze all occupied orbitals (occupation number different from zero, as reported in the *INPORB* file).
FREEZE	Implicit frozen core option. The default is not to freeze any orbitals, except for the localisations of the virtual space (see keywords PAO and VIRTUAL) where the default is to freeze all occupied orbitals (occupation number different from zero, as reported in the *INPORB* file). The definition of core orbitals is taken from program SEWARD.
OCCUPIED	Requests that the occupied orbitals should be localised. This is the default except for PAO where the default is virtual.
VIRTUAL	Requests that the virtual orbitals should be localised. The default is to localise the occupied orbitals, except for PAO where the default is virtual.

ALL Requests that all orbitals should be localised. The default is to localise the occupied orbitals, except for PAO where the default is virtual.

PIPEk-Mezey Requests Pipek-Mezey localisation. This is the default.

BOYS Requests Boys localisation. The default is Pipek-Mezey.

EDMIston-Ruedenberg Requests Edmiston-Ruedenberg localisation. The default is Pipek-Mezey. Note that this option requires that the Cholesky (or RI/DF) representation of the two-electron integrals has been produced by `SEWARD`.

CHOLesky Requests Cholesky localisation (non-iterative). The default is Pipek-Mezey. This and PAO are the only options that can handle point group symmetry. The decomposition threshold is by default 1.0d-8 but may be changed through the THREshold keyword.

PAO Requests PAO localisation (non-iterative) using Cholesky decomposition to remove linear dependence. The default is Pipek-Mezey. This and Cholesky are the only options that can handle point group symmetry. The decomposition threshold is by default 1.0d-8 but may be changed through the THREshold keyword.

SKIP Leaves the input orbitals unchanged. It is turned off by default.

ITERations The following line specifies the maximum number of iterations to be used by the iterative localisation procedures. The default is 100.

THREshold The following line specifies the convergence threshold used for changes in the localisation functional. The default is 1.0d-6. For Cholesky and PAO methods, it is the decomposition threshold and the default is 1.0d-8.

THRGradient The following line specifies the convergence threshold used for the gradient of the localisation functional. The default is 1.0d-2.

THRRotations The following line specifies the screening threshold used in the Jacobi sweep optimization algorithm. The default is 1.0d-10.

CHOStart Requests that iterative localisation procedures use Cholesky orbitals as initial orbitals. The default is to use the orbitals from *`INPORB`* directly.

ORDEr Requests that the localised orbitals are ordered in the same way as the Cholesky orbitals would be. This is mainly useful when comparing orbitals from different localisation schemes. The ordering is done according to maximum overlap with the Cholesky orbitals. The default is not to order.

DOMAin Requests orbital domains and pair domains are set up and analyzed. The default is not to set up domains.

THRDomain The following line specifies two thresholds to be used in defining orbital domains. The first is the Mulliken population threshold such that atoms are included in the domain until the population (divided by 2) is larger than this number (default: 9.0d-1). The second threshold is used for the Pulay completeness check of the domain (default: 2.0d-2).

THRPairdomain The following line specifies three thresholds to be used for classifying pair domains: R1, R2, and R3. (Defaults: 1.0d-10, 1.0d1, and 1.5d1.) If R is the smallest distance between two atoms in the pair domain (union of the individual orbital domains), then pair domains are classified according to: R≤R1: strong pair, R1<R≤R2: weak pair, R2<R≤R3: distant pair, and R3<R: very distant pair.

LOCNatural orbitals This keyword is used to select atoms for defining the localised natural orbitals (LNOs), thus a set of localised orbitals with well-defined occupation numbers. All other options specified in the `LOCALISATION` program input apply (e.g., input orbitals, localisation method, etc.). On the next line give the number of atoms that identify the region of interest and the threshold used to select the localised orbitals belonging to this region (recommended values > 0.2 and < 1). An additional line gives the names of the (symmetry unique) atoms as defined in the `SEWARD` input. The keyword LOCN is used to define suitable occupation numbers for RASSCF active orbitals that have been localised. It has proven useful in Effective Bond Order (EBO) analysis. Here is a sample input for a complex containing an iron-iron multiple bond.

```
LOCN
2  0.3
Fe1  Fe2
```

In this example, the (localised) orbitals constructed by the `LOCALISATION` program are subdivided in two groups: those having less than 0.3 total Mulliken population on the two iron atoms, and the remaining orbitals, obviously localised on the iron-iron region. The resulting density matrices for the two subsets of orbitals are then diagonalized separately and the corresponding (localised) natural orbitals written to *`LOCORB`* with the proper occupation numbers. Note that the two sets of LNOs are mutually non-orthogonal.

LOCCanonical orbitals This keyword is used to select atoms for defining the localised canonical orbitals (LCOs), thus a set of localised orbitals with well-defined orbital energies (eigenvalues of a local Fock matrix). Please, refer to the analogous keyword LOCN in this manual for more details and input examples.

Limitations

The limitations on the number of basis functions are the same as specified for `SEWARD`.

Input examples

This input is an example of the Boys localisation of the CO molecule. Note that no symmetry should not be used in any calculation of localised orbitals except for Cholesky and PAO orbitals.

```
 &GATEWAY
Coord = $MOLCAS/Coord/CO.xyz
Basis = STO-3G
Group = C1
 &SEWARD ; &SCF
 &LOCALISATION
Boys
```

This input is an example of the Projected Atomic Orbital localisation of the virtual orbitals of the CO molecule. The threshold for the Cholesky decomposition that removes linear dependence is set to 1.0d-14.

```
 &GATEWAY
Coord = $MOLCAS/Coord/CO.xyz
Basis = STO-3G
Group = C1
 &SEWARD ; &SCF
 &LOCALISATION
PAO
Threshold = 1.0d-14
```

This input is an example of the Cholesky localisation (using default 1.0d-8 as threshold for the decomposition) of the valence occupied orbitals of the CO molecule. Orbital domains are set up and analyzed.

```
 &GATEWAY
Coord = $MOLCAS/Coord/CO.xyz
Basis = STO-3G
Group = C1
 &SEWARD ; &SCF
 &LOCALISATION
Cholesky
Freeze
Domain
```

8.25 LOPROP

The program LOPROP is a tool to compute molecular properties based on the one-electron density or transition-density and one-electron integrals like charges, dipole moments and polarizabilities. LOPROP allows to partition such properties into atomic and interatomic contributions. The method requires a subdivision of the atomic orbitals into occupied and virtual basis functions for each atom in the molecular system. It is a requirement for the approach to have any physical significance that the basis functions which are classified as ”occupied” essentially are the atomic orbitals of each species. It is therefore advisable to use an ANO type basis set, or at least a basis set with general contraction.

The localization procedure is organized into a series of orthogonalizations of the original basis set, which will have as a final result a localized orthonormal basis set. **Note that this module does not operate with symmetry.**

A static property, which can be evaluated as an expectation value, like a charge, a component of the dipole moment or an exchange-hole dipole moment, is localized by transforming the integrals of the property and the one-electron density matrix to the new basis and restricting the trace to the subspace of functions of a single center or the combination of two centers.

The molecular polarizability is the first order derivative of the dipole moment with respect to an electric field and the localized molecular polarizability can be expressed in terms of local responses. In practical terms a calculation of localized polarizabilities will require to run seven energy calculations. The first one is in the absence of the field and the other six calculations are in the presence of the field in the ± x,y,z axis respectively.

For a detailed description of the method and its implementation see [64].

8.25.1 Dependencies

The dependencies of the LOPROP module is the union of the dependencies of the SEWARD, and the program used to perform the energy calculation, namely the SCF, MBPT2, RASSCF, or CASPT2 module. The user can also provide LOPROP with a density matrix as input; then LOPROP only depends on SEWARD. The one-electron transition density matrix can also be localized to compute, for example, Förster transition probabilities; then LOPROP depends on RASSI to compute the transition density.

8.25.2 Files

The files of the LOPROP module is the union of the files of the SEWARD module, and the SCF or MBPT2, or RASSCF, or CASPT2 module. An exception is if a density matrix is given as input or when a transition density matrix is localized, see below.

Input files

File	*Contents*
USERDEN	The density matrix given as input when the keyword USERDENSITY is included in the input. The density matrix should be of the following form: triangularly stored ((1,1),(2,1),(2,2),(3,1) etc.) with all off-diagonal elements multiplied by two.
USERDEN1	The density matrix for a field-perturbed calculation (X = +delta)
USERDEN2	The density matrix for a field-perturbed calculation (X = -delta)
USERDEN3	The density matrix for a field-perturbed calculation (Y = +delta)
USERDEN4	The density matrix for a field-perturbed calculation (Y = -delta)
USERDEN5	The density matrix for a field-perturbed calculation (Z = +delta)
USERDEN6	The density matrix for a field-perturbed calculation (Z = -delta)

TOFILE	The one-electron transition density matrix, which optionally can be put to disk by `RASSI`, see its manual pages.

Output files

In addition to the standard output unit `LOPROP` will generate the following file.

File	*Contents*
MpProp	File with the input for `NEMO`.

8.25.3 Input

This section describes the input to the `LOPROP` program. The program name is:

```
&LOPROP
```

Keywords

There are no compulsory keywords.

Keyword	*Meaning*
NOFIeld	The calculation is run in the absence of a field and only static properties like charges and dipole moments are computed. The default is to go beyond the static properties.
DELTa	The magnitude of the electric field in the finite field perturbation calculations to determine the polarizabilities. Default value is 0.001 au.
ALPHa	A parameter in the penalty function used for determining the charge fluctuation contribution to the polarizabilities. See eq. 17 in [64]. The default value of 7.14 is good for small molecules (less than 50 atoms). For larger molecules, a smaller alpha (e.g. 2.0) may be needed for numerical stability.
BOND	Defines the maximum allowed bond length based on the ratio compared to Bragg-Slater radii. All contributions in bonds longer than this radius will be redistributed to the two atoms involved in the bond, so the the total molecular properties are left unaltered. The default value is 1.5.

MPPROP Defines the maximum l value for the multipole moments written to the MpProp file. If the value specified is larger than the highest multipole moment calculated it will be reset to this value, which is also the default value. The 'MULTipoles' keyword in Seward can change the default value.

EXPANSION CENTER Defines which points will be used as the expansion centers for the bonds. The next line must contain either 'MIDPoint' in order just to use the midpoint of the bond or 'OPTImized' in order to let LoProp move the expansion center along the bond. The latter is still highly experimental!

USERDENSITY No density matrix is computed instead it is read as an input from the file *USERDEN*. This enables LOPROP to obtain localized properties for densities that currently can not be computed with MOLCAS. If the keyword NOFIELD is not given, six additional files are required (*USERDEN1*-*USERDEN6*), each containing the density matrix of a perturbed calculation, see above. Observe the form of *USERDEN*, see above.

TDENSITY This keyword signals that the one-electron density matrix which is to be read comes from the TOFILE file generated by RASSI. The keyword is followed by two integers that gives number of initial and final state of the transition. For example, if it is the transition density between the first and second state which should be localized, the integers should be 1 and 2. The keyword implies NOFIELD

XHOLE The exchange hole dipole moment is computed, localized and given as additional output. This quantity can be used to compute local dispersion coefficients according to Becke and Johnson.[65] The numerical integration routine in MOLCASis used.

Input example

Below follows an example input to determine the localized charges, and dipole moments of acetone at the CASSCF level of theory.

```
 &GATEWAY
Title = acetone
Coord = $MOLCAS/Coord/Acetone.xyz
Basis = ANO-L-VDZP
Group = C1
 &SEWARD
 &SCF
Occupation = 15
 &RASSCF
SPIN       = 1
SYMMETRY   = 1
NACTEL     = 4 0 0
INACTIVE   = 13
RAS2       = 4
 &LOPROP
NoField
Expansion Center
```

```
Optimized
Bond       = 1.5
MpProp     = 2
```

In case the density matrix is given as input the input is of the form below (where $CurrDir is a variable defined by the user pointing to the directory where the input density is).

```
 &Gateway
Coord = Water.xyz
Basis = 6-31G*
Group = C1
 &Seward
>>COPY $CurrDir/Density $WorkDir/$Project.UserDen
 &LoProp
UserDensity
```

8.26 MBPT2

8.26.1 Description

The MBPT2 program of the Molcas program system computes the second order correlation energy and the reference weight for a closed-shell Hartree-Fock reference wave function, based on a Møller-Plesset partitioning of the Hamiltonian and canonical orbitals.

If SEWARD performed a Cholesky decomposition of the two-electron integrals prior to running the MBPT2 program, Cholesky vectors will be employed for computing the second order energy correction. This is done by first transforming the Cholesky vectors to MO basis and subsequently computing the $(ai|bj)$ integrals. These integrals are either computed, stored on disk, and then read back in to memory during the energy evaluation (i.e. mimicking a conventional calculation) or they may be computed on-the-fly. The user may choose either algorithm through the Cholesky-specific options described below.

If SEWARD did not perform a Cholesky decomposition, the transformation of the two-electron integrals in AO basis ($\mathcal{O}(N^4)$, where N is the number of basis functions) to the exchange operator matrices $\mathbf{K}^{ij}$ in MO basis ($\mathcal{O}(O^2)$ matrices of size V^2, where O and V denote the number of occupied and virtual orbitals, respectively), is either done conventionally, using the two-electron integral file *ORDINT*, which was generated in a previous step by the SEWARD integral code.

8.26.2 Dependencies

The MBPT2 program requires the communications file *RUNFILE*. It contains specifications processed by SEWARD, the Hartree-Fock canonical orbitals, eigenvalues and energy generated by SCF. For Cholesky-based calculations, all Cholesky related files (see the manual pages for SEWARD) must be available, whereas for conventional (not integral-direct) calculations the two-electron integral file *ORDINT* is required. Hence, before running MBPT2, a SEWARD and a SCF run have to be performed.

8.26.3 Files

Input files

MBPT2 will use the following input files: *ONEINT*, *ORDINT*,*RUNFILE*. For Cholesky runs: *CHVEC*, *CHORST*, *CHRED* and *CHOR2F* (for more information see 7.1.2).

Output files

File	*Contents*
RUNFILE	File for communication of auxiliary information.

8.26.4 Input

Below follows a description of the input to MBPT2. The input for each module is preceded by its name like:

```
&MBPT2
```

No compulsory keywords are required for MBPT2. The reference statement mentioned above is sufficient for a default MBPT2 run.

Optional keywords

Keyword	*Meaning*
TITLE	The line following this line is regarded as a title line
PRINT	Specifies the general print level of the calculation. An integer has to be supplied as argument. The default value, 0, is recommended for production calculations.
FREEZE	Specifies the total number of frozen occupied orbitals. The lowest-energy occupied orbitals are then automatically identified and frozen. The keyword takes as argument one integer. Incompatible with the FROZEN keyword.
FROZEN	Specifies the number of frozen occupied orbitals in each of the irreducible representations (irreps) of the subgroup of D_{2h} in which the system is represented. The counting of the orbitals follows the *increasing* orbital energy within each irrep, with those orbitals being frozen first that correspond to lowest orbital energies. The keyword takes as argument *nIrrep* (# of irreps) integers. Incompatible with the FREEZE keyword. Default is to freeze non-valence orbitals.

DELEted

Specifies the number of deleted orbitals in each of the irreducible representations (irreps) of the subgroup of D_{2h} in which the system is represented. The counting of the orbitals follows the *decreasing* orbital energy within each irrep, with those orbitals being deleted first that correspond to highest orbital energies. The keyword takes as argument *nIrrep* (# of irreps) integers.
NOTE: Those orbitals, which have been deleted already in the `SCF` calculation (cf. SPDElete, OVLDelete of the `SCF` program description) are never seen by the `MBPT2` program and hence are **not** to be deleted again with the present option.

SFROzen

Allows to specify specific orbitals to freeze in each of the irreducible representations (irreps) of the subgroup of D_{2h} in which the system is represented. In the 1st line after the keyword the number of orbitals to freeze for each irrep is specified (*nIrrep* (# of irreps) integers). The next $\leq$ *nIrrep* lines reference the orbitals to freeze for the related irrep, following an enumeration of the individual orbitals of 1, 2, 3,... according to *increasing* orbital energy. Note that the orbital reference numbers obey the original ordering and also include those orbitals which may have been frozen already by the FROZen or FREEze options. If the corresponding irrep does not contain any specific orbitals to freeze (i.e. a zero was supplied for this irrep in the 1st line), no line orbital reference input line is supplied for that irrep.

SDELeted

Allows to specify specific orbitals to delete in each of the irreducible representations (irreps) of the subgroup of D_{2h} in which the system is represented. In the 1st line after the keyword the number of orbitals to delete for each irrep is specified (*nIrrep* (# of irreps) integers). The next $\leq$ *nIrrep* lines reference the orbitals to delete for the related irrep, following an enumeration of the individual orbitals of 1, 2, 3,... according to *increasing* orbital energy. Note that the orbital reference numbers obey the original ordering. If the corresponding irrep does not contain any specific orbitals to freeze (i.e. a zero was supplied for this irrep in the 1st line), no line orbital reference input line is supplied for that irrep.

GHOStdelete

Excludes from PT2 treatment orbitals localized on ghost atoms. A threshold for this selection must be specified.

LUMOrb

Molecular orbital coefficients and energies read from *`INPORB`* file rather than *`RunFile`*.

EREF

Specifies the value of the reference energy. Available only in combination with LumOrb. Default value of the reference energy is set to zero.

TEST

If this keyword is specified the input is checked without performing any calculation.

T1AM

Singles amplitudes/energy introduced according to Thouless formula.

An INPORB file containing MOs different from HF orbitals is required.

LOVMP2 "Freeze-and-Delete" type of MP2, available only in connection with Cholesky or RI. An example of input for the keyword LOVM is the following:

```
LovMP2
2  0.2  (nCenters,thrs)
C1 N    (Center labels)
DoMP2
```

In this case, both occupied and virtual orbitals (localized by the program) are divided in two groups: those (A) mainly located on the two (symmetry independent) centers C1 and C2, and the remaining ones (B), which are obviously "outside" this region. The value of the threshold (between 0 and 1) is used to perform this selection (in the example, 20% of the gross Mulliken population of a given orbital on the specified atoms). By default, the MP2 calculation is performed only for the correlating orbitals associated with the region A ("active site"). The keyword DOMP2 is optional and forces the program to perform also an independent MP2 calculation on the "frozen region" (B). Alternatively, one can specify the keyword VIRALL in order to use all virtual orbitals as correlating space for the occupied orbitals of the active site.

FNOMP2 Performs a Frozen Natural Orbital (FNO) MP2 calculation, available only in combination with Cholesky or RI integral representation. An example of input for the keyword FNOM is the following:

```
FNOMp2
0.4
DoMP2
```

The keyword FNOM has one compulsory argument (real number in]0,1]) specifying the fraction of virtual orbitals (in each irrep) to be retained in the FNO-MP2 calculation. The keyword DOMP2 is optional and used to compute the (estimated) correction for the truncation error.

PRPT Multipole moments (dipoles and quadrupoles) are calculated and printed. The moments are calculated by using a variational one-particle MP2 density matrix. The calculation of the density matrix substantially increases the computational effort compared to an ordinary energy calculation. If the call to `MBPT2` is followed by a `LOPROP` call the variational MP2 density matrix will automatically be passed on to that module when this keyword is active.

GRDT Variational one and two-particle MP2 densities are calculated to prepare for analytical gradient calculations. The default for subsequent gradient calculations are changed from numerical to analytical when this keyword is invoked. When using `MBPT2` in a `SLAPAF`-loop with only C1 symmetry analytical gradients are automatically default and this keyword is not needed. GRDT prints Multipole moments and

	prepare for `LOPROP` in the exact same way as PRPT. Use of this keyword therefore makes it redundant (but harmless) to also specify the keyword PRPT.
NOGRdt	Disables the calculation of variational densities for analytical gradients. This is useful to cancel the implicit GRDT added when using `MBPT2` inside a `SLAPAF`-loop, if no analytic gradients are actually needed. Note that using the Numerical keyword in `GATEWAY` already disables GRDT, so NOGRDT is only needed in some advanced situations.

Optional keywords specific to Cholesky calculations

Observe that these keywords are disregarded if the integrals were not Cholesky decomposed by `SEWARD`. Furthermore, they are disregarded for algorithm 0 (see below).

Keyword	*Meaning*
CHOAlgorithm	Takes as argument one positive integer specifying the algorithm to use for Cholesky MP2. Options: 0 [generate MO integrals on disk from Cholesky vectors], 1 [compute integrals on-the-fly, minimal operation count, level 2 BLAS], 2 [compute integrals on-the-fly, not minimal operation count, level 3 BLAS], Default is 2.
VERBose	Increases printing from the Cholesky MP2 routines, although not by much. Default is (almost) no printing.
DECOmpose	Requests Cholesky decomposition of the $(ai\|bj)$ integrals. Unless user-defined (see below), the threshold used is identical to that used by `SEWARD` for decomposing the two-electron integrals. Default is to not decompose.
THRCholesky	Specifies the threshold for $(ai\|bj)$ Cholesky decomposition. Takes as argument one real number. Default is the threshold used by `SEWARD` for decomposing the two-electron integrals.
NODEcompose	Turns off Cholesky decomposition of the (ai—bj) integrals. Default is to not decompose.
SPAN	Specifies the span factor used for $(ai\|bj)$ Cholesky decomposition. Takes as argument one real number. Default is the span factor used by `SEWARD` for decomposing the two-electron integrals.
MXQUal	Specifies the max. number of qualified diagonals treated during $(ai\|bj)$ Cholesky decomposition. Takes as argument one integer. Default is 10% of the max. rank of $(ai\|bj)$, although never more than 200.
PRESort	Presort the MO Cholesky vectors according to the batches over occupied orbitals. This will reduce the amount of I/O performed during on-the-fly assembly of the $(ai\|bj)$ integrals. This keyword is obsolete.

Limitations

The maximum number of selectively frozen SFRO or selectively deleted orbitals SDEL in each symmetry is limited to 50.
The limitations on the number of basis functions are the same as specified for SEWARD.

Input example

```
 &MBPT2
Title
 H2O:  O(9.5/4.2), H(4/2)
* The lowest energy occupied orbital in the repr. no.1 will be frozen in
* MBPT2 calculations. The number of representations is 4 and all zeros
* must be explicitly given
Frozen
1 0 0 0
* Two highest energy external orbitals in the repr. no.3 will be deleted
* in MBPT2 calculations. The number of representations is 4 and all
* zeros must be explicitly given
Deleted
0 0 2 0
* One occupied orbital in symmetry no.1 will be additionally frozen by
* using the SFRO option. Let it be the third SCF occupied orbital in
* this symmetry
sFrozen
1 0 0 0   // Gives the number of frozen orbitals in each symmetry
3         // Gives the frozen orbital reference number in symmetry no. 1
* Two external orbitals in symmetry no.1 and one external orbital in
* symmetry 3 will be deleted. In symmetry 1 let it be the second and
* third external orbitals, and in symmetry 3 the third (already deleted
* in by using the option DELE) external orbital
sDeleted
2 0 1 0   // Gives the number of orbitals to be deleted in each symmetry
2 3       // Gives the reference numbers of external orbitals in sym. 1
3         // Gives the reference number of the external orb. in sym. 3
```

8.27 MCKINLEY (a.k.a. DENALI)

8.27.1 Description

This program calculates the second and first order derivatives used for calculating second order derivatives of the energies with perturbation dependent basis sets for SCF and CASSCF wavefunctions. Vibronic frequencies are calculated by the MCLR program which MCKINLEY will start automatically.

MCKINLEY is written such that properties can be computed for any kind of valence basis function that SEWARD will accept. Note, that MCKINLEY can't handle ECP type basis sets.

MCKINLEY is able to compute the following integral derivatives:

- overlap integrals,
- kinetic energy integrals,
- nuclear attraction integrals,

- electron repulsion integrals,
- external electric field integrals,

MCKINLEY employs two different integration schemes to generate the one- and two-electron integral derivatives. The nuclear attraction and electron repulsion integrals are evaluated by a modified Rys-Gauss quadrature [66]. All other integral derivatives are evaluated with the Hermite-Gauss quadrature. The same restriction of the valence basis sets applies as to SEWARD. None of the integral 2nd derivatives are written to disk but rather combined immediately with the corresponding matrix from the wavefunction. Integral 1st derivative information is preprocessed and written to disc for later use in MCLR.

Note that MCKINLEY for frequency calculations normally automatically starts the MCLR with a default input. To override this check the keyword NOMCLR.

At present the following limitations are built into MCKINLEY:

Max number of unique basis functions:	2000
Max number of symmetry independent centers:	500
Highest angular momentum:	15
Highest symmetry point group:	D_{2h}

8.27.2 Dependencies

MCKINLEY depends on the density, Fock matrices and Molecular orbitals generated by SCF or RASSCF. In addition it needs the basis set specification defined in SEWARD. These dependencies, however, are totally transparent to the user.

8.27.3 Files

Input files

MCKINLEY will use the following input files: *RYSRW*, *ABDATA*,*RUNFILE* (for more information see 7.1.2).

Output files

In addition to the standard output unit MCKINLEY will generate the following files.

File	*Contents*
MCKINT	File with information needed by the MCLR response program.

8.27.4 Input

Below follows a description of the input to MCKINLEY.

The input for each module is preceded by its name like:

```
&MCKINLEY
```

Optional keywords

Keyword	*Meaning*
ONEONLY	Compile only the nuclear repulsion and one-electron integrals contribution. The default is to compute all contributions.
CUTOFF	Threshold for ignoring contributions follows on the next line. The default is 1.0D-7. The prescreening is based on the 2nd order density matrix and the radial overlap contribution to the integral derivatives.
SHOWHESSIAN	The Hessian contributions will be printed.
PERTURBATION	This key word takes one of the arguments HESSian or GEOMetry on the following line. HESSian makes MCKINLEY compute the full geometrical Hessian, which is required for a subsequent calculation of the vibronic frequencies with the MCLR program. GEOMetry calculates only those parts of the geometrical Hessian that correspond to symmetry allowed displacements (could be used in a second order geometry optimization). The default is to compute the Hessian for vibrational frequency calculations. Note that in the case of numerical differentiation only symmetry allowed displacements will be computed regardless of this keyword.
NOMCLR	Request that this calculation is not followed by the execution of the MCLR module.
VERBOSE	The output will be a bit more verbose.

The following is an example of an input which will work for almost all practical cases.

```
 &MCKINLEY
Perturbation
Hessian
```

8.28 MCLR

The MCLR program in MOLCAS performs response calculations on single and multiconfigurational SCF wave functions with the technique described in [66]. The right hand side (RHS) and thus the perturbation has to be defined through a preceding MCKINLEY calculation. Second order derivatives are obtained from a MCKINLEY and a consecutive MCLR calculation, with a geometrical displacement as the external perturbation. If the response of a geometrical perturbation is calculated, harmonic frequencies corresponding to the most abundant masses are printed. MCLR also calculates isotope shifted frequencies. Per default, vibrational frequencies are calculated for all possible single isotopic substitutions. MCLR always calculates the response of an electric field and prints the polarizability.

Note that the user should not normally, for frequency calculations, request the execution of this module since this will be automatic.

The MCLR code also calculates the Lagrangian multipliers required for a SA-MCSCF single state gradient [67], where the RHS is generated by the program itself. Through an MCLR and a consecutive ALASKA calculation, analytical gradients of a SA-MCSCF state may be obtained. **Note that ALASKA will automatically run the MCLR module!** Thus, with SLAPAF geometry optimizations of excited MCSCF states can be performed.

The MCLR program is based on the split GUGA formalism. However, it uses determinant based algorithms to solve the configuration interaction problem [68], in analogy to how it is done in the RASSCF. For spin symmetric wave function ($M_S = 0$) the time reversal symmetry is used, and the innermost loops are performed in combinations instead of determinant.

The upper limit to the size of the CI wave function that can be handled with the present program is about the same as for the RASSCF. The present version of the code is just able to handle CASSCF wave function, RASSCF wave function will soon be included.

The orbital handling is based on a one index transformation technique. The integrals is the transformed to occupied orbitals in two indexes, this can be done directly or from disk based integrals generated by SEWARD.

8.28.1 Dependencies

To start the MCLR module the one-electron integrals generated by SEWARD have to be available. Moreover, MCLR requires the wave function from a SCF or RASSCF calculation and apart from in an evaluation of SA-MCSCF gradients, it also requires the differentiated integrals from MCKINLEY.

8.28.2 Files

Input files

MCLR will use the following input files: *ONEINT*, *ORDINT*,*RUNFILE*, *ABDATA*, *RYSRW*, *JOBIPH* (for more information see 7.1.2).

Output files

File	*Contents*
MCKINT	Communication file between MCLR and MCKINLEY and RASSI
UNSYM	ASCII file where all essential information, like geometry, Hessian normal modes and dipole derivatives are stored.
MLDNFQ	Molden input file for harmonic frequency analysis.

Scratch files

File	*Contents*
TEMP0x	x=1,8 used for for integral transformation and storing half transformed integrals.
REORD	Used for storing data used in the transformation of CI vectors from determinant base to CSF base.
TEMPCIV	Exchange file for temporary storing the CI vectors during the PCG.
RESP	Binary file where the solution of the response equations are stored.
JOPR	Used for half transformed integrals in direct mode.
KOPR	Used for half transformed integrals in direct mode.

8.28.3 Input

This section describes the input to the MCLR program in the Molcas program system. The input for each module is preceded by its name like:

```
&MCLR
```

Optional keywords

A list of these keywords is given below:

Keyword	*Meaning*
SALA	Makes MCLR compute the Lagrangian multipliers for a state average MCSCF wave function. These multipliers are required by ALASKA to obtain analytical gradients for an excited state, when the excited state is determined by a SA optimization. SALA has to be followed by an integer on the next line, specifying the excited state for which the gradient is required.
NAC	Like SALA, but for computing nonadiabatic couplings. It must be followed by two integers on the next line, specifying the states between which the coupling is required. Note that, unlike SALA, the numbering here is absolute, regardless of which roots are included in the state average.
EXPDimension	Here follows the dimension of the explicit Hamiltonian used as preconditioner in the Preconditioned conjugate gradient algorithm. Default 100.
ITERations	Specify the maximum number of iterations in the PCG. Default 200.

LOWMEMORY Lowers the amount of memory used, by paging out the CI vectors on disk. This will lower the performance, but the program will need less memory.

PRINT Raise the print level, default 0.

RASSI This keyword is used for transforming the CI vectors to split GUGA representation, and transforming the orbital rotations to AO basis, to make the response accessible for state interaction calculations.

SEWARD Specify one particle operators, used as right hand side, form the *ONEINT* file constructed by `SEWARD` The keyword is followed by one row for each perturbation: LABEL symmetry Component

ENDSEWARD Marks the end of perturbation specifications read from `SEWARD` *ONEINT* file.

THRESHOLD Specify the convergence threshold for the PCG. Default is 1.0e-4.

DISOTOPE Calculates frequencies modified for double isotopic substitution.

THERMOCHEMISTRY Request an user specified thermochemical analysis. The keyword must be followed by a line containing the Rotational Symmetry Number, a line containing the Pressure (in atm), and lines containing the Temperatures (in K) for which the thermochemistry will be calculated. The section is ended by the keyword "End of PT".

TIME Calculates the time dependent response of an electric periodic perturbation. The frequency of the perturbation should be specified on the following line. Used to calculated time dependent polarizabilities and required in a `RASSI` calculation of two photon transition moments.

MASS Used to generate single and double (in conjunction with DISO) isotope shifted frequencies, with the isotope masses specified by the user. This implementation can be useful for example in calculating intermolecular frequencies which are contaminated by the BSSE. By setting the corresponding masses to the very large numbers, ghost orbitals can be used in the frequency calculation. MASS needs the atomic label and the new mass in units of u (real), for each element of the molecule.

Input example

A default input for a harmonic frequency calculation.

```
 &MCLR
```

An input for a harmonic frequency calculation with modified isotopic masses for hydrogen and oxygen.

```
 &MCLR
```

```
MASS
H   = 2.0079
O   = 150000.998
```

Thermochemistry for an asymmetric top (Rotational Symmetry Number = 1), at 1.0 atm and 273.15, 298.15, 398.15 and 498.15 K.

```
 &MCLR
THERmochemistry
 1
 1.0
 273.15 ;  298.15 ;  398.15 ;  498.15
End of PT
```

The time dependent response is calculated for a perturbation of frequency 0.2 au.

```
 &MCLR
TIME = 0.2
```

The time dependent response is calculated for a perturbation of frequency 0.2 au.

The input:

```
 &MCLR
SALA   = 2
```

computes the Lagrangian multipliers for state number 2 in the SA root. Note, that 2 refers to the SA root. Thus, if the ground state is not included in the SA, the numbering of roots in the CI root and SA root differ. With the following RASSCF input:

```
 &RASSCF
CiRoot
 2 3
 2 3
 1 1
RlxRoot = 2
```

SALA 2 yields the gradient for CI root number 3. Geometry optimization of an excited SA-CASSCF state can be done normally using EMIL commands, and requires the use of the RLXR keyword in the RASSCF input to specify the selected root to be optimized. An explicit input to MCLR is not required but can be specified if default options are not appropriate.

8.29 MCPDFT

The MCPDFT program in Molcas performs multiconfiguration pair-density functional theory (MC-PDFT) calculations, as described in [69, 70]. The MC-PDFT method involves two steps: (i) a CASSCF, RASSCF, or GASSCF wave function calculation to obtain the kinetic energy, classical Coulomb energy, total electron density, and on-top pair density; (ii) a post-SCF calculation of the remaining energy using an on-top density functional. In the current implementation, the on-top pair density functional is obtained by "translation" (t) of exchange-correlation functionals. Four translated functionals are currently available: tLSDA, tPBE, tBLYP, and trevPBE, in addition to the "fully-translated" (ft) variants[71] : ftLSDA, ftPBE, ftBLYP, and ftrevPBE. As multiconfigurational wave functions are used as input quantities, spin and space symmetry are correctly conserved.

The molecular orbitals and one- and two-body density matrices are read from the *JOBIPH* (or *JOBOLD*) file generated during a RASSCF run.

8.29.1 Dependencies

To start the MCPDFT module, the one-electron and two-electron integrals generated by SEWARD are required. For MC-PDFT calculations it is suggested to use a fine or ultrafine grid via the following input specifications (see the SEWARD section for further details):

```
&SEWARD
grid input
grid=ultrafine
end of grid input
```

Additionally, a *JOBIPH* file must be supplied to provide the molecular orbitals and one- and two-body density matrices.

8.29.2 Files

Input files

MCPDFT will use the following input files: *ONEINT*, *ORDINT*,*RUNFILE*, *JOBOLD*, *JOBIPH*

If Cholesky options are selected, additional Cholesky-related files will also be used. Only *JOBIPH* or *JOBOLD* is needed; the code will first look for *JOBOLD* first and *JOBIPH* second.

Output files

File	*Contents*
JOBPDFT	This file is written in binary format and has the same structue of the *JOBIPH* file.
RUNFILE	The *RUNFILE* is updated with information from the MC-PDFT calculation.
MCDENS	This ASCII file is generated for MC-PDFT calculations. It contains spin densities, total density and on-top pair density values on grid (coordinates in a.u.).

8.29.3 Input

This section describes the input to the MCPDFT program in the MOLCAS program system. The input starts with the program name

```
&MCPDFT
```

The KSDFT is the only required keyword.

Keyword	*Meaning*
KSDFT	The functional choice follows. Currently available functionals are: tPBE, tBLYP, tLSDA, trevPBE, ftPBE, ftBLYP, ftLSDA, and ftrevPBE.

Input example

The following example shows the input to the RASSCF and MCPDFT programs for a calculation on the water molecule. The tPBE functional is used. The calculation is performed in C_{2v} symmetry (symmetries: a_1, b_2, b_1, a_2, where the two last species are antisymmetric with respect to the molecular plane). Inactive orbitals are $1a_1$ (oxygen $1s$) $2a_1$ (oxygen 2s) and $1b_1$ (the π lone-pair orbital). Two bonding and two anti-bonding OH orbitals are active, a_1 and b_2 symmetries. The calculation is performed for the $^1A_1$ ground state. Note that no information about basis set, geometry, etc has to be given. Such information is supplied by the SEWARD integral program via the one-electron integral file *ONEINT*.

```
  &RASSCF
 Title= Water molecule. Active orbitals OH and OH* in both symmetries
 Spin     = 1
 Symmetry = 1
 Inactive = 2 0 1 0
 Ras2     = 2 2 0 0
  &MCPDFT
  KSDFT=TPBE
```

The first RASSCF run is a standard CASSCF calculation that leads to variationally optimized orbitals and CI coefficients. The MC-PDFT run will use the orbitals and density matrices optimized during the preceeding RASSCF run.

8.30 MKNEMO

The MKNEMO module generates intermolecular potential between two subsystems and saves all informations in the NEMO file format.

8.30.1 Description

According to the NEMO model of interaction between two subsystems, the MKNEMO module splits super-system into two clusters: A and B. The subsystems A and B are defined in the main coordinate system, R. Calculations of the interaction potential between two subsystems are performed for different configurations. At the first step one has to transform both subsystems to the first configuration. An identical transformation to the first configuration is only allowed for one of the subsystems if any coordinates of atoms in the A-subsystem are the same as any atom's coordinates of the B-subsystem. Generally, we define T_A and T_B as transformation operations of the first and second subsystem to the first configuration(R_A, R_B) from the main coordinate system (R):

$$T_A : R \rightarrow R_A,$$

$$T_B : R \to R_B.$$

Any other configuration can be obtained by transformation ie., translation or rotation of one of the subsystems. For any configuration, one has to calculate the total energy of super-system, A-subsystem with the virtual orbitals of the B-subsystem, and the B-subsystem with virtual orbitals of the A-subsystem at first (unperturbed theory) and second (perturbation) level of theory.

The MKNEMO is written in such way that at first step user has to:

1. define different molecules in global coordinate system, R and the molecules can overlay.
2. Define the clusters, A and B, using translation and rotation operations applied for the molecules and for the clusters themselves.
3. Define all possible displacements of any cluster to obtain new configuration.

In the second step user has to provide any input of MOLCAS module which is able to calculate the total energy of the super-system, A-subsystem, and B-subsystem on the first and second level of theory for a given configuration. After any calculation of total energy, one has to call proper block of MKNEMO module, GetE, to save energy in the MKNEMO.Conf file. Finally, in the third step, user has to generate new configuration, according to displacement transformations. All three steps are placed in the do-while loop.

8.30.2 Files

Standard input

The MKNEMO obeys all rules for format of MOLCAS's input except order of MKNEMO's blocks in an input. The input is always preceded by the dummy namelist reference `&MkNemo &End` and ended by `End Of Input`.

Example:

```
&MkNemo &End
  .................
 End of input
```

The MKNEMO defines *transformation* as translation,T, or rotation, R, operation in a format:

```
[ x y z angle]
```

where the $[x\ y\ z]$ is a 3D-vector of translation, or the $[x, y\ z]$ is a 3D-vector of rotation if the *angle* parameter is presented, and the *angle* is an optional parameter which is an angle of rotation around this vector in degrees. Generally, translation and rotation operation do not commute, since that the MKNEMO first applys transformation from left to right, i.e.: product $T\ R$ means that the MKNEMO will apply first rotation and then translation.

The input of MKNEMO module has been split into four groups of keywords:

- **Mole**cules, **Clus**ters, and **Displ**acement,
- **GetE**energy,

- **Next**,
- **Test**.

All keywords can be provided in a full name but only first 4 characters (bold characters) are recognize by MKNEMO.

Keyword *Meaning*

MOLE, CLUS, AND DISP The keywords must be provided in right order in the input file. And the blocks of keywords, MOLE, CLUS, and DISP, cannot be split between separated MKNEMO inputs.

The definition of a **Mole**cule has format:

```
Mole : MoleculeName
  AtomLabel  x  y  z
  .........  .. .. ..
  AtomLabel  x  y  z
End
```

where the **Mole** is keyword which marks beginning of a molecule's block, the MoleculeName is an unique name of molecule, the AtomLabel is the label of atom, and x, y, and z are coordinates of atoms. The name of the molecule is case sensitive, but atom's label is not.

In the **Clus**ter's block, user defines a cluster in format:

```
Clus : ClusterName  ClusterTransformation
  MoleculeName  MoleculeTransformation
  ............  ......................
  MoleculeName
End
```

where the **Clus** keyword marks beginning of cluster's block, the ClusterName[MoleculeName] is an unique name of cluster[molecule], and the ClusterTransformation[MoleculeTransformation] is an optional argument which defines a transformation of the cluster[molecule]. The cluster and molecule names are case sensitive. The MoleculeName must be defined in a **Mole** block.

The DISPlacement block contains information about transformations of one of the clusters in the format:

```
Disp
 ClusterName  NumberOfSteps Transformation
 ...........  ............. ......... .....
 ClusterName  NumberOfSteps Transformation
End
```

where the ClusterName is a name of one of the clusters which has been defined in **Clus** block, the NumberOfSteps is a number of steps in which transformation will be reached, the Transformation is a translation or rotation. Any kind of transformations must be provided line by line in the **Disp**lacement block and number of transformations is not limited. It means that any row of the DISPlacement block contains information about different transformations. Any new configuration is simply generated from the previous configuration. In this point we can construct final transformation, from the starting configuration to current configuration, as a product of all previous transformations for given subsystem. The `MKNEMO` will store final transformation in order $T\ R$

Any atomic coordinates and vectors of transformations must be provided in a.u. units. The coordinates of transformation vector can be separated by space or a comma .Moreover, the **Mole**cule blocks must be provided first, then the **Clus**ter blocks must appear, and finally **Disp**lacement block. In a mixed order, the `MKNEMO` will not be able to recognize a label of molecule[cluster] defined below a block which is using it.

An execution of `MKNEMO` module within defined **Mole**, **Clus**, and **Disp** blocks in an input will generate a two coordinate files, named MKNEMO.Axyz and MNEMO.Bxyz. Those files contain coordinates of atoms for clusters B and A respectively, and can be used directly in the `SEWARD` and `GATEWAY` (see documentation of `GATEWAY` for COORD keyword).

By default, the `SEWARD` or `GATEWAY` will apply symmetry, so **user must be aware that the displacement transformation can break symmetry of the system and the `MKNEMO` does not control it**. If you do not want use symmetry see documentation of `SEWARD` or `GATEWAY` for details.

Example:

```
&MkNemo&End
  * Molecules definitions
  Mole : H2o
   H   1.43  0.0  1.07
   H  -1.43  0.0  1.07
   O   0.00  0.0  0.00
  End
  Mole : Cm3+
   Cm  0.0 0.0 0.0
  End
  * Clusters definitions
  Clus : Cm3+H2o
   H2o  [0.0 0.0 -1.0] [0.0 1.0 0.0 180.0]
   Cm3+
  End
  Clus : H2O [0.0 0.0 2.0]
   H2o  [0.0 0.0 1.0]
  End
  Disp
    Cm3+H2o   : 3 [0.0,0.0,3.0]
    Cm3+H2o   : 1 [0.0,3.0 0.0]
    H2O       : 2 [0.0 0.0,1.0 90.0]
  End
End Of Input
```

In this example, we define two molecules, H2o and Cm3+. Then we define a Cm3+H2o cluster which has been build form H2o and Cm3+ molecule. The H2o molecule has been rotated around Y-axis by the 180 degree and translated along Z-axis by 2 a.u.. The Cm3+ molecule stays unchanged. The second cluster, named H2O has been constructed from translated H2o molecule. The H2o molecule has been translated along Z-axis by 1 a.u.. Then the H2O cluster has been translated along Z-direction by 2 a.u. In the **Disp** block Cm3+H2o subsystem is translated by vector [0,0,3] in the three steps. Then, in the second row we define translation of H2O cluster by vector [0,3,0] in one step. Finally we rotate H2O cluster by 90 degree around [0,0,1] vector in the two steps. The total number of different configurations is simply a sum of steps: 9=3+1+2+first configuration.

GETE

The **GetE**nergy block is used to read total energy stored at RUNFILE, and to save it into the MKNEMO.Conf file. The argument of GetEnergy block must be present and it must be a label from the list below. Use

S1 to save the energy of super-system at the first level of theory,

S2 to save the energy of super-system at the second level of theory,

A1 to save the energy of the A-subsystem with virtual orbitals of B-subsystem at the first level of theory,

A2 to save the energy of the A-subsystem with virtual orbitals of B-subsystem at the second level of theory,

B1 to save the energy of the B-subsystem with virtual orbitals of A-subsystem at the first level of theory,

B2 to save the energy of the B-subsystem with virtual orbitals of A-subsystem at the second level of theory.

Please note, that MKNEMO does not have any possibility to check what kind of total energy was computed in the previous step by any MOLCAS module. The user has to pay attention on what kind of energy was computed in the previous step.

Example:

```
&MkNemo&End
  GetE
    A1
End Of Input
```

In this case the total energy which has been computed by a MOLCAS module will be saved as energy of the A-subsystem with virtual orbitals of B-subsystem at the first level of theory.

NEXT

The **Next** block is used to save all information about potential curve from previous step into the MKNEMO.Nemo file (the command Next will move data from MKNEMO.Conf file into MKNEMO.Nemo file and will delete MKNEMO.Conf file) and to continue or break an

EMIL's loop. **This block cannot be used before Mole, Clus, and Disp blocks.**

Example:

```
&MkNemo&End
  Next
End Of Input
```

TEST The **TEST** block CAN BE ONLY USED to save verification data for MOLCAS command *verify*.

Example:

```
&MkNemo&End
  Test
End Of Input
```

Finally the structure of a standard input file for `MKNEMO` module has the following form:

```
* Loop over configurations
>>>>>>>>>>>>>>>>>>> Do While <<<<<<<<<<<<<<<<<<<<
  &MkNemo&End
    * Molecules definitions
    Mole : MoleculeName
      AtomLabel  x  y  z
      .........  .. .. ..
      AtomLabel  x  y  z
    End
    ....................
    Mole : MoleculeName
      AtomLabel  x  y  z
      .........  .. .. ..
      AtomLabel  x  y  z
    End
    *
    Clus : ClusterName  ClusterTransformation
      MoleculeName  MoleculeTransformation
      ............  ......................
      MoleculeName
      MoleculeName
    End
    Clus : ClusterName  ClusterTransformation
      MoleculeName  MoleculeTransformation
      ............
      MoleculeName
    End
    Disp
      ClusterName  NumberOfSteps [x y z alpha]
      ClusterName  NumberOfSteps [x y z]
      ...........  ............. .............
      ClusterName  NumberOfSteps [x y z alpha]
    End
  End Of Input
  *************** SUPER-SYSTEM CALCULATION *********************
  * Calculation of integrals
  &Seward
    coord=$Project.MkNemo.Axyz
    coord=$Project.MkNemo.Bxyz
    basis=.......
     ................................
  * Energy calculation on the first level of the theory
  &Scf
     ...............................
  * Save energy
  &MkNemo
    GetE=S1
  * Energy calculation on the second level of the theory
  &MBPT2
     ..............................
  * Save energy
  &MkNemo
    GetE=S2
  *************** A-SUBSYSTEM CALCULATION ********************
  * Calculation of integrals
  &Seward
    coord=$Project.MkNemo.Axyz
    coord=$Project.MkNemo.Bxyz
    * the B-subsytem has charge equal to zero
    BSSE=2
    basis=........
     ...............................
  * Energy calculation on the first level of the theory
  &Scf
     ..............................
  * Save energy
  &MkNemo&End
    GetE=A1
  * Energy calculation on the second level of the theory
  &MBPT2
     ..............................
  * Save energy
  &MkNemo
    GetE=A2
  *************** B-SUBSYSTEM CALCULATION ********************
  * Calculation of integrals
  &Seward
    coord=$Project.MkNemo.Axyz
    coord=$Project.MkNemo.Bxyz
    * the A-subsytem has charge equal to zero
    BSSE=1
    basis=........
     ...............................
  * Energy calculation on the first level of the theory
  &Scf
     ..............................
  * Save energy
  &MkNemo
    GetE=B1
  * Energy calculation on the second level of the theory
  &MBPT2
     ..............................
  * Save energy and take next configuration
  &MkNemo
    GetE=B2; Next
>>>>>>>>>>>>>>>>>>> EndDo <<<<<<<<<<<<<<<<<<<
```

Example:

```
*
* Loop over all configurations
*
>>>>>>>>>>>>>>>>>>> Do While <<<<<<<<<<<<<<<<<<<
  *
  * H2O and H2O clusters
  *
  &MkNemo&End
    * Molecules definitions
    Mole : H2O
     H   1.43  0.0  1.07
     H  -1.43  0.0  1.07
```

```
   O   0.00  0.0  0.00
  End
  * Clusters definitions
  Clus : H2O
   H2O : [0.0 1.0 0.0 180.0]
  End
  Clus : h2o [ 0.0 0.0 2.0]
   H2O
  End
  Disp
    h2o : 10 [0.0  0.0, 5.0       ]
    h2o : 10 [0.0, 0.0, 20.0      ]
    h2o : 18 [0.0  0.0  1.0  180.0]
  End
End Of Input
*************** SUPER-SYSTEM CALCULATION *********************
* Calculation of integrals
&Seward
  NEMO
  Title=Sypersystem
  Douglas-Kroll
  ANGM= 0.0 0.0 0.0; AMFI
  COORD=$Project.MkNemo.Axyz;Coord=$Project.MkNemo.Bxyz
  basis=H.ano-rcc...2s1p.,O.ano-rcc.Roos..4s3p2d1f.
* Energy calculation on the first level of the theory
&Scf
  Title=Supersystem; Occupied=10; Iterations=30; Disk=1 0
* Save energy
&MkNemo
  GetE=S1
* Energy calculation on the second level of the theory
&MBPT2
  Title=Sypersystem; Threshold=1.0d-14 1.0d-14 1.0d-14
* Save energy
&MkNemo
  GetE=S2
*************** A-SUBSYSTEM CALCULATION *********************
* Calculation of integrals
&Seward
  NEMO
  Title=A-system
  Douglas-Kroll
  ANGM= 0.0 0.0 0.0; AMFI
  COORD=$Project.MkNemo.Axyz;Coord=$Project.MkNemo.Bxyz
  basis=H.ano-rcc...2s1p.,O.ano-rcc.Roos..4s3p2d1f.
  BSSE=2
* Energy calculation on the first level of the theory
&Scf
  Title=A-subsystem; Occupied=5; Iterations=30; Disk=1 0
* Save energy
&MkNemo
  GetE=A1
* Energy calculation on the second level of the theory
&MBPT2
  Title=A-subsystem; Threshold=1.0d-14 1.0d-14 1.0d-14
* Save energy
&MkNemo
  GetE=A2
*************** B-SUBSYSTEM CALCULATION *********************
* Calculation of integrals
&Seward
  NEMO
  Title=A-system
  Douglas-Kroll
  ANGM= 0.0 0.0 0.0; AMFI
  COORD=$Project.MkNemo.Axyz;Coord=$Project.MkNemo.Bxyz
```

```
    basis=H.ano-rcc...2s1p.,O.ano-rcc.Roos..4s3p2d1f.
    BSSE=1
  * Energy calculation on the first level of the theory
   &Scf
     Title=B-subsystem; Occupied=5; Iterations=30; Disk=1 0
    * Save energy
   &MkNemo
     GetE=B1
  * Energy calculation on the second level of the theory
   &MBPT2
     Title=B-subsytem; Threshold= 1.0d-14 1.0d-14 1.0d-14
  * Save energy and take next configuration
   &MkNemo
    GetE=B2; Next
>>>>>>>>>>>>>>>>>>>> EndDo <<<<<<<<<<<<<<<<<<<<
```

In this example we calculate potential energy curve for interaction between two water clusters. The A-cluster, H2O, was rotated around Y-axis about 180 degrees. The B-subsystem, h2o,has been translated along Z-axis by 2 a.u.. In the **Disp** block we have defined 20 translation operations for h2o cluster and 18 rotation operations for H2O cluster. For energy calculations of super-system, A-subsystem, and B-subsystem, at first level of theory we used SCF module, and MBPT2 at second level of theory, respectively. After a calculation of energy we save calculated results using keyword **GetE** with proper argument in the MKNEMO.Conf file of MKNEMO module. Finally, by calling block **Next** of MKNEMO, we save all informations about potential for given configuration and we generate new configuration. This procedure will be repeated for all translations and rotations defined in the Displacement block.

Input files

Apart from the standard input unit MKNEMO will use the following input files.

File	*Contents*
MKNEMO.Input	A MKNEMO's input file contains the latest preprocessed input.
MKNEMO.Restart	The MKNEMO.Restart is a restart file, which will be generated by MKNEMO at the first run if the file does not exist. Any call of **group of command: Mole, Clus, and Disp** will be updated and the restart file is saved in user's $CurrDir. If MKNEMO calculation crashes, one can fix a reason of crash, copy restart and MKNEMO.Nemo files to $WorkDir, and run the calculation again. The MKNEMO will restart calculation from the last point which has been finished successfully. If the MKNEMO.Nemo file will not be copied the MKNEMO will generate a new one and will overwrite the file in your $CurrDirr if any exist. Beware of it. The restart file is formated : `<Restart>   RowInDisp   Step'sNum LoopControl</Restart>` where the RowInDisp is the index of currently used row in the **Disp** block and the Step'sNum is the current number of step for a given displacement's row. Ex. If a displacement row, RowInDisp, is 3, which

corresponds to a displacement row, H2O 4 [0,4,0], and Step'sNum is equal to 2 then it means that current displacement vector is [0,2,0]. The LoopControl parameter is a return code. The command **Next** will read this value and use it to continue looping or breaking a loop.

MKNEMO.Conf The MKNEMO.Conf is a file which stores block **Mole**, **Clus**, and **Eneries** in similar format like it is define in the input of theMKNEMO, but within XML format. The propose of this file is to share definition of molecules, clusters, and energies between different blocks of namelist, &MkNemo. A format of this file is:

```
* Configuration definition - contains informations
* about configuration
<Configuration>
  * Definition of molecule
  <Molecule Name=''Name of molecule''>
    labelOfAtom x  y  z
    .......... .. .. ..
    labelOfAtom x  y  z
  </Molecule>
  .................................
  <Molecule Name=''Name of molecule''>
    labelOfAtom x  y  z
    .......... .. .. ..
    labelOfAtom x  y  z
  </Molecule>
  * Definition of cluster
  <Cluster Name=''Name of cluster A'' Transformation=''x y z q0 q1 q2 q3''>
    labelOfMolecule x' y' z'  q0' q1' q2' a3'
    .............. .. .. ..  ..  ..  ..  ..
    labelOfMolecule x' y' z'  q0' q1' q2' a3'
  </Cluster>
  <Cluster Name=''Name of cluster B'' Transformation=''x y z q0 q1 q2 q3''>
    labelOfMolecule x' y' z'  q0' q1' q2' a3'
    .............. .. .. ..  ..  ..  ..  ..
    labelOfMolecule x' y' z'  q0' q1' q2' a3'
  </Cluster>
  * Eneriegies definition
  <Energies>
    EnegyLabel MethodLabel Energy
    ......... .......... ......
    EnegyLabel MethodLabel Energy
  </Energies>
</Configuration>
```

where the EnergyLabel is one of labels defined in the **GetE** block, the MethodLabel is a name of method which has been used to calculate energy, and Energy is a vector of eigenvalues. The **Next** command will save energy information into a MKNEMO.Nemo file, and will clear this file. Hacking hint: If you want to use RASSI then do not use call of command **GetE** but postprocess output and print eigenvalues to the MKNEMO.Conf file in the right format (use ! in user input to execute shell command for postprocessing of output)

Files of the SEWARD, SCF, RASSCF, MBPT2, MOTRA, CCSDT, and CASPT2 modules are needed to get total energy on each level of theory for subsystems and super-system.

Output files

In addition to the standard output unit MKNEMO will generate the following files.

File *Contents*

MKNEMO.Axyz,MKNEMO.Bxyz The MKNEMO.*xyz file is a file of coordinates in format:

```
NumberOfAtoms
AdditionalLine
AtomLabel x  y  z
......... .. .. ..
AtomLabel x  y  z
```

where the NumberOfAtoms is a number of atoms in the file, the AdditionalLine is a line where one can provide unit of coordinate (currently MKNEMO supports only a.u.), the AtomLabel is a label of atom, and x, y, z is a vector of coordinates.

RUNFILE A file with informations needed by the block of MOLCAS.

MKNEMO.Nemo On this file MKNEMO will store all information about intermolecular potential in the NEMO file format. This format is used by NEMO to fit intermolecular potential to the NEMO model. The format of this file is defined as follows:

```
<Nemo>
  * Definition of configuration
  <Configuration>
    ............................
  </Configuration>
  ............................
  <Configuration>
    ............................
  </Configuration>
</Nemo>
```

Where configuration block is defined like in the MKNEMO.Conf.

8.30.3 Dependencies

The MKNEMO depends on the modules of MOLCAS program, which calculate the total energy of the system.

8.31 MOTRA

The program MOTRA is used to transform one- and two-electron integrals from AO to MO basis. It reads the one-electron file and the file of ordered and symmetry blocked two-electron integrals generated by SEWARD.

The two-electron integral transformation is performed one symmetry block at a time, as a series of four sequential one-index transformations. The process includes a sorting of the half transformed integrals prior to the second half transformation. This step is performed in core if there is space enough of memory available to keep one symmetry block of integrals. Otherwise the half transformed integrals are written out on an temporary file. The result of the transformation is two files, *TRAONE* and *TRAINT* which contain the transformed one- and two- electron integrals, respectively.

The one-electron transformation is performed for the kinetic integrals and the bare nuclei Hamiltonian. If there are frozen orbitals MOTRA replaces the bare nuclei Hamiltonian by an effective Fock operator, which incorporates the interaction between the frozen (core) electrons and the remaining electrons. In practice this means that in any subsequent calculation (for example MRCI, CPF or MBPT) the effect of the frozen orbitals is incorporated into the one-electron Hamiltonian, and these orbitals need not be explicitly accounted for. The total energy of the frozen electrons is added to the nuclear-nuclear repulsion energy and transferred from MOTRA to the subsequent program(s).

The two-electron transformation is performed from the list of ordered integrals generated by SEWARD (file *ORDINT*).

8.31.1 Dependencies

To run the program MOTRA successfully the one- and two-electron integrals are needed. In addition, a set of MO coefficients must be available. The latter may be obtained by any wave function optimization program.

8.31.2 Files

Input files

The following is a list of input files

File	*Contents*
ONEINT	One-electron integral file generated by SEWARD.
*ORDINT**	Ordered two-electron integrals generated by SEWARD.
INPORB	If MO's are read in formatted form.
JOBIPH	If molecular orbitals are read from a RASSCF interface.

In general, input orbitals are supplied in the form of a formatted ASCII file, but can also be taken directly from the binary interface file, *JOBIPH*, created by the RASSCF program. The selection in controlled by input options.

Output files

The program MOTRA creates two files: The first carries all basic information and a list of transformed one-electron integrals. The second file includes the transformed two-electron integrals.

The following is a list of output files

File	*Contents*
TRAONE	Auxiliary data and transformed one-electron integrals.
*TRAINT**	Transformed two-electron integrals.

8.31.3 Input

This section describes the input to the MOTRA program in the MOLCAS program system. The input for each module is preceded by its name like:

```
&MOTRA
```

Compulsory keywords

The following keywords are compulsory.

Keyword	*Meaning*
LUMORB	Specifies that the molecular orbitals are read from a formatted file produced by one of the wave function generating programs. **Note** that either of LUMORB or JOBIPH should be specified. LUMORB is the default keyword. No additional input is required.
JOBIPH	Specifies that the molecular orbitals are read from a RASSCF job interface file. MOTRA will in this case read the average orbitals. No additional input is required.

When natural orbitals from a RASSCF (or a state averaged CASSCF) calculation are to be used in MOTRA, they can be produced, or extracted from an existing *JOBIPH* file, by RASSCF, using keyword OUTORBITALS.

Optional keywords

There are a few useful optional keywords that can be specified. The following is a list

Keyword	*Meaning*
AUTO	This keyword specified automatic deletion of orbitals based on occupation numbers. The following line contain one threshold per symmetry, and all orbitals with occupation numbers smaller that the threshold will be deleted. If AUTO and DELETE are both specified, the larger number will be used.
DELETED	Specifies the number of virtual orbitals that are not to be used as correlating orbitals in the subsequent CI calculation. The last orbitals in each symmetry are deleted. The default is no deleted orbitals. One additional line with the number of deleted orbitals in each symmetry (free format).

FROZen — Specifies the number of doubly occupied orbitals that are left uncorrelated in subsequent correlation calculation(s). Additional orbitals can be frozen in these programs, but from an efficiency point of view it is preferable to freeze orbitals in the transformation. One additional line with the number of frozen orbitals in each symmetry (free format). For more details on freezing orbitals in MOTRA see the program description. The frozen orbitals are the first in each symmetry block. Default is to freeze the core (but not semi-core) orbitals.

ONEL — Specifies that only one-electron integrals are to be transformed. No additional input is required.

PRINt — Specifies the print level in the program. The default (1) does not print the orbitals that are used in the transformation, but they appear at print level 2. Beware of large print levels since vast amounts of output may be produced. The value is read from the line after the keyword, in free format.

RFPErt — Add a constant reaction field perturbation to the bare nuclei Hamiltonian. The perturbation is read from the file *RUNOLD* (if not present defaults to *RUNFILE*) and is the latest self consistent perturbation generated by one of the programs SCF or RASSCF.

CTONly — Specifies that Cholesky vectors are to be transformed without subsequent calculation of the two-el integrals. It requires as input one of the two following strings: ”pqK” or ”Kpq”, which indicate the storage format as L(pq,K) or L(K,pq), respectively. The former is the default option. Transformed vectors are stored in the files *_CHMOT*, one for each compound symmetry. Available only in combination with Cholesky or RI integral representation.

DIAGonal integrals — Activates the evaluation of the diagonal integrals in MO basis. Requires the keyword CTONly. The file *DIAGINT* is generated which contains these integrals.

TITLe — This keyword should be followed by exactly one title line.

Input example

```
 &MOTRA
Title  =  Water molecule.
* Don’t correlate 1s on oxygen
Frozen =  1 0 0 0
Lumorb
```

8.32 MPPROP

8.32.1 Description

The MPPROP is a general distributed multipole expansion, and a first order polarizabilty analyzis program. It will use the one electron integrals to generate the distribution. The order of the distributed multipole expansion is defined by SEWARD. In order to generate distributed multipoles of higher order than 2. One has to use the Keyword MULTipoles in SEWARD. SEWARD also needs the Keyword NEMO to arrange the integrals in correct order.

8.32.2 Dependencies

The MPPROP program requires the one-electron integral file *ONEINT* and the communications file *RUNFILE*, which contains among others the multipole integrals produced by SEWARD.

8.32.3 Files

Below is a list of the files that are used/created by the program MPPROP.

Input files

File	*Contents*
ONEINT	One-electron integral file generated by the program SEWARD.
RUNFILE	File for communication of auxiliary information generated by the different programs e.g. SEWARD.
INPORB	*SCFORB* or *RASORB* file containing the orbitals of a previous SCF run or a RASSCF run , which are used now as vectors in the MPPROP run.

Output files

File	*Contents*
MPPROP	The distributed multipole expansion.

8.32.4 Input

Below follows a description of the input to MPPROP. The keywords are always significant to four characters, but in order to make the input more transparent, it is recommended to use the full keywords. The MPPROP program section of the MOLCAS input starts with the program:

```
&MPPROP
```

There are no compulsory keywords.

Optional general keywords

Keyword *Meaning*

BONDs

Use this Keyword to define bond between atoms. This Keyword should be followed by a line of atomlabels separated by a space. The following line can define another bond. This Keyword should be ended by a END statement in the last line. The example below means that O1 will bond to H1 and H2. It does not mean that H1 is bonded to H2.

```
BOND
O1 H1 H2
END
```

TITLe

This Keyword specifies the title of the molecule. This will be recognized by the Nemo package. And you are requested to use this Keyword. It is defined in the program as a Character*80

LUMOrb

This Keyword tells MPPROP to use an INPORB file for the one electron densities.

TYPE

This is to specify the typen of the atom. Where the first number is the atomnumber m, counted in the order it was defined in SEWARD. The second number is the type of the atom n.

```
TYPE
m n
```

POLArizability

This specifies if the polarizability should be calculated or not.

```
POLArizability
m
```

- **m=0** Means no polarizability should be calculated.
- **m=1** (Default) Means polarizability should be calculated.

NONEarestAtom

The program is written in the way that multipoles should be moved to the nearest atom if the nearest atom is closer than any of the bonding atoms.Note that the move will be to atoms and not nearest bond.

ALLCenter

This Keywords means that all centers are considered for the distributed multipole expansion.

Limitations

The limitations on the order of the multipole expansion is defined by SEWARD. While the polarizability can only be calculated directly in the program for an scf wavefunction. And it is limited to first order polarizability

Input examples

First we have the bare minimum of input. This will work well for all systems.

```
  &MPPROP
```

The next example is a bit more elaborate and show how to use a few of the keywords. The system is formic-acid.

```
  &Gateway
 Title
 Fa
 NEMO
 Basis set
 C.ANO-L...3s2p1d.
 C1       2.15211991525414      -3.97152266198745       4.15134452433510
 End of Basis set
 Basis set
 O.ANO-L...4s3p1d.
 O2      3.99101917304681      -2.23465022227817       3.72611355598476
 O3      2.36712399248396      -5.81178517731397       5.48680572323840
 End of Basis set
 Basis set
 H.ANO-L...3s1p.
 H4       0.43787447048429      -3.44210745229883       3.08410918233085
 End of Basis set
 Basis set
 H.ANO-L...3s2p.
 H5      5.46574083366162      -2.78397269852552       4.70186773165853
 End of Basis set
  &Seward
 & Scf
 Title
 Formic-acid
 Occupied
   12
  &MPPROP
 Title
 Formic-acid
 lumorb
 POLArizability
 1
 BONDs
 C1 O2 O3 H4
 O2 H5
 End
 TYPE
 2 1
 3 2
 4 1
 5 2
 End
```

8.33 MRCI

The MRCI program generates Multi Reference SDCI or ACPF[41] wavefunctions. ACPF is a modification of the CPF[39] method which allows more than one reference configuration. The program is based on the Direct CI method[42], and with the coupling coefficients generated with the Graphical Unitary Group Approach[43]–[44]. (See program description for GUGA). If requested, MRCI computes matrix elements of those one-electron properties for which it can find integrals in the *ONEINT* file. It also generates natural orbitals that can be fed into the property program to evaluate certain one electron properties. The natural orbitals are also useful for Iterated Natural Orbital (INO) calculations.

The MRCI code is a modification of an MRCI program written by M. Blomberg and P. E. M. Siegbahn (Institute of Physics, Stockholm University, Sweden), which has later been extensively modified (P.-Å. Malmqvist)

The program can calculate several eigenvectors simultaneously, not necessarily those with lowest eigenvalue. However, in the ACPF case, only one single eigenvector is possible.

Orbital subspaces

The orbital space is divided into the following subspaces: Frozen, Inactive, Active, Secondary, and Deleted orbitals. Within each symmetry type, they follow this order.

- **Frozen:** Frozen orbitals are always doubly occupied, i.e., they are not correlated. Orbitals may be frozen already in the integral integral transformation step, program MOTRA, but can also be specified in the input to the MRCI program. The former method is more efficient, and has the effect that the frozen orbitals are effectively removed from the subsequent MRCI calculation.

- **Inactive:** Inactive orbitals are doubly occupied in all reference configurations, but excitations out of this orbital space are allowed in the final CI wavefunction, i.e., they are correlated but have two electrons in all reference configurations. Restrictions may be applied to excitation from some inactive orbitals, see keyword NoCorr in the GUGA input section.

- **Active:** Active orbitals are those which may have different occupation in different reference configurations. Restrictions may be applied to occupation of some active orbitals, see keyword OneOcc in the GUGA input section.

- **Secondary:** This subspace is empty in all reference configurations, but may be populated with up to two electrons in the excited configurations. This subspace is not explicitly specified, but consists of the orbitals which are left over when other spaces are accounted for.

- **Deleted:** This orbital subspace does not participate in the CI wavefunction at all. Typically the 3s,4p,... components of 3d,4f..., or orbitals that essentially describe core correlation, are deleted. Similar to freezing, deleting can be done in MOTRA, which is more efficient, but also as input specifications to the MRCI program.

Since ordinarily the frozen and deleted orbitals were handled by MOTRA and the subdivision into inactive and active orbitals were defined in GUGA, the only time one has to specify orbital spaces in the input to MRCI is when additional frozen or deleted orbitals are required without recomputing the transformed integrals.

8.33.1 Dependencies

The program needs the coupling coefficients generated by the program GUGA and transformed one- and two-electron integrals generated by the program MOTRA.

8.33.2 Files

Input files

File	*Contents*
CIGUGA	Coupling coefficients from GUGA.
*TRAINT**	Transformed two-electron integrals from MOTRA.
TRAONE	Transformed one-electron integrals from MOTRA.
ONEINT	One-electron property integrals from SEWARD.
MRCIVECT	Used for input only in restart case.

Output files

File	*Contents*
CIORBnn	One or more sets of natural orbitals, one for each CI root, where nn stands for 01,02, etc.
MRCIVECT	CI vector, for later restart.

Note that these file names are the FORTRAN file names used by the program, so they have to be mapped to the actual file names. This is usually done automatically in the Molcas system. However, in the case of several different numbered files *CIORBnn* only the first will be defined as default, with the FORTRAN file name *CIORB* used for *CIORB01* .

8.33.3 Input

This section describes the input to the MRCI program in the Molcas program system, with the program name:

```
&MRCI
```

Keywords

Keyword	*Meaning*
TITLe	The line following this keyword is treated as title line
SDCI	This keyword is used to perform an ordinary Multi-Reference Singles and Doubles CI, MR-SDCI, calculation. This is the default assumption of the program. Note that SDCI and ACPF are mutually exclusive.
ACPF	This keyword tells the program to use the Average Coupled Pair Functional, ACPF, when computing the energy and natural orbitals. Note that SDCI and ACPF are mutually exclusive.
GVALue	The coefficient g which is used in the ACPF functional. The default value is = 2.0/(Nr of correlated electrons).
NRROots	Specifies the number of CI roots (states) to be simultaneously optimized. The default is 1.
ROOTs	Specifies which root(s) to converge to. These are defined as the ordinal number of that eigenvector of the reference CI which is used as start approximation. The default is the sequence 1,2,3... The values are entered on the next line(s). If the number of roots is larger than 1, it must first have been entered using keyword NRROOTS. The keywords ROOTS and SELECT are mutually exclusive.
SELEct	Another way of specifying the roots: instead of using ordinal numbers, the roots selected will be those NRROOTS which have largest projections in a selection space which is specified on the next lines, as follows: One line gives NSEL, the number of vectors used to define the selection space. For each selection vector, program reads the number of CSF-s (NC), and # NC pairs of CSEL (text strings) and SSEL (coefficients). The text string is composed of the digits 0,1,2,3 and denotes the GUGA case numbers of the active orbitals, defining uniquely a CSF belonging to the reference space. The keywords ROOTS and SELECT are mutually exclusive.
RESTart	Restart the calculation from a previous calculation. No additional input is required. The *MRCIVECT* file is required for restarted calculations.
THRPrint	Threshold for printout of the wavefunction. All configurations with a coefficient greater than this threshold are printed. The default is 0.05. .
ECONvergence	Energy convergence threshold. The result is converged when the energy of all roots has been lowered less than this threshold in the last iteration. The default is 1.0d-8.

PRINt Print level of the program. Default is 5.

MAXIterations Maximum number of iterations. Default 20. The maximum possible value is 49.

MXVEctors Maximum number of trial vector pairs (CI+sigma) kept on disk. Default is MAX(NRROOTS,10). It should never be smaller than NRROOTS. A good value is 3*NRROOTS or more.

TRANsition This keyword is relevant to a multi-root calculation. In addition to properties, also the transition matrix elements of various operators, for each pair of wave functions, will be computed.

FROZen Specify the number of orbitals to be frozen in **addition** to the orbitals frozen in the integral transformation. Default is 0 in all symmetries.

DELEted Specify the number of orbitals to be deleted in **addition** to the orbitals deleted in the integral transformation. Default is 0 in all symmetries.

REFCi Perform only reference CI.

PRORbitals Threshold for printing natural orbitals. Only orbitals with occupation number larger than this threshold appears in the printed output. Default is 1.0d-5.

Input example

```
 &MRCI
Title
 Water molecule. 1S frozen in transformation.
Sdci
```

8.34 MULA

The MULA calculates intensities of vibrational transitions between electronic states.

8.34.1 Dependencies

The MULA program may need one or more UNSYM files produced by the MCLR program, depending on input options.

8.34.2 Files

Input files

File	*Contents*
UNSYM	Output file from the MCLR program

Output files

File	*Contents*
plot.intensity	Contains data for plotting an artificial spectrum.

8.34.3 Input

The input for MULA begins after the program name:

```
&MULA
```

There are no compulsory keyword.

Keywords

Keyword	*Meaning*
TITLe	Followed by a single line, the title of the calculation.
FORCe	A force field will be given as input (or read from file), defining two oscillators for which individual vibrational levels and transition data will be computed.
ATOMs	Followed by one line for each individual atom in the molecule. On each line is the label of the atom, consisting of an element symbol followed by a number. After the label, separated by one or more blanks, one can optionally give a mass number; else, a standard mass taken from the file data/atomic.data. After these lines is one single line with the keyword "END of atoms".
INTErnal	Specification of which internal coordinates that are to be used in the calculation. Each subsequent line has the form 'BOND *a b*' or 'ANGLE *a b c*' or or 'TORSION *a b c d*' or or 'OUTOFPL *a b c d*', for bond distances, valence angles, torsions (e.g. dihedral angles), and out-of-plane angles. Here, *a*. . . *d* stand for atom labels. After these lines follows one line with the keyword "END of internal".

MODEs — Selection of modes to be used in the intensity calculation. This is followed by a list of numbers, enumerating the vibrational modes to use. The modes are numbered sequentially in order of vibrational frequency. After this list follows one line with the keyword "END of modes".

MXLEvels — Followed by one line with the maximum number of excitations in each of the two states.

VARIational — If this keyword is included, a variational calculation will be made, instead of using the default double harmonic approximation.

TRANsitions — Indicates the excitations to be printed in the output. Followed by the word FIRST on one line, then a list of numbers which are the number of phonons – the excitation level – to be distributed among the modes, defining the vibrational states of the first potential function (force field). Then similarly, after a line with the word SECOND, a list of excitation levels for the second state.

ENERgies — The electronic T_0 energies of the two states, each value is followed by either "eV" or "au".

GEOMetry — Geometry input. Followed by keywords FILE, CARTESIAN, or INTERNAL. If FILE, the geometry input is taken from UNSYM1 and UNSYM2. If CARTESIAN or INTERNAL, two sections follow, one headed by a line with the word FIRST, the other with the word SECOND. For the CARTESIAN case, the following lines list the atoms and coordinates. On each line is an atom label, and the three coordinates (x, y, z). For the INTERNAL case, each line defines an internal coordinate in the same way as for keyword INTERNAL, and the value.

MXORder — Maximum order of transition dipole expansion. Next line is 0, if the transition dipole is constant, 1 if it is a linear function, etc.

OSCStr — If this keyword is included, the oscillator strength, instead of the intensity, of the transitions will calculated.

BROAdplot — Gives the peaks in the spectrum plot an artificial halfwidth. The default lifetime is $130 \cdot 10^{-15}$ s but this can be changed with keyword LIFEtime followd by the value.

NANOmeters — If this keyword is included, the plot file will be in nanometers. Default is in eV.

CM-1 — If this keyword is included, the plot file will be in cm^{-1}. Default is in eV.

PLOT — Enter the limits (in eV, cm^{-1}, or in nm) for the plot file.

VIBWrite — If this keyword is included, the vibrational levels of the two states will be printed in the output.

VIBPlot — Two files, plot.modes1 and plot.modes2, will be generated, with pictures of the normal vibrational modes of the two electronic states.

HUGElog — This keyword will give a much more detailed output file.

SCALe — Scales the Hessians, by multiplying with the scale factors following this keyword.

DIPOles — Transition dipole data. If MXORDER=0 (see above), there follows a single line with x, y, z components of the transition dipole moment. If MXORDER=1 there are an additional line for each cartesian coordinate of each atom, with the derivative of the transition dipole moment w.r.t. that nuclear coordinate.

NONLinear — Specifies non-linear variable substitutions to be used in the definition of potential surfaces.

POLYnomial — Gives the different terms to be included in the fit of the polynomial to the energy data.

DATA — Potential energy surface data.

Input example

```
  &MULA
 Title
  Water molecule
 Atoms
  O1
  H2
  H3
 End Atoms
 Internal Coordinates
  Bond  O1 H2
  Bond  O1 H3
  Angle H3 O1 H2
 End Internal Coordinates
 MxLevels
   0  3
 Energies
  First
   0.0 eV
  Second
   3.78 eV
 Geometry
  Cartesian
  First
   O1     0.0000000000      0.0000000000     -0.5000000000
   H2     1.6000000000      0.0000000000      1.1000000000
   H3    -1.6000000000      0.0000000000      1.1000000000
  End
  Second
   O1     0.0000000000      0.0000000000     -0.4500000000
   H2     1.7000000000      0.0000000000      1.0000000000
   H3    -1.7000000000      0.0000000000      1.0000000000
  End
 ForceField
  First state
  Internal
  0.55 0.07 0.01
  0.07 0.55 0.01
  0.01 0.01 0.35
 Second state
 Internal
  0.50 0.03 0.01
  0.03 0.50 0.01
  0.01 0.01 0.25
DIPOles
  0.20 0.20 1.20
BroadPlot
LifeTime
 10.0E-15
NANO
PlotWindow
 260 305
End of input
```

```
 &MULA
TITLe
 Benzene
ATOMs
  C1
  C2
  C3
  C4
  C5
  C6
  H1
  H2
  H3
  H4
  H5
  H6
End of Atoms
GEOMetry
 file
INTERNAL COORDINATES
 Bond    C1 C3
 Bond    C3 C5
 Bond    C5 C2
 Bond    C2 C6
 Bond    C6 C4
 Bond    C1 H1
 Bond    C2 H2
 Bond    C3 H3
 Bond    C4 H4
 Bond    C5 H5
 Bond    C6 H6
 Angle   C1 C3 C5
 Angle   C3 C5 C2
 Angle   C5 C2 C6
 Angle   C2 C6 C4
 Angle   H1 C1 C4
 Angle   H2 C2 C5
 Angle   H3 C3 C1
 Angle   H4 C4 C6
 Angle   H5 C5 C3
 Angle   H6 C6 C2
 Torsion C1 C3 C5 C2
 Torsion C3 C5 C2 C6
 Torsion C5 C2 C6 C4
 Torsion H1 C1 C4 C6
 Torsion H2 C2 C5 C3
 Torsion H3 C3 C1 C4
```

```
  Torsion H4 C4 C6 C2
  Torsion H5 C5 C3 C1
  Torsion H6 C6 C2 C5
 END INTERNAL COORDINATES
 VIBPLOT
  cyclic 4 1
 ENERGIES
  First
   0.0 eV
  Second
   4.51 eV
 MODES
  14 30 5 6 26 27 22 23 16 17 1 2 9 10
 END
 MXLE - MAXIMUM LEVEL of excitation (ground state - excited state)
   2 2
 MXOR - MAXIMUM ORDER in transition dipole.
   1
 OscStr
 Transitions
  First
   0
  Second
   0 1 2
 FORCEFIELD
  First
    file
  Second
    file
 DIPOLES
  file
```

8.35 NEMO

8.35.1 Description

The NEMO is a potential analyzis package that calculates interaction energies between molecules. The package uses input files from MPPROP and MKNEMOPOT. The package was originally a set of programs that has been totaly rewritten and put together into one program. The package are capable of doing fitting of potential surfaces, energy optimization between molecules, calculate some specific potential curves and generate simulation parameters for rigid molecules.

The theoretical background stands in perturbation theory. The interaction energy between two molecules can be described by three quantum chemical calculations. One quantum chemical calculation for each of the monomers and one calculation for the two molecules together i.e. the dimer. The energy for the two monomers are then subtracted from the dimer calculation. That is done for each configuration, i.e. coordinate set for a dimer calculation, given by the input to MKNEMOPOT. The calculations are set up by the MKNEMOPOT package and to performe those calculations it is recommended to read the manual for the MKNEMOPOT package. The interaction energy can also be described in the classical energy terms electrostatic, induction dispersion, repulsion and chargetransfer. Where, a good description the first three energy terms can be given by distributed multipole expansions and distributed polarizabilities. The last two energy terms are harder to predict and are of quantum chemical origin. The reason for calculating the interaction energy quantum chemically is that this reference energy will be used for the description of the repulsion(/chargetransfer) parameters. The repulsive reference energy term is achieved by subtracting the energy for the electrostatic, induction and dispersion from the reference energy. Note here that the dispersion energy is only added if the reference energy is performed with a method that includes true dynamic correlation, i.e. when the energy includes the London dispersion term. The reference energy will also include a charge transfer term if it is defined by the user. An estimation of the repulsive energy term can now be fitted to the reference repulsive energy term by using the FITPar subprogram in the NEMO program. The fitted parameters are classified in elements and type. Where a hydrogen atom is element 1 and can be classified in different types depending on their chemical environment. This information is supplied with the MpProp file together with coordinates, multipole and polarizabilities of a molecule. The MpProp file is an output from the MPPROP or LOPROP program. A MpProp file does not always contain all the information needed to run the NEMO program. Thus, it is important have look directly in the file and do your prefered changes before using it. It can for example be to change the type of a hydrogen atom. If we take the ethanol molecule as an example. It is composed of two carbons, five hydrogens and one oxygen atoms. Here we can define three different type of hydrogens that are bonded to C1, C2 and O1 respectively. The two carbons in the molecule can of course also be defined to be of different type. For each defined type there excists two corresponding parameters for the repulsion energy. These are the ones that are varied to in the fitting procedure.

The input file comming from the MKNEMOPOT program can contain a cluster definition. A cluster is defined as a supermolecule containing one/several different/equal molecules. The interaction energy is thus defined as the interaction between different clusters.

The POTSurf subprogram produces potential energy curves between two clusters. This is normally used to compare the fitted potential with the result from a quantum chemical calculation. Whats happening is that one of clusters are translated and rotated to a certain position. The moved cluster is then translated along a displacement vector.

In the DIMEr subprogram an optimizition/minimizition of the energy between two/several molecules is performed. The routine is not good and practical for many molecules. Because, it was originally written to do the job for two molecules which works pretty good.

The SIMPar program can produce input files for the MOLSIM package.

8.35.2 Dependencies

The NEMO program requires a nemo library. The library is just a concatenation of several different *nemo* files. In order to run the FitPar subprogram in NEMO a *NEMO* file is required. The *NEMO* file is either autogenerated through the MKNEMOPOT or it might be generated by hand from some other potential.

8.35.3 Files

Below is a list of the files that are used/created by the program NEMO.

Input files

File *Contents*

NEMO This file will be opened in the $WorkDir/ directory and is composed of several *Nemo* files generated by `MKNEMOPOT`.

ATOMPAR This file will be opened in the $WorkDir/ directory and it holds the atomic parameters for repulsion, scaling constants for the dispersion, valence of the atoms. It will originaly be stored in the $MOLCAS/nemo_libary directory. It's definition is: two dummy lines, nElements=103 of lines and all this taken nType=4 times. The signifacant nElements of lines will hold 12 columns. Where the first column is the element number, the second column is the element label, the third column

Columns in the ATOMPAR file

- **Column=1** element number
- **Column=2** element label
- **Column=3** Alpha
- **Column=4** Kappa
- **Column=5** Charge Transfer Alpha
- **Column=6** Charge Transfer Kappa
- **Column=7** Valence of the atom
- **Column=8** RepExp an integer for the rˆ-n type potential.
- **Column=9** RepFac
- **Column=10** DispFac
- **Column=11** K1/Sigma
- **Column=12** K2/Epsilon

Output files

File *Contents*

POTSURF This file holds the potential curve. The columns of the PotSurf file will be:

- **Column=1** Coordinate 1

	- **Column=2** Electrostatic+Induction+Repulsion - **Column=3** Electrostatic+Induction+Repulsion+Dispersion - **Column=4** Electrostatic - **Column=5** Induction - **Column=6** Dispersion - **Column=7** Repulsion - **Column=8** Charge Transfer
MOLSIM	The input file in molsim format for the particle part.
MOLSIMLIB	The library file in molsim format for the repulsive and dispersive part.
ATOMFIT	This is the same file as ATOMPAR, but it is written to the $WorkDir directory

8.35.4 Input

Below follows a description of the input to NEMO. The keywords are always significant to four characters, but in order to make the input more transparent, it is recommended to use the full keywords. The NEMO program section of the MOLCAS input is bracketed by a preceding dummy namelist reference

```
/&NEMO /&END
```

and an "end of input" statement

```
End of Input
```

Argument(s) to a keyword are always supplied on the next line of the input file, except explicitly stated otherwise.

Optional general keywords

Keyword	*Meaning*

ALPHA	Use this Keyword to define the alpha parameter for a specific atom and atomtype. The keyword should be followed by a line/lines composed of the element number, the atomtype and the value for alpha.This Keyword should be ended by a END statement in the last line. The example below means that uran type 1 will have the value 0.1 . The alpha parameter will be used in the exponent for the repulsion.
	`ALPHa`
	`92 1 0.1`
	`END`

KAPPa

Use this Keyword to define the kappa parameter for a specific atom and atomtype. The keyword should be followed by a line/lines composed of the element number, the atomtype and the value for kappa.This Keyword should be ended by a END statement in the last line. The example below means that uran type 1 will have the value 10.0 . The kappa parameter will be used as a prefactor to the exponent expression for the repulsion.

```
KAPPa
92 1 10.0
END
```

ALCT

This keyword is for the charge transfer term that can be used if one specifies that in the NEMO keyword. The energy term is exactly the same expression as the repulsion, but with a minus sign instead. Use this Keyword to define the charge transfer alpha parameter for a specific atom and atomtype. The keyword should be followed by a line/lines composed of the element number, the atomtype and the value for charge transfer alpha.This Keyword should be ended by a END statement in the last line. The example below means that uran type 1 will have the value 0.1 . The charge transfer alpha parameter will be used in the exponent for the repulsion.

```
ALCT
92 1 0.1
END
```

KACT

This keyword is for the charge transfer term that can be used if one specifies that in the NEMO keyword. The energy term is exactly the same expression as the repulsion, but with a minus sign instead. Use this Keyword to define the charge transfer kappa parameter for a specific atom and atomtype. The keyword should be followed by a line/lines composed of the element number, the atomtype and the value for charge transfer kappa. This Keyword should be ended by a END statement in the last line. The example below means that uran type 1 will have the value 10.0 . The charge transfer kappa parameter will be used as a prefactor to the exponent expression for the repulsion.

```
KACT
92 1 10.0
END
```

REPFactor

If a repulsion of type sqrt(Factor1*Factor2)r-̂n is to be used. Check the NEMO keyword for information. This keyword is specified in the same way as kappa.

DISPFactor

Two factors are multiplied with the dispersion energy. They work in the same way as the REPFactor does and are specified in the same way.

VALEnce

Set the number of valence electrons. The keyword should be followed by a line/lines composed of the element number, the atomtype and the value for kappa.This Keyword should be ended by a END statement in the last line. The example below means that oxygen type 2 will have 6 valence electrons.

```
VALEnce
8 2 6.0
END
```

NOISOTROPICPOLARIZABILITIES The default is to use isotropic polarizabilities for the induction energy. This is due to the fact that we use Thole damping as default, which require isotropic polarizabilities.

NOMOVE The default interactions sites are not placed in the atoms. If this keyword is used the interactions sites are not moved to a new location.

NOQUADRUPOLEDELETE The default is to replace the quadrupoles with local dipoles to get the correct total quadrupole. If this keyword is used, the quadrupoles will be truncated at the dipole level.

NODAMPING As default the Thole damping is used, but using this heyword that is overruled.

REPLACE Use this keyword to specify that some atomic quadrupoles should be replaced by charges.

MOLD The new local atomic dipole will be used when calculating the new interaction center. The default is to use the original local atomic dipole.

NOLM The new local atomic quadrupole will be used when estamating the size of the atom. This is used when calculating the repulsion and dispersive energy. The default is to use the original local atomic quadrupole which is the correct way.

RETY REpTYpe: The keyword should be followed by a line, specifying the expression to use for the repulsion type.

Optional RETY paramters

- **m=0** (Default) Here the exponent is described by -r12/sqrt(Tr(Q1)/3/qv1+Tr(Q2)/3/qv2)/(aplha1+alpha2) .
- **m=1** Here the exponent is described by -r12/(alpha1*sqrt(Tr(Q1)/3/qv1)+alpha2*sqrt(Tr(Q2)/3/qv2)) .

NEMO The keyword should be followed by a line, what kind of energy expression to use. The parameters for the energies are read from the *nemo* and *ATOMPAR*

Optional NEMO paramters

- **m=0** (Default) Electrostatic, inductive, dispersive and a exponetial repulsion energy term is used.
- **m=1** Here a sqrt(Factor1*Factor2)*r^-n type repulsion is added to the default energy.
- **m=2** Here dispersion factors are used to scale the energy.
- **m=3** This number means that default energy is used, plus the repulsive term of type 1 and the dispersive scaling of type 2.
- **m=4** An charge transfer term is added to the default energy, which has the same expression as the repulsion term only differing in the sign.

SEED — The seed to the random generator.

FITPar — This is the start keyword for the subprogram `FITPAR`. It should consist of the Keyword plus a END statement. Inbetween there should be `FITPAR` specific keywords. The subprogram to do the fitting of parameters.

DIMEr — This is the start keyword for the subprogram `DIMER`. It should consist of the Keyword plus a END statement. Inbetween there should be `DIMEr` specific keywords. The subprogram do an energy minimisation for two monomers.

POTSurf — This is the start keyword for the subprogram `POTSURF`. It should consist of the Keyword plus a END statement. Inbetween there should be `POTSURF` specific keywords. The subprogram generates potential curves.

SIMPar — This is the start keyword for the subprogram `SIMPAR`. It should consist of the Keyword plus a END statement. Inbetween there should be `SIMPAR` specific keywords.

Optional FITPar specific keywords

These keywords should begin by a FITPar keyword and end with a END statement.

Keyword — *Meaning*

NUAL — NO UPDATE ALPHA. This keyword should be followed by a line/lines specifying the element and type of the atomic parameter that should not be updated during the fitting. The example says that the oxygen type 2 atomic parameter should not be updated.

```
NUAL
8 2
END
```

NUKA NO UPDATE KAPPA. This keyword should be followed by a line/lines specifying the element and type of the atomic parameter that should not be updated during the fitting. The example says that the oxygen type 2 atomic parameter should not be updated.

```
NUKA
8 2
END
```

NUAC NO UPDATE CHARGE TRANSFER ALPHA. This keyword should be followed by a line/lines specifying the element and type of the atomic parameter that should not be updated during the fitting. The example says that the oxygen type 2 atomic parameter should not be updated.This only works for NEMO type 4. Check the NEMO keyword.

```
NUAC
8 2
END
```

NUKC NO UPDATE CHARGE TRANSFER KAPPA. This keyword should be followed by a line/lines specifying the element and type of the atomic parameter that should not be updated during the fitting. The example says that the oxygen type 2 atomic parameter should not be updated.This only works for NEMO type 4. Check the NEMO keyword.

```
NUKC
8 2
END
```

NUSI Not implemented. For future use.

NUEP Not implemented. For future use.

NURE NO UPDATE REPULSION FACTOR. This keyword should be followed by a line/lines specifying the element and type of the atomic parameter that should not be updated during the fitting. The example says that the oxygen type 2 atomic parameter should not be updated. This only works for NEMO type 1 and 3. Check the NEMO keyword.

```
NUKC
8 2
END
```

NUDI NO UPDATE DISPERSION FACTOR. This keyword should be followed by a line/lines specifying the element and type of the atomic parameter that should not be updated during the fitting. The example says that the oxygen type 2 atomic parameter should not be updated. This only works for NEMO type 2 and 3. Check the NEMO keyword.

```
NUKC
8 2
END
```

GLOBal The keyword should be followed by a line specifying the number of globalsteps.

MACRo The keyword should be followed by a line specifying the number of macrosteps.

MICRo The keyword should be followed by a line specifying the number of microsteps.

TEMP The keyword should be followed by a line specifying the temperature for the weighting procedure. See the keyword WEIG.

SCFFit By default the program tries to fit the second energy term in the NEMO file. Using this keyword the program uses the first energy term witch is a SCF type energy.

CONVergence The keyword should be followed by a line specifying the number for the convergence radii.

RFACtor The keyword should be followed by a line specifying the number for the scaling constant in the least square fit.

WEIGht The keyword should be followed by a line specifying the number of the weight type

Optional WEIGht paramters

- **m=0** (Default) Weight=Min(2,Exp(-0.2*(E(dimer)-E(Monomer1)-E(Monomer2)))
- **m=1** Weight=exp(-(E(dimer)-E(Monomer1)-E(Monomer2))/kT)

ERROr The keyword should be followed by a line specifying the number of the error type

Optional ERROr paramters

- **m=0** (Default) Error=Weight*(Exp(0.15D0*(E(estimated)-E(reference)))-1)**2
- **m=1** Error=Weight*(E(reference)-E(estimated))**2

DISFactor The keyword should be followed by a line specifying a scaling constant for the dispersion energy. (Default 1.0)

LINEarsearch The keyword can contain any of the keywords FORCe, SIMPlex, ITERation and CONVergence. It should also finnish by an END statement.

SIMPlex Keyword for the simplex method.

FORCe Keyword for a steepest descent type method.

ITERation	The keyword should be followed by a line specifying the number of interations.
CONVergence	The keyword should be followed by a line specifying the number for the convergence.

Optional DIMEr specific keywords

These keywords should begin by a DIMEr keyword and end with a END statement.

Keyword	*Meaning*
MOLEcules	The keyword should be followed by a line specifying a molecule by name exactly as they are named in the nemo file. All other molecular based keywords will be given to this molecule. That until a new molecule name is given with this keyword.
METHod	Specifies the method to be used for the file to be opened. The program will find another method if the specified method cannot be found in the MPPROP file.
MACRosteps	The keyword should be followed by a line specifying the number of macrosteps.
MICRosteps	The keyword should be followed by a line specifying the number of microsteps.
STARt	The keyword should be followed by a line specifying two numbers. The first number is search radii for coordinates and the second number is the search radii for the angles. In the first macrostep.
RFACtor	The keyword should be followed by a line specifying the number of the scaling factor for the search radii each macrostep.
CONVergence	The keyword should be followed by a line specifying the number for the convergence radii.
DISFac	The keyword should be followed by a line specifying a scaling constant for the dispersion energy. (Default 1.0)

Optional POTSurf specific keywords

These keywords should begin by a POTSur keyword and end with a END statement.

Keyword	*Meaning*
MOLEcule	Specifies the start and the title of a new molecule. This means every keyword after this MOLEcule keyword will belong to the last specified MOLEcule.

METHod Specifies the method to be used for the file to be opened. The program will find another method if the specified method cannot be found in the MPPROP file.

CLUSter This keyword should be followed by a line that gives an integer number of witch cluster the lates molecule belongs to. Only the integer numbers 1 and 2 are valid for the PotSurf module.

TROR This keyword should be followed by a line that gives six numbers. The six numbers describes the translation in polar coordinates and the rotation in the three euler angles for the molecule given by the latest MOLEcule keyword. The sequence of the numbers are the following: R Theta Phi Alpha Beta Gamma (See Arfken for definitions)

POTEntial The keyword should be followed by one line specifying three numbers. The numbers gives the displacement vector in spherical poolar coordinates for the second cluster when calculating the potential energy. The numbers are given in the following order: R Theta Phi

NPOInts The keyword should be followed by one line specifying the number of points in the potential.

TRANslation The keyword should be followed by a line specifying up to five numbers. The first number specifies the type of potential coordinates. In order to visulize the potential curve one has to define a translation coordinate. The first column of the PotSurf file will consist of a coordinete specified by the iTrType parameter. The other parameters jTrType, kTrType ... are specified below.

Optional TRANslation paramters

- **iTrType=0** The coordinate will be the length of the translation vector. (Default)
- **iTrType=1** jTrType=coordinte (1=X,2=Y and 3=Z) index of kTrType=molecule given by the order of the apperence in the input section.
- **iTrType=2** jTrType=Atom1 and kTrType=Atom2 on molecule=lTrType and mTrType respectively. The molecules are given by the order of the apperence in the input section. The potential coordinate will be the distance between Atom1 and Atom2. Note that if the potential coordinate is constant if the molcules belong to the same cluster.

DISFactor The keyword should be followed by a line specifying a scaling constant for the dispersion energy. (Default 1.0)

Optional SIMPar specific keywords

These keywords should begin by a DIMEr keyword and end with a END statement.

Keyword	*Meaning*
MOLEcules	Specifies the start and the title of a new molecule. This means that every keyword after this MOLEcule keyword will belong to the last specified MOLEcule.
METHod	Specifies the method to be used for the file to be opened. The program will find another method if the specified method cannot be found in the MPPROP file.
MOLSim	Tells the program to generate Molsim parameters and input files.

EQUAlatoms	This keyword should be followed by a line specifying two atom numbers that should treated as equal. The atomic numbers are the numbers in sequence as they are found in the MPPROP file. For example, a water molecule in gasphase has the two hydrogen atoms equal by symmetry. They should thus be treated equally for the analysis in a simulation program. If the MPPROP file has the atoms in the sequence O H H the example below makes the two hydrogen equal
	`EQUA`
	`2 3`

NUMBer	The keyword should by a line giving the number of latest molecule that will be used in the latter simulation. This information will be written in the MOLSIM file.
DISFactor	The keyword should be followed by a line specifying a scaling constant for the dispersion energy. (Default 1.0)

Limitations

The program package has no internal degrees of freedom. The program cannot handle interactions including quadrupoles and higher. The program cannot handle hyperpolarizabilities. For the time being we cannot handle more than two clusters.

8.36 NUMERICAL_GRADIENT

The NUMERICAL_GRADIENT module is a program which numerically evaluates the gradient of the energy with respect to nuclear perturbations.

Note that this module is automatically invoked by the ALASKA module if the wave function method is MBPT2, CCSDT, CASPT2, MS-CASPT2, or a calculation using the Cholesky decomposition. The user should normally never request the execution of this module; instead it is advised to use the NUMErical keyword in Alaska, if it is necessary to force the use of numerical gradients rather than analytical ones.

The module is parallelized over the displacements, which in case of large jobs gives a linear speed up compared to a serial execution, although in order to obtain this it is important to choose the number of nodes such that the number of contributing perturbations is a multiple of the number of nodes. For a given molecule the number of perturbations equals the number of atoms times 6 (a perturbation with plus and minus delta for each of the three axes). Symmetry can of course reduce this number. If the request of execution originates from the SLAPAF module further reduction in perturbations is achieved due to the utilization of rotational and translational invariance.

8.36.1 Dependencies

The dependencies of the NUMERICAL_GRADIENT module is the union of the dependencies of the SEWARD, SCF, RASSCF, MBPT2, MOTRA, CCSDT, and CASPT2 modules.

8.36.2 Files

The files of the NUMERICAL_GRADIENT module is the union of the files of the SEWARD, SCF, RASSCF, MBPT2, MOTRA, CCSDT, and CASPT2 modules.

8.37 POLY_ANISO

This module is still undocumented

8.38 QMSTAT

8.38.1 Description

QMSTAT couples a quantum chemical region to a statistically mechanically described surrounding thus creating an effective Hamiltonian for the quantum chemical region $H_{eff.}$. This way solvent effects can be accounted for.

The surrounding is discrete in contrast to the continuum models, such as PCM (also available in MOLCAS see SEWARD). The explicit representation of the solvent enables a more accurate description of the solvation, but also makes the model more complex and significantly less “black-box”. For example, the interaction within the statistical mechanical surrounding has to be accounted for with an accurate enough force-field. In its present implementation QMSTAT only treats water as described by an early version of NEMO, which includes polarization of the molecules [72]. Also, the interaction between the quantum chemical region (typically the solute) and the surrounding (typically the solvent) has to be considered in more detail than in a continuum model. Several approaches to discrete (or explicit) solvation are thus possible. The approach in QMSTAT is summarized below, see also [73, 74, 75, 76].

To include entropic effects to the solvation phenomena, QMSTAT uses the Metropolis–Monte Carlo simulation technique. This means that random steps are taken in the space of solute–solvent configurations, some of which are accepted, others rejected, on account of the usual energy difference criteria. This implies that at each step, an energy has to be evaluated. Using normal quantum chemical methods, this is usually too restrictive, since roughly one million Monte Carlo steps are required to converge the statistical mechanical treatment. QMSTAT proceeds by doing simplifications to the quantum chemistry, not the statistical mechanics, as is the more common way forward. QMSTAT is therefore a *hybrid* QM/MM methods (according to one existing terminology).

Two simplified quantum chemical methods are presently available: orbital basis Hartree–Fock and a state basis formulation, which is approximate to the CASSCF method. Both formulations uses the fact that there is only minor differences in the electronic structure for the different configurations in the Monte Carlo simulation. Therefore, a basis as general as the standard atomic orbital basis sets is redundant. QMSTAT constructs either a more compact orbital basis or a more compact basis in terms of states to expand the solvated wave function. This requires some work before the simulation, but has the advantage that during the simulation, less computational work is needed.

Finally, a comment on the way the energy is computed for a given configuration is needed. The evaluation of the interactions between the solvent molecules is prescribed by the construction of the force-field and are relatively simple. The interaction between the quantum chemical region and the solvent is formulated to include electrostatic and non-electrostatic interactions. The former is described in a multi-center multipole expanded way, while the latter models the effect the antisymmetry principle between solute and solvent electrons has on the solute electronic structure. Its formulation is similar to pseudo-potentials. Also a phenomenological term for the dispersion is added. Long range electrostatics, finally, is described with a dielectric cavity model.

8.38.2 Dependencies

The dependencies of QMSTAT differ for the two quantum chemical methods. In the Hartree–Fock description, SEWARD, FFPT, SCF, AVERD, MPPROP and MOTRA typically have to precede. If an orbital basis is taken from somewhere else FFPT, SCF and AVERD are not mandatory. For the RASSI alternative, typically SEWARD, SCF, RASSCF, MPPROP and RASSI precede QMSTAT.

8.38.3 Files

Below is a list of the files that are used/created by the program QMSTAT.

Input files

File	*Contents*
ONEINT	One-electron integral file generated by the program SEWARD.
RUNFILE	File for communication of auxiliary information generated by the program SEWARD.
RUNFILEW	File for communication of auxiliary information generated by the program SEWARD for the solvent molecule.

AVEORB	(Only for Hartree–Fock alternative). Average orbitals generated by AVERD. If other orbitals are to be used, they should be given the above name; in other words, the orbitals must not be created by AVERD, it is only customary.
SOLORB	Solvent orbitals generated by SCF.
TRAONE	(Only for Hartree–Fock alternative). Molecular orbital transformed one-electron integrals generated by MOTRA.
TRAINT	(Only for Hartree–Fock alternative). Molecular orbital transformed two-electron integral generated by MOTRA.
MPPROP	File generated by MPPROP.
DIFFPR	Exponents and Prefactors for a Slater desciption of the Electrostatics to take into account the penetration effects due to the overlap.File generated by MPPROP.
RASSIM	(Only for the RASSI alternative). The transition density matrix generated by RASSI. The keyword TOFILE has to be given in the input to RASSI.
EIGV	(Only for the RASSI alternative). Information about the eigenvectors and their energy generated by RASSI (TOFILE needed).
*ADDON**	File with additional one-electron perturbation to be added to the Hamiltonian matrix. This file is only required if EXTERNAL is used.

Output files

File	*Contents*
*STFIL**	Start files in which solvent configurations are stored at intervals during the simulation. They enable the simulation to restart, hence they can also be as input to Qmstat.
*SAFIL**	Sampling files in which a selection of configurations are stored for analysis. They can in some applications also act as input to Qmstat, usually in free-energy perturbation calculations.
*EXTRA**	Extract files which are formatted files in which data from the analysis of the sampling files are stored.

8.38.4 Input

The complexity inherit in a discrete solvent model engenders a, potentially, complex input. To (hopefully) make the input transparent the set of keywords are ordered in several tiers. Below all keywords and their sub- and subsubkeywords are presented. A keyword with several tiers should typically be of the following form

```
SIMUlation
...(keywords on higher tier)
END simulation
```

Also consult the input example below and the examples in section 10.6 for guidance. Mandatory keywords are highlighted.

Keyword	*Meaning*
TITLe	Title to the calculation.
SIMUlation	Keywords relating to the how the simulation is to be performed and under which conditions. • **RADIus** Initial radius of the dielectric cavity. The radius is also specified on the startfile and has higher priority than the radius given with the present keyword. • **PERMittivity** Permittivity of the dielectric continuum. 80 on default. • **TEMPerature** Temperature in Kelvin. Default is 300. • **PRESsure** Macroscopic pressure in atmosphere. Default is 1 atm. • **SURFace** Surface tension parameter for the cavity. Default is for air-water interface. • **TRANslation** Maximal translation in the simulation (in a.u.)Default is 0.0 a.u. • **ROTAtion** Maximal angle for rotation of solvent around molecular axes. Default is 0^o. • **CAVIty** Maximal modification of radius of dielectric cavity. Default is 0.0 a.u. • **FORCe** Force constant for the harmonic potential that presents a bias in the simulation for configurations with the QM-region close to the center of the cavity. Default is 0.001. • **BREPulsion** Parameter for the Repulsion energy that keeps the QM-region away from the boundary. Default is 0.0 a.u. • **SEED** Seed to the pseudo-random number generator. • **PARAlleltemp** A parallel tempering procedure is performed to boost sampling. It is mainly used in systems with small transition elements in the Markov chain, which will give difficult samplings. Three lines follow: First line gives the number

of different temperatures to perform the simulation, $NTemp$. In the second line $Ntemp$ integers are given, each of these specify a file to store the configuration for each temperature. Third line gives the $NTemp$ temperatures used for the tempering procedure.

- **END_Simulation Parameters** Marks the end of the input to the simulation parameters.

THREshold Followed by three numbers. First the threshold for the induced dipoles in the generalized self-consistent field method for the solution of the mutual polarization problem is specified. Second the the threshold for the energy in the same method is given. Finally the maximum number of iterations in the method is specified. Defaults are 0.0001 0.0000001 and 30.

STEPs Followed by two entries. Number of macrosteps and number of microsteps. The total number of steps is the product of the two numbers above. At the end of each macrostep the relevant STFIL is up-dated. Default is 1 and 1.

RUN Specify type of simulation. 'QMEQ' means quantum chemical equilibration; only the startfile is up-dated. 'QMPR' means quantum chemical production; startfile is up-dated and sampfile constructed. **Observe** that if 'QMPR' is specified a line with two entries follows in which the interval of sampling is specified and on which sampfile (1-7) the data is to be stored. 'ANAL' means an analysis of the stored results is to be performed.

PRINt Print level. 1 is default and anything above this number can generate large outputs. No higher than 10 is recommended for non-developers.

EXTErnal An external perturbation is to be added to the Hamiltonian in the Rassi alternative. The arguments are number of perturbation matrices, N, followed by N lines. Each line has the form: c_i a scalar with which the perturbation can be scaled, V_i is a character string with the label of the perturbation as given on SEWARD's one-electron integral file, nc_i is the component number of the perturbation. A final expression for the perturbation would be: $c_1V_1(nc_1) + c_2V_2(nc_2) + \cdots + c_NV_N(nc_N)$.

CONFiguration Keywords relating to from which source the initial solvent configuration is to be obtained. **It is mandatory to specify a source**.

- **ADD** Followed by one number specifying how many solvent molecules that are to be added at random to the cavity. This is the worst way to start a simulation since it will take a lot of time to equilibrate the system.
- **FILE** Signify that start configuration is to be read from some file.
  - **STARtfile** Read solvent configuration from startfile.

    - **SCRAtch** Read solvent configuration from startfile and place the QM-region as given on RUNFILE.
    - **COPY** Read solvent and QM configuration from startfile. This is he keyword to use if a simulation is to be restarted. **Observe** that consistent startfile and RUNFILE must be used.
    - **CM_ _** Read solvent configuration from startfile and place the QM in the center of mass of the QM placed on startfile. For any of the previous keywords two numbers are given, N_{in} and N_{out} which specify from which startfile QMSTAT is supposed to read and write, respectively
  - **SAMPfile** Read solvent configurations put on a sampfile and analyze them. Two numbers are given, N_{in} and N_{extr} which specify from which sampfile QMSTAT is supposed to read and on which extract file the results are to be put.
- **INPUt** The starting configuration is to be read from the input. The coordinates are given after the keyword COORdinates in the second tier to the SOLVent keyword. One number as argument: the startfile to which configurations are written.
- **END_Configuration** Marks the end of the input to the initial configuration.

EDIT Signify that a startfile is to be edited. If this keyword is given, then no simulation will be performed.

- **DELEte** Two rows follow; on the first N_{in} and N_{out} are given which specify the startfile to read from and write to, respectively; on the second the number of solvent molecules to delete. The solvent molecules farthest away from origin are deleted.
- **ADD** The form of the arguments as DELEte above, only the second row give number of molecules to add. **Observe** that the keyword RADIus will with the present keyword specified give the radius of the cavity of the edited startfile.
- **QMDElete** Delete the QM-region and substitute it by water molecules. One row follows with two numbers, which specify the startfile to read from and write to, respectively.
- **DUMP** Dump startfile coordinates in a way suitable for graphical display. Two rows follow; on the first a character string with the format the coordinated will be dumped; on the second N_{in} specifies the startfile to read. Currently there is only one format for this keyword: MOLDen.
- **END_EditStartFile** Marks the end of the input to edit the startfile.

QMSUrrounding Keywords that are related to the interaction between surrounding and the quantum chemical region.

- **DPARameters** Parameters for the dispersion interaction. Follow N lines, which N the number of atoms in the QM-region. The general form for each line is: $d1$ and $d2$ where $d1$ is the dispersion parameter between one atom of the QM-region and the water oxygen, and $d2$ is the same but regarding to the hydrogen of the water.The order of the QM atoms is given by RUNFILE.
- **ELECtrostatic** Parameters to describe the electrostatic penetration using Slater integrals.
  - **THREsholds** Two number follow. First, the cutoff (distance Quantum Site-Classical molecule) to evaluate penetration effects. Default is 6 a.u. Second, difference between two Slater exponents to not be consider the same value. Default is 0.001.
  - **NOPEnetration** No electric penetration is considered in the calculations. Penetration is considered by default.
  - **QUADrupoles** Electrostatic Penetration computed in quadrupoles. Default is that penetration is computed up to dipoles.
  - **END Electrostatic** Marks the end of the input to the electrostatic penetration computed by Slater.
- **XPARameters** Parameters to describe the repulsion energy.
  - **S2** The parameter for the $\sim S^2$ term. Default zero.
  - **S4** The parameter for the $\sim S^4$ term. Default zero.
  - **S6** The parameter for the $\sim S^6$ term. Default zero.
  - **S10** The parameter for the $\sim S^10$ term. Default zero.
  - **CUTOff** Two numbers follow. The first is the cut-off radius such as if any distance from the given solvent molecule is longer than this number, the overlap term is set to zero. The second is a cut-off radius such as if any distance from the given solvent molecule is shorter than this number the energy is set to infinity, or practically speaking, this configuration is rejected with certainty. Defaults are 10.0 a.u. and 0.0 a.u.
  - **END XParameters** Marks the end of the input to the repulsive parameters.
- **DAMPing**
  - **DISPersion** Input parameters to a dispersion damping expression. The parameters are number obtain from a quantum chemical calculation. All lines have the form: C_{val}, Q_{xx}, Q_{yy}, Q_{zz} where C_{val} is the valence charge and Q_{**} are diagonal terms in the quadrupole tensor. First two lines are for the hydrogen atom then the oxygen atom in a water molecule. Next follows as many lines as atoms in the QM region. All these quantities can be obtained from a calculation with `MPPROP`. The numbers are given as input so that the user can if it is found to be needed, modify the damping. Default is no damping. The order of the atoms in the QM region is given by RUNFILE.

  - **FIELd** The electric field between QM region and surrounding is damped. Three numbers are arguments:C_O, C_H, N where they are parameters to a field damping expression ($E = \tilde{E}(1 - e^{C_x R})^N$) where x is O if the point in the surrounding is on a oxygen atom, H if on a hydrogen atom; R is the distance between the point in the QM region and the points in the surrounding.
  - **END Damping** Marks the end of the input to the Damping parameters.

- **END QmSurrounding** Marks the end of the input related to the interaction between surrounding and the quantum chemical region.

SOLVENT

Keywords that govern the solvent-solvent interaction and some other initial data. Most of these numbers are presently fixed and should not be altered.

- **COORdinates** If solvent coordinates are to be given explicitly in input. First line gives number of particles to add. Then follows three times that number lines with coordinates for the oxygen atom and the hydrogen atoms. If the keyword SINGLE-POINT has been given the present keyword assumes a different meaning (see description of SINGLE-POINT).
- **CAVRepulsion** Two parameters that regulate the repulsion with the boundary of the cavity. Defaults are 30.0 and 0.06.
- **ATCEchpol** Five numbers follow: number of atoms, centers, charges, polarizabilities and slater sites. Defaults are 3, 5, 4, 3 and 5, respectively.
- **CHARge** Four numbers follow: the partial charge on the hydrogen atoms and the partial charge on the pseudo-centers.
- **POLArizability** Three numbers follow: the polarizability on the oxygen atom and on the two hydrogen atoms.
- **SLATer** Magnitude of Slater Prefactors and exponents. One mumber follow: 0 is slater description of electrostatics up to charges, 1 up to dipoles. Then it follows N times (where N is the number of Slater centers) three lines if description up to charge. First line Slater exponent for charges, second line Slater Prefactor and third line nuclear charge of the center. If the description goes up to dipole, N times five lines follows. First two lines are the same as charge description, third line is Slater exponent for dipole, fourth line is the three Slater Prefactors for the dipole (one for each cartesian coordinate) and fith line is the nuclear charge of the center. Defaults: See papers of Karlstrom. If the number of Slater sites is modified this keyword should be after ATCECHPOL
- **END Solvent** Marks the end of the input that govern the solvent-solvent interaction.

RASSISECTION

This section provides the information needed to perform QMSTAT calculations using the RASSI-construction of the wave function.

- **JOBFiles** First number give the number of JOB-files that was generated by `RASSCF` (*i. e.* how many RASSCF calculations that preceded `QMSTAT`). The following numbers (as many as the number of JOB-files) specify how many states each calculation rendered. So for example if a State-Average (SA) RASSCF calculation is performed with two states, the number should be 2.
- **EQSTate** Which state interacts with the surrounding. Should be 1 if it is the ground state, which also is the default.
- **MOREduce** A Reduction of the Molecular Orbitals is performed. One number as argument: the threshold giving the value of the lowest occupation number of the selected natural orbitals [75].
- **CONTract** The RASSI state basis are contracted. One number as argument: the threshold giving the value of the lowest RASSCF overlap for the RASSI state basis [75].
- **LEVElshift** Introduce levelshift of RASSI states. Three lines must be written. First line gives the number of levelshifts to perform. Then follows the states to levelshift (as many as the number of levelshifts). Finally, the value of the levelshift for each state is given.
- **CISElect** The QM solvent overlap is used as the criterion to choose the state that interacts with the surrounding. Three lines follow. One entire: among how many states can be chosen the interacting state, N. The second line, N entries giving the number of each state. Finally, N scaling factors, one for each state, of the overlap.
- **END RassiSection** Marks the end of the input that govern the Rassi calculations.

SCFSECTION

This section provides additional information to perform QMSTAT calculations using the SCF-construction of the wave function.

- **ORBItals** Two numbers are required: how many orbitals that are to be used how many occupied orbitals there are in the QM region. as a basis in which to solve the Hartree-Fock equation, and
- **END ScfSection** Marks the end of the input that govern the Scf calculations.

SINGLE-POINT

This keywords signals that a set of single point calculations should be performed; this is typically what one needs when fitting parameters. The keyword gives the COORDINATES keyword in the SOLVENT section a new meaning. The first row then gives the number of points in which a single-point calculation should be performed and the coordinates that follow give the coordinates for the water monomer. QMSTAT then run each solute-monomer solvent configuration specified and the energy (among other things) is computed. The keyword thus overrides the usual meaning of the input. **Observe** that the permittivity has to be set to 1 if one attempts to reproduce a quantum chemical supermolecular potential.

EXTRact Section Give details about the analysis performed to the results stored in the sampfile.

- **TOTAl energy** The total energy of the whole system is extracted.
- **DIPOle** The three components and the total dipole of the QM-region are extracted.
- **QUADrupole** The six components and the quadrupole of the QM-region are extracted.
- **EIGEn** The Eigenvalues of the RASSI matrix and the eigenvectors are extracted. Follow by a number and a ”YES” or ”NON” statement. The number gives the highest state where the eigenvalue is extracted. YES means that also the corresponding eigenvectors are extracted.
- **EXPEctation values** The expectation values of H_0 and main perturbations: V_{el}, V_{pol} and V_{n-el} are extracted. If keyword EIGEn is specified it is done for the same states as this keyword, otherwise the extraction is performed for the equilibrated state. **Observe** that the expectation values are for the final wave function of the QM-region in solution, so H_0 is not the same as for the isolated QM-region.
- **ELOCal** The local expectation values of V_{el} and V_{pol} for the multipole expansion sites are extracted. Two lines follow. First, gives for how many sites these values will be extracted, N. Second line, N entries giving the number of each site. If keyword EIGEn is specified the extraction is done for the same states as this keyword, otherwise it is performed for the equilibrated state.
- **MESP** The Main Electrostatic potential, field and field gradients will be obtained in order to produce perturbation integrals that will be used to optimize the intramolecular geometry of the QM system. **Observe** that this keyword will change the one electron integrals file, so it is advised to make a copy of the original file. After running this option ALASKA and SLAPAF must be running with the new one electron integrals file in order to produce the gradients and a new geometry in the geometry optimization procedure.
- **END ExtractSection** Marks the end of the input that give details about the analysis performed.

Input example

The following input uses the Rassi alternative and restarts from startfile.0 and write to startfile.1 every 1000th step, where the total number of steps is 200*1000. A set of parameters are given which are for an organic molecule with one carbon, one oxygen and two hydrogen atoms. The order in the previous SEWARD and RASSCF calculations for the atoms is carbon, oxygen, hydrogen 1 and hydrogen 2. The dispersion is damped. Finally, there are sixteen RASSCF calculations preceeding and the last two are state-average since two states are collected from these files; the ground state interacts with the surrounding.

```
 &QmStat &End
Simulation       * Simulation parameters.
Translation
0.03             * Maximun translation step of water.
Rotation
1.0              * Maximun rotation step of water.
Cavity
0.05             * Maximun variation of the cavity radius for step.
End Simulation
Steps            * Number of macro and microsteps.
200 1000
Configuration    * How the start configuration is readed.
Start            * The cordinates are taken form a startfile.
Copy             * The coordinates of the QM region are the same as in the startfile.
0 1
End Configuration
QmSurrounding
DParameters      * Dispersion parameters.
35.356 4.556     * Carbon_QM-Oxygen_wat     Carbon_QM-Hydrogen_wat.
16.517 2.129     * Oxygen_QM-Oxygen_wat     Oxygen_QM-Hydrogen_wat.
10.904 1.405     * Hydrogen1_QM-Oxygen_wat  Hydrogen1_QM-Hydrogen_wat.
10.904 1.405     * Hydrogen2_QM-Oxygen_wat  Hydrogen2_QM-Hydrogen_wat.
XParameters      * QM-Solvent Repulsion Parameters.
S2
-0.375
S6
1.7
End XParameters
Damping          * Dispersion Damping.
Dispersion
-6.64838476  -5.22591434  -4.32517889 -4.58504467     * Water Hydrogen.
-.34146881   -0.21833165  -0.22092206 -0.21923063     * Water Oxygen.
-4.23157193  -1.91850438  -2.28125523  -1.91682521    * Quamtum Carbon.
-6.19610865  -3.90535461  -4.73256142  -3.77737447    * Quantum Oxygen.
-.57795931   -0.42899268  -0.43228880  -0.43771290    * Quantum Hydrogen 1.
-.57795931   -0.42899268  -0.43228880  -0.43771290    * Quantum Hydrogen 2.
End Damping
End QmSurrounding
RassiSection
JobFiles          * Number of JobFiles.
16
1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2      * One state is collected form all JobFiles
*                                      except from the two last ones, which two
*                                      are collected.
EqState           * The state interacting with the surrounding.
1
End RassiSection
End of Input
```

8.39 QUATER

8.39.1 Dependencies

The QUATER is free-standing and does not depend on any other program.

8.39.2 Files

Input files

The calculation of vibrational wave functions and spectroscopic constants uses no input files (except for the standard input).

8.39.3 Input

This section describes the input to the QUATER program in the MOLCAS program system. The program name is

```
&QUATER
```

Keywords

Keyword	*Meaning*
NOROTATION	No rotation is performed by the program. Only the rotation matrix is printed out.
NOTRANSLATION	No translation is performed by the program.
DEBUG	Turn on DEBUG printout
AXIS	Define the old frame of reference
NEWAXIS	Define the new frame of reference
GEO1	Define the first geometry
GEO2	Define the second geometry
XYZ1	Define the origin and two axes for the orientation of the first geometry by the index of three atoms of this geometry.
XYZ2	Define the origin and two axes for the orientation of the second geometry by the index of three atoms of this geometry.
END	End of input

QUATER will perform a vib-rot analysis and compute spectroscopic constants.

Input example

```
&QUATER
GEO1
   19
titre
 C     0.000000     0.000000     0.000000
 O     0.000000     0.000000     1.400000
 H     0.895670     0.000000     1.716663
 C    -0.683537    -1.183920    -0.483333
 H    -0.513360     0.889165    -0.363000
 C    -0.683537    -1.183920    -1.933333
 H    -0.170177    -2.073085    -0.120333
 H    -1.710256    -1.183920    -0.120333
 C     0.683537    -1.183920    -2.416667
 H    -1.196896    -2.073085    -2.296333
 H    -1.196896    -0.294755    -2.296333
 C     1.367073     0.000000    -1.933333
 H     1.196896    -2.073085    -2.053667
 H     0.683537    -1.183920    -3.505667
 C     1.367073     0.000000    -0.483333
 H     2.393792     0.000000    -2.296333
 H     0.853714     0.889165    -2.296333
 H     1.880433    -0.889165    -0.120333
 H     1.880433     0.889165    -0.120333
END
GEO2
   23
titre
 C     0.000000     0.000000     0.000000
 H     0.000000     0.000000     1.089000
 C     1.367075     0.000000    -0.483328
 H    -0.334267    -0.970782    -0.363000
 C     1.367081     0.000000    -1.933328
 H     1.880433     0.889165    -0.120326
 H     1.880433    -0.889165    -0.120326
 C     0.683546     1.183920    -2.416664
 H     2.393801     0.000000    -2.296324
 H     0.853722    -0.889165    -2.296330
 C    -0.683529     1.183920    -1.933336
 H     1.196904     2.073085    -2.053662
 O     0.683551     1.183920    -3.816664
 C    -0.683535     1.183920    -0.483336
 H    -1.196887     2.073085    -2.296338
 H    -1.196887     0.294755    -2.296338
 O    -0.023570     2.327015    -0.016667
 H    -1.710255     1.183920    -0.120340
 H     0.237132     1.957142    -4.132332
 C    -0.023576     2.327015     1.383333
 H     0.489783     3.216180     1.746335
 H    -1.050296     2.327015     1.746329
 H     0.489783     1.437850     1.746335
END
XYZ1
15 12 9
XYZ2
11 14 1
END
```

This input will perform the alignment of the second geometry (GEO2) on the first one (GEO1). Atom number 11 (C11) of the second geometry will be moved to the position of atom number 15 of the first geometry (C15). The vector C11 C14 in GEO1 will be aligned with the vector C15 C12 of GEO1. Finally the plane 11 14 1 of GEO1 will be aligned with the plane 15 12 9 of GEO2.

8.40 RASSCF

The RASSCF program in MOLCAS performs multiconfigurational SCF calculations using the Restricted Active Space [77] or the Generalized Active Space [10] SCF construction of the wave function. RASSCF is an extension of the Complete Active Space (CAS) approach, in which the wave function is obtained as a full CI expansion in an active orbital space [78, 79]. The RASSCF method is based on a partitioning of the occupied molecular orbitals into the following groups:

- **Inactive orbitals:** Orbitals that are doubly occupied in all configurations.
- **Active orbitals:** These orbitals are subdivided into three separate groups:
  - **RAS1 orbitals:** Orbitals that are doubly occupied except for a maximum number of holes allowed in this orbital subspace.
  - **RAS2 orbitals:** In these orbitals all possible occupations are allowed (former Complete Active Space orbitals).
  - **RAS3 orbitals:** Orbitals that are unoccupied except for a maximum number of electrons allowed in this subspace.

CASSCF calculations can be performed with the program, by allowing orbitals only in the RAS2 space. A single reference SDCI wave function is obtained by allowing a maximum of 2 holes in RAS1 and a maximum of 2 electrons in RAS3, while RAS2 is empty (the extension to SDT- and SDTQ-CI is obvious). Multireference CI wave functions can be constructed by adding orbitals also in RAS2.

The RASSCF program is based on the split GUGA formalism. However, it uses determinant based algorithms to solve the configuration interaction problem [68]. To ensure a proper spin function, the transformation to a determinant basis is only performed in the innermost loops of the program to evaluate the σ-vectors in the Davidson procedure and to compute the two-body density matrices. The upper limit to the size of the CASSCF wave function that can be handled with the present program is about 10^7 CSFs and is, in general, limited by the dynamic work array available to the program.

The orbital optimization in the RASSCF program is performed using the super-CI method. The reader is referred to the references [77, 80] for more details. A quasi-Newton (QN) update method is used in order to improve convergence. No explicit CI-orbital coupling is used in the present version of the program, except for the coupling introduced in the QN update.

Convergence of the orbital optimization procedure is normally good for CASSCF type wave functions, but problems can occur in calculations on excited states, especially when several states are close in energy. In such applications it is better to optimize the orbitals for the average energy of several electronic states. Further, convergence can be slower in some cases when orbitals in RAS1 and RAS3 are included. The program is not optimal for SDCI calculations with a large number of orbitals in RAS1 and RAS3.

As with other program modules, please observe that the input is preprocessed and may therefore differ in some respects to the input file prepared by the user. In most cases, this does not imply any functional changes as compared to the user's requests. However, when the input has some minor mistakes or contradictory requests, it can be modified when it is felt that the correction is beneficial. Also, see below for the keyword EXPERT. Without this keyword, the program is assuming more flexibility to optimize the calculation, e.g. by using CIRESTART, if the RASSCF module is called during a numerical differentiation, even if the input requested doing CI calculations from scratch. Using keyword EXPERT, such automatic modification of the user's input is no longer done, and the input is obeyed exactly (when possible).

It is best to provide a set of good input orbitals. (The program can be started from scratch by using CORE, but this should be used only if other possibilities fail). They can either be from some other type of calculation, for example `SCF`, or generated by `GUESSORB`, or from a previous `RASSCF` calculation on the same system. In the first case the orbitals are normally given in formatted form, file *`INPORB`*, in the second case they can also be read from a `RASSCF` input unit *`JOBOLD`*. Input provides both possibilities. Some care has to be taken in choosing the input orbitals, especially for the weakly occupied ones. Different choices may lead to convergence to different local minima. One should therefore make sure that the input orbitals have the correct general structure. A good strategy is often to start using a smaller basis set (MB or DZ) and once the orbitals have been defined, increase the basis set and use `EXPBAS` to generate input orbitals.

When we speak of files like *`INPORB`* or *`JOBIPH`*, please note that these can be regarded as generic names. You may have various files with different file names available, and when invocating the `RASSCF` program, these can be linked or copied (See EMIL command LINK and COPY) so that the program treats them as having the names INPORB or JOBIPH. Alternatively, by the keywords FILEORB and IPHNAME, you can instruct the program to use other file names.

There are two kinds of specifications to make for orbitals: One is the coefficient arrays that describe the molecular orbitals, commonly called 'CMO data'. The other kind is the number of inactive, RAS1, etc. orbitals of each symmetry type, which will be called 'orbital specifications'. The program can take either or both kinds of data from *`INPORB`*, *`JOBIPH`* or runfile. The program selects where to fetch such data, based on rules and input keywords. Avoid using conflicting keywords: the program may sometimes go ahead and resolve the problem, or it may decide to stop, not to risk wasting the user's time on a calculation that was not intended. This decision may be in error.

The orbital specification by keyword input is easy: See keywords FROZEN, INACTIVE, etc. If any such keyword is used, then all the orbital specifications are assumed to be by input, and any such input that is lacking is determined by default rules. These are that there are no such orbitals, with the exception of DELETED: If earlier calculations deleted some orbitals for reason of (near) linear dependence, then these will continue being deleted in subsequent calculations, and cannot be 'undeleted'. Another special case occurs if both CHARGE and NACTEL are given in the input and there is no symmetry, then the default value of INACTIVE will be automatically determined.

If no such keyword has been given, but keyword LUMORB is used to instruct the program to fetch CMO data from *`INPORB`*, then also the orbitals specs are taken from *`INPORB`*, if (as is usually the case) this file contains so-called **typeindex** information. The `GV` program may have been used to graphically view orbital pictures and pick out suitable active orbitals, etc., producing a file with extension '.GvOrb'. When this is used as *INPORB* file, the selected orbitals will be picked in the correct order.

An *INPORB* file with typeindex can also be used to provide orbital specs while the CMO data are taken from another source (*JOBOLD*, *RUNFILE*, ...). This is achieved by TYPEINDEX, and you can look in the manual for this keyword to see an explanation of how the typeindex is written. (This is usually done by the program generating the file, but since these are ASCII files, you may find it expedient to look at, or edit, the typeindex).

In case both keywords, such as INACTIVE, **and** LUMORB, is given, this is of course the very common case that CMO data are read from *INPORB* but orbital specs are given by input. This is perhaps the most common usage. However, when the *INPORB* file is a produced by LUSCUS, it happens frequently that also keyword specs are left in the input, since the user knows that these merely duplicate the specs in *INPORB*. But the latter may also imply a reordering of the orbitals. For this reason, when the keyword input merely duplicates the number of inactive, etc., that is also specified by typeindex, then the typeindex input overrides, to produce the correct ordering. If they do **not** match precisely, then the CMO data are read, without reordering, and the keyword input (as usual) takes precedence.

The CMO data are obtained as follows: With the following keywords, it is assumed that the user knows what he wants.

- CORE: (A bad choice, but here for completeness). Creates orbitals from scratch.
- LUMORB or FILEORB: Try *INPORB*, or fail.
- JOBIPH: Try *JOBOLD*, if not usable, try *JOBIPH*, or fail.

If none of these keywords were used, then the user accepts defaults, namely

1. look for RASSCF orbitals on *RUNFILE*
2. look for SCF orbitals on *RUNFILE*
3. look for GUESSORB orbitals on *RUNFILE*
4. If still nothing found, create orbitals from scratch.

As for earlier versions, notice the possibility to read the orbitals on *JOBIPH*, at a later time, by using the keywords OUTORBITAL and ORBONLY. This results in editable ASCII files, with names like *Project.RasOrb* (or *Project.RasOrb5* for the fifth root). Such orbitals will be produced by default for the lowest roots – up to the tenth, named now, e.g., *Project.RasOrb.5*. There is a keyword MAXORB to produce more (or fewer) such files.

The RASSCF program has special input options, which will limit the degrees of freedoms used in the orbital rotations. It is, for example, possible to impose averaging of the orbital densities in π symmetries for linear molecules. Use the keyword AVERAGE for this purpose. It is also possible to prevent specific orbitals from rotating with each other. The keyword is SUPSYM. This can be used, for example, when the molecule has higher symmetry than one can use with the MOLCAS system. For example, in a linear molecule the point group to be used is C_{2v} or D_{2h}. Both $\sigma-$ and $\delta-$orbitals will then appear in irrep 1. If the input orbitals have been prepared to be adapted to linear symmetry, the SUPSYM input can be used to keep this symmetry through the iterations. The program will do this automatically with the use of the input keyword LINEAR. Similarly, for single atoms, spherical symmetry can be enforced by the keyword ATOM.

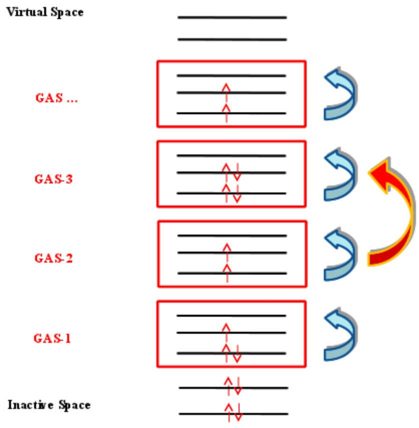


Figure 8.1: *Pictorial representation of GAS active space.*

8.40.1 GASSCF method

In certain cases it is useful/necessary to enforce restrictions on electronic excitations within the active space beyond the ones accessible by RASSCF. These restrictions are meant to remove configurations that contribute only marginally to the total wave function. In MOLCAS this is obtained by the GASSCF approach [10]. GASSCF is a further generalization of the active space concept. This method, like RASSCF, allows restrictions on the active space, but they are more flexible than in RASSCF. These restrictions allow GASSCF to be applied to larger and more complex systems at affordable cost. If the active space is well chosen and the restrictions are not too severe, MCSCF methods recover most of the static correlation energy, and part of the dynamic correlation energy. In the GASSCF method, instead of three active spaces, an in-principle arbitrary number of active spaces (GAS1, GAS2, ...) may be chosen by the user. Instead of a maximum number of holes in RAS1 and particles in RAS3, accumulated minimum and maximum numbers of electrons are specified for GAS1, GAS1+GAS2, GAS1+GAS2+GAS3, etc. in order to define the desired CI expansion(Fig 8.1). All intra-space excitations are allowed (Full-CI in subspaces). Constraints are imposed by user choice on inter-space excitations.

When and how to use the GAS approach? We consider three examples: (1) an organometallic material with separated metal centers and orbitals not delocalized across the metal centers. One can include the near degenerate orbitals of each center in its own GAS space. This implies that one may choose as many GAS spaces as the number of multiconfigurational centers. (2) Lanthanide or actinide metal compounds where the f-electrons require a MC treatment but they do not participate in bonding neither mix with d orbitals. In this case one can put the f orbitals and their electrons into one or more separated GAS spaces and not allow excitations from and/or to other GAS spaces. (3) Molecules where each bond and its correlating anti-bonding orbital could form a separate GAS space as in GVB approach. Finally, if a wave function with a fixed number of holes in one or more orbitals is desired, without interference of configurations where those orbitals are fully occupied the GAS approach is the method of choice instead of the RAS approach. There is no rigorous scheme to choose a GAS partitioning. The right GAS strategy is system-specific. This makes the method versatile but at the same time it is not a black box method. An input example follows:

```
&RASSCF
nActEl
 6 0 0
FROZen
0 0 0 0 0 0 0 0
INACTIVE
2 0 0 0 2 0 0 0
GASScf
3
 1 0 0 0 1 0 0 0
2 2
 0 1 0 0 0 1 0 0
4 4
 0 0 1 0 0 0 1 0
6 6
DELEted
0 0 0 0 0 0 0 0
```

In this example the entire active space counts six active electrons and six active orbitals. These latter are partitioned in three GAS spaces according to symmetry consideration and in the spirit of the GVB strategy. Each subspace has a fixed number of electrons, TWO, and no interspace excitations are allowed. This input shows clearly the difference with the RAS approach. Also for the GASSCF variant a slower convergence might occur.

RASSCF output orbitals

The RASSCF program produces a binary output file called *JOBIPH*, which can be used in subsequent calculations. Previously, this was usually a link, pointing to whichever file the user wanted, or by default to the file *$Project.JobIph* if no such links had been made. This default can be changed, see keyword NewIph and IphName. For simplicity, we refer to this as *JOBIPH* in the manual.The job interface, *JOBIPH*, contains four different sets of MO's and it is important to know the difference between the sets:

1. **Average orbitals:** These are the orbitals produced in the optimization procedure. Before performing the final CI wave function they are modified as follows: inactive and secondary orbitals are rotated (separately) such as to diagonalize an effective Fock operator, and they are then ordered after increasing energy. The orbitals in the different RAS subspaces are rotated (within each space separately) such that the corresponding block of the state-average density matrix becomes diagonal. These orbitals are therefore called "pseudo-natural orbitals". They become true natural orbitals only for CAS type wave functions. These orbitals are not ordered. The corresponding "occupation numbers" may therefore appear in the output in arbitrary order. The final CI wave function is computed using these orbitals. They are also the orbitals found in the printed output.

2. **Natural orbitals:** They differ from the above orbitals, in the active subspace. The entire first order density matrix has been diagonalized. Note that in a RAS calculation, such a rotation does not in general leave the RAS CI space invariant. One set of such orbitals is produced for each of the wave functions in an average RASSCF calculation. The main use of these orbitals is in the calculation of one-electron properties. They are extracted by default (up to ten roots) to the working directory from *JOBIPH* and named *$Project.RasOrb.1*, *$Project.RasOrb.2*, etc. Each set of MO's is stored together

with the corresponding occupation numbers. The natural orbitals are identical to the average orbitals only for a single state CASSCF wave function.

3. **Canonical orbitals:** This is a special set of MO's generated for use in the `CASPT2` and `CCSDT` programs. They are obtained by a specific input option to the `RASSCF` program. They are identical to the above orbitals in the inactive and secondary subspaces. The active orbitals have been obtained by diagonalizing an effective one-electron Hamiltonian, a procedure that leaves the CI space invariant only for CAS type wave functions.

4. **Spin orbitals:** This set of orbitals is generated by diagonalizing the first order spin density matrix and can be used to compute spin properties.

8.40.2 Dependencies

To start the `RASSCF` module at least the one-electron and two-electron integrals generated by `SEWARD` have to be available (exception: See keyword ORBONLY). Moreover, the `RASSCF` requires a suitable start wave function such as the orbitals from a RHF-SCF calculation or produced by `GUESSORB`.

8.40.3 Files

Input files

`RASSCF` will use the following input files: *`ONEINT`*, *`ORDINT`*,*`RUNFILE`*, *`INPORB`*, *`JOBIPH`* (for more information see 7.1.2).

A number of additional files generated by `SEWARD` are also used by the `RASSCF` program. The availability of either of the files named *`INPORB`* and *`JOBOLD`* is optional and determined by the input options LUMORB and JOBIPH, respectively.

Output files

File	*Contents*
`JOBIPH`	This file is written in binary format and carries the results of the wave function optimization such as MO- and CI-coefficients. If several consecutive RASSCF calculations are made, the file names will be modified by appending '01','02' etc.
`RUNFILE`	The *`RUNFILE`* is updated with information from the RASSCF calculation such as the first order density and the Fock matrix.
`MD_CAS.x`	Molden input file for molecular orbital analysis for CI root x.
`RASORB`	This ASCII file contains molecular orbitals, occupation numbers, and orbital indices from a `RASSCF` calculation. The natural orbitals of individual states in an average-state calculation are also produced, and are named *`RASORB.1`*, *`RASORB.2`*, etc.

8.40.4 Input

This section describes the input to the RASSCF program in the MOLCAS program system. The input starts with the program name

```
&RASSCF
```

There are no compulsory keywords, but almost any meaningful calculation will require some keyword. At the same time, most choices have default settings, and many are able to take relevant values from earlier calculations, from available orbital files, etc.

Optional keywords

There is a large number of optional keywords you can specify. They are used to specify the orbital spaces, the CI wave function etc., but also more arcane technical details that can modify e.g. the convergence or precision. The first 4 characters of the keyword are recognized by the input parser and the rest is ignored. If not otherwise stated the numerical input that follows a keyword is read in free format. A list of these keywords is given below:

Keyword *Meaning*

TITLe Follows the title for the calculation in a single line

SYMMetry Specify the selected symmetry type (the irrep) of the wave function as a number between 1 and 8 (see SYMMETRY keyword in GATEWAY section). Default is 1, which always denote the totally symmetric irrep.

SPIN The keyword is followed by an integer giving the value of spin multiplicity ($2S + 1$). Default is 1 (singlet).

CHARge Specify the total charge on the system as an integer. If this keyword is used, the NACTEL keyword should not be used, unless the symmetry group is C1 and INACTIVE is not used (in this case the number of inactive orbitals will be computed from the total charge and active electrons). Default value: 0

RASScf Specify two numbers: maximum number of holes in RAS1 and the maximum number of electrons occupying the RAS3 orbitals Default values are: 0,0 See also keyword CHARGE and NACTEL. The specification using RASSCF, and CHARGE if needed, together replace the single keyword NACTEL.

NACTel Requires one or three numbers to follow, specifying

1. the total number of active electrons (all electrons minus twice the number of inactive and frozen orbitals)
2. the maximum number of holes in RAS1
3. the maximum number of electrons occupying the RAS3 orbitals

If only one number is given, the maximum number of holes in RAS1 and of electrons in RAS3 are both set to zero. Default values are: x,0,0, where x is the number needed to get a neutral system. See also keywords CHARGE and RASSCF, which offer an alternative specification.

CIROot

Specifies the CI root(s) and the dimension of the starting CI matrix used in the CI Davidson procedure. This input makes it possible to perform orbital optimization for the average energy of a number of states. The first line of input gives two or three numbers, specifying the number of roots used in the average calculation (NROOTS), the dimension of the small CI matrix in the Davidson procedure (LROOTS), and possibly a non-zero integer IALL. If $IALL.ne.1$ or there is no IALL, the second line gives the index of the states over which the average is taken (NROOTS numbers, IROOT). **Note** that the size of the CI matrix, LROOTS, must be at least as large as the highest root, IROOT. If, **and only if**, NROOTS>1 a third line follows, specifying the weights of the different states in the average energy. If IALL=1 has been specified, no more lines are read. A state average calculation will be performed over the NROOTS lowest states with equal weights. energy. Examples:

```
CIRoot= 3 5; 2 4 5; 1 1 3
```

The average is taken over three states corresponding to roots 2, 4, and 5 with weights 20%, 20%, and 60%, respectively. The size of the Davidson Hamiltonian is 5. Another example is:

```
CIRoot= 19 19 1
```

A state average calculation will be performed over the 19 lowest states each with the weight 1/19 Default values are NROOTS = LROOTS = IROOT = 1 (ground state), which is the same as the input:

```
CIRoot= 1 1; 1
```

CISElect

This keyword is used to select CI roots by an overlap criterion. The input consists of three lines per root that is used in the CI diagonalization (3*NROOTS lines in total). The first line gives the number of configurations used in the comparison, `nRef`, up to five. The second line gives `nRef` reference configuration indices. The third line gives estimates of CI coefficients for these CSF's. The program will select the roots which have the largest overlap with this input. Be careful to use a large enough value for LROOTS (see above) to cover the roots of interest.

ATOM

This keyword is used to get orbitals with pure spherical symmetry for atomic calculations (the radial dependence can vary for different irreps though). It causes super-symmetry to be switched on (see SUPSym keyword) and generates automatically the super-symmetry vector needed. Also, at start and after each iteration, it sets to zero any CMO coefficients with the wrong symmetry. Use this keyword instead of SUPSym for atoms.

LINEar This keyword is used to get orbitals with pure rotational symmetry for linear molecules. It causes super-symmetry to be switched on (see SUPSym keyword) and generates automatically the super-symmetry vector needed. Also, at start and after each iteration, it sets to zero any CMO coefficients with the wrong symmetry. Use this keyword instead of SUPSym for linear molecules.

RLXRoot Specifies which root to be relaxed in a geometry optimization of a state average wave function. Thus, the keyword has to be combined with CIRO. In a geometry optimization the following input

```
CIRoot= 3 5; 2 4 5; 1 1 3
RLXRoot= 4
```

will relax CI root number 4.

MDRLxroot Selects a root from a state average wave function for gradient computation in the first step of a molecular dynamics simulation. The root is specified in the same way as in the RLXR keyword. In the following steps the trajectory surface hopping can change the root if transitions between the states occur. This keyword is mutually exclusive with the RLXR keyword.

EXPErt This keyword forces the program to obey the input. Normally, the program can decide to change the input requests, in order to optimize the calculation. Using the EXPERT keyword, such changes are disallowed.

RFPErt This keyword will add a constant reaction field perturbation to the Hamiltonian. The perturbation is read from the *RUNOLD* (if not present defults to *RUNFILE*) and is the latest self-consistent perturbation generated by one of the programs SCF or RASSCF.

NONEquilibrium Makes the slow components of the reaction field of another state present in the reaction field calculation (so-called non-equilibrium solvation). The slow component is always generated and stored on file for equilibrium solvation calculations so that it potentially can be used in subsequent non-equilibrium calculations on other states.

RFROot Enter the index of that particular root in a state-average calculation for which the reaction-field is generated. It is used with the PCM model.

CIRFroot Enter the relative index of one of the roots specified in CISElect for which the reaction-field is generated. Used with the PCM model.

NEWIph The default name of the *JOBIPH* file will be determined by any already existing such files in the work directory, by appending '01', '02' etc. so a new unique name is obtained.

FROZen Specifies the number of frozen orbitals in each symmetry. (see below for condition on input orbitals). Frozen orbitals will not be modified in the calculation. Only doubly occupied orbitals can be left frozen. This input can be used for example for inner shells of heavy atoms to reduce the basis set superposition error. Default is 0 in all symmetries.

INACtive

Specify on the next line the number of inactive (doubly occupied) orbitals in each symmetry. Frozen orbitals should not be included here. Default is 0 in all symmetries, but if there is no symmetry (C1) and both CHARGE and NACTEL are given, the number of inactive orbitals will be calculated automatically.

RAS1

On the next line specify the number of orbitals in each symmetry for the RAS1 orbital subspace. Default is 0 in all symmetries.

RAS2

On the next line specify the number of orbitals in each symmetry for the RAS2 orbital subspace. Default is 0 in all symmetries.

RAS3

On the next line specify the number of orbitals in each symmetry for the RAS3 orbital subspace. Default is 0 in all symmetries.

DELEted

Specify the number of deleted orbitals in each symmetry. These orbitals will not be allowed to mix into the occupied orbitals. It is always the last orbitals in each symmetry that are deleted. Default is 0 in all symmetries, unless orbitals wer already deleted by previous programs due to near-linear dependence.

GASScf

Needed to perform a Generalized Active Space (GASSCF) calculation. It is followed by an integer that defines the number of active subspaces, and two lines for each subspace. The first line gives the number of orbitals in each symmetry, the second gives the minimum and maximum number of electrons in the accumulated active space.

An example of an input that uses this keyword is the following:

```
GASSCF
 5
 2 0 0 0 2 0 0 0
 4 4
 0 2 0 0 0 2 0 0
 8 8
 0 0 2 0 0 0 2 0
 12 12
 0 0 0 1 0 0 0 1
 14 14
 4 2 2 1 4 2 2 1
 20 20
```

In the example above (20in32), excitations from one subspace to another are not allowed since the values of MIN and MAX for GSOC are identical for each of the five subspaces.

JOBIph

Input molecular orbitals are read from an unformatted file with FORTRAN file name *JOBOLD*. **Note**, the keywords Lumorb, Core, and Jobiph are mutually exclusive. If none is given the program will search for input orbitals on the runfile in the order: `RASSCF`, `SCF`, `GUESSORB`. If none is found, the keyword CORE will be activated.

IPHName

Override the default choice of name of the *JOBIPH* file by giving the file name you want. The name will be truncated to 8 characters and converted to uppercase.

LUMOrb

Input molecular orbitals are read from a formatted file with FORTRAN file name *INPORB*. **Note**, the keywords Lumorb, Core, and Jobiph are mutually exclusive. If none is given the program will search for input orbitals on the runfile in the order: RASSCF, SCF, GUESSORB. If none is found, the keyword CORE will be activated.

FILEorb

Override the default name (*INPORB*) for starting orbital file by giving the file name you want.

CORE

Input molecular orbitals are obtained by diagonalizing the core Hamiltonian. This option is only recommended in simple cases. It often leads to divergence. **Note**, the keywords Lumorb, Core, and Jobiph are mutually exclusive.

ALPHaOrBeta

With UHF orbitals as input, select alpha or beta as starting orbitals. A positive value selects alpha, a negative value selects beta. Default is 0, which fails with UHF orbitals. This keyword does not affect the spin of the wave function (see the SPIN keyword).

TYPEIndex

This keyword forces the program to use information about subspaces from the *INPORB* file.

User can change the order of orbitals by editing of ”Type Index” section in the *INPORB* file. The legend of the types is:

- **F** - Frozen
- **I** - Inactive
- **1** - RAS1
- **2** - RAS2
- **3** - RAS3
- **S** - Secondary
- **D** - Deleted

ALTEr

This keyword is used to change the ordering of MO in *INPORB* or *JOBOLD*. The keyword requires first the number of pairs to be interchanged, followed, for each pair, the symmetry species of the pair and the indices of the two permuting MOs. Here is an example:

```
ALTEr= 2; 1 4 5; 3 6 8
```

In this example, 2 pairs of MO will be exchanged: 4 and 5 in symmetry 1 and 6 and 8 in symmetry 3.

CLEAnup

This input is used to set to zero specific coefficients of the input orbitals. It is of value, for example, when the actual symmetry is higher than given by input and the trial orbitals are contaminated by lower symmetry mixing. The input requires at least one line per symmetry specifying the number of additional groups of orbitals to clean. For each group of orbitals within the symmetry, three lines follow. The first line indicates the number of considered orbitals and the specific number of the orbital (within the symmetry) in the set of input orbitals. Note the input lines can not be longer than 72 characters and the program expects as many continuation lines as are needed. The second line indicates the number of coefficients belonging to the prior orbitals which are going to be set to zero and which coefficients. The third line indicates the number of the coefficients of all the complementary orbitals of the symmetry which are going to be set to zero and which are these coefficients. Here is an example of what an input would look like:

```
CLEAnup
2
   3 4 7 9; 3 10 11 13; 4 12 15 16 17
   2 8 11; 1 15; 0
0; 0; 0
```

In this example the first entry indicates that two groups of orbitals are specified in the first symmetry. The first item of the following entry indicates that there are three orbitals considered (4, 7, and 9). The first item of the following entry indicates that there are three coefficients of the orbitals 4, 7, and 9 to be set to zero, coefficients 10, 11, and 13. The first item of the following entry indicates that there are four coefficients (12, 15, 16, and 17) which will be zero in all the remaining orbitals of the symmetry. For the second group of the first symmetry orbitals 8 and 11 will have their coefficient 15 set to zero, while nothing will be applied in the remaining orbitals. If a geometry optimization is performed the keyword is inactive after the first structure iteration.

CIREstart — Starting CI-coefficients are read from a binary file *JOBOLD*.

ORBOnly — This input keyword is used to get a formated ASCII file (*RASORB*, *RASORB.2*, etc) containing molecular orbitals and occupations reading from a binary *JobIph* file. The program will not perform any other operation. (In this usage, the program can be run without any files, except the *JOBIPH* file).

CIONly — This keyword is used to disable orbital optimization, that is, the CI roots are computed only for a given set of input orbitals.

CHOInput — This marks the start of an input section for modifying the default settings of the Cholesky RASSCF. Below follows a description of the associated options. The options may be given in any order, and they are all optional except for ENDChoinput which marks the end of the CHOInput section.

NoLK — Available only within ChoInput. Deactivates the “Local Exchange” (LK) screening algorithm [81] in computing the Fock matrix. The loss of speed compared to the default algorithm can be substantial, especially for electron-rich systems. Default is to use LK.

DMPK Available only within ChoInput. Modifies the thresholds used in the LK screening. The keyword takes as argument a (double precision) floating point (non-negative) number used as correction factor for the LK screening thresholds. The default value is 1.0d-1. A smaller value results in a slower but more accurate calculation.
Note.: The default choice of the LK screening thresholds is tailored to achieve as much as possible an accuracy of the converged RASSCF energies consistent with the choice of the Cholesky decomposition threshold.

NODEcomposition Available only within ChoInput. Deactivates the Cholesky decomposition of the inactive AO 1-particle density matrix. The inactive Exchange contribution to the Fock matrix is therefore computed using inactive canonical orbitals instead of (localized) "Cholesky MOs" [62]. This choice tends to lower the performances of the LK screening. Default is to perform this decomposition in order to obtain the Cholesky MOs.

TIME Activates printing of the timings of each task of the Fock matrix build. Default is to not show these timings.

MEMFraction Set the fraction of memory to use as global Cholesky vector buffer. Default: for serial runs 0.0d0; for parallel runs 0.3d0.

OFEMBEDDING Performs a Orbital-Free Embedding (OFE)RASSCF calculation, available only in combination with Cholesky or RI integral representation. The runfile of the environment subsystem renamed AUXRFIL is required. An example of input for the keyword OFEM is the following:

```
OFEMbedding
 ldtf/pbe
dFMD
 1.0   1.0d2
FTHAw
 1.0d-4
```

The keyword OFEM requires the specification of two functionals in the form fun1/fun2, where fun1 is the functional used for the Kinetic Energy (available functionals: Thomas-Fermi, with acronym LDTF, and the NDSD functional), and where fun2 is the xc-functional (LDA, LDA5, PBE and BLYP available at the moment). The OPTIONAL keyword DFMD has two arguments: first, the fraction of correlation potential to be added to the OFE potential; second, the exponential decay factor for this correction (used in PES calculations). The OPTIONAL keyword DFMD specifies the fraction of correlation potential to be added to the OFE potential. The OPTIONAL keyword FTHA is used in a freeze-and-thaw cycle (EMIL Do While) to specify the (subsystems) energy convergence threshold.

ITERATIONS Specify the maximum number of RASSCF iterations, and the maximum number of iterations used in the orbital optimization (super-CI) section. Default and maximum values are 200,100.

LEVShft Define a level shift value for the super-CI Hamiltonian. Typical values are in the range 0.0 – 1.5. Increase this value if a calculation diverges. The default value 0.5, is normally the best choice when Quasi-Newton is performed.

THRS Specify convergence thresholds for: energy, orbital rotation matrix, and energy gradient. Default values are: 1.0e-08, 1.0e-04, 1.0e-04.

TIGHt Convergence thresholds for the Davidson diagonalization procedure. Two numbers should be given: THREN and THFACT. THREN specifies the energy threshold in the first iteration. THFACT is used to compute the threshold in subsequent iterations as THFACT*DE, where DE is the RASSCF energy change. Default values are 1.0d-04 and 1.0d-3.

NOQUne This input keyword is used to switch off the Quasi-Newton update procedure for the Hessian. Pure super-CI iterations will be performed. (Default setting: QN update is used unless the calculation involves numerically integrated DFT contributions.)

QUNE This input keyword is used to switch on the Quasi-Newton update procedure for the Hessian. (Default setting: QN update is used unless the calculation involves numerically integrated DFT contributions.)

CIMX Specify the maximum number of iterations allowed in the CI procedure. Default is 100 with maximum value 200.

HEXS Highly excited states. Will eliminate the maximum occupation in one or more RAS/GAS's thereby not eliminating all roots below. Very helpful for core excitations where the ground-state input can be used to eliminate unwanted roots. Works with RASSI. First input is the number of RAS/GAS where the maximum occupation should be eliminated. Second is the RAS/GAS or RAS/GAS's where maximum occupation will not be allowed.

SDAV Here follows the dimension of the explicit Hamiltonian used to speed up the Davidson CI iteration process. An explicit H matrix is constructed for those configurations that have the lowest diagonal elements. This H-matrix is used instead of the corresponding diagonal elements in the Davidson update vector construction. The result is a large saving in the number if CI iterations needed. Default value is the smallest of 100 and the number of configurations. Increase this value if there is problems converging to the right roots.

SXDAmp A variable called SXDAMP regulates the size of the orbital rotations. Use keyword SXDAmp and enter a real number. The default value is 0.0002. Larger values can give slow convergence, very low values may give problems e.g. if some active occupations are very close to 0 or 2.

SUPSym This input is used to restrict possible orbital rotations. It is of value, for example, when the actual symmetry is higher than given by input.

Each orbital is given a label IXSYM(I). If two orbitals in the same symmetry have different labels they will not be allowed to rotate into each other and thus prevents from obtaining symmetry broken solutions. Note, however, that the starting orbitals must have the right symmetry. The input requires one or more entries per symmetry. The first specifies the number of additional subgroups in this symmetry (a 0 (zero) denotes that there is no additional subgroups and the value of IXSYM will be 0 (zero) for all orbitals in that symmetry). If the number of additional subgroups is not zero there are additional entries for each subgroup: The dimension of the subgroup and the list of orbitals in the subgroup counted relative to the first orbital in this symmetry. Note, the input lines can not be longer than 180 characters and the program expects continuation lines as many as there are needed. As an example assume an atom treated in C_{2v} symmetry for which the d_{z^2} orbitals (7 and 10) in symmetries 1 may mix with the s orbitals. In addition, the d_{z^2} and $\mathrm{d}_{x^2-y^2}$ orbitals (8 and 11) may also mix. Then the input would look like:

```
SUPSym
2
    2 7 10; 2 8 11
0; 0; 0
```

In this example the first entry indicates that we would like to specify two additional subgroups in the first symmetry (total symmetric group). The first item in the following two entries declares that each subgroup consists of two orbitals. Orbitals 7 and 10 constitute the first group and it is assumed that these are orbitals of d_{z^2} character. The second group includes the $\mathrm{d}_{x^2-y^2}$ orbitals 8 and 11. The following three entries denote that there are no further subgroups defined for the remaining symmetries. Ordering of the orbitals according to energy is deactivated when using SUPSym. If you activate ordering using ORDEr, the new labels will be printed in the output section. If a geometry optimization is performed the reordered matrix will be stored in the ***RUNFILE*** file and read from there instead of from the input in each new structure iteration.

HOME

With this keyword, the root selection in the Super-CI orbital update is by maximum overlap rather than lowest energy.

VB

Using this keyword, the CI optimization step in the RASSCF program will be replaced by a call to the *CASVB* program, such that fully variational valence bond calculations may be carried out. The `VB` keyword can be followed by any of the directives described in section 8.4 and should be terminated by `ENDVB`. Energy-based optimization of the VB parameters is the default, and the output level for the main *CASVB* iterations is reduced to −1, unless the print level for *CASVB* print option 6 is ≥2.

PRINt

The keyword is followed by a line giving the print levels for various logical code sections. It has the following structure: IW IPR IPRSEC(I), I=1,7

- IW - logical unit number of printed output (normally 6).
- IPR - is the overall print level (normally 2).
- IPRSEC(I) - gives print levels in different sections of the program.
    1. Input section
    2. Transformation section
    3. CI section
    4. Super-CI section
    5. Not used
    6. Output section
    7. Population analysis section

The meaning of the numbers: 0=Silent, 1=Terse, 2=Normal, 3=Verbose, 4=Debug, and 5=Insane. If input is not given, the default (normally=2) is determined by a global setting which can be altered bubroutine call. (Programmers: See programming guide). The local print level in any section is the maximum of the IPRGLB and IPRSEC() setting, and is automatically reduced e.g. during structure optimizations or numerical differentiation. Example:

```
Print= 6 2 2 2 2 3 2 2 2 2
```

MAXOrb

Maximum number of RasOrb files to produce, one for each root up to the maximum.

OUTOrbitals

This keyword is used to select the type of orbitals to be written in a formated ASCII file. By default a formated *RASORB* file containing average orbitals and subsequent *RASORB.1*, *RASORB.2*, etc, files containing natural orbitals for each of the computed (up to ten) roots will be generated in the working directory. An entry follows with an additional keyword selecting the orbital type. The possibilities are:

AVERage orbitals: this is the default option. This keyword is used to produce a formated ASCII file of orbitals (*RASORB*) which correspond to the final state average density matrix obtained by the RASSCF program. The inactive and secondary orbitals have been transformed to make an effective Fock matrix diagonal. Corresponding diagonal elements are given as orbital energies in the RASSCF output listing. The active orbitals have been obtained by diagonalizing the sub-blocks of the average density matrix corresponding to the three different RAS orbital spaces, thereby the name pseudo-natural orbitals. They will be true natural orbitals only for a CASSCF wave function.

CANOnical orbitals: Use this keyword to produce the canonical orbitals. They differ from the natural orbitals, because also the active part of the Fock matrix is diagonalized. Note that the density matrix is no longer diagonal and the CI coefficients have not been transformed to this basis. This option substitutes the previous keyword CANOnical.

NATUral orbitals: Use this keyword to produce the true natural orbitals. The keyword should be followed by a new line with an integer specifying the maximum CI root for which the orbitals and occupation numbers are needed. One file for each root will be generated up to the specified number. In a one state RASSCF calculation this number is always 1, but if an average calculation has been performed, the NO's can be obtained for all the states included in the energy averaging. Note that the natural orbitals main use is as input for property calculations using `SEWARD`. The files will be named ***RASORB***, ***RASORB.2***, ***RASORB.3***, etc. This keyword is on by default for up to ten roots.

SPIN orbitals. This keyword is used to produce a set of spin orbitals and is followed by a new line with an integer specifying the maximum CI root for which the orbitals and occupation numbers are needed. One file for each root will be generated up to the specified number. Note, for convenience the doubly occupied and secondary orbitals have been added to these sets with occupation numbers 0 (zero). The main use of these orbitals is to act as input to property calculations and for graphical presentations. This keyword is on by default for up to ten roots.

An example input follows in which five files are requested containing natural orbitals for roots one to five of a RASSCF calculation. The files are named ***RASORB.1***, ***RASORB.2***, ***RASORB.3***, ***RASORB.4***, and ***RASORB.5***, respectively for each one of the roots. Although this is the default, it can be used complemented by the ORBOnly keyword, and the orbitals will be read from a JobIph file from a previous calculation, in case the formated files were lost or you require more than ten roots. As an option the MAXOrb can be also used to increase the number of files over ten.

```
OUTOrbital= Natural; 15
```

ORBListing

This keyword is followed with a word showing how extensive you want the orbital listing to be in the printed output. The alternatives are:

- **NOTHing:** No orbitals will be printed (suggested for numerical CASPT2 optimization). (Also, the old VERYbrief will be accepted).
- **FEW:** The program will print the occupied orbitals, and any secondary with less than 0.15 a.u. orbital energy. (Old BRIEF also accepted).
- **NOCOre:** The program will print the active orbitals, and any secondary with less than 0.15 a.u. orbital energy.
- **ALL:** All orbitals are printed. (Old LONG also accepted).

(unless other limits are specified by the PROR keyword).

ORBAppear

This keyword requires an entry with a word showing the appearance of the orbital listing in the printed output. The alternatives are:

- **COMPact:** The format of the orbital output is changed from a tabular form to a list giving the orbital indices and MO-coefficients. Coefficients smaller than 0.1 will be omitted.
- **FULL:** The tabular form will be chosen.

PROR

This keyword is used to alter the printout of the MO-coefficients. Two numbers must be given of which the first is an upper boundary for the orbital energies and the second is a lower boundary for the occupation numbers. Orbitals with energy higher than the threshold or occupation numbers lower that the threshold will not be printed. By default these values are set such that all occupied orbitals are printed, and virtual orbitals with energy less than 0.15 au. However, the values are also affected by use of OUTPUT.

PRSD

This keyword is used to request that not only CSFs are printed with the CI coefficients, but also the determinant expansion.

ORDEr

This input keyword is used to deactivate or activate ordering of the output orbitals according to energy. One number must be given: 1 if you want ordering and 0 if you want to deactivate ordering. Default is 1 and with SUPSym keyword default is 0.

PRSP

Use this keyword to get the spin density matrix for the active orbitals printed.

PRWF

Enter the threshold for CI coefficients to be printed (Default: 0.05).

A general comment concerning the input orbitals: The orbitals are ordered by symmetry. Within each symmetry block the order is assumed to be: frozen, inactive, active, external (secondary), and deleted. Note that if the Spdelete option has been used in a preceding `SCF` calculation, the deleted orbitals will automatically be placed as the last ones in each symmetry block.

For calculations of a molecule in a reaction field see section 8.16.1 of the present manual and section 10.6 of the examples manual.

Input example

The following example shows the input to the `RASSCF` program for a calculation on the water molecule. The calculation is performed in C_{2v} symmetry (symmetries: a_1, b_2, b_1, a_2, where the two last species are antisymmetric with respect to the molecular plane). Inactive orbitals are $1a_1$ (oxygen $1s$) $2a_1$ (oxygen 2s) and $1b_1$ (the π lone-pair orbital). Two bonding and two anti-bonding OH orbitals are active, a_1 and b_2 symmetries. The calculation is performed for the $^1A_1$ ground state. Note that no information about basis set, geometry, etc has to be given. Such information is supplied by the `SEWARD` integral program via the one-electron integral file *`ONEINT`*.

```
 &RASSCF
Title= Water molecule. Active orbitals OH and OH* in both symmetries
Spin     = 1
```

```
Symmetry = 1
Inactive = 2 0 1 0
Ras2     = 2 2 0 0
```

More advanced examples can be found in the tutorial section of the manual.

8.41 RASSI

The RASSI (RAS State Interaction) program forms overlaps and other matrix elements of the Hamiltonian and other operators over a wave function basis, which consists of RASSCF wave functions, each with an individual set of orbitals. It is extensively used for computing dipole oscillator strengths, but any one-electron operator, for which the SEWARD has computed integrals to the *ORDINT* file, can be used, not just dipole moment components.

Also, it solves the Schrödinger equation projected on the space spanned by these wave functions, i.e., it forms non-interacting linear combinations of the input state functions, and computes matrix elements over the resulting eigenbasis as well.

Finally, using these spin-free eigenstates as a basis, it can compute spin-orbit interaction matrix elements, diagonalize the resulting matrix, and compute various matrix elements over the resulting set of spin-orbit eigenstates.

If only matrix elements of some one-electron operator(s), such as the dipole transition moments, are required, the calculation of Hamiltonian matrix elements and the transformation to the eigenbasis of this matrix can be skipped. However, if any states have the same symmetry and different orbitals, it is desirable to use the transitions strengths as computed between properly non-interacting and orthonormal states. The reason is that the individually optimized RASSCF states are interacting and non-orthogonal, and the main error in the computed transition matrix elements is the difference in electronic dipole moment times the overlap of any two states involved. For excited states, the overlap is often in the order of 10%.

Please note: Due to the increasing number of calculations done with a hundred input states, or more, there has been a demand to change the output. Until MOLCAS6.2, the default assumption has been to print all expectation values and matrix elements that can be computed from the selection of one-electron integrals. From 6.4, this is requested by keywords, see the keyword list below for XVIN, XVES, XVSO, MEIN, MEES, and MESO.

Apart from computing oscillator strengths, overlaps and Hamiltonian matrix elements can be used to compute electron transfer rates, or to form quasi-diabatic states and reexpress matrix elements over a basis of such states.

The CSF space of a RASSCF wave function is closed under deexcitation. For any given pair of RASSCF wave functions, this is used in the way described in reference [82] to allow the pair of orbital sets to be transformed to a biorthonormal pair, while simultaneously transforming the CI expansion coefficients so that the wave functions remain unchanged. The basic principles are the same as in the earlier program [83], but is adapted to allow RASSCF as well as CASSCF wave functions. It uses internally a Slater determinant expansion. It can now use spin-dependent operators, including the AMFI spin-orbit operator, and can compute matrix elements over spin-orbit states, i.e. the eigenstates of the sum of the spin-free hamiltonian and the spin-orbit operator.

One use of the RASSI eigenstates is to resolve ambiguities due to the imperfect description of highly excited states. Association between individually optimized states and the exact electronic eigenstates is often not clear, when the calculation involves several or many excited states. The reason is that the different states each use a different set of orbitals. The State Interaction calculation gives an unambiguous set of non-interacting and orthonormal eigenstates to the projected Schrödinger equation, and also the overlaps between the original RASSCF wave functions and the eigenstates. The latter is a very efficient diagnostic, since it describes the RASSCF states in terms of one single wave-function basis set.

To make the last point clear, assume the following situation: We have performed three RASSCF calculations, one where we optimize for the lowest state, one for the first excited state, and one for the 2nd excited state in the same symmetry. The active orbitals are fairly much mixed around, so a simple inspection of the CI coefficient is insufficient for comparing the states. Assume that for each state, we have calculated the three lowest CI roots. It can now happen, that the 2nd root of each calculation is a fair approximation to the exact 2nd eigenstate, and the same with the 3rd, or possibly that the order gets interchanged in one or two of the calculation. In that case, a RASSI calculation with these 9 states will give three improved solutions close to the original ones, and of course 6 more that are considered to be the removed garbage. The overlaps will confirm that each of the input states consists mainly of one particular out of the three lowest eigenstates. This situation is the one we usually assume, if no further information is available.

However, it happens that the active orbitals of the three calculations do not span approximately the same space. The orbital optimization procedure has made a qualitatively different selection of correlating orbitals for the three different calculation. Then the `RASSI` calculation may well come out with 4 lowest roots that overlap strongly with the original RASSCF states. This may change the assignments and may also give valuable information about the importance of some state. The natural orbitals of the eigenstates will show that the active space used in the RASSCF was in some way inappropriate.

Another bothersome situation is also solved by the RASSI method. The analysis of the original states in terms of RASSI eigenstates may reveal that the three optimized RASSCF states consists mainly of TWO low RASSI eigenstates! This is because the RASSCF optimization equations are non-linear and may sometimes offer spurious extra solutions. Two of the calculations are in this case to be regarded qualitatively, as two different (local) solutions that approximate (imperfectly) the same excited state. Also in this case, the natural orbitals will probably offer a clue to how to get rid of the problem. Extra solutions rarely occur for low states in CASSCF calculations, provided a generous active space can be afforded. Problems occur when the active space is too small, and in particular with general RASSCF calculations.

A further application is the preparation of a suitable orbital basis for a subsequent CI calculation. Note that such an application also allows the use of badly converged RASSCF wave functions, or of RASSCF wave functions containing multiple minima solutions close to a common exact eigenstate. In effect, the `RASSI` program cleans up the situation by removing the errors due to bad convergence (pushing the errors into a garbage part of the spectrum). This requires that the set of input states (9 in this example) provides flexibility enough to remove at least a major part of the error. As one would expect, this is usually true: The erratic non-convergent, or the too slowly convergent, error mode is to a large extent spanned by the few lowest RASSCF wave functions.

Finally, there are situations where there is no problem to obtain adiabatic RASSCF solutions, but where it is still imperative to use RASSI natural orbitals in a subsequent CI. Consider the case of transition metal chemistry, where there is in general two or more electronic states involved. These states are supposed to interact strongly, at least within some range of interatomic distances. Here, an MCSCF solution, such as RASSCF, will have at least two very different solutions, one associated with each configuration of the transition metal atom. Using one set of orbitals, one electronic state has a reasonably described potential energy curve, while other states get pushed far up in energy. Using another set of orbitals, another state gets correctly described. In no calculation with a single orbital set do we obtain the avoided crossings, where one switches from one diabatic state to another. The only way to accomplish this is via a RASSI calculation. In this case, it is probably necessary also to shift the energies of the RASSCF states to ensure that the crossing occur at the correct places. The shifts can be determined by correcting the atomic spectrum in the separated-atoms limit.

Note, however, that most of the problems described above can be solved by performing state-averaged RASSCF calculations.

8.41.1 Dependencies

The RASSI program needs one or more *JOBIPH* files produced by the RASSCF program. Also, it needs a *ONEINT* file from SEWARD, with overlap integrals and any one-electron property integrals for the requested matrix elements. If Hamiltonian matrix elements are used, also the *ORDINT* file is needed.

8.41.2 Files

Input files

File	*Contents*
*ORDINT**	Ordered two-electron integral file produced by the SEWARD program. In reality, this is up to 10 files in a multi-file system, named ORDINT, ORDINT1,...,ORDINT9. This is necessary on some platforms in order to store large amounts of data.
ONEINT	The one-electron integral file from SEWARD
JOBnnn	A number of *JOBIPH* files from different RASSCF jobs. An older naming convention assumes file names JOB001, JOB002, etc. for these files. They are automatically linked to default files named *$Project.JobIph*, *$Project.JobIph01*, *$Project.JobIph02*, etc. in directory *$WorkDir*, unless they already exist as files or links before the program starts. You can set up such links yourself, or else you can specify file names to use by the keyword IPHNames.
JOBIPHnn	A number of *JOBIPH* files from different RASSCF jobs. The present naming convention assumes file names JOBIPH, JOBIPH01, etc. for

such files, when created by subsequent `RASSCF` runs, unless other names were specified by input. They are automatically linked to default files named *$Project.JobIph*, *$Project.JobIph01*, *$Project.JobIph02*, etc. in directory *$WorkDir*, unless they already exist as files or links before the program starts. You can set up such links yourself, or else you can specify file names to use by the keyword IPHNames.

Output files

File	*Contents*
SIORBnn	A number of files containing natural orbitals, (numbered sequentially as *SIORB01*, *SIORB02*, etc.) If you do not like these names, you must set up links yourselves, except for the first one: SIORB01 is automatically linked to a default file named $Project.SiOrb in directory $WorkDir, if it does not already exists as a file or a link before the program starts. You must set up the other files yourself.
TOFILE	This output is only created if TOFIle is given in the input. It will contain the transition density matrix computed by `RASSI`. Currently, this file is only used as input to `QMSTAT`.
EIGV	Like *TOFILE* this file is only created if TOFIle is given in the input. It contains auxiliary information that is picked up by `QMSTAT`.

8.41.3 Input

This section describes the input to the `RASSI` program in the Molcas program system, with the program name:

```
&RASSI
```

When a keyword is followed by additional mandatory lines of input, this sequence cannot be interrupted by a comment line. The first 4 characters of keywords are decoded. An unidentified keyword makes the program stop.

Keywords

Keyword	*Meaning*
CHOInput	This marks the start of an input section for modifying the default settings of the Cholesky RASSI. Below follows a description of the associated options. The options may be given in any order, and they are all optional except for ENDChoinput which marks the end of the CHOInput section.

NoLK Available only within ChoInput. Deactivates the “Local Exchange” (LK) screening algorithm [81] in computing the Fock matrix. The loss of speed compared to the default algorithm can be substantial, especially for electron-rich systems. Default is to use LK.

DMPK Available only within ChoInput. Modifies the thresholds used in the LK screening. The keyword takes as argument a (double precision) floating point (non-negative) number used as correction factor for the LK screening thresholds. The default value is 1.0d-1. A smaller value results in a slower but more accurate calculation.
Note: the default choice of the LK screening thresholds is tailored to achieve as much as possible an accuracy of the RASSI energies consistent with the choice of the Cholesky decomposition threshold.

NODEcomposition Available only within ChoInput. The inactive Exchange contribution to the Fock matrix is computed using inactive canonical orbitals instead of (localized) “Cholesky MOs”. This choice is effective only in combination with the LK screening. Default is to use Cholesky MOs. **Note:** the Cholesky MOs in RASSI are computed by decomposing the density type supermatrix $\mathbf{D} = (\mathbf{C}_A, \mathbf{C}_B)(\mathbf{C}_A, \mathbf{C}_B)^T$ where $\mathbf{C}$ is the corresponding canonical MOs matrix for the state A and B. When computing the coupling between 2 different states A and B, only for the first state we use pure Cholesky MOs. The invariance of the Fock matrix is then ensured by rotating the orbitals of B according to the orthogonal matrix defined in A through the Cholesky localization. These orbitals used for B are therefore called “pseudo Cholesky MOs”.

TIME Activates printing of the timings of each task of the Fock matrix build. Default is to not show these timings.

MEMFraction Set the fraction of memory to use as global Cholesky vector buffer. Default: for serial runs 0.0d0; for parallel runs 0.3d0.

MEIN Demand for printing matrix elements of all selected one-electron properties, over the input RASSCF wave functions.

MEES Demand for printing matrix elements of all selected one-electron properties, over the spin-free eigenstates.

MESO Demand for printing matrix elements of all selected one-electron properties, over the spin-orbit states.

PROPerty Replace the default selection of one-electron operators, for which matrix elements and expectation values are to be calculated, with a user-supplied list of operators.

From the lines following the keyword the selection list is read by the following *FORTRAN* code:
READ(*,*) NPROP,(PNAME(I),ICOMP(I),I=1,NPROP)

NPROP is the number of selected properties, PNAME(I) is a character string with the label of this operator on SEWARD's one-electron integral file, and ICOMP(I) is the component number.

The default selection is to use dipole and/or velocity integrals, if these are available in the *ONEINT* file. This choice is replaced by the user-specified choice if the PROP keyword is used. Note that the character strings are read using list directed input and thus must be within single quotes, see sample input below. For a listing of presently available operators, their labels, and component conventions, see SEWARD program description.

SOCOupling — Enter a positive threshold value. Spin-orbit interaction matrix elements over the spin components of the spin-free eigenstates will be printed, unless smaller than this threshold. The value is given in cm^{-1} units. The keyword is ignored unless an SO hamiltonian is actually computed.

SOPROPerty — Enter a user-supplied selection of one-electron operators, for which matrix elements and expectation values are to be calculated over the of spin-orbital eigenstates. This keyword has no effect unless the SPIN keyword has been used. Format: see PROP keyword.

SPINorbit — Spin-orbit interaction matrix elements will be computed. Provided that the ONEL keyword was not used, the resulting Hamiltonian including the spin-orbit coupling, over a basis consisting of all the spin components of wave functions constructed using the spin-free eigenstates, will be diagonalized. NB: For this keyword to have any effect, the SO integrals must have been computed by SEWARD! See AMFI keyword in SEWARD documentation.

ONEL — The two-electron integral file will not be accessed. No Hamiltonian matrix elements will be calculated, and only matrix elements for the original RASSCF wave functions will be calculated. ONEE is a valid synonym for this keyword.

J-VAlue — For spin-orbit calculations with single atoms, only: The output lines with energy for each spin-orbit state will be annotated with the approximate J and Omega quantum numbers.

OMEGa — For spin-orbit calculations with linear molecules, only: The output lines with energy for each spin-orbit state will be annotated with the approximate Omega quantum number.

NROF jobiphs — Number of *JOBIPH* files used as input. This keyword should be followed by the number of states to be read from each *JOBIPH*. Further, one line per *JOBIPH* is required with a list of the states to be read from the particular file. See sample input below. Alternatively, the first line can contain the number of *JOBIPH* used as input followed by the word ALL, indicating that all states will be taken from each file. In this case no further lines are required. For *JOBIPH* file names, see the Files section. Note: If this keyword is missing, then by default all files named 'JOB001', 'JOB002', etc. will be used, and all states found on these files will be used.

IPHNames — Followed by one entry for each *JOBIPH* file to be used, with the name of each file. Note: This keyword presumes that the number of *JOBIPH* files have already been entered using keyword NROF. For default *JOBIPH* file names, see the Files section. The names will be truncated to 8 characters and converted to uppercase.

SHIFt — The next entry or entries gives an energy shift for each wave function, to be added to diagonal elements of the Hamiltonian matrix. This may be necessary e.g. to ensure that an energy crossing occurs where it should. NOTE: The number of states must be known (See keyword NROF) before this input is read. In case the states are not orthonormal, the actual quantity added to the Hamiltonian is 0.5D0*(ESHFT(I)+ESHFT(J))*OVLP(I,J). This is necessary to ensure that the shift does not introduce artificial interactions. SHIFT and HDIAG can be used together.

HDIAg — The next entry or entries gives an energy for each wave function, to replace the diagonal elements of the Hamiltonian matrix. Non-orthogonality is handled similarly as for the SHIFT keyword. SHIFT and HDIAG can be used together.

NATOrb — The next entry gives the number of eigenstates, for which natural orbitals will be computed. They will be written, formatted, commented, and followed by natural occupancy numbers, on one file each. For file names, see the Files section. The format allows their use as standard orbital input files to other Molcas programs.

ORBItals — Print out the Molecular Orbitals read from each *JOBIPH* file.

OVERlaps — Print out the overlap integrals between the various orbital sets.

CIPRint — Print out the CI coefficients read from *JOBIPH*.

THRS — The next line gives the threshold for printing CI coefficients. The default value is 0.05.

DIPR — The next entry gives the threshold for printing dipole intensities. Default is 1.0D-5.

QIPR — The next entry gives the threshold for printing quadrupole intensities. Default is 1.0D-5. Will overwrite any value chosen for dipole intesities.

QIALL — Print all quadrupole intensities.

RFPE — RASSI will read from *RUNOLD* (if not present defaults to *RUNFILE*) a response field contribution and add it to the Fock matrix.

HCOM — The spin-free Hamiltonian is computed.

HEXT — It is read from the following few lines, as a triangular matrix: One element of the first row, two from the next, etc, as list-directed input of reals.

HEFF — A spin-free effective Hamiltonian is read from *JOBIPH* instead of being computed. It must have been computed by an earlier program. Presently, this is done by a multi-state calculation using CASPT2. In the future, other programs may add dynamic correlation estimates in a similar way.

EJOB — The spin-free effective Hamiltonian is assumed to be diagonal, with energies being read from a *JOBMIX* file from a multi-state CASPT2 calculation. In the future, other programs may add dynamic correlation estimates in a similar way.

TOFILE — Signals that a set of files with data from RASSI should be created. This keyword is necessary if QMSTAT is to be run afterwards.

XVIN — Demand for printing expectation values of all selected one-electron properties, for the input RASSCF wave functions.

XVES — Demand for printing expectation values of all selected one-electron properties, for the spin-free eigenstates.

XVSO — Demand for printing expectation values of all selected one-electron properties, for the spin-orbit states.

EPRG — This computes the g matrix and principal g values for the states lying within the energy range supplied on the next line. A value of 0.0D0 or negative will select only the ground state, a value E will select all states within energy E of the ground state. The states should be ordered by increasing energy in the input. The angular momentum and spin-orbit coupling matrix elements need to be available (use keywords SPIN and PROP). For a more detailed description see ref [84].

MAGN — This computes the magnetic moment and magnetic susceptibility. On the next two lines you have to provide the magnetic field and temperature data. On the first line put the number of magnetic field steps, the starting field (in Tesla), size of the steps (in Tesla), and an angular resolution for sampling points in case of powder magnetization (for a value of 0.0d0 the powder magnetization is deactivated). The second line reads the number of temperature steps, the starting temperature (K), and the size of the temperature steps (K). The angular momentum and spin-orbit coupling matrix elements need to be available (use keywords SPIN and PROP). For a more detailed description see ref [85].

HOP — Enables a trajectory surface hopping (TSH) algorithm which allow non-adiabatic transitions between electronic states during molecular dynamics simulation with DYNAMIX program. The algorithm computes the scalar product of the amplitudes of different states in two consecutive steps. If the scalar product deviates from the given threshold a transition between the states is invoked by changing the root for the gradient computation. The current implementation is working only with SA-CASSCF.

STOVerlaps — Computes only the overlaps between the input states.

TRACk — Tries to follow a particular root during an optimization. Needs two ***JOBIPH*** files (see NrOfJobIphs) with the same number of roots. The first file corresponds to the current iteration, the second file is the one from the previous iteration (taken as a reference). With this keyword `RASSI` selects the root from the first ***JOBIPH*** with highest overlap with the root that was selected in the previous iteration. It also needs MDRlxRoot, rather than RlxRoot, to be specified in `RASSCF`. No other calculations are done by `RASSI` when Track is specified.

DQVD — Perfoms DQΦ diabatization[86] by using properties that are computed with `RASSI`. One must compute 7 properties with RASSI in order for this keyword to work (x,y,z,xx,yy,zz,1/r), and they must be specified in that order. At present, this keyword also requires ALPHa and BETA, where ALPHa is the parameter in front of rr and BETA is the parameter in front of 1/r. When ALPHa and BETA are equal to zero, this method reduces to Boys localized diabatization[87]. At present, this method only works for one choice of origin for each quantity.

ALPHa — ALPHa is the prefactor for the quadrupole term in DQΦ diabatization. This keyword must be used in conjunction with DQVD and BETA. You must specify a real number (e.g. $\alpha = 1.0$ not $\alpha = 1$).

BETA — BETA is the prefactor for the electrostatic potential term in DQΦ diabatization. This keyword must be used in conjunction with DQVD and ALPHa. You must specify a real number (e.g. $\beta = 1.0$ not $\beta = 1$).

TRDI — Prints out the components and the module of the transition dipole vector. Only vectors with sizes large than 1.0D-4 a.u. are printed. See also the TDMN keyword.

TDMN — Prints out the components and the module of the transition dipole vector. On the next line, the minimum size, in a.u., for the dipole vector to be printed must be given.

Input example

¿¿COPY ”Jobiph file 1” JOB001 ¿¿COPY ”Jobiph file 2” JOB002 ¿¿COPY ”Jobiph file 3” JOB003

```
  &RASSI
 NR OF JOBIPHS= 3 4 2 2    -- 3 JOBIPHs. Nr of states from each.
```

```
1 2 3 4; 3 4; 3 4         -- Which roots from each JOBIPH.
CIPR; THRS= 0.02
Properties= 4; 'MltPl  1'  1   'MltPl  1'  3    'Velocity'  1 'Velocity'  3
* This input will compute eigenstates in the space
* spanned by the 8 input functions. Assume only the first
* 4 are of interest, and we want natural orbitals out
NATO= 4
```

8.42 RPA

This module is still undocumented

8.43 SCF

8.43.1 Description

The SCF program of the MOLCAS program system generates closed-shell Hartree-Fock, open-shell UHF, and Kohn Sham DFT wave functions.

The construction of the Fock matrices is either done conventionally from the two-electron integral file *ORDINT*, which was generated in a previous step by the SEWARD integral code, or alternatively (only for closed shell calculations) integral-direct by recomputing all the two-electron integrals when needed [88]. The later route is recommended for large basis sets or molecules, when the two-electron integral file would become extensively large. It is automatically taken, when the SCF program cannot find any *ORDINT* file in the work directory. The direct Fock matrix construction employs an efficient integral prescreening scheme, which is based on differential densities [89, 90]: only those AO integrals are computed, where the estimated contractions with the related differential density matrix elements give significant (Coulomb or exchange) contributions to the (differential) two-electron part of the Fock matrix. Integral prescreening is performed at two levels, (i) at the level of shell quadruples, and (ii) at the level of individual primitive Gaussians. Prescreening at the level of contracted functions is not supported, because this would be inefficient in the context of a general contraction scheme. In order to work with differential density and Fock matrices, a history of these entities over previous iterations has to be kept. All these matrices are partly kept in memory, and partly held on disk. The SCF program either works with simple differences of the actual and the previous density, or alternatively with minimized densities, obtained from linear combinations of the actual density and all the previous minimized densities.
Besides the conventional and the fully-direct algorithms there is also a semi-direct path, which allows for the storage of some of the AO integrals during the first iteration, which then are retrieved from disk in subsequent iterations. That path is taken, if the keyword DISK with an appropriate argument specifying the amount of AO integrals to store is found on the input stream. The semi-direct path is recommended for medium sized problems, where the two-electron integral file would become a bit too large (but not orders of magnitude).

The program contains a feature that allows you to make the orbitals partially populated during the aufbau procedure. This feature in not primarily intended to accelerate the convergence but rather to ensure that you do get convergence in difficult cases. The orbitals are populated with with electrons according to

$$\eta_i = \frac{2}{1 + e^{(\varepsilon_i - \varepsilon_f)/kT}} \tag{8.11}$$

where ε_i is the orbital energy of orbital i and ε_f is the Fermi energy. In this "Fermi aufbau" procedure the temperature is slowly lowered until it reaches a minimum value and then kept constant until a stable closed shell configuration is determined. Then normal SCF iterations will be performed with the selected closed shell configuration. For systems that are not really closed shell systems, for example diradicals, you might end up in the situation that the program does not find any stable closed shell configuration. In that case it will continue to optimize the closed shell energy functional with partial occupation numbers. If this is the case, this is probably what you want, and such orbitals would be ideal as starting orbitals for an MCSCF calculation.

The initial orbital guess is either obtained by diagonalizing the bare nuclei Hamiltonian, from an initial guess produced by the module GUESSORB or from orbitals of a previous Hartree-Fock SCF calculation. These starting orbitals are automatically located in the order

1. SCF orbitals from a previous calculation located in the *RUNFILE*
2. SCF orbitals from a previous calculation located in a formatted orbitals file, *INPORB*.
3. initial guess orbitals from module GUESSORB located in the *RUNFILE* and

The program has three types of convergence accelerating schemes: (i) dynamic damping [91], (ii) the C^2-DIIS method using the orbital gradient as error vector [92], and (iii) a combined second-order update/C^2-DIIS procedure. The latter eliminates the Brillouin violating elements of the Fock matrix by proper orbital rotations and hence avoids diagonalization of the Fock matrix: the approximate inverse Hessian is updated (BFGS) in a first step, and then the new orbital displacement vector is obtained from the updated Hessian using C^2-DIIS extrapolation [93]. Dynamic damping gives substantial improvements in highly anharmonic regions of the energy hyper surface, while the second-order update/C^2-DIIS procedure exhibits excellent convergence for less anharmonic regions. By default, dynamic damping is used during the first few iterations. When the change in the density between two subsequent iterations drops below a certain threshold the second-order update/C^2-DIIS procedure kicks in. It is also possible to use the older first order C^2-DIIS scheme instead of the second-order update/C^2-DIIS procedure by setting the density threshold for the latter to zero in the corresponding input card (keyword QNRTHRESHOLD).

By default SCF behaves in different ways depending on what kind of start orbitals are found according to

1. No start orbitals are found. In this case the core hamiltonian is diagonalized and these orbitals are used as start. The "Fermi aufbau" procedure is used until a stable configuration is found.
2. Start orbitals from GUESSORB are found. In this case the HOMO LUMO gap is analyzed and if it is small the "Fermi aufbau" procedure is used until a stable configuration is found. Otherwise the configuration suggested by GUESSORB is used.
3. Start orbitals from a previous SCF calculation is found. The configuration from the previous SCF calculation is used, unless some problem is detected such as partial occupation numbers from an unconverged calculation. In the latter case "Fermi aufbau" is used.

4. Start orbitals from an *INPORB* is in the same way as for start orbitals from an SCF calculation, see last point.

This behavior can be changed by suitable keywords described below.

One of the main objects of the SCF program in the context of the MOLCAS program system is to generate starting orbitals for subsequent MCSCF calculations. Two options are available to improve the canonical Hartree-Fock orbitals in this respect.
(i) It is possible to specify pseudo occupation numbers that are neither zero nor two, thus simulating to some extent an open shell system. The resulting wavefunction does not have any physical meaning, but will provide better starting orbitals for open shell systems.
(ii) Usually, the lowest virtual canonical Hartree-Fock orbitals are too diffuse as correlating orbitals in an MCSCF calculation. If the keyword IVO is encountered in the input stream, the SCF program will diagonalize the core Hamiltonian matrix within the virtual space and write the resulting more compact eigenvectors to the *SCFORB* and *RUNFILE* files, rather than the virtual eigenvectors of the Fock matrix. It should be noted, that this option must never be used, if the SCF wave function itself is used subsequently as a reference function: no MP2 or coupled cluster calculations after an SCF run with IVO !
A further method to generate starting orbitals for MCSCF calculations is to perform an SCF calculation for a slightly positively charged moiety.

8.43.2 Dependencies

The SCF program requires the one-electron integral file *ONEINT* and the communications file *RUNFILE*, which contains among others the basis set specifications processed by SEWARD. For conventional (not integral-direct) runs the two-electron integral file *ORDINT* is required as well. All these files are generated by a preceding SEWARD run.

8.43.3 Files

Below is a list of the files that are used/created by the program SCF.

Input files

SCF will use the following input files: *ONEINT*, *ORDINT*,*RUNFILE*, *INPORB* (for more information see 7.1.2).

Output files

File	*Contents*
SCFORB	SCF orbital output file. Contains the canonical Hartree-Fock orbitals for closed shell calculations. If the IVO option was specified, the virtual orbitals instead are those that diagonalize the bare nuclei Hamiltonian within that subspace.

UHFORB	Contains the canonical Hartree-Fock orbitals for open shell calculations.
UNAORB	This file is produced if you make a UHF calculation and it contain natural orbitals.
MD_SCF	Molden input file for molecular orbital analysis.

8.43.4 Input

Below follows a description of the input to SCF.

The input for each module is preceded by its name like:

```
&SCF
```

Argument(s) to a keyword, either individual or composed by several entries, can be placed in a separated line or in the same line separated by a semicolon. If in the same line, the first argument requires an equal sign after the name of the keyword.

Basic general keywords

Below is a list of keywords that should cover the needs of most users.

Keyword	*Meaning*
TITLe	One line for the title
UHF	Use this keyword to run Unrestricted Hartree-Fock code. Note that current implementation of UHF code has some restrictions, and not all features of SCF program are supported.
ZSPIN	Use this keyword to specify the difference in the number of α and β electrons in the system. The default is 0 or 1 depending on if there is an even or odd number of electrons. Any value different from 0 requires the UHF keyword. This keyword is not needed when you specify the number of electrons with the keyword OCCUpied.
SPIN	Alternative way of specifying the electronic spin of the system. The keyword is followed by an integer giving the value of spin multiplicity $(2S + 1)$. Default is 1 (singlet) or 2 (doublet) depending on if there is an even or odd number of electrons. Any value different from 1 requires the UHF keyword.
KSDFT	Use this Keyword to do the Density Functional Theory Calculations. This Keyword should be followed by functional Keyword: BLYP, B3LYP, B3LYP5, HFB, HFS, LDA, LDA5, LSDA, LSDA5, SVWN, SVWN5, TLYP, PBE, PBE0, M06, M062X, M06HF, M06L. Example: KSDFT=B3LYP

CHARge

Use this keyword to set the number of electrons in the system. This number is defined by giving the net charge of the system. If this keyword is not specified, the molecule is assumed to have net charge zero. The input is given as

```
Charge=n
```

where n is the charge of the system.

OCCUpied

Use this keyword to set the number of electrons in the system. This number is defined by giving the number of electron pairs per irreducible representation of the subgroup of D_{2h} used in the calculation. You can use one and only one of the keywords, CHARge and OCCUpied for this purpose. If neither of these keywords are specified CHARge is assumed with a net charge of zero. It should be noted that the "fermi aufbau" procedure is not used when you specify this keyword. The input for one of the point groups D_2, C_{2h} or C_{2v} is given as

```
OCCUpied= n1 n2 n3 n4
```

where n1 is the number of electron pairs (occupied orbitals) in the first irreducible representation, etc.

If UHF keyword was specified, occupation numbers must be specified in two lines: for alpha and beta spins

FERMi

Use this keyword to specify that you want to use the "Fermi aufbau" procedure for the first few iterations to ensure convergence. The orbitals will be partially populated according to a Fermi population. The input is gives as

```
Fermi= m
```

where m is the temperature parameter according to

- m=0: No temperature is used. Not recommended.
- m=1: A low temperature is used and will yield swift convergence for well behaved systems.
- m=2: A medium low temperature is used and will yield swift and safe convergence for most systems. This is the default value.
- m=3: A medium temperature is used and you will obtain good convergence for closed shell systems. If the system is not a closed shell system, the temperature dependent aufbau procedure may not terminate. This will result in a density matrix with fractional occupation numbers.
- m=4: A medium high temperature is used and the temperature dependent aufbau procedure will most probably not terminate. This is useful for generating starting orbitals for an MCSCF calculation.
- m=5: A high temperature is used. Behaves as m=4 only more so.

It should be noted that only dynamic damping is used until the program have found a stable closed shell configuration. When this have happened the more efficient methods: the ordinary C2-DIIS and the second order update/C2-DIIS procedure, are enabled.

CHOLesky — SCF will use Cholesky (or RI/DF) representation of the two-electron integrals to compute the corresponding contributions to the Fock or KS matrices. The default (LK) algorithm is used. The configuration may be tailored using the ChoInput section. Default is to not use Cholesky unless the Cholesky (or RI/DF) representation of the two-electron integrals has been produced by SEWARD.

CHOInput — This marks the start of an input section for modifying the default settings of the Cholesky SCF. Below follows a description of the associated options. The options may be given in any order, and they are all optional except for ENDChoinput which marks the end of the CHOInput section.

NoLK — Available only within ChoInput. Deactivates the “Local Exchange” (LK) screening algorithm [81] in computing the Fock matrix. The loss of speed compared to the default algorithm can be substantial, especially for electron-rich systems. Default is to use LK.

DMPK — Available only within ChoInput. Modifies the thresholds used in the LK screening. The keyword takes as argument a (double precision) floating point (non-negative) number used as correction factor for the LK screening thresholds. The default value is 1.0d0. A smaller value results in a slower but more accurate calculation.
Note: The default choice of the LK screening thresholds is tailored to achieve as much as possible an accuracy of the converged SCF energy consistent with the choice of the Cholesky decomposition threshold.

NODEcomposition — Available only within ChoInput. Deactivates the Cholesky decomposition of the AO 1-particle density matrix. The Exchange contribution to the Fock matrix is therefore computed using occupied canonical orbitals instead of (localized) “Cholesky MOs” [62]. This choice tends to lower the performances of the LK screening. Default is to perform this decomposition in order to obtain the Cholesky MOs.

TIME — Activates printing of the timings of each task of the Fock matrix build. Default is to not show these timings.

MEMFraction — Set the fraction of memory to use as global Cholesky vector buffer. Default: for serial runs 0.0d0; for parallel runs 0.3d0.

CONStraints — Performs a Constrained (Natural Orbitals) SCF calculation, available only in combination with Cholesky or RI integral representation. An example of input for the keyword CONS is the following:

```
CONStraints
```

```
 2  3
 1 -1
 1  1  1
ADDCorrelation
pbe
SAVErage
```

The keyword CONS has two compulsory arguments: the number of constrained NOs (in each irrep) to be used in the CNO-SCF calculation, followed by one line per irrep specifying the spin configuration of the so-called (+) wavelet (-1 –¿ beta, 1 –¿ alpha) The OPTIONAL keyword ADDC is used to include a correlation energy correction through a DFT functional specified as argument (LDA, LDA5, PBE and BLYP available at the moment) The OPTIONAL keyword SAVE forces the program to use spin-averaged wavelets.

OFEMbedding Performs a Orbital-Free Embedding (OFE)SCF calculation, available only in combination with Cholesky or RI integral representation. The runfile of the environment subsystem renamed AUXRFIL is required. An example of input for the keyword OFEM is the following:

```
OFEMbedding
 ldtf/pbe
dFMD
 1.0   1.0d2
FTHAw
 1.0d-4
```

The keyword OFEM requires the specification of two functionals in the form fun1/fun2, where fun1 is the functional used for the Kinetic Energy (available functionals: Thomas-Fermi, with acronym LDTF, and the NDSD functional), and where fun2 is the xc-functional (LDA, LDA5, PBE and BLYP available at the moment). The OPTIONAL keyword dFMD has two arguments: first, the fraction of correlation potential to be added to the OFE potential (zero for KSDFT and one for HF); second, the exponential decay factor for this correction (used in PES calculations). The OPTIONAL keyword FTHA is used in a freeze-and-thaw cycle (EMIL Do While) to specify the (subsystems) energy convergence threshold.

ITERations Specifies the maximum number of iterations. The default is 400 which is also the largest number you can specify.

CORE The starting vectors are obtained from a diagonalization of the core Hamiltonian.

LUMORB The starting vectors are taken from a previous *SCFORB* file called *INPORB*.

FILEORB The starting vectors are taken from a previous *SCFORB* file, specified by user.

GSSRunfile The starting vectors are taken from the orbitals produced by GUESSORB.

HLGAp

This keyword is used to make the program level shift the virtual orbitals in such a way that the HOMO LUMO gap is at least the value specified on the next line. This will help convergence in difficult cases but may lead to that it converges to an excited configuration. A suitable value is 0.2.

Advanced general keywords

Keyword *Meaning*

SCRAmble

This keyword will make the start orbitals slightly scrambled, accomplished by making a few small random orbital rotations. How much the orbitals are scrambled is determined by the parameter read on the next entry. A reasonable choice for this parameter is 0.2 which correspond to maximum rotation angle of $\arcsin 0.2$. Using this keyword may be useful for UHF calculations with same number of α and β electrons that are not closed shell cases.

ORBItals

Specifies the number of orbitals in the subspace of the full orbital space defined by the basis set, in which the SCF energy functional is optimized. The size of this subspace is given for each of the irreducible representations of the subgroup of D_{2h}. If this keyword is not specified when starting orbitals are read, the full orbital space is assumed. The keyword takes as argument *nIrrep* (# of irreps) integers. **Note** that this keyword is only meaningful when the `SCF` program is fed with input orbitals (cf. LUMORB).

FROZen

Specifies the number of orbitals not optimized during iterative procedure. The size of this subspace is given for each of the irreducible representations of the subgroup of D_{2h}. If this keyword is not specified the number of frozen orbitals is set to zero for each irreducible representation. If the starting vectors are obtained from a diagonalization of the bare nuclei Hamiltonian the atomic orbitals with the lowest one-electron energy are frozen. If molecular orbitals are read from ***`INPORB`*** the frozen orbitals are those that are read in first in each symmetry. The keyword takes as argument *nIrrep* (# of irreps) integers.

OVLDelete

Specifies the threshold for deleting near linear dependence in the basis set. The eigenvectors of the overlap matrix with eigenvalues less than that threshold are removed from the orbital subspace, and do not participate in the optimization procedure. The default value is 1.0d-5. The keyword takes as argument a (double precision) floating point number. Note that the ***`SCFORB`*** file will contain the deleted orbitals as a complementary set to the actual SCF orbitals! In future use of this orbital file the complementary set should always be deleted from use.

PRORBITALS

Specifies which orbitals are to be printed in the log file (standard output). The keyword takes as argument two integers. The possible values are:

0 — No orbitals printed.

1 — orbitals with orbital energies smaller than $2E_{\text{homo}} - E_{\text{lumo}}$ are printed.

2 — followed by real number (ThrEne) — orbitals with orbital energies smaller than ThrEne are printed.

Default value is 1.

Second (optional) argument specifies a format:

0 — No orbitals printed

1 — Print only one-electron energies and Occupation numbers

2 — Short print format

3 — Extended print format

Default value is 3 for small numbers of MOs and 2 for number of MOs> 256.

PRLSCF

Specifies the general print level of the calculation. An integer has to be supplied as argument. The default value, 1, is recommended for production calculations.

ROBU

Robust LDF integral representation (non-hybrid KS-DFT only). Requires Local Density Fitting (LDF) in SEWARD. This is the default for LDF.

NR-2

Nonrobust LDF integral representation with 2-index integrals only (non-hybrid KS-DFT only). Requires Local Density Fitting (LDF) in SEWARD. Default is robust integral representation.

NR-3

Nonrobust LDF integral representation with 3-index integrals only (non-hybrid KS-DFT only). Requires Local Density Fitting (LDF) in SEWARD. Default is robust integral representation.

XIDI

Use exact integral diagonal blocks with LDF. Reduces the risk of negative eigenvalues of the approximate integral matrix. Default is to not use exact integral diagonal blocks.

THRESHOLDS

Specifies convergence thresholds. Four individual thresholds are specified as arguments, which have to be fulfilled simultaneously to reach convergence: EThr, DThr and FThr specify the maximum permissible difference in energy, density matrix elements and Fock matrix elements, respectively, in the last two iterations. The DltNTh finally specifies the norm of the orbital displacement vector used for the orbital rotations in the second-order/C^2-DIIS procedure. The corresponding values are read in the order given above. The default values are 1.0d-9, 1.0d-4, 1.5d-4, and 0.2d-4, respectively. **Note** that these thresholds automatically define the threshold used in the direct Fock matrix construction to estimate individual contributions to the Fock matrix such that the computed energy will have an accuracy that is better than the convergence threshold.

NODIis Disable the DIIS convergence acceleration procedure.

DIISthr Set the threshold on the change in density, at which the DIIS procedure is turned on. The keyword takes as argument a (double precision) floating point number. The default value is 0.15.

QNRThr Set the threshold on the change in density, at which the second-order/C^2-DIIS procedure kicks in. The keyword takes as argument a (double precision) floating point number. The default value is 0.15. **Note:** the change in density has to drop under both the DIIS thr and the QNRThr threshold, for the second-order/C^2-DIIS to be activated. If the latter is set to zero the older first order C^2-DIIS procedure will be used instead.

C1DIis Use C^1-DIIS for convergence acceleration rather that C^2-DIIS which is the default (not recommended).

NODAmp Disable the Damping convergence acceleration procedure.

OCCNumbers Gives the option to specify occupation numbers other than 0 and 2. This can be useful for generating starting orbitals for open shell cases. It should be noted however, that it is still the closed shell SCF energy functional that is optimized, thus yielding unphysical energies. Occupation numbers have to be provided for all occupied orbitals. In the case of UHF calculation occupation numbers should be specified on two different entries: for alpha and beta spin.

IVO Specifies that the virtual orbitals are to be improved for subsequent MCSCF calculations. The core Hamiltonian is diagonalized within the virtual orbital subspace, thus yielding as compact orbitals as possible with the constraint that they have to be orthogonal to the occupied orbitals. **Note** that this option must not be used whenever the Hartree-Fock wavefunction itself is used as a reference in a subsequent calculation.

NOMInimization Program will use density differences $D^{(\mathrm{k})} - D^{(\mathrm{k}-1)}$ rather than minimized differences.

ONEGrid Disable use of a smaller intermediate grid in the integration of the DFT functional during the first SCF iterations.

RFPErt This keyword will add a constant reaction field perturbation to the bare nuclei hamiltonian. The perturbation is read from *RUNOLD* (if not present defaults to *RUNFILE*) and is the latest self consistent perturbation generated by one of the programs SCF or RASSCF.

STAT This keyword will add an addition print outs with statistic information.

For calculations of a molecule in a reaction field see section 8.16.1 of the present manual and section 10.6 of the examples manual.

DFT functionals: Below is listed the keywords for the DFT functionals currently implemented in the package.

Keyword	*Meaning*
LSDA, LDA, SVWN	Vosko, Wilk, and Nusair 1980 correlation functional fitting the RPA solution to the uniform electron gas [94] (functional III in the paper).
LSDA5, LDA5, SVWN5	Functional V from the VWN80 paper [94] which fits the Ceperley-Alder solution to the uniform electron gas.
HFB	Becke's 1988 exchange functional which includes the Slater exchange along with corrections involving the gradient of the density [95].
HFS	$\rho^{4/3}$ with the theoretical coefficient of 2/3 also known as Local Spin Density exchange [96, 97, 98].
HFB86	Becke's 1986 two-parameter exchange functional which includes the Slater exchange along with corrections involving the gradient of the density [99, 100].
HFO	Handy's stand-alone OPTX exchange functional [101]
BLYP	Becke's 1988 exchange functional which includes the Slater exchange along with corrections involving the gradient of the density [95]. Correlation functional of Lee, Yang, and Parr, which includes both local and non-local terms [102, 103].
BPBE	Becke's 1988 exchange functional which includes the Slater exchange along with corrections involving the gradient of the density [95] , combined with the GGA correlation functional by Perdew, Burke and Ernzerhof [104]
B3LYP	Becke's 3 parameter functional [105] with the form $A * E_X^{Slater} + (1-A) * E_X^{HF} + B * \Delta E_X^{Becke} + E_C^{VWN} + C * \Delta E_C^{non-local} \qquad (8.12)$ where the non-local correlation functional is the LYP functional and the VWN is functional III (not functional V). The constants A, B, C are those determined by Becke by fitting to the G1 molecule set, namely A=0.80, B=0.72, C=0.81.
B3LYP5	Becke's 3 parameter functional [105] with the form $A * E_X^{Slater} + (1-A) * E_X^{HF} + B * \Delta E_X^{Becke} + E_C^{VWN} + C * \Delta E_C^{non-local} \qquad (8.13)$ where the non-local correlation functional is the LYP functional and the VWN is functional V. The constants A, B, C are those determined by Becke by fitting to the G1 molecule set, namely A=0.80, B=0.72, C=0.81.

B2PLYP

Grimme's double-hybrid density functional [106] based on Becke;s 1988 exchange and LYP correlation GGA functionals with the form

$$A * E_X^{Slater} + (1-A) * E_X^{HF} + A * \Delta E_X^{Becke} + C * E_C^{LYP} + (1-C) * E_C^{PT2} \tag{8.14}$$

The constants A, and C are A=0.47, C=0.73. The SCF program computes only the DFT part of the B2PLYP energy. In order to get the PT2 term, one has to run the MBPT2 program on converged B2PLYP orbitals, and scale the MP2 correlation energy by the factor (1-C)=0.27.

B86LYP

Becke's 1986 exchange [99, 100] functional combined with the LYP correlation [102, 103]

BWig

Becke's 1988 GGA exchange functional combined with the local Wigner correlation functional [107]

GLYP

Gill's 1996 GGA exchange functional [108] combined with the combined with the LYP correlation [102, 103]

OLYP

Handy's OPTX exchange functional [101] combined with the LYP correlation [102, 103]

OPBE

Handy's OPTX exchange functional [101] combined with the PBE correlation[104]

O3LYP

A hybrid density functional based on the OPTX exchange [109] , with the form

$$A * E_X^{HF} + B * E_X^{Slater} + C * \Delta E_X^{OPTX} + 0.19 * E_C^{VWN} + 0.81 * \Delta E_C^{LYP} \tag{8.15}$$

The constants A, B, C are as provided in Ref. [109] : A=0.1161, B=0.9262, C=0.8133.

KT3

The exchange-correlation functional by Keal and Tozer, 2004 [110, 111]

TLYP

$$E_X^{HF} + E_C^{non-local} \tag{8.16}$$

where the non-local correlation functional is the LYP functional

PBE

The Perdew, Burke, Ernzerhof GGA functional 1996 [104].

PBE0

The Perdew, Ernzerhof, Burke non-empirical hybrid functional 1996 [112].

PBEsol

The Perdew et al. 2008 modification of PBE for solids

RGE2

The regularized gradient approximation (RGE2)exchange functional by Ruzsinszky, Csonka, and Scuseria, 2009 that contains higher-power s terms in the exchange functional, as compared to the PBEsol. It is coupled with the PBEsol correlation [113]

PTCA	The correlation functional by Tognetti, Cortona, and Adamo combined with the PBE exchange [114]
SSB	The exchange functional SSB-sw by Swart, Sola, and Bickelhaupt, 2008 [115] that switches between the OPTX exchange for small values of the reduced density gradient and the PBE exchange for the large ones. It is combined with the PBE correlation functional.
M06	The M06 functional of the Minnesota 2006 class of functionals by Zhao and Truhlar [116, 117, 118, 119]
M06L	The M06-L functional of the Minnesota 2006 class of functionals by Zhao and Truhlar [116, 117, 118, 119]
M06HF	The M06-HF functional of the Minnesota 2006 class of functionals by Zhao and Truhlar [116, 117, 118, 119]
M062X	The M06-2X functional of the Minnesota 2006 class of functionals by Zhao and Truhlar [116, 117, 118, 119]

Keywords for direct calculations

Note again that the threshold for contributions to the Fock matrix depends on the convergence thresholds mentioned above. The choice between the conventional and direct SCF methods is based on the presence of a two-electron integral file (file *ORDINT*). The keyword Direct in the SEWARD input controls that no two-electron integral file is to be generated and that integral direct algorithms can be used in subsequent modules. Thus, *the choice between conventional and direct SCF is done already in the input for the integral program* SEWARD. The direct (or semi-direct) path will be taken whenever there are no two-electron integrals available.

Keyword	*Meaning*
CONVentional	This option will override the automatic choice between the conventional and the direct SCF algorithm such that the conventional method will be executed regardless of the status of the *ORDINT* file.
DISK	This option enables/disables the semi-direct algorithm. It requires two arguments which specifies the max Mbyte of integrals that are written on disk during the first iteration (and retrieved later in subsequent iterations) and the size of the corresponding I/O buffer in kbyte. The default values are 2000 MByte and 512 kByte. In case the specified disk space is zero and the I/O buffer is different from zero it will default to a semi-direct SCF with in-core storage of the integrals. The size of the memory for integrals storage is the size of the I/O buffer. If the size of the disk is non-zero and the I/O buffer size is zero the latter will be reset to the default value.

THIZe — This option specifies a threshold for two-electron integrals. Only integrals above this threshold (but not necessarily all of those) are kept on disk for the semi-direct algorithm. The keyword takes as argument a (double precision) floating point number.

SIMPle — If this option is specified, only a simple prescreening scheme, based solely on the estimated two-electron integral value will be employed (no density involved).

Limitations

The limitations/MODULE on the number of basis functions are the same as specified for SEWARD.

Input examples

First we have the bare minimum of input. This will work well for almost all systems containing an even number of electrons.

```
&SCF
```

The next example is almost as simple. Here we have an open shell case, i.e. you have an odd number of electrons in the neutral system and you need to generate starting orbitals for RASSCF. In this case we recommend that you perform a calculation on the cation with the input below.

```
&SCF; Charge= 1
```

The next example explains how to run UHF code for a nitrogen atom:

```
&SCF; UHF; ZSPIN=3
```

The next example is a bit more elaborate and show how to use a few of the keywords. The system is water that have the electron configuration $1a_1^2 2a_1^2 3a_1^2 1b_1^2 1b_2^2$.

```
 &SCF; Title= Water molecule. Experimental equilibrium geometry. The symmetries are a1, b2, b1 and a2.
Occupied= 3 1 1 0
Threshold= 0.5D-9 0.5D-6 0.5D-6 0.5D-5
* semi-direct algorithm writing max 128k words (1MByte) to disk
* the size of the I/O buffer by default (512 kByte)
Disk= 1 0
Ivo
```

8.44 SEWARD

The SEWARD module generates one- and two-electron integrals needed by other programs. The two-electron integrals may optionally be Cholesky decomposed. In addition, it will serve as the input parser for parameters related to the specification of the quadrature grid used in numerical integration in association with DFT and reaction field calculations.

In the following three subsection we will in detail describe the input parameters for analytic integration, numerical integration, and reaction fields.

8.44.1 Analytic integration

Any conventional ab initio quantum chemical calculation starts by computing overlap, kinetic energy, nuclear attraction and electron repulsion integrals. These are used repeatedly to determine the optimal wave function and the total energy of the system under investigation. Finally, to compute various properties of the system additional integrals may be needed, examples include multipole moments and field gradients.

Description

`SEWARD` is able to compute the following integrals:

- kinetic energy,
- nuclear attraction,
- two electron repulsion (optionally Cholesky decomposed),
- n'th (default n=2) order moments (overlap, dipole moment, etc.),
- electric field (generated at a given point by all charges in the system),
- electric field gradients (spherical gradient operators),
- linear momentum (velocity),
- orbital angular momentum,
- relativistic mass-velocity correction (1st order),
- one-electron Darwin contact term,
- one-electron relativistic no-pair Douglas-Kroll
- diamagnetic shielding,
- spherical well potential (Pauli repulsion),
- ECP and PP integrals,
- modified kinetic energy and multipole moment integrals (integration on a finite sphere centered at the origin) for use in the variational R-matrix approach,
- external field (represented by a large number of charges and dipoles),
- angular momentum products, and
- atomic mean-field integrals (AMFI) for spin-orbit coupling.

Note that while SEWARD *compute these integrals the input to select them and their settings are put in the input of* GATEWAY*.*

In general these integrals will be written to a file, possibly in the form of Cholesky vectors (two-electron integrals only). However, SEWARD can also compute the orbital contributions and total components of these properties if provided with orbital coefficients and orbital occupation numbers.

To generate the one- and two-electron integrals SEWARD uses two different integration schemes. Repulsion type integrals (two- electron integrals, electric field integrals, etc.) are evaluated by the reduced multiplication scheme of the Rys quadrature [120]. All other integrals are computed by the Gauss-Hermite quadrature. SEWARD use spherical Gaussians as basis functions, the only exception to this are the diffuse/polarization functions of the 6-31G family of basis sets. The double coset [121] formalism is used to treat symmetry. SEWARD is especially designed to handle ANO-type basis sets, however, segmented basis sets are also processed.

At present the following limitations are built into SEWARD:

Max number of unique basis functions:	2000
Max number of symmetry independent centers:	500
Highest angular momentum:	15
Highest symmetry point group:	D_{2h}

Dependencies

SEWARD usually runs after program GATEWAY. In the same time, any input used in GATEWAY can be placed into SEWARD input. However, it is recommended to specify all information about the molecule and the basis set in GATEWAY input.

SEWARD does normally not depend on any other code, except of GATEWAY. There are two exceptions however. The first one is when SEWARD is used as a property module. In these cases the file *INPORB* has to have been generated by a wave function code. The second case, which is totally transparent to the user, is when SEWARD picks up the new Cartesian coordinates generated by SLAPAF during a geometry optimization.

Files

Input Files Apart form the standard input file SEWARD will use the following input files: *RYSRW*, *ABDATA*, *RUNFILE*, *INPORB* (for calculation of properties) (7.1.2). In addition, SEWARD uses the following files:

File	*Contents*
BASLIB	The default directory for one-particle basis set information. This directory contains files which are part of the program system and could be manipulated by the user in accordance with the instructions in the section 8.49 and following subsections. New basis set files can be added to this directory by the local MOLCAS administrator.
QRPLIB	Library for numerical mass-velocity plus Darwin potentials (used for ECPs).

Output files In addition to the standard output file `SEWARD` may generate the following files: *`ONEINT`*, *`ORDINT`*, *`CHVEC`*, *`CHRED`*, *`CHORST`*, *`CHOMAP`*, *`CHOR2f`* (7.1.2).

Input

Below follows a description of the input to `SEWARD`. Observe that if nothing else is requested `SEWARD` will by default compute the overlap, the dipole, the quadrupole, the nuclear attraction, the kinetic energy, the one-electron Hamiltonian, and the two-electron repulsion integrals.

The input for each module is preceded by its name like:

```
&SEWARD
```

Argument(s) to a keyword, either individual or composed by several entries, can be placed in a separated line or in the same line separated by a semicolon. If in the same line, the first argument requires an equal sign after the name of the keyword.

General keywords

Keyword	*Meaning*
TITLE	One line of title card follows.
TEST	`SEWARD` will only process the input and generate a non-zero return code.
ONEONLY	`SEWARD` will not compute the two-electron integrals.
NODKROLL	`SEWARD` will not compute Douglas-Kroll integrals.
DIRECT	Prepares for later integral-direct calculations. As with keyword ONEONLY, `SEWARD` will evaluate no two-electron integrals.
EXPERT	Sets “expert mode”, in which various default settings are altered. Integral-direct calculations will be carried out if the two-electron integral file is unavailable.
CHOLESKY	`SEWARD` will Cholesky decompose the two-electron integrals using default configuration (in particular, the decomposition threshold is 1.0d-4) of the decomposition driver. The decomposition threshold can be changed using keyword THRC. Default is to not decompose.
1CCD	`SEWARD` will Cholesky decompose the two-electron integrals using the one-center approximation. The decomposition threshold can be changed using keyword THRC. Default is to not decompose.
THRCHOLESKY	Specify decomposition threshold for Cholesky decomposition of two-electron integrals on the next line. Default value: 1.0d-4.

SPAN Specify span factor ($0 < \text{span} \leq 1$) for Cholesky decomposition of two-electron integrals on the next line. Span=1 implies full pivoting, Span=0 means no pivoting. If the span factor is too low, numerical errors may cause negative diagonal elements and force the program to quit; if the span factor is too large, the execution time may increase. Default value: 1.0d-2.

LOW CHOLESKY `SEWARD` will Cholesky decompose the two-electron integrals using low accuracy (threshold 1.0d-4) configuration of the decomposition driver. Default is to not decompose.

MEDIum CHOLESKY `SEWARD` will Cholesky decompose the two-electron integrals using medium accuracy (threshold 1.0d-6) configuration of the decomposition driver. Default is to not decompose.

HIGH CHOLESKY `SEWARD` will Cholesky decompose the two-electron integrals using high-accuracy (threshold 1.0d-8) configuration of the decomposition driver. Default is to not decompose.

LDF Local Density Fitting: `SEWARD` will fit each AO product using only auxiliary functions centered on the two parent atoms. Fitting coefficients are computed and stored for use in later modules (only non-hybrid KS-DFT at the moment). Requires an auxiliary basis set (generated by RICD in Gateway or externally defined). Can not be combined with Cholesky and (global) DF/RI. LDF is turned off by default.

LDF1 Local Density Fitting using auxiliary functions centered on the two parent atoms of each AO product. Equivalent to keyword LDF. Equivalent to keyword LDF.

LDF2 LDF using auxiliary functions centered on the two parent atoms of each AO product as well as the two-center functions required to achieve the given target accuracy.

TARGet ACCURACY Specify the target accuracy for LDF2 on the next line. Default value: threshold used to generate the aCD or acCD auxiliary basis set (or 1.0d-4 in case of externally defined auxiliary basis sets).

APTH Specify the LDF/LDF2 atom pair prescreening threshold on the next line, thus defining which atom pairs are considered significant. Default value: the target accuracy or the cutoff threshold for primitive integral evaluation, whichever is smaller.

CLDF Constrained LDF/LDF2: specify constraint order on the next line (-1 for unconstrained, 0 for charge constraint). Default: unconstrained LDF.

FAKE CD/RI If CD/RI vectors are already available, `SEWARD` will not redo work!

JMAX The integer entry on the next line is the highest rotational quantum number for which `SEWARD` will compute the rotational energy within the rigid rotor model. The default value is 5.

SYMMETRY — See the the description in the manual for the program `GATEWAY`

BASIS SET — See the the description in the manual for the program `GATEWAY`

ZMAT — See the the description in the manual for the program `GATEWAY`

XBAS — See the the description in the manual for the program `GATEWAY`

XYZ — See the the description in the manual for the program `GATEWAY`

NOGUESSORB — Disable automatic generation of starting orbitals with the GuessOrb procedure.

NODELETE — Do not delete any orbitals automatically.

SDELETE — Set the threshold for deleting orbitals based on the eigenvalues of the overlap matrix. All eigenvalues with eigenvectors below this threshold will be deleted. If you want no orbitals deleted use keyword NODELETE.

TDELETE — Set the threshold for deleting orbitals based on the eigenvalues of the kinetic energy matrix. All eigenvalues with eigenvectors above this threshold will be deleted. If you want no orbitals deleted use keyword NODELETE.

ECPSHOW — Force `SEWARD` to print ECP parameters.

AUXSHOW — Force `SEWARD` to print auxiliary basis set parameters.

BSSHOW — Force `SEWARD` to print basis set parameters.

VERBOSE — Force `SEWARD` to print a bit more verbose.

EMBEDDING — Reads in an embedding potential from a file. It can also write out the density and the electrostatic potential on a grid. It is a block keyword which MUST be ended with ENDEMBEDDING.

The subkeywords are:

EMBInput — Specifies the path to the file which contains the embedding potential (e.g. `EMBI=myEmbPot.dat`). The file contains a potential given on a grid. It has the number of grid points in the first line. Then in five columns data for each grid point is given (x, y, z, weight of this grid point, value of the potential). Default is `EMBPOT`.

OUTGrid — Specifies the path to a file containing a grid on which the output is produced. It is only needed if you want to have the data on a grid different from the one given in EMBINPUT. The columns for the potential and weights are not needed in this file (and also not read in).

WRDEnsity — Switches on the calculation of the final electron density on a grid. The output file path must be specified along with this keyword (e.g. `WRDE=myDens.dat`).

WREPotential Switches on the calculation of the electrostatic potential on a grid. The output file path must be specified along with this keyword (e.g. `WREP=myESP.dat`).

ENDEmbedding Ends the EMBEDDING section. This keyword MUST be present.

The EMBEdding feature is currently only supported by the SCF part of MOLCAS.

Keywords associated to one-electron integrals

Keyword	*Meaning*
MULTIPOLES	Entry which specifies the highest order of the multipole for which integrals will be generated. The default center for the dipole moment operator is the origin. The default center for the higher order operators is the center of the nuclear mass. The default is to do up to quadrupole moment integrals (2).
CENTER	This option is used to override the default selection of the origin of the multipole moment operators. On the first entry add an integer entry specifying the number of multipole moment operators for which the origin of expansion will be defined. Following this, one entry for each operator, the order of the multipole operator and the coordinates of the center (in au) of expansion are specified.
SDIPOLE	Requests computation of velocity integrals.
RELINT	Requests the computation of mass-velocity and one-electron Darwin contact term integrals for the calculation of a first order correction of the energy with respect to relativistic effects.
RXXPYY	Request arbitrary scalar relativistic Douglas-Kroll-Hess (DKH) correction to the one-electron Hamiltonian and the so-called picture-change correction to the property integrals (multipole moments and electronic potential related properties). Here XX represents the order of the DKH correction to the one-electron Hamiltonian and yy the order of the picture-change correction. The character P denotes the parameterization used in the DKH procedure. The possible parametrizations P of the unitary transformation used in the DKH transformation supported by MOLCAS are: `P=O:` Optimum parametrization (OPT) `P=E:` Exponential parametrization (EXP) `P=S:` Square-root parametrization (SQR) `P=M:` McWeeny parametrization (MCW) `P=C:` Cayley parametrization (CAY)

	Hence, the proper keyword for the 4th order relativistically corrected one-electron Hamiltonian and 3rd order relativistically corrected property integrals in the EXP parameterization would read as R04E03. If yy is larger than XX it is set to XX. If yy is omitted it will default to same value as XX. Recommended orders and parametrization is R02O02. Since the EXP parameterization employs a fast algorithm, it is recommended for high order DKH transformation.
RX2C	Request the scalar relativistic X2C (eXact-two-Component) corrections to the one-electron Hamiltonian as well as the property integrals.
RBSS	Request the scalar relativistic BSS (Barysz-Sadlej-Snijders) corrections to the one-electron Hamiltonian as well as the property integrals. The non-iterative scheme is employed for the construction of BSS transformation.
RLOCal	Request local approximation to the relativistic exact decoupling approaches such as X2C, BSS and DKH. This option cannot be used together with point group symmetry.
NOAMfi	Explicit request for no computation of atomic mean-field integrals.
AMFI	Explicit request for the computation of atomic mean-field integrals (used in subsequent spin-orbit calculations). These integrals are computed by default for the ANO-RCC and ANO-DK3 basis sets.
Grid Input	Specification of numerical quadrature parameters, consult the numerical quadrature section of this manual.

Additional keywords for property calculations

Keyword	*Meaning*
VECTors	Requests a property calculation. For this purpose a file, *`INPORB`*, must be available, which contains the MO's and occupation numbers of a wave function. A custom filename can be given with FileOrb.
FILEorb	The next line specifies the filename containing the orbitals and occupation numbers from which the properties will be computed. By default a file named *`INPORB`* will be used.
ORBCon	The keyword will force `SEWARD` to produce a list of the orbital contributions to the properties being computed. The default is to generate a compact list.
THRS	The real entry on the following line specifies the threshold for the occupation number of an orbital in order for the OrbCon option to list the contribution of that orbital to a property. The default is 1.0d-6.

Keywords for two-electron integrals

Keyword	*Meaning*
NOPACK	The two-electron integrals will not be packed. The default is to pack the two-electron integrals.
PKTHRE	An entry specifies the desired accuracy for the packing algorithm, the default is 1.0d-14.
STDOUT	Generate a two-electron integral file according to the standard of version 1 of MOLCAS. The default is to generate the two-electron integrals according to the standard used since version 2 of MOLCAS.
THRESHOLD	Threshold for writing integrals to disk follows. The default is 1.0d-14.
CUTOFF	Threshold for ignoring the calculation of integrals based on the pair prefactor follows. The default is 1.0d-16.

Keywords associated to electron-molecule scattering calculations within the framework of the R-matrix method This section contains keyword which control the radial numerical integration of the diffuse basis functions describing the scattered electrons in the variational R-matrix approach. The activation of this option is controlled by that the center of the diffuse basis is assigned the unique atom label DBAS.

Keyword	*Meaning*
RMAT	Radius of the R-matrix sphere (in Bohr). This sphere is centered at the coordinate origin. The default is 10 Bohr.
RMEA	Absolute precision in radial integration. The default is 1d-9.
RMER	Relative precision in radial integration. The default is 1d-14.
RMQC	Effective charge of the target molecule. This is the effective charge seen by the incident electron outside of the R-matrix sphere. The default is 0d0.
RMDI	Effective dipole of the target molecule. This is the effective dipole seen by the incident electron outside of the R-matrix sphere. The default is (0d0,0d0,0d0).
RMEQ	Minimal value of the effective charge of the target molecule to be considered. This is also the minimal value of the components of the effective dipole to be considered. Default is 1d-8
RMBP	Parameter used for test purposes in the definition of the Bloch term. Default is 0d0.

CELL	Defines the three vectors of the unit cell ($\vec{e_1}$,$\vec{e_2}$,$\vec{e_3}$). The optional keyword *Angstrom* before the definition of vectors would read data in Å. Must consist of three entries (four in the case of Å) which correspond to coordinates of the vectors. All the atoms which are defined after that key are considered as the atoms of the cell.
SPREAD	Three integer numbers n_1, n_2, n_3 which define the spread of the unit cell along the unit cell vectors. For example, 0 0 2 would add all cell's atoms translated on $-2\vec{e_3}$, $-\vec{e_3}$, $\vec{e_3}$, $2\vec{e_3}$. This key must be placed **before** the definition of the unit cell atoms.

Below follows an input for the calculation of integrals of a carbon atom. The comments in the input gives a brief explanation of the subsequent keywords.

```
 &SEWARD
Title= This is a test deck!
* Remove integrals from a specific irreps
Skip= 0 0 0 0 1 1 1 1
* Requesting only overlap integrals.
Multipole= 0
* Request integrals for diamagnetic shielding
DSHD= 0.0 0.0 0.0; 1; 0.0 0.0 0.0
* Specify a title card
* Request only one-electron integrals to be computed
OneOnly
* Specify group generators
Symmetry= X Y Z
* Enable an inline basis set
Expert
* Specify basis sets
Basis set
C.ANO-L...6s5p3d2f.
Contaminant d
C  0.0 0.0 0.0
End of basis
```

The basis set label and the all electron basis set library The label, which defines the basis set for a given atom or set of atoms, is given as input after the keyword Basis set. It has the following general structure (notice that the last character is a period):

atom.type.author.primitive.contracted.aux.

where the different identifiers have the following meaning:

Identifier	*Meaning*
`atom`	Specification of the atom by its chemical symbol.
`type`	Gives the type of basis set (ANO, STO, ECP, etc.) according to specifications given in the basis set library, *vide supra*. Observe that the upper cased character of the type label defines the file name in the basis directory.
`author`	First author in the publication where that basis set appeared.

`primitive` Specification of the primitive set (e.g. 14s9p4d3f).

`contracted` Specification of the contracted set to be selected. Some basis sets allow only one type of contraction, others all types up to a maximum. The first basis functions for each angular momentum is then used. **Note**, for the basis set types ANO and ECP, on-the-fly decontraction of the most diffuse functions are performed in case the number of contracted functions specified in this field exceeds what formally is specified in the library file.

`aux` Specification of the type of AIMP, for instance, to choose between non-relativistic and relativistic core AIMP's.

Only the identifiers `atom`, `type`, and `contracted` have to be included in the label. The others can be left out. However, the periods have to be kept. Example — the basis set label '`C.ano-s...4s3p2d.`' will by MOLCAS be interpreted as '`C.ano-s.Pierloot.10s6p3d.4s3p2d.`', which is the first basis set in the ANO-S file in the basis directory that fulfills the specifications given.

More information about basis set format can be found in the section Advanced examples.

8.44.2 Numerical integration

Various Density Functional Theory (DFT) models can be used in MOLCAS . Energies and analytical gradients are available for all DFT models. In DFT the exact exchange present in HF theory is replaced by a more general expression, the exchange-correlation functional, which accounts for both the exchange energy, $\mathrm{E}_X[P]$ and the electron correlation energy ,$\mathrm{E}_C[P]$.

Description

We shall now describe briefly how the exchange and correlation energy terms look like. The functionals used in DFT are integrals of some function of the electron density and optionally the gradient of the electron density

$$E_X[P] = \int f(\rho_\alpha(r), \rho_\beta(r), \nabla\rho_\alpha(r), \nabla\rho_\beta(r))dr \tag{8.17}$$

The various DFT methods differ in which function, f, is used for $\mathrm{E}_X[P]$ and for $\mathrm{E}_C[P]$. In MOLCAS pure DFT methods are supported, together with hybrid methods, in which the exchange functional is a linear combination of the HF exchange and a functional integral of the above form. The latter are evaluated by numerical quadrature. In the `SEWARD` input the parameters for the numerical integration can be set up. In the `SCF` and `RASSCF` inputs the keywords for using different functionals can be specified. Names for the various pure DFT models are given by combining the names for the exchange and correlation functionals.

The DFT gradients has been implemented for both the fixed and the moving grid approach [122, 123, 124]. The latter is known to be translationally invariant by definition and is recommended in geometry optimizations.

Input

Below follows a description of the input to the numerical integration utility in the `SEWARD` input.

Compulsory keywords

Keyword	*Meaning*
GRID INPUT	This marks the beginning of the input to the numerical integration utility.
END OF GRID-INPUT	This marks the end of the input to the numerical integration utility.

Optional keywords

Keyword	*Meaning*
GRID	It specifies the quadrature quality. The possible indexes that can follow are COARSE, SG1GRID, FINE, ULTRAFINE following the Gaussian98 convention. Default is FINE.
RQUAD	It specifies the radial quadrature scheme. Options are LOG3 (Mura and Knowles)[125], BECKE (Becke)[126], MHL (Murray et al.)[127], TA (Treutler and Ahlrichs, defined for H-Kr)[128], and LMG (Lindh et al.)[129], respectively. The default is MHL.
GGL	It activates the use of Gauss and Gauss-Legendre angular quadrature. Default is to use the Lebedev angular grid.
LEBEDEV	It turns on the Lebedev angular grid.
LOBATTO	It activates the use of Lobatto angular quadrature. Default is to use the Lebedev angular grid.
LMAX	It specifies the angular grid size. Default is 29.
NGRID	It specifies the maximum number of grid points to process at one instance. Default is 128 grid points.
NOPRUNNING	It turns off the the angular prunning. Default is to prune.
NR	It is followed by the number of radial grid points. Default is 75 radial grid points.
FIXED GRID	Use a fixed grid in the evaluation of the gradient. This corresponds to using the grid to numerically evaluate the analytic gradient expression. Default is to use a moving grid.
MOVING GRID	Use a moving grid in the evaluation of the gradient. This correspond to evaluating the gradient of the numerical expression of the DFT energy. This is the default.

THREshold It is followed by the value for the the radial threshold. Default value is 1.0D-13.

T_X Threshold for screening in the assembling of the density on the grid. Default value is 1.0D-18.

T_Y Threshold for screening in the assembling of the integrals. Default value is 1.0D-11.

NOSCreening Turn off any screening in the numerical integration.

CROWding The crowding factor, according to MHL, used in the pruning of the angular grid close to the nuclei. Default value 3.0.

The SCF and RASSCF programs have their own keywords to decide which functionals to use in a DFT calculation.

Below follows an example of a DFT calculation with two different functionals.

```
 &GATEWAY
Basis set
H.3-21G.....
H1 0.0  0.0 0.0
End of basis
 &SEWARD
Grid input
 RQuad= Log3; nGrid= 50000; GGL; lMax= 26; Global
End of Grid Input
 &SCF; Occupations=1; KSDFT=LDA5; Iterations= 1 1
 &SCF; Occupations= 1; KSDFT=B3LYP; Iterations= 1 1
```

8.44.3 Relativistic operators

The current different implementation of all relativistic operators in MOLCAS as described in the following subsubsections has been programmed and tested in Ref.[130]

Using the Douglas–Kroll–Hess Hamiltonian

For all-electron calculations, the preferred way is to use the scalar-relativistic Douglas–Kroll–Hess (DKH) Hamiltonian, which, in principle, is available up to arbitrary order in Molcas; for actual calculations, however, the standard 2nd order is usually fine, but one may use a higher order that 8th order by default to be on the safe side.

The arbitrary-order Hamiltonian is activated by setting

```
RXXPyy
```

somewhere in the SEWARD input, where the XX denotes the order of the DKH Hamiltonian in the external potential. I.e., for the standard 2nd-order Hamiltonian you may use R02O. Note in particular that the parametrization P does not affect the Hamiltonian up to fourth order. Therefore, as long as you run calculations with DKH Hamiltonians below 5th order you may use any symbol for the parametrization as they would all yield the same results.

The possible parametrizations P of the unitary transformation used in the DKH transformation supported by MOLCAS are:

P=O: Optimum parametrization (OPT)

P=E: Exponential parametrization (EXP)

P=S: Square-root parametrization (SQR)

P=M: McWeeny parametrization (MCW)

P=C: Cayley parametrization (CAY)

Up to fourth order (XX=04) the DKH Hamiltonian is independent of the chosen parametrization. Higher-order DKH Hamiltonians depend slightly on the chosen parametrization of the unitary transformations applied in order to decouple the Dirac Hamiltonian. Since the EXP parameterization employs a fast algorithm [131], it is recommended for high-order DKH transformation.

For details on the arbitrary-order DKH Hamiltonians see [132] with respect to theory, [133] with respect to aspects of implementation, and [134] with respect to general principles of DKH. The current version of MOLCAS employs different algorithms, but the polynomial cost scheme of the DKH implementation as described in [131] is used as the default algorithm. The implementation in MOLCAS has been presented in [130].

For details on the different parametrizations of the unitary transformations see [135].

Douglas–Kroll–Hess transformed properties

As mentioned above, four-component molecular property operators need to be DKH transformed as well when going from a four-component to a two- or one-component description; the results do not coincide with the well-known corresponding nonrelativistic expressions for a given property but are properly picture change corrected.

The transformation of electric-field-like molecular property operators can be carried out for any order smaller or equal to the order chosen for the scalar-relativistic DKH Hamiltonian. In order to change the default transformation of order 2, you may concatenate the input for the DKH Hamiltonian by 2 more numbers specifying the order in the property,

```
RxxPyy
```

where yy denotes the order of the Hamiltonian starting with first order 01. The DKH transformation is then done automatically for all one-electron electric-field-like one-electron property matrices.

Also note that the current implementation of both the Hamiltonian and the property operators is carried out in the full, completely decontracted basis set of the molecule under consideration. The local nature of the relativistic contributions is not yet exploited and hence large molecules may require considerable computing time for all higher-order DKH transformations.

For details on the arbitrary-order DKH properties see [136] with respect to theory and [137, 130] with respect to implementation aspects.

Using the X2C/Barysz–Sadlej–Snijders Hamiltonian

Exact decoupling of the relativistic Dirac Hamiltonian can be achieved with infinite-order approaches, such as the so-called X2C (exact–two–component) and BSS (Barysz–Sadlej–Snijders) method. In Molcas, both methods are available for all-electron calculations. The evaluation of transformation matrices employs a non-iterative scheme.

The exact decoupling Hamiltonian is activated by setting either `RX2C` or `RBSS` somewhere in the `SEWARD` input, where `RX2C` and `RBSS` denote using the scalar (one-component) X2C and BSS Hamiltonian respectively. The one-electron Hamiltonian as well as the property integrals will be transformed according to the given exact decoupling method. In other words, all property integrals are by default picture change corrected.

The computation time of the X2C/BSS method is almost the same as of the DKH method at 8th order, while X2C is a little bit faster than BSS since the additional free-particle Foldy–Wouthuysen transformation is skipped in the X2C approach[130]. For molecules including only light atoms, the DKH method with low orders (<8) is enough to account for the relativistic effects.

The differences between different exact decoupling relativistic methods are very small compared to errors introduced by other approximations, such as the basis set incompleteness, approximate density functionals, etc. Therefore, any exact decoupling model is acceptable for the treatment of relativistic effects in molecular calculations.

For details on the exact decoupling approaches see [130] with respect to theories and comparison of numerical results, [138, 139, 140] for the X2C method, and [141, 142] for the BSS method.

Local approximation to relativistic decoupling

The computational cost for relativistic transformations increases rapidly if the molecule becomes larger. The local DLU scheme [143] was proposed to reduce the computational cost based on the atomic decomposition of the (unitary) decoupling transformation. It is important to note that the DLU scheme can be applied to any kind of relativistic approaches mentioned above (i.e., DKH, BSS, and X2C). It was found [143] that the DLU approximation introduces very small errors for total energies, which are less than 0.01 milli-Hartree for molecules including heavy atoms. The local approximation is activated by setting `RLOCal` somewhere in the `SEWARD` input.

The direct local approximation to the Hamiltonian, called DLH in Ref. [143] may be activated by setting `RLOCal=DLH`. However, as DLH is not superior to the DLU scheme, but introduces slightly larger errors, it is not recommended.

The picture-change effect for molecular properties is automatically taken care of when a local approximation is employed for the transformed operator. The default order of DKH transformed properties is set to the same as the order of the DKH Hamiltonian.

It is important to note the local approximation cannot be used together with point group symmetry in the current implementation. Because the relativistic transformation is applied in the molecular orbital (MO) representation instead of the atomic orbital (AO) representation. Thus, the program will report an error and exit if symmetry is used.

8.45 SINGLE_ANISO

The SINGLE_ANISO program is a routine which allows the non-perturbative calculation of effective spin (pseudospin) Hamiltonians and static magnetic properties of mononuclear complexes and fragments completely *ab initio*, including the spin-orbit interaction. As a starting point it uses the results of RASSI calculation for the ground and several excited spin-orbital multiplets. A short description of methodology and applications can be found in [144], [145]. The second version of the SINGLE_ANISO program is able to calculate the following quantities:

- Parameters of pseudospin magnetic Hamiltonians (the methodology is described in [146]):
    1. First rank (linear after pseudospin) Zeeman splitting tensor $g_{\alpha\beta}$, its main values, including the sign of the product $g_X \cdot g_Y \cdot g_Z$, and the main magnetic axes.
    2. Second rank (bilinear after pseudospin) zero-field splitting tensor $D_{\alpha\beta}$, its main values and the anisotropy axes. The anisotropy axes are given in two coordinate systems: a) in the initial Cartesian coordinate system (*x, y, z*) and b) in the coordinate system of the main magnetic axes (*Xm, Ym, Zm*).
    3. Higher rank ZFS tensors (D^4, D^6, ... , etc.) and Zeeman splitting tensors (G^3, G^5, ... , etc.) for complexes with moderate and strong spin-orbit coupling.
    4. Angular moments along the main magnetic axes.
- All (27) parameters of the *ab initio* Crystal field acting on the ground atomic multiplet of lanthanides, and the decomposition of the CASSCF/RASSI wave functions into functions with definite projections of the total angular moment on the quantization axis.
- Static magnetic properties:
    1. Van Vleck susceptibility tensor $\chi_{\alpha\beta}(T)$
    2. Powder magnetic susceptibility function $\chi(T)$
    3. Magnetization vector $\vec{M}(\vec{H})$ for specified directions of the applied magnetic field $\vec{H}$
    4. Powder magnetization $M_{mol}(H)$

The magnetic Hamiltonians are defined for a desired group of N electronic states obtained in RASSI calculation to which a pseudospin $\tilde{S}$ (it reduces to a true spin S in the absence of spin-orbit coupling) is subscribed according to the relation $N = 2\tilde{S}$+1. For instance, the two wave functions of a Kramers doublet correspond to $\tilde{S}$=1/2. The implementation is done for $\tilde{S}$=1/2, 1, 3/2, ... ,15/2.

The calculation of magnetic properties takes into account the contribution of excited states (the ligand-field and charge transfer states of the complex or mononuclear fragment included in the RASSI calculation) via their thermal population and Zeeman admixture. The intermolecular exchange interaction between magnetic molecules in a crystal can be taken into account during the simulation of magnetic properties by a phenomenological parameter zJ specified by the user (see keyword MLTP).

8.45.1 Dependencies

The SINGLE_ANISO program takes all needed *ab initio* information from the *RUNFILE*: i.e. matrix elements of angular momentum, spin-orbit energy spectrum and mixing coefficients, number of mixed states and their multiplicity, etc. In order to find the necessary information in the *RUNFILE*, the keywords MEES and SPIN are mandatory for RASSI. The SEWARD keyword ANGM is also compulsory.

8.45.2 Files

Input files

File	*Contents*
RUNFILE	The file of communication between different modules in MOLCAS. Its presence is mandatory.

Output files

File	*Contents*
$PROJECT.ANISO	This file is intended to be an input for the future POLY_ANISO module in MOLCAS (in development). The creation of this file is done via a specific keyword.

8.45.3 Input

Normally SINGLE_ANISO runs without specifying any of the following keywords. The only unknown variable for SINGLE_ANISO is the dimension (multiplicity) of the pseudospin. By default one multiplet is selected, which has the dimension equal to the multiplicity of the ground term. For example, in cases where spin-orbit coupling is weak, the multiplicity of the effective spin Hamiltonian is usually the same as the multiplicity of the lowest term, while in the cases with strong anisotropy (lanthanide or actinide complexes, Co^{2+} complexes, etc...) the lowest energy levels of the complexes form a group of states which can differ quite strong from the spin multiplicity of the lowest term. In these cases the user should specify the multiplicity corresponding to a chosen value of pseudospin ($2\tilde{\mathrm{S}}+1$). For instance, in Dy^{3+} the spin of the ground state term is $S = 5/2$, but in many situations only the ground Kramers doublet is considered; then the user should set the multiplicity of the pseudospin equal to 2 (see MLTP keyword). The calculation of the parameters of the crystal field corresponding to the ground atomic multiplet for lanthanides should be requested by the CRYS keyword.

```
&SINGLE_ANISO
```

Argument(s) to a keyword are always supplied on the next line of the input file.

Optional general keywords to control the input

Keyword *Meaning*

TITLE

One line following this one is regarded as title.

TYPE

Specifies which magnetic properties must be computed. The program will read one of the following numbers (1-7):

- 1: – the g- and D-tensors (only)
- 2: – the powder magnetic susceptibility $\chi T(T)$, the magnetic susceptibility tensor $\chi_{\alpha\beta}T(T)$, magnetic susceptibility in the direction of the main magnetic axis: $\chi_X T(T)$, $\chi_Y T(T)$, $\chi_Z T(T)$
- 3: – the powder molar magnetization $M(H)$, the magnetization vectors $M_X(H), M_Y(H), M_Z(H)$ -for certain directions of the field H_i.
- 4: – 1 + 2
- 5: – 1 + 3
- 6: – 2 + 3
- 7: – 1 + 2 + 3. This is the default value.

MLTP

The number of molecular multiplets (i.e. groups of spin-orbital eigenstates) for which g, D and higher magnetic tensors will be calculated (default MLTP=1). The program reads two lines: the first is the number of multiplets (NMULT) and the second the array of NMULT numbers specifying the dimension of each multiplet. The default is to select one multiplet which has the dimension equal to the multiplicity of the ground term. In cases of strong spin-orbit coupling the usage of this keyword is mandatory. Example:

```
MLTP
4
4 4 2 2
```

SINGLE_ANISO will compute the g tensor for 4 groups of states: 2 groups having the effective spin $\tilde{\text{S}}$=|3/2>, and other 2 groups of states being Kramers doublets.

TINT

Specifies the temperature points for the evaluation of the magnetic susceptibility. The program will read four numbers: T_{min}, T_{max}, nT, and δT_0.

- T_{min} – the minimal temperature (Default 0.0K)
- T_{max} – the maximal temperature (Default 300.0K)
- nT – number of temperature points (Default 101)
- δT_0 – the minimal temperature above the 0.0K that helps to avoid infinite values during calculation (Default 0.001K).

Example:

```
TINT
0.0  330.0  331  0.0001
```

SINGLE_ANISO will compute temperature dependence of the magnetic susceptibility in 331 points evenly distributed in temperature interval: 0.0K – 330.0K. δT_0=0.0001K.

HINT Specifies the field points for the evaluation of the magnetization in a certain direction. The program will read four numbers: H_{min}, H_{max}, nH, and δH_0.

- H_{min} – the minimal field (Default 0.0T)
- H_{max} – the maximal filed (Default 10.0T)
- nH – number of field points (Default 101)
- δH_0 – the minimal field above the 0.0T that helps to avoid infinite values during calculation (Default 0.001T).

Example:

```
HINT
0.0  20.0  201  0.0001
```

SINGLE_ANISO will compute the molar magnetization in 201 points evenly distributed in field interval: 0.0T – 20.0T. δH_0=0.0001T.

TMAG Specifies the temperature at which the field-dependent magnetization is calculated. Default is 2.0 K Example:

```
TMAG
1.8
```

ENCU This flag is used to define the cut-off energy for the lowest states for which Zeeman interaction is taken into account exactly. The contribution to the magnetization coming from states that are higher in energy than E (see below) is done by second order perturbation theory. The program will read two integer numbers: NK and MG. Default values are: $NK = 100$,$MG = 100$.

$$E = NK \cdot k_{Boltz} \cdot TMAG + MG \cdot \mu_{Bohr} \cdot H_{max}$$

The field-dependent magnetization is calculated at the temperature value TMAG. Example:

```
ENCU
250  150
```

If $H_{max} = 10$T and TMAG=1.8K, then the cut-off energy is:

$$E = 100 \cdot 250 \cdot k_{Boltz} \cdot 1.8 + 150 \cdot \mu_{Bohr} \cdot 10 = 1013.06258(cm^{-1})$$

This means that the magnetization coming from all spin-orbit states with energy lower than $E = 1013.06258(cm^{-1})$ will be computed exactly.

MVEC

Defines the number of directions for which the magnetization vector will be computed. On the first line below the keyword, the number of directions should be mentioned (NDIR. Default 0). The program will read NDIR lines for spherical coordinates specifying the direction i of the magnetic field (θ_i and ϕ_i). These values should be in radians. Example:

```
MVEC
4
0.000000 0.000000
1.570796 0.000000
1.570796 1.570796
0.425741 0.418747
```

MAVE

Specifies the number of directions of the applied magnetic field for the computation of the powder molar magnetization. The program will read two numbers: N_θ and N_ϕ.

- N_θ – number of θ points in the interval $(0, \pi/2)$. (Default 12)
- N_ϕ – number of ϕ points in the interval $(0, 2\pi)$. (Default 24)

The number of directions over which the actual averaging will take place is roughly the product of N_θ and N_ϕ.

TEXP

This keyword allows computation of the magnetic susceptibility $\chi T(T)$ at experimental points. On the line below the keyword, the number of experimental points NT is defined, and on the next NT lines the program reads the experimental temperature (in K) and the experimental magnetic susceptibility (in $cm^3 K mol^{-1}$). TEXP and TINT keywords are mutually exclusive. The magnetic susceptibility routine will also print the standard deviation from the experiment.

HEXP

This keyword allows computation of the molar magnetization $M_{mol}(H)$ at experimental points. On the line below the keyword,the number of experimental points NH is defined, and on the next NH lines the program reads the experimental field intensity (Tesla) and the experimental magnetization (in μ_{Bohr}). HEXP and HINT are mutually exclusive. The magnetization routine will print the standard deviation from the experiment.

ZJPR

This keyword specifies the value (in cm^{-1}) of a phenomenological parameter of a mean molecular field acting on the spin of the complex (the average intermolecular exchange constant). It is used in the calculation of all magnetic properties (not for spin Hamiltonians) (Default is 0.0)

PRLV

This keyword controls the print level.

- 2 – normal. (Default)
- 3 or larger (debug)

POLY

The program SINGLE_ANISO will prepare an input file (binary) for the future POLY_ANISO program. The default is not to create it.

CRYS This keyword will enable computation of all 27 Crystal-Field parameters acting on the ground atomic multiplet of a lanthanide. On the next line the program will read the chemical symbol of the lanthanide. By default the program will not compute the parameters of the Crystal-Field.

QUAX This keyword controls the quantization axis for the computation of the Crystal-Field parameters acting on the ground atomic multiplet of a lanthanide. On the next line, the program will read one of the three values: 1, 2 or 3.

- 1 – quantization axis is the main magnetic axis Zm of the ground pseudospin multiplet, whose size is specified within the MLTP keyword. (Default)
- 2 – quantization axis is the main magnetic axis Zm of the entire atomic multiplet $|J, M_J>$.
- 3 – the direction of the quantization axis is given by the user: on the next line the program will read three real numbers: the projections (p_x, p_y, p_z) of the specified direction on the initial Cartesian axes. Note that $p_x^2 + p_y^2 + p_z^2 = 1$.

An input example

```
&SINGLE_ANISO
TITLe
magnetic properties for Co complex
TYPE
2
MVEC
3
0.000000 0.000000
1.570796 0.000000
1.570796 1.570796
MLTP
3
4 4 2
ZJPR
-0.2
ENCU
250 400
HINT
0.0  20.0  100  0.01
TINT
0.0  330.0  331  0.01
MAVE
16 18
```

8.46 SLAPAF

Provided with the first order derivative with respect to nuclear displacements the program is capable to optimize molecular structures with or without constraints for minima or transition states. This will be achieved with a quasi-Newton approach in combination with 2nd ranks updates of the approximate Hessian or with the use of a computed (analytic or numerical) Hessian. Note that *if* a computed Hessian is available on the *RUNFILE* then it will be used rather than the approximate Hessian generated by SLAPAF. **On completion of an optimization SLAPAF will automatically execute a single energy evaluation**.

8.46.1 Description

SLAPAF has three different ways in selecting the basis for the displacements during the optimization. The first format require user input (not recommended), whereas the two other options are totally black-boxed. The formats are:

1. the old format as in Molcas-3, which is user specified. The internal coordinates are here represented as linear combination of internal coordinates (such as bonds, angles, torsions, out of plane angles, Cartesian coordinates and separation of centers of mass) and the linear combinations are totally defined by user input. This format does also require the user to specify the Hessian (default a diagonal matrix). This option *allows* for frozen internal coordinates.

2. the second format is an automatic option which employs the Cartesian eigenvectors of the approximative Hessian (generated by the Hessian model functional [147]).

3. the third format (this is the recommend and default) is an automatic option which utilizes linear combinations of some curvilinear coordinates (stretches, bends, and torsions). This implementation has two variations. The first can be viewed as the conventional use of non-redundant internal coordinates [148, 149, 150]. The second variation is a force constant weighted (FCW) redundant space (the HWRS option) version of the former implementation [151].

All three formats of internal coordinates can be used in combinations with constraints on the molecular parameters or other type of constraints as for example energy differences.

The displacements are symmetry adapted and any rotation and translation if present is deleted. The relaxation is symmetry preserving.

8.46.2 Dependencies

SLAPAF depends on the results of ALASKA and also possibly on MCKINLEY and MCLR.

8.46.3 Files

Input files

Apart from the standard input file SLAPAF will use the following input files.

File	*Contents*

RUNFILE	File for communication of auxiliary information. If a computed Hessian is available on this file it will be used rather than the approximate Hessian generated by SLAPAF.
RUNFILE2	File for communication of auxiliary information of the ”ground state” in case of minimum energy cross point optimizations.
RUNOLD	File for communication of auxiliary information for reading an old Hessian matrix from a previous geometry optimization.

Output files

In addition to the standard output file SLAPAF will use the following output files.

File	*Contents*
RUNFILE	File for communication of auxiliary information.
RUNFILE2	File for communication of auxiliary information of the ”ground state” in case of minimum energy cross point optimizations.
MD_GEO	Molden input file for geometry optimization analysis.
MD_MEP	Molden input file for minimum energy path (MEP).
MD_SADDLE	Molden input file for energy path (MEP) of a Saddle TS optimization.
MD_IRC	Molden input file for intrinsic reaction coordinate analysis of a TS.
MD_FREQ	Molden input file for harmonic frequency analysis.
UNSYM	ASCII file where all essential information, like geometry, Hessian normal modes and dipole derivatives are stored.
STRUCTURE	Output file with a statistics of geometry optimization convergence.

8.46.4 Input

SLAPAF will as standard provided with an energy and a corresponding gradient update the geometry (optimize). Possible update methods include different quasi-Newton methods. The program will also provide for updates of the Hessian. The program has a number of different variable metric methods available for the Hessian update. This section describes the input to the SLAPAF program.

This section describes the input to the SLAPAF program in the Molcas program system. The input starts with the program name

```
&SLAPAF
```

There are no compulsory keywords

Optional convergence control keywords

Keyword	*Meaning*
ITERations	Maximum number of iterations which will be allowed in the relaxation procedure. Default is 500 iterations, however, if environment variable MOLCAS_MAXITER has been exported by the user this is the assumed default value.
THRShld	Enter two real numbers which specifies the convergence criterion with respect to the energy change and the norm of the gradient. The defaults are 0.0 and 3.0D-4 au for Gaussian convergence criteria (which normally do not consider the energy change), and 1.0D-6 and 3.0D-4 for Baker criteria (see the BAKER keyword).
BAKEr	Activate convergence criteria according to Baker [152]. Default is to use the convergence criteria as in the Gaussian program [153].
MAXStep	This keyword is followed by the value which defines the seed of largest change of the internal coordinates which will be accepted. A change which is larger is reduced to the max value. The value is dynamically modified each iterations. The default value is 0.3 au or rad.
CNWEight	Sets the maximum weight assigned to the fulfillment of the constraints, relative to the step taken in the complementary space for energy minimization. The step in the constraint space is truncated to be at most as large as the step in the minimization space, or half the maximum total step, whichever is larger, multiplied by this value. Default is 1.0.
TOLErance	Controls how strictly the constraints (if any) must be satisfied at convergence. The default value is very large, such that this criterion is always met, and only the gradient and maximum step (or energy difference) control convergence. If you set this keyword to some value, a constrained optimization will only converge if the maximum error in any constraint is lower than this number (in atomic units, and radians).

Optional coordinate selection keywords

Keyword	*Meaning*
CARTesian	Activate `SLAPAF` to use the eigenvectors of the approximative Hessian expressed in Cartesian as the definition of the internal coordinates. The default is to use the FCW non-redundant internal coordinates. The Hessian will be modeled by the Hessian Model Functional.
INTErnal	This marks the start of the definition of the internal coordinates. This section is always ended by the keyword End of Internal. For a complete description of this keyword see the section 8.46.4. This option will also use a diagonal matrix as default for the Hessian matrix. The default is to use the FCW non-redundant internal coordinates.

HWRS	Use the force constant weighted (FCW) redundant space version of the nonredundant internal coordinates. This is the default. The Hessian will be modeled by the Hessian Model Functional.
NOHWRs	Disable the use of the force constant weighted redundant space version of the nonredundant internal coordinates. The default is to use the HWRS option. The Hessian will be modeled by the Hessian Model Functional.
FUZZ	When automatically generating the primitive internal coordinates, the system may end up in disconnected fragments, in which case additional bonds are defined between the fragments. This keyword controls how many inter-fragment bonds are added. Bonds are generated between the closest atoms of two fragments, and all pairs of atoms in separate fragments at a distance up to the specified value longer. The value can be followed with the unit BOHR or ANGSTROM. The default is 0.5 a.u.

Optional Hessian update keywords

Keyword	*Meaning*
HUPDate	Method used for updating the Hessian matrix. It must be one of: • None — No update is applied. • BFGS — Activate update according to Broyden–Fletcher–Goldfarb–Shanno. This is the default. • MSP — Activate the Murtagh–Sargent–Powell update according to Bofill [154]. This update is preferred for the location of transition states. • EU — Activate the EU update according to Bofill [155]. This update can be used for the location of transition states. • TS-BFGS — Activate the TS-BFGS update according to Bofill [155]. This update can be used for the location of minima or transition states.
UORDer	Order the gradients and displacements vectors according to Schlegel prior to the update of the Hessian. Default is no reorder.
WINDow	Maximum number of previous iterations to include in the Hessian update. Default is 5.

Optional optimization procedure keywords

Keyword	*Meaning*
NOLIne	Disable line search. Default is to use line search for minima.
RATIonal	Activate geometry optimization using the restricted step Rational Functional optimization [156, 157], this is the default.

C1-DIIS Activate geometry optimization using the C1-GDIIS method [158, 159, 160]. The default is to use the Rational Functional approach.

C2-DIIS Activate geometry optimization using the C2-GDIIS method [92]. The default is to use the Rational Functional approach.

DXDX This option is associated to the use of the C1- and C2-GDIIS procedures. This option will activate the computation of the so-called error matrix elements as $e = \delta x^\dagger \delta x$, where δx is the displacement vector.

DXG This option is associated to the use of the C1- and C2-GDIIS procedures. This option will activate the computation of the so-called error matrix elements as $e = \delta x^\dagger g$, where δx is the displacement vector and g is the gradient vector.

GDX See above.

GG This option is associated to the use of the C1- and C2-GDIIS procedures. This option will activate the computation of the so-called error matrix elements as $e = g^\dagger g$, where g is the gradient vector. This is the default.

NEWTON Activate geometry optimization using the standard quasi-Newton approach. The default is to use the Rational Functional approach.

RS-P-RFO Activate RS-P-RFO [157] as default for TS-search. Default is RS-I-RFO.

TS Keyword for optimization of transition states. This flag will activate the use of the mode following rational functional approach [161]. The mode to follow can either be the one with the lowest eigenvalue (if positive it will be changed to a negative value) or by the eigenvector which index is specified by the MODE keyword (see below). The keyword will also activate the Murtagh-Sargent-Powell update of the Hessian and inactivate line search. This keyword will also enforce that the Hessian has the right index (i.e. one negative eigenvalue).

MODE Specification of the Hessian eigenvector index, this mode will be followed by the mode following RF method for optimization of transition states. The keyword card is followed by a single card specifying the eigenvector index.

FINDTS Enable a constrained optimization to release the constraints and locate a transition state if negative curvature is encountered and the gradient norm is below a specific threshold (see the GNRM option). Keyword TSCONSTRAINTS should be used in combination with FINDTS.

TSCONSTRAINTS Specify constraints that will be active during the initial stage of an optimization with FINDTS. When negative curvature and low gradient are encountered, these constraints will be released and other constraints will remain active. If this block is not given in the input, all constraints will be released. The syntax of this keyword is exactly like normal constraints, and it must be ended with End of TSConstraints (see section 8.46.4 below).

GNRM

Modify the gradient norm threshold associated with the FINDTS option. The actual threshold is specified on the subsequent line. The default value is 0.2.

MEP-search

Enable a minimum energy path (MEP) search. MEP is a valid synonym.

NMEP

Maximum number of points to find in a minimum energy path search or intrinsic reaction coordinate analysis. Synonym of NIRC.

MEPStep

The keyword is used to specify the step length done in the MEP search or IRC analysis. The step length can be followed with the unit BOHR or ANGSTROM. The default is 0.1 a.u. (in normalized mass-weighted coordinates). Synonym of IRCStep.

MEPType

Specifies what kind of constraint will be used for optimizing the points during the MEP search or IRC analysis. The possibilities are SPHERE, the default, which uses the Sphere constraint (each structure is at a given distance in coordinate space from the reference), or PLANE which uses the Transverse constraint (each structure is at a given distance from the hyperplane defined by the reference and the path direction). The reference structure changes at each step, according to the MEPAlgorithm keyword. Synonym of IRCType.

MEPAlgorithm

Selects the algorithm for a MEP search or IRC analysis. The possibilities are GS for the González-Schlegel algorithm, the default, or MB for the Müller-Brown algorithm. Synonym of IRCAlgorithm.

REFErence

The keyword is followed by a list of the symmetry unique coordinates (in au) of the origin of the hyper sphere. The default origin is the structure of the first iteration.

GRADient of reference

The keyword is followed by a list of the gradient vector components. This keyword is compulsory when using the Transverse kind of constraint. The optimization is performed in a space orthogonal to the given vector.

IRC

The keyword is used to perform an intrinsic reaction coordinate (IRC) analysis of a transition state structure. The analysis will follow the reaction path forward and backward until the energy increases. The keyword requires that the starting structure be that of a transition state and that the reaction vector be specified explicitly (check the keyword REACtion vector) or implicitly if it can be found on *`RUNOLD`*. Note that the user should not specify any explicit constraints!

NIRC

Maximum number of points to find in an intrinsic reaction coordinate analysis or minimum energy path search. Synonym of NMEP.

IRCStep — The keyword is used to specify the step length done in the IRC analysis or MEP search. The step length can be followed with the unit BOHR or ANGSTROM. The default is 0.1 a.u. (in normalized mass-weighted coordinates). Synonym of MEPStep.

IRCType — Specifies what kind of constraint will be used for optimizing the points during the IRC analysis or MEP search. The possibilities are SPHERE, the default, which uses the Sphere constraint (each structure is at a given distance in coordinate space from the reference), or PLANE which uses the Transverse constraint (each structure is at a given distance from the hyperplane defined by the reference and the path direction). The reference structure changes at each step, according to the IRCAlgorithm keyword. Synonym of MEPType

IRCAlgorithm — Selects the algorithm for a MEP search or IRC analysis. The possibilities are GS for the González-Schlegel algorithm, the default, or MB for the Müller-Brown algorithm. Synonym of MEPAlgorithm.

REACtion vector — The keyword is followed by the reaction vector specified as the Cartesian vector components on each of the symmetry unique atoms.

Optional force constant keywords

Keyword — *Meaning*

OLDForce — The Hessian matrix is read from the file ***RUNOLD***. This Hessian is either an analytic or approximative Hessian updated by Slapaf. Note that for this option to work properly the type of internal coordinates must be the same!

FCONstant — Input of Hessian in internal coordinates. There are two different syntaxes.

1. The keyword is followed by an entry with the number of elements which will be set (observe that the update will preserve that the elements H_{ij} and H_{ji} are equal). The next entries will contain the value and the indices of the elements to be replaced.
2. The keyword if followed by the label Square or Triangular. The subsequent line specifies the rank of the Hessian. This is then followed by entries specifying the Hessian in square or lower triangular order.

XFCOnstant — Input of an external Hessian matrix in cartesian coordinates. The syntax is the same as for the FCONSTANT keyword.

NUMErical — This invokes as calculation of the force constant matrix by a two-point finite difference formula. The resulting force constant matrix is used for an analysis of the harmonic frequencies. **Observe** that in case of the use of internal coordinates defined as Cartesian coordinates that these has to be linear combinations which are free from

	translational and rotational components for the harmonic frequency analysis to be valid. **Alternative:** see keyword RowH in the section about Internal coordinates.
CUBIc	This invokes a calculation of the 2nd and the 3rd order force constant matrix by finite difference formula.
DELTa	This keyword is followed by a real number which defines the step length used in the finite differentiation. Default: 1.0D-2.
PRFC	The eigenvalues and eigenvectors of the Hessian matrix are printed. The internal coordinates definitions are also printed.
RHIDden	Define the hidden atoms selection radius in order to improve a QM/MM Hessian. It can be followed by Angstrom.

Optional miscellaneous keywords

Keyword	*Meaning*
CTOF	Coordinates TO Follow defines an internal coordinate whose values will be printed in the output during the optimization. Both the original and the new values will be printed. The keyword must be followed by the definition on the primitive coordinate.
RTRN	Maximum number of atoms for which bond lengths, angles and dihedral angles are listed, and the radius defining the maximum length of a bond follows. The latter is used as a threshold when printing out angles and dihedral angles. The length can be followed by Bohr or Angstrom which indicates the unit in which the length was specified, the default is Bohr. The default values are 15 and 3.0 au.
THERmochemistry	Request frequencies to be computed followed by an user specified thermochemical analysis. The keyword must be followed by different entries containing the Rotational Symmetry Number, the Pressure (in atm), and one entry per Temperature (in K) for which the thermochemistry will be calculated. The section is ended by the keyword End of PT.
DISOtope	Calculates frequencies modified for double isotopic substitution.
TRACk	Tries to follow electronic states during an optimization, by computing state overlaps with RASSI at each step. Root numbers selected with RlxRoot in RASSCF or with the "EDiff" constraint are only fixed in the first iteration, then the best-matching states are chosen.
LASTenergy	Specifies the quantum chemical method requested for the Last_Energy module (e.g., SCF, CASSCF, CASPT2, etc.) The keyword must be followed by the name of the module. Moreover, the EMIL command COPY needs to be used in the global input to provide a file named LASTEN, containing the input for the specified module.

NOLAst energy Disables the call to the Last_Energy module when convergence is achieved.

Example: A complete set of input decks for a CASSCF geometry optimization. These are the input decks for the optimization of the enediyne molecule.

```
 &GATEWAY
Title= Enediyne
Coord= $MOLCAS/Coord/enediyne.xyz
Basis= ANO-L-VQZP
Group= x z
> DoWhile
 &SEWARD
 &SCF
ITERATIONS= 30; Occupied= 9 8 2 1; Thresholds= 1.0d-8 1.0d-3 1.5d-3 0.2d-3; IVO
 &RASSCF
Symmetry= 1; Spin= 1
NactEl= 12 0 0; Inactive= 7 7 0 0; Ras2= 3 3 3 3
Iterations= 50 50; CiRoot= 1 1; 1; Thrs= 1.0e-08 1.0e-05 1.0e-05
Lumorb
 &SLAPAF; Iterations= 20
> EndDo
```

Example: Thermochemistry for an asymmetric top (Rotational Symmetry Number = 1), at 1.0 atm and 273.15, 298.15, 398.15 and 498.15 K.

```
 &SLAPAF; THERmochemistry= 1; 1.0; 273.15; 298.15; 398.15; 498.15; End of PT
End of input
```

Definition of internal coordinates or constraints

The input section defining the internal coordinates always start with the keyword Internal coordinates and the definition of the constraints starts with the keyword Constraints. Note that the latter is an input section for the GATEWAY module.

The input is always sectioned into two parts where the first section defines a set of primitive internal coordinates and the second part defines the actual internal coordinates as any arbitrary linear combination of the primitive internal coordinates that was defined in the first section. In case of constraints the second part does also assign values to the constraints.

In the first section we will refer to the atoms by their atom label (SEWARD will make sure that there is no redundancy). In case of symmetry one will have to augment the atom label with a symmetry operation in parenthesis in order to specify a symmetry related center. Note that the user only have to specify distinct internal coordinates (ALASKA will make the symmetry adaptation).

In the specification below *rLabel* is a user defined label with no more than 8 (eight) characters. The specifications atom1, atom2, atom3, and atom4 are the unique atom labels as specified in the input to SEWARD.

The primitive internal coordinates are defined as

- ***rLabel* = Bond atom1 atom2** — a primitive internal coordinate *rLabel* is defined as the bond between center atom1 and atom2.

- ***rLabel* = Angle atom1 atom2 atom3** — a primitive internal coordinate *rLabel* is defined as the angle between the bonds formed from connecting atom1 to atom2 and connecting atom2 to atom3.
- ***rLabel* = LAngle(1) atom1 atom2 atom3** — a primitive internal coordinate *rLabel* is defined as the linear angle between the bonds formed from connecting atom1 to atom2 and connecting atom2 to atom3. To define the direction of the angle the following procedure is followed.

  1. – *the three centers are linear*,
     (a) form a reference axis, **R1**, connecting atom1 and atom3,
     (b) compute the number of zero elements, *nR*, in the reference vector,
         i. – *nR=0*, a first perpendicular direction to the reference axis is formed by

         $$\mathbf{R2} = (R1x, R1y, -R1z)$$

         followed by the projection

         $$\mathbf{R2} = \mathbf{R2} - \frac{\mathbf{R2}\cdot\mathbf{R1}}{\mathbf{R1}\cdot\mathbf{R1}}\mathbf{R1}.$$

         The second perpendicular direction completes the right-handed system.
         ii. – *nR=1*, a first perpendicular direction to the reference axis is defined by setting the element in **R2** corresponding to the zero entry in **R1** to unity. The second perpendicular direction completes the right-handed system.
         iii. – *nR=2*, a first perpendicular direction to the reference axis is defined by setting the element corresponding to the first zero entry in **R1** to unity. The second perpendicular direction completes the right-handed system.
  2. – *the three centers are nonlinear*, the first perpendicular direction is the one which is in the plane formed by atoms atom1, atom2, and atom3. The second perpendicular direction is taken as the direction perpendicular to the same plane.

  The direction of the bend for **LAngle(1)** is taken in the direction of the first perpendicular direction, etc.

- ***rLabel* = LAngle(2) atom1 atom2 atom3** — a primitive internal coordinate *rLabel* is defined as the linear angle between the bonds formed from connecting atom1 to atom2 and connecting atom2 to atom3. The definition of the perpendicular directions is as described above. The direction of the bend for **LAngle(2)** is taken in the direction of the second perpendicular direction.
- ***rLabel* = Dihedral atom1 atom2 atom3 atom4** — a primitive internal coordinate *rLabel* is defined as the angle between the planes formed of atom1, atom2 and atom3, and atom2, atom3 and atom4, respectively.
- ***rLabel* = OutOfP atom1 atom2 atom3 atom4** — a primitive internal coordinate *rLabel* is defined as the angle between the plane formed by atom2, atom3, and atom4 and the bond formed by connecting atom1 and atom4.
- ***rLabel* = Dissoc (n1+n2) atom1 atom2 atom3 ... atomN** — a primitive internal coordinate *rLabel* is defined as the distance between the center of masses of two sets of centers. The first center has n1 members and the second has n2. The input contains the labels of the atoms of the first group followed immediately by the labels of the second group. This option is not available for constraints.

- ***rLabel* = Cartesian i atom1** — a primitive internal coordinate *rLabel* is defined as the pure Cartesian displacement of the center labeled atom1. The label i is selected to x, y, or z to give the appropriate component.

- ***rLabel* = Ediff [i j]** — the energy difference between states i and j (if provided, the brackets indicate they are optional, do not include the brackets). If i and j are not provided, the difference is between the "current" state and the state provided on *RUNFILE2*. This is only used in constrained optimization in which crossings or conical intersections are located. If this constraint is used, the average energy of the two states will be optimized, subject to the constraint. If the value is 0.0 and the spin and spatial symmetry of both states is the same, a conical intersection will be searched. In this case, SLAPAF will request an analytical calculation of the nonadiabatic coupling vector, if available. If it is not available, or if *RUNFILE2* is being used (i and j not provided), the branching space update method of Maeda et al. will be used [162].

- ***rLabel* = Sphere** — the radius of the hypersphere defined by two different molecular structures (the origin is the first structure) in relative mass-weighted coordinates. This is only used in constrained optimization in which minimum reaction paths (MEP) or intrinsic reaction coordinate (IRC) paths are followed. The units of the radius is in mass-weighted coordinates divided with the square root of the total mass of the molecule.

- ***rLabel* = Transverse** — a level of "orthogonality". This is used to perform an optimization in a space orthogonal to a given vector. Recommended value 0.0. Requires usage of GRAD keyword.

- ***rLabel* = Fragment atom1 atom2 atom3 atom4** — a dummy internal coordinate *rLabel* is defined. This translates to that a set of internal coordinates are generated automatically according to a standard Z-matrix format to define all degrees of freedom of the fragment defined by the list of atoms on the same line. These internal coordinates will be automatically fixed in the geometry optimizations to the values of starting structure. Note, the values of these do not need to be explicitly defined and set in the VALUES section. Note, too, that the generation of the internal coordinates is done according to the order in which atom1, atom2, etc. are given; for some systems, especially with linear angles, it may be preferable to define the coordinates manually.

The second section starts with the label VARY or in the case of constraints with the label VALUES.

In case of a definition of **internal coordinates** in this section the user specifies all symmetric internal coordinates excluding translation and rotation using a list of expressions like

label = f1 *rLabel1* + f2 *rLabel2* + ...

which defines an internal coordinate *label* as the linear combination of the primitive internal coordinates *rLabel1*, *rLabel2*, ... with the coefficients f1, f2, ..., respectively. If the internal coordinate just corresponds to the primitive internal coordinate, the same label can be used

label

If some internal coordinates are chosen to be fixed they should be defined after the label FIX. The fixed internal coordinate are defined with expressions as in the section VARY. Observe:

using expression can introduce linear dependence and/or undefined nuclear coordinates, so use with care.
For the internal coordinates defined after VARY (and FIX, if present) a numerical estimation of rows and columns of the hessian matrix can be performed. The *label* of internal coordinates (max 10) must be specified after keyword ROWH. Keywords NUMERICAL and ROWH are mutually exclusive.

In case of a definition of **constraints** the sections contains either a direct reference to a *rLabel* as in

rLabel = *rValue* [Angstrom,Degrees] [Soft,Hard] [Phantom]

or one can also use expressions like

f1 *rLabel1* +/- f2 *rLabel2* +/- .... = *Value* [Angstrom,Degrees] [Soft,Hard] [Phantom]

where *rValue* is the desired value of the constraint in au or rad, or in angstrom or degrees if the corresponding keyword is added. The “Hard” and “Soft” keywords are only meaningful for numerical differentiation: the coordinates corresponding to soft constraints are differentiated, those of hard constraints are not [17]. By default almost all constraints are hard, only constraints of the type “Sphere”, “Transverse” and “Ediff” default to soft. The “Hard” and “Soft” keywords override the default. When using constraints in combination with the FINDTS keyword, one should use soft constraints, at least for the constraint most similar to the expected reaction vector. Constraints defined in TSCONSTRAINTS (recommended) are automatically considered soft.

The “Phantom” modifier can be used to ignore a constraint in the optimization. A phantom constraint will only be considered for numerical differentiation. Phantom constraints are useful in combination with the KEEPOLDGRADIENT keyword of ALASKA. Using NGEXCLUDE in GATEWAY is equivalent to phantom constraints, and it is the preferred way to set up composite gradients [17].

Alternatively, if the current value of an internal coordinate is to be used, i.e. no change is to be allowed (frozen), this is expressed as

rLabel = FIX [Soft,Hard] [Phantom]

Note that a coordinate of type “Fragment” does not need to appear in the VALUES section, but if it does it must be assigned the value “FIX”.

Example: A definition of user specified internal coordinates of benzene. The molecule is in D_{6h} and since MOLCAS only uses up to D_{2h} the FIX option is used to constrain the relaxation to the higher point group. **Observe** that this will only restrict the nuclear coordinates to D_{6h}. The electronic wavefunction, however, can have lower symmetry.

```
Internal coordinates
r1 = Bond C1 C2
r2 = Bond C1 H1
r3 = Bond C2 H2
r4 = Bond C2 C2(x)
f1 = Angle H1 C1 C2
f2 = Angle H2 C2 C1
Vary
a = 1.0 r1 +  1.0 r4
b = 1.0 r2 +  1.0 r3
c = 1.0 f1 +  1.0 f2
Fix
a = 1.0 r1 + -1.0 r4
```

```
b = 1.0 r2 + -1.0 r3
c = 1.0 f1 + -1.0 f2
End of Internal
```

Example: A input for the optimization of water constraining the structure to be linear at convergence.

```
 &GATEWAY
    Title= H2O geom optim, using the ANO-S basis set.
    Coord=$MOLCAS/Coord/Water.xyz
    Basis=ANO-S-VDZ
    Group= c1
    Constraints
        a1 = langle(1) H2 O1 H3
    Values
        a1 = 179.99 degrees
    End of Constraints
>>> DO WHILE <<<
 &SEWARD; &SCF
 &SLAPAF
>>> END DO <<<
```

Example: A complete set of input decks for a UHF transition structure geometry optimization of an identity hydrogen transfer reaction (HO + H_2O -> H_2O + OH).

```
 &GATEWAY
ZMAT
O.STO-3G....
H.STO-3G....
H1
Z2    1  1.0
O3    1  1.15    2   92.
O4    1  1.15    2   92.     3  180.
H5    3  0.98    4  105.4    2  120.
H6    4  0.98    3  105.4    2  120.
>>> DO WHILE <<<
 &SEWARD;
 &SCF; UHF
 &SLAPAF; TS; PRFC
    Internal
      bOO4  = Bond O3 O4
      bOH5  = Bond H5 O3
      bOH6  = Bond H6 O4
      bOH1  = Bond O3 H1
      aOOH5 = Angle O4 O3 H5
      aOOH6 = Angle O3 O4 H6
      aHOH1 = Angle H5 O3 H1
      dH6  = Dihedral H6 O4 O3 H5
      dH1  = Dihedral O4 H5 O3 H1
    Vary; bOO4; bOH5; bOH6; bOH1; aOOH5; aOOH6; aHOH1; dH6; dH1
    RowH; bOH1
    End of Internal
>>> ENDDO <<<
```

Example: Optimization of a minimum energy conical intersection point, using automatic calculation of analytical gradients and nonadiabatic coupling.

```
&GATEWAY
  Coord = acrolein.xyz
  Basis = cc-pVDZ
  Group = NoSymm
  Constraints
    a = Ediff 1 2
```

```
   Values
     a = 0.0
   End of constraints
 >>> DoWhile
   &SEWARD
   >>> If (iter = 1)
     &SCF
     &MBPT2
       PrPt
   >>> EndIf
   &RASSCF
     FileOrb = $Project.MP2Orb
     Charge = 0
     NActEl = 6 0 0
     RAS2 = 5
     CIRoot = 4 4 1
   &SLAPAF
 >>> EndDo
```

8.47 SURFACEHOP

This module deals with surface hop semiclassical molecular dynamics (SHMD) and has to be used together with module DYNAMIX. Its purpose is the calculation of the relax root for the next step of the SHMD. In this moment the implemented algorithm under this module is the Tully's fewest switches[163], along with the Hammes-Schiffer/Tully scheme[164] and the decoherence correction proposed by Granucci and Persico[165].

Output files

File	*Contents*
RUNFILE	Surface hop information such as Amatrix and CI coefficients for previous steps are stored in this file.

8.47.1 Input

```
&Gateway
coord=$Project.xyz
basis=6-31G*
group=nosym

>> FOREACH A in (1..400)

&Seward

&rasscf
 jobiph
 cirestart
 nactel = 6 0 0
 inactive = 23
 ras2 = 6
 ciroot = 2 2 1
 prwf = 0.0
 mdrlxroot = 2

&Surfacehop
 tully
 decoherence = 0.1
 psub

&alaska

&Dynamix
 velver
 dt = 41.3
 velo = 1
 thermo = 0
>>> End Do
```

General keywords

Keyword	*Meaning*
TULLY	This keyword enables the Tully–Hammes-Schiffer integration of the TDSE for the Tully Surface Hop Algorithm. If you use this keyword you should not use the HOP keyword in `DYNAMIX`.
DECOHERENCE	This keyword must be used after the TULLY keyword. It enables the decoherence correction in the population density matrix as reported by Persico–Granucci. The value is called decay factor and it is usually 0.1 hartree. It can be seen as how strongly this correction is applied. It is recommendable to leave it to 0.1, unless you really know what you're doing.
SUBSTEP	This keyword must be used after the TULLY keyword. This keyword specifies how many steps of integration we use to interpolate/extrapolate between two Newton's consecutive steps. The default is usually a good compromise between quickness and precision (200 substeps each femtoseconds of MD).
PSUB	This keyword must be used after the TULLY keyword. To print in MOLCAS output D matrix, A matrix, B matrix, probabilities, randoms, population and energies at each substep (quite verbose, but gives you a lot of useful information).

DMTX This keyword must be used after the TULLY keyword. With this keyword you can start your calculation with an initial A matrix (population density matrix). It is a complex matrix. In the first line after the keyword you must specify its dimension N. Then N lines (N values each line) with the real part of the matrix followed by N more lines with the imaginary part.

FRANDOM This keyword must be used after the TULLY keyword. It fixes the random number to one provided by the user, in case a deterministic trajectory is needed

ISEED This keyword must be used after the TULLY keyword. The initial seed number is read from the input file. Then, seed numbers are modified (in a deterministic way), saved in the ***RunFile*** and read in the next call to the module. This way, MD simulations are reproducible.

MAXHOP This keyword must be used after the TULLY keyword. It specifies how many non-adiabatic transitions are allowed between electronic states.

H5RESTART This keyword allows to restart a surface hopping trajectory calculation from an HDF5 file. The name of the restart file is given on the next line.

Input examples

This example shows an excited state CASSCF MD simulation of a methaniminium cation using the Tully Surface Hop algorithm. Within the SURFACEHOP module The keyword TULLY enables the TDSE integration. The options used in this case are: (SUBSTEP=200) to specify 200 substep of electronic integration between Newton's, (DECOHERENCE=1) to deal with the decoherence using a decay constant of 0.1 hartree and (PSUB) to print the substeps matrices verbosely into the Molcas log.

```
&GATEWAY
 COORD
 6
 Angstrom
 C  0.00031448  0.00000000  0.04334060
 N  0.00062994  0.00000000  1.32317716
 H  0.92882820  0.00000000 -0.49115611
 H -0.92846597  0.00000000 -0.49069213
 H -0.85725321  0.00000000  1.86103989
 H  0.85877656  0.00000000  1.86062860
 BASIS= 3-21G
 GROUP= nosym
>> FOREACH ITER in (1 .. 1000)
&SEWARD
>> IF ( $ITER = 1 )
&RASSCF
 LUMORB
 FileOrb= $Project.GssOrb
 Symmetry= 1
 Spin= 1
 nActEl= 2 0 0
```

```
  Inactive= 7
  RAS2= 2
  CIroot= 3 3 1
 >> COPY $Project.JobIph $Project.JobOld
 >> ENDIF
 &RASSCF
  JOBIPH; CIRESTART
  Symmetry= 1
  Spin= 1
  nActEl= 2 0 0
  Inactive= 7
  RAS2= 2
  CIroot= 3 3 1
  MDRLXR= 2
 >> COPY $Project.JobIph $Project.JobOld
 &surfacehop
  TULLY
  SUBSTEP = 200
  DECOHERENCE = 0.1
  PSUB
 &ALASKA
 &Dynamix
  VELVer
  DT= 10.0
  VELO= 3
  THER= 2
  TEMP=300
 >> END DO
```

vibrot.tex $this file belongs to the Molcas repository$

8.48 VIBROT

The program VIBROT is used to compute a vibration-rotation spectrum for a diatomic molecule, using as input a potential computed over a grid. The grid should be dense around equilibrium (recommended spacing 0.05 au) and should extend to large distance (say 50 au) if dissociation energies are computed.

The potential is fitted to an analytical form using cubic splines. The ro-vibrational Schrödinger equation is then solved numerically (using Numerov's method) for one vibrational state at a time and for a number of rotational quantum numbers as specified by input. The corresponding wave functions are stored on file *VIBWVS* for later use. The ro-vibrational energies are analyzed in terms of spectroscopic constants. Weakly bound potentials can be scaled for better numerical precision.

The program can also be fed with property functions, such as a dipole moment curve. Matrix elements over the ro-vib wave functions for the property in question are then computed. These results can be used to compute IR intensities and vibrational averages of different properties.

VIBROT can also be used to compute transition properties between different electronic states. The program is then run twice to produce two files of wave functions. These files are used as input in a third run, which will then compute transition matrices for input properties. The main use is to compute transition moments, oscillator strengths, and lifetimes for ro-vib levels of electronically excited states. The asymptotic energy difference between the two electronic states must be provided using the ASYMPTOTIC keyword.

8.48.1 Dependencies

The VIBROT is free-standing and does not depend on any other program.

8.48.2 Files

Input files

The calculation of vibrational wave functions and spectroscopic constants uses no input files (except for the standard input). The calculation of transition properties uses *VIBWVS* files from two preceding VIBROT runs, redefined as *VIBWVS1* and *VIBWVS2*.

Output files

VIBROT generates the file *VIBWVS* with vibrational wave functions for each v and J quantum number, when run in the wave function mode. If requested VIBROT can also produce files *VIBPLT* with the fitted potential and property functions for later plotting.

8.48.3 Input

This section describes the input to the VIBROT program in the Molcas program system. The program name is

```
&VIBROT
```

Keywords

The first keyword to VIBROT is an indicator for the type of calculation that is to be performed. Two possibilities exist:

Keyword	*Meaning*
ROVIbrational spectrum	VIBROT will perform a vib-rot analysis and compute spectroscopic constants.
TRANsition moments	VIBROT will compute transition moment integrals using results from two previous calculations of the vib-rot wave functions. In this case the keyword Observable should be included, and it will be interpreted as the transition dipole moment.

Note that only one of the above keywords can be used in a single calculation. If none is given the program will only process the input section.

After this first keyword follows a set of keywords, which are used to specify the run. Most of them are optional.

The compulsory keywords are:

Keyword	*Meaning*
ATOMs	Gives the mass of the two atoms. Write mass number (an integer) and the chemical symbol Xx, in this order, for each of the two atoms in free format. If the mass numbers is zero for any atom, the mass of the most abundant isotope will be used. All isotope masses are stored in the program. You may introduce your own masses by giving a negative integer value to the mass number (one of them or both). The masses (in unified atomic mass units, or Da) are then read on the next (or next two) entry(ies). The isotopes of hydrogen can be given as H, D, or T.
POTEntial	Gives the potential as an arbitrary number of lines. Each line contains a bond distance (in au) and an energy value (in au). A plot file of the potential is generated if the keyword Plot is added after the last energy input. One more entry should then follow with three numbers specifying the start and end value for the internuclear distance and the distance between adjacent plot points. This input must only be given together with the keyword RoVibrational spectrum.

In addition you may want to specify some of the following optional input:

Keyword	*Meaning*
TITLe	One single title line
GRID	The next entries give the number of grid points used in the numerical solution of the radial Schrödinger equation. The default value is 199. The maximum value that can be used is 4999.
RANGe	The next entry contains two distances Rmin and Rmax (in au) specifying the range in which the vibrational wave functions will be computed. The default values are 1.0 and 5.0 au. Note that these values most often have to be given as input since they vary considerably from one case to another. If the range specified is too small, the program will give a message informing the user that the vibrational wave function is large outside the integration range.
VIBRational	The next entry specifies the number of vibrational quanta for which the wave functions and energies are computed. Default value is 3.
ROTAtional	The next entry specifies the range of rotational quantum numbers. Default values are 0 to 5. If the orbital angular momentum quantum number (m_ℓ) is non zero, the lower value will be adjusted to m_ℓ if the start value given in input is smaller than m_ℓ.
ORBItal	The next entry specifies the value of the orbital angular momentum (0,1,2, etc). Default value is zero.
SCALe	This keyword is used to scale the potential, such that the binding energy is 0.1 au. This leads to better precision in the numerical procedure and is strongly advised for weakly bound potentials.

NOSPectroscopic Only the wave function analysis will be carried out but not the calculation of spectroscopic constants.

OBSErvable This keyword indicates the start of input for radial functions of observables other than the energy, for example the dipole moment function. The next line gives a title for this observable. An arbitrary number of input lines follows. Each line contains a distance and the corresponding value for the observable. As for the potential, this input can also end with the keyword Plot, to indicate that a file of the function for later plotting is to be constructed. The next line then contains the minimum and maximum R-values and the distance between adjacent points. When this input is given with the top keyword RoVibrational spectrum the program will compute matrix elements for vibrational wave functions of the current electronic state. Transition moment integrals are instead obtained when the top keyword is Transition moments. In the latter case the calculation becomes rather meaningless if this input is not provided. The program will then only compute the overlap integrals between the vibrational wave functions of the two states. The keyword Observable can be repeated up to ten times in a single run. All observables should be given in atomic units.

TEMPerature The next entry gives the temperature (in K) at which the vibrational averaging of observables will be computed. The default is 300 K.

STEP The next entry gives the starting value for the energy step used in the bracketing of the eigenvalues. The default value is 0.004 au (88 cm^{-1}). This value must be smaller than the zero-point vibrational energy of the molecule.

ASYMptotic The next entries specifies the asymptotic energy difference between two potential curves in a calculation of transition matrix elements. The default value is zero atomic units.

ALLRotational By default, when the Transition moments keyword is given, only the transitions between the lowest rotational level in each vibrational state are computed. The keyword AllRotational specifies that the transitions between all the rotational levels are to be included. Note that this may result in a very large output file.

PRWF Requests the vibrational wave functions to be printed in the output file.

Input example

```
  &VIBROT
RoVibrational spectrum
Title = Vib-Rot spectrum for FeNi
Atoms = 0 Fe 0 Ni
Potential
 1.0 -0.516768
```

```
  1.1 -0.554562
  [...]
 Plot  = 1.0 10.0 0.1
 Grid  = 150
 Range = 1.0 10.0
 Vibrations = 10
 Rotations  = 2 10
 Orbital    = 2
 Observable
  Dipole Moment
  1.0 0.102354
  1.1 0.112898
  [...]
 Plot  = 1.0 10.0 0.1
```

Comments: The vibrational-rotation spectrum for FeNi will be computed using the potential curve given in input. The 10 lowest vibrational levels will be obtained and for each level the rotational states in the range J=2 to 10. The vib-rot matrix elements of the dipole function will also be computed. A plot file of the potential and the dipole function will be generated. The masses for the most abundant isotopes of Fe and Ni will be selected.

8.49 The Basis Set Libraries

The basis sets library contains both all-electron and effective core potentials. They will be briefly described below and we refer to the publications for more details. The user can also add new basis sets to the basis directory and the structure of the file will therefore be described below.

Dummy atoms

Note that to use dummy atoms the user should employ the basis set label ”X....”. This will signify centers associated with no charge and no basis functions.

The All Electron Basis Set Library

The basis set library of MOLCAS contains an extensive set of basis sets both segmented and generally contracted. The files in the basis directory are named in upper case after the basis type label (see below). Three sets of generally contracted basis sets have been especially designed for MOLCAS. They are based on the Atomic Natural Orbital (ANO) concept and are labeled ANO-X (X=S, L, or RCC). They have been designed to give a balanced description of the atoms in ground, excited, and ionized states. A more detailed description of these basis sets is given below. A fourth basis set, which is especially designed for the calculation of electric properties of molecules (POL) will also be described.

In addition to this, an subset of segmented standard basis sets are included, for example, STO-3G, 3-21G 4-31G, 6-31G, 6-31G*, 6-31G**, cc-pVXZ (X=D,T,Q), and aug-cc-pVXZ (X=D,T). In addition, the library also contains different variants of the Turbomole RI basis sets. For additional all electron basis set we recommend a visit to the EMSL Gaussian Basis Set Order Form (*http://www.emsl.pnl.gov/forms/basisform.html*). All basis sets are stored in the directory *basis_library*. The different types of available basis sets can be found in the file *basistype.tbl* in this directory. Aliases for the names are listed in the file *basis.tbl*. However, the best way to find out which basis sets are available is to issue the command molcas help basis X where X is the atom. Note that a short hand notation can be used for most basis sets: for example ANO-L-VTZP will give a basis set of valence triple zeta accuracy with polarization functions.

Small ANO basis sets — ANO-S The smallest of the Atomic Natural Orbital (ANO) basis sets are available for the atoms H–Kr. They have been constructed as eigenfunctions of a density matrix averaged over several electronic configurations. The ground state of the atom was included for all atoms, and dependent on the particular atom one or more of the following states were included: valence excited states, ground state for the anion and ground state for the cation. The density matrices were obtained by the SCF, SDCI or MCPF methods for 1 electron, 2 electron and many electron cases respectively. The emphasis have been on obtaining good structural properties such as bond-lengths and -strengths with as small contracted sets as possible. The quality for electric properties such as polarizabilities have been sacrificed for the benefit of the properties mentioned above. See [54] for further discussions. These basis sets are recommended for large molecules where the more extended ANO-L basis sets require to much computational times. One should, however, remember that for a given contraction it is only the time needed to generate the integrals (or Cholesky vectors) that is affected and it is usually preferred to use the more accurate ANO-L (or ANO-RCC) basis sets.

For information about the primitive basis set we refer to the library. The maximum number of ANO's given in the library is:

- 4s3p for H–He.
- 6s4p3d for Li–Be.
- 7s6p3d for B–Ne.
- 7s5p3d for Na–Mg.
- 7s7p4d for Al–Ar.
- 7s7p4d for K–Ca.
- 8s7p7d4f for Sc–Zn.
- 9s9p5d for Ga–Kr.

However, such contractions are unnecessarily large. Almost converged results (compared to the primitive sets) are obtained with the basis sets:

- 3s2p for H–He.
- 4s3p2d for Li–Ne.
- 5s4p3d for Na–Ar.
- 6s5p4d for K–Ca.
- 7s5p4d3f for Sc–Zn.

- 6s5p4d for Ga–Kr.

The results become more approximate below the DZP size:

- 2s1p for H–He.
- 3s2p1d for Li–Ne.
- 4s3p2d for Na–Ar.
- 5s4p3d for K–Ca.
- 6s4p3d2f for Sc–Zn.
- 5s4p3d for Ga–Kr.

Large ANO basis sets — ANO-L The large ANO basis sets for atoms H–Zn, excluding K and Ca, have been constructed by averaging the corresponding density matrix over several atomic states, positive and negative ions and the atom in an external electric field [51, 52, 53]. The different density matrices have been obtained from correlated atomic wave functions. Usually the SDCI method has been used. The exponents of the primitive basis have in some cases been optimized. The contracted basis sets give virtually identical results as the corresponding uncontracted basis sets for the atomic properties, which they have been optimized to reproduce. The design objective has been to describe the ionization potential, the electron affinity, and the polarizability as accurately as possible. The result is a well balanced basis set for molecular calculations.

For information about the primitive basis set we refer to the library. The maximum number of ANO's given in the library is:

- 6s4p3d for Hydrogen.
- 7s4p3d for Helium.
- 7s6p4d3f for Li–Be.
- 7s7p4d3f for B–Ne.
- 7s7p5d4f for Na–Ar.
- 8s7p6d5f4g for Sc–Zn

However, such contractions are unnecessarily large. Almost converged results (compared to the primitive sets) are obtained with the VQZP basis sets:

- 3s2p1d for H–He.
- 5s4d3d2f for Li–Ne.
- 6s5p4d3f for Na–Ar.
- 7s6p5d4f3g for Sc–Zn

The results become more approximate below the size:

- 3s2p for H–He.
- 4s3p2d for Li–Ne
- 5s4p2d for Na–Ar.
- 6s5p4d3f for Sc–Zn

It is recommended to use at least two polarization (3d/4f) functions, since one of them is used for polarization and the second for correlation. If only one 3d/4f-type function is used one has to decide for which purpose and adjust the exponents and the contraction correspondingly. Here both effects are described jointly by the two first 3d/4f-type ANO's (The same is true for the hydrogen 2p-type ANO's). For further discussions regarding the use of these basis sets we refer to the literature [51, 52, 53].

Relativistic ANO basis sets — ANO-RCC Extended relativistic ANO-type basis sets are available for the atoms H–Cm. These basis sets have been generated using the same principles as described above for the ANO-L basis sets with the difference that the density matrices have been computed using the CASSCF/CASPT2 method. The basis have been contracted using the Douglas-Kroll Hamiltonian and should therefore only be used in calculations where scalar relativistic effects are included. Seward will automatically recognize this and turn on the DK option when these basis sets are used [1, 2, 3, 4]. The basis sets contain functions for correlation of the semi-core electrons. The new basis sets are called ANO-RCC. More details about the construction and performance is given in the header for each basis set in the ANO-RCC library. Basis sets are available for all atoms up to Cm.

Scalar relativistic effect become important already in the second row of the periodic systems. It is therefore recommended to use these basis sets instead of ANO-L in all calculations.

For information about the primitive basis set we refer to the library. The maximum number of ANOs given in the library is:

- 6s4p3d1f for Hydrogen.
- 7s4p3d2f for Helium.
- 8s7p4d2f1g for Li–Be.
- 8s7p4d3f2g for Be–Ne.
- 17s12p5d4f for Na.
- 9s8p5d4f for Mg–Al.
- 8s7p5d4f2g for Si–Ar
- 10s9p5d3f for K
- 10s9p6d2f for Ca
- 10s10p8d6f4g2h for Sc–Zn

- 9s8p6d4f2g for Ga–Kr
- 10s10p5d4f for Rb–Sr
- 10s9p8d5f3g for In–Xe
- 12s10p8d4f for Cs–Ba
- 11s10p8d5f3g for La
- 12s11p8d7f4g2h for Ce–Lu
- 11s10p9d8f4g2h for Hf–Hg
- 11s10p9d6f4g for Tl–Rn
- 12s11p8d5f for Fr-Ra
- 13s11p10d8f6g3h for Ac-Pa
- 12s10p9d7f5g3h for U-Cm

However, such contractions are unnecessarily large. Almost converged results (compared to the primitive sets) are usually obtained with basis sets of QZP quality. You can get a feeling for the convergence from the test results presented in the header of each basis set in the library. One should also remember that larger basis sets are needed for the correlation of semi-core electrons.

Below is a list of the core electrons correlated for each atom.

Li–B:	1s
C–Ne:	No core correlation
Na:	2s,2p
Mg–Al:	2p
Si–Ar:	No core correlation
K:	3s,3p
Ca–Zn:	3p
Ga–Ge:	3d
As–Kr:	No core correlation
Rb–Sr:	4p
In–Xe:	4d
Cs–Ba:	5p
La–Lu:	5s,5p
Hf–Re:	4f,5s,5p
Os–Hg:	5s,5p
Tl–Rn:	5d
Fr–Ra:	6p
Ac-Cm:	6s,6p

Basis set label in input:
The general label is given as for the other ANO basis sets:
Atom.ano-rcc...contracted set. (Note the last dot!). A short hand notation is also possible: *Atom.ANO-RCC-label*, where *label* is one of MB,VDZ,VDZP,VTZP, or VQZP. A translation between the two possibilities can be found in file: *`$MOLCAS/basis_library/basis.tbl`*

Polarized basis sets The so-called polarized basis sets are purpose oriented, relatively small GTO/CGTO sets devised for the purpose of accurate calculations of dipole electric properties of polyatomic molecules [166, 167, 168, 169, 170]. For each row of the periodic table the performance of the basis sets has been carefully examined in calculations of dipole moments and dipole polarizabilities of simple hydrides at both the SCF and correlated levels of approximation [166, 167, 168, 169, 170]. The corresponding results match within a few percent the best available experimental data. Also the calculated molecular quadrupole moments turn out to be fairly close to those computed with much larger basis sets. According to the present documentation the polarized basis GTO/CGTO sets can be used for safe accurate predictions of molecular dipole moments, dipole polarizabilities, and also molecular quadrupole moments by using high-level correlated computational methods. The use of the polarized basis sets has also been investigated in calculations of weak intermolecular interactions. The interaction energies, corrected for the basis set superposition effect (BSSE), which is rather large for these basis sets, turn out to be close to the best available data. In calculations for molecules involving the 4th row atoms, the property data need to be corrected for the relativistic contribution. The corresponding finite perturbation facility is available [171, 172].

It is recommended to use these basis sets with the contraction given in the library. It is of course possible to truncate them further, for example by deleting some polarization functions, but this will lead to a deterioration of the computed properties.

Structure of the all electron basis set library

The start of a given basis set in the library is given by the line

```
/label
```

where “label” is the basis set label, as defined below in the input description to SEWARD. Then follows two lines with the appropriate literature reference for that basis set. These cards are read by SEWARD and must thus be included in the library, and may not be blank. Next is a set of comment lines, which begin with an asterisk in column 1, giving some details of the basis sets. A number of lines follow, which specifies the basis set:

1. Charge of the atom and the highest angular momentum. For each angular momentum (l) then follows.
2. Number of primitives and contracted functions for angular momentum l (must be identical to those given in the basis set label) .
3. Exponents of the primitive functions .
4. The contraction matrix (with one CGTO per column). Note that all basis sets are given in the generally contracted format, even if they happen to be segmented. Note that the number of CGTOs must correspond to the data given in the label .

The following is an example of an entry in a basis set library.

```
* This is the Huzinaga 5s,2p set contracted to 3s,2p        -- Comment
* according to the Dunning paper.                           -- Comment
/H.TZ2P.Dunning.5s2p.3s2p.                                  -- Label
Exponents  : S. Huzinaga, J. Chem. Phys., 42, 1293(1965).   -- First ref line
```

```
Coefficients: T. H. Dunning, J. Chem. Phys., 55, 716(1971). -- Second ref line
 1.0 1                                                      -- Charge, sp
 5 3                                                        -- 5s->3s
 52.56 7.903 1.792 0.502 0.158                              -- s-exponents
 0.025374  0.0  0.0       -- contr. matrix
 0.189684  0.0  0.0       -- contr. matrix
 0.852933  0.0  0.0       -- contr. matrix
 0.0    1.0  0.0       -- contr. matrix
 0.0    0.0  1.0       -- contr. matrix
 2  2                                                       -- 2p->2p
 1.5  0.5                                                   -- p-exponents
 1.0 0.0             -- contr. matrix
 0.0 1.0             -- contr. matrix
```

The ECP Library

Molcas is able to perform *effective core potential* (ECP) calculations and *embedded cluster* calculations. In ECP calculations, only the *valence* electrons of a molecule are explicitly handled in a quantum mechanical calculation, at a time that the *core* electrons are kept frozen and are represented by ECP's. (An example of this is a calculation on HAt in which only the 5d, 6s and 6p electrons of Astatine and the one of Hydrogen are explicitly considered.) Similarly, in *embedded cluster* calculations, only the electrons assigned to a piece of the whole system (the *cluster*) are explicitly handled in the quantum mechanical calculation, under the assumption that they are the only ones relevant for some local properties under study; the rest of the whole system (the *environment*) is kept frozen and represented by embedding potentials which act onto the *cluster*. (As an example, calculations on a TlF_{12}^{11-} cluster embedded in a frozen lattice of $KMgF_3$ can be sufficient to calculate spectroscopical properties of Tl^+-doped $KMgF_3$ which are due to the Tl^+ impurity.)

In order to be able to perform ECP calculations in molecules, as well as *embedded cluster* calculations in ionic solids, with the Ab Initio Model Potential method (AIMP) [173, 174, 175, 176, 177, 178] Molcas is provided with the library *ECP* which includes nonrelativistic and relativistic *core* ab initio model potentials and *embedding* ab initio model potentials representing both complete-cations and complete-anions in ionic lattices [174, 179].

Before we continue we should comment a little bit on the terminology used here. Strictly speaking, ECP methods are all that use the frozen-core approximation. Among them, we can distinguish two families: the 'pseudopotential' methods and the 'model potential' methods. The pseudopotential methods are ultimately based on the Phillips-Kleinman equation [180] and handle valence nodeless pseudo orbitals. The model potential methods are based on the Huzinaga equation [181, 182] and handle node-showing valence orbitals; the AIMP method belongs to this family. Here, when we use the general term ECP we will be referring to the more particular of AIMP. According to its characteristics, the AIMP method can be also applied to represent frozen-ions in ionic lattices in embedded cluster calculations; in this case, we will not be very strict in the nomenclature and we will also call ECP's to the frozen-ion (embedding) *ab initio* model potentials.

The effective potentials in the libraries include the effects of the atomic core wave functions (embedding ion wave functions) through the following operators:

- a local representation of the core (ion) Coulomb operator,
- a non-local spectral representation of the core (ion) exchange operator,

- a core (ion) projection operator,
- a spectral representation of the relativistic mass-velocity and Darwin operators corresponding to the valence orbitals, if the Cowan-Griffin-based scalar relativistic CG-AIMP method [175] is used.
- a spectral representation of the relativistic no-pair Douglas-Kroll operators, if the scalar relativistic no-pair Douglas-Kroll NP-AIMP method [176, 177, 178] is used.

Given the quality and non-parametric nature of the operators listed above, the flexibility of the basis sets to be used with the AIMP's is crucial, as in any *ab initio* method.

The valence basis sets included in the libraries have been obtained by energy minimization in atomic valence-electron calculations, following standard optimization procedures. All the experience gathered in the design of molecular basis sets starting from all-electron atomic basis sets, and in particular from segmented minimal ones, is directly applicable to the AIMP valence basis sets included in the libraries. They are, for non-relativistic and relativistic Cowan-Griffin AIMPs, minimal basis sets with added functions, such as polarization and diffuse functions; in consequence, the minimal sets should be split in molecular calculations in order to get reasonable sets (a splitting pattern is recommended in the library for every set); the splitting can be done by means of 'the basis set label'. For the relativistic no-pair Douglas-Kroll AIMPs contracted valence basis sets are given directly in a form which is recommended in molecular calculations, i.e. they are of triple zeta quality in the outer shells and contain polarization functions. In both cases these *valence* basis sets contain very *inner* primitive GTF's: They are necessary since, typical to a model potential method, the valence orbitals will show correct nodal structure. Finally, it must be noted that the core AIMP's can be safely mixed together with all-electron basis sets.

In AIMP *embedded cluster calculations*, the cluster basis set, which must be decided upon by the user, should be designed following high quality standard procedures. Very rigid cluster basis sets should not be used. In particular, the presence of the necessary embedding projection operators, which prevent the cluster densities from collapsing onto the crystal lattice, demands flexible cluster bases, including, eventually, components outside the cluster volume.[183] The use of flexible cluster basis sets is then a necessary requirement to avoid artificial frontier effects, not ascribable to the AIMP embedding potentials. This requirement is unavoidable, anyway, if good correlated wave functions are to be calculated for the cluster. Finally, one must remember that the AIMP method does exclude any correlation between the cluster electronic group and the embedding crystal components; in other words, only intra-cluster correlation effects can be accounted for in AIMP embedded cluster calculations. Therefore the cluster-environment partition and the choice of the cluster wave function must be done accordingly. In particular, the use of one-atom clusters is not recommended.

Core- and embedding- AIMP's can be combined in a natural way in valence-electron, embedded cluster calculations. They can be used with any of the different types of wave functions that can be calculated with MOLCAS.

Core AIMP's The list of core potentials and valence basis sets available in the `ECP` library follows. Although AIMP's exist in the literature for different core sizes, this library includes only those recommended by the authors after numerical experimentation. Relativistic CG-AIMP's and NP-AIMP's, respectively, and nonrelativistic NR-AIMP's are included. Each entry of the CG-AIMP's and the NR-AIMP's in the list is accompanied with a recommended contraction pattern (to be used in the fifth field). The NP-AIMP basis sets are given explicitly in the recommended contraction pattern. For the third-row transition metals two NP-AIMP basis sets are provided which differ in the number of primitive and contracted f GTFs. For further details, please refer to the literature.[178] For more information about a particular entry consult the *ECP* library.

The ECP libraries have also been extended to include the so-called nodeless ECPs or pseudo potentials based on the Phillips-Kleinman equation [180]. These are included both as explicit and implicit operators. Following the work by M. Pelissier and co-workers [184] the operators of nodeless ECPs can implicitly be fully expressed via spectral representation of operators. The explicit libraries are the ECP.STOLL and ECP.HAY-WADT files, all other files are for the implicitly expressed operator. In the list of nodeless ECPs the Hay and Wadt's family of ECPs (LANL2DZ ECPs) [185, 186, 187] has been included in addition to the popular set of the so-called Stoll and Dolg ECPs [188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212]. Both of them in either the explicit form labeled as HAY-WADT and STOLL, or in the implicit form labeled as HW and DOLG. The latter include the recently developed ANO-basis sets for actinides [212].

Structure of the ECP libraries

The start of a given basis set and AIMP is identified by the line

```
/label
```

where "label" is defined below, in the input description to SEWARD. Then, comment lines, effective charge, and basis set follow, with the same structure that the all-electron Basis Set Library (see items 1. to 4. in Sec. 8.49.) Next, the AIMP/ECP/PP is specified as follows:

5. The pseudo potential approach [213, 214, 215], see eqs. (3) and (4) in Ref. [216], with the following lines:

   (a) The keyword PP On the same line follows the atomic symbol of the element, the number of core electrons (N_c) and L, where $L-1$ is the largest angular momentum orbital belonging to the core. This line is followed by $L+1$ identical sections. The first of these sections is the so-called L potential and the subsequent sections corresponds to the S-L, P-L, D-L, etc. potentials. Each sections start with a line specifying the number of Gaussian terms in the potential. This line is then followed by a single line for each Gaussian specifying the powers (n_{kl}), the Gaussian exponent (ζ_{kl}), and the associated coefficient (d_{kl}).
   Note that the pseudo potential input is mutually exclusive to the M1, M2, COREREP, and PROJOP keywords!

6. The Coulomb local model potential, eq.(6) in Ref. [173] with the following lines:

   (a) The keyword M1, which identifies the terms with $n_k = 0$.

   (b) The number of terms. If greater than 0, lines 6c and 6d are read.

   (c) The exponents α_k.

   (d) The coefficients A_k (divided by the negative of the effective charge).

   (e) The keyword M2, which identifies the terms with $n_k = 1$.

   (f) The number of terms. If greater than 0, lines 6g and 6h are read.

   (g) The exponents α_k.

   (h) The coefficients A_k (divided by the negative of the effective charge).

7. A line with the keyword COREREP followed by another one with a real constant. This is not used now but it is reserved for future use.

8. The projection operator, eq.(3) in Ref. [173] with the following lines:

   (a) The keyword PROJOP.

   (b) The maximum angular momentum (l) of the frozen core (embedding) orbitals. Lines 8c to 8f are repeated for each angular momentum l.

   (c) The number of primitives and the number of orbitals (more properly, degenerate sets of orbitals or l-shells) for angular momentum l. As an option, these two integers can be followed by the occupation numbers of the l-shells; default values are 2 for l=0, 6 for l=1, etc.

   (d) The projection constants, $-2\varepsilon_c$.

   (e) The exponents of the primitive functions.

   (f) The coefficients of the orbitals, one per column, using general contraction format.

9. The spectral representation operator, eq.(7) in Ref. [173] for NR-AIMP, eq.(3) in Ref. [175] for relativistic CG-AIMP, and eqs.(1) and (7) in Ref. [178] for relativistic NP-AIMP, with the following lines:

   (a) The keyword SPECTRAL REPRESENTATION OPERATOR.

   (b) One of the keywords VALENCE, CORE, or EXTERNAL. VALENCE indicates that the set of primitive functions specified in the basis set data will be used for the spectral representation operator; this is the standard for ab initio *core* model potentials. CORE means that the set of primitives specified in the PROJOP section will be used instead; this is the standard for complete-ion ab initio *embedding* model potentials. EXTERNAL means that a set of primitives specific for the spectral representation operator will be provided in the next lines. In this case the format is one line in which an integer number specifies the highest angular momentum of the external basis sets; then, for each angular momentum the input is formated as for lines 2, 3, and 4 in Sec. 8.49.

   (c) The keyword EXCHANGE.

   (d) For relativistic AIMPs one of the keywords NOPAIR or 1STORDER RELATIVISTIC CORRECTION. NOPAIR indicates that scalar relativistic no-pair Douglas-Kroll AIMP integrals are to be calculated. 1STORDER RELATIVISTIC CORRECTION means that Cowan-Griffin-based scalar relativistic AIMP, CG-AIMP's, are used. In the latter case, in the next line a *keyword* follows which, in the library ***QRPLIB***, identifies the starting of the numerical mass-velocity plus Darwin potentials (eq.(2) in Ref. [175]). (In ***QRPLIB*** a line with "*keyword* MV&DW POTENTIALS START" must exist, followed by the number of points in the radial logarithmic grid, the values of the radial coordinate r, and, for each valence orbital, its label (2S, 4P, etc), and the values of the mass-velocity plus Darwin potentials at the corresponding values of r; these data must end up with a line "*keyword* MV&DW POTENTIALS END".)

(e) The keyword End of Spectral Representation Operator.

Below is an example of an entry in the *ECP* library for an AIMP.

```
/S.ECP.Barandiaran.7s6p1d.1s1p1d.6e-CG-AIMP.        -- label (note that type is ECP)
Z.Barandiaran and L.Seijo, Can.J.Chem. 70(1992)409. -- 1st ref. line
core[Ne] val[3s,3p]  (61/411/1*)=2s3p1d recommended -- 2nd ref. line
*SQR-SP(7/6/1)                 (61/411/1)           -- comment line
  6.000000         2                                -- eff. charge & highest ang.mom.
                                                    -- blank line
    7    1                                          -- 7s -> 1s
   1421.989530                                      -- s-exponent
   211.0266560                                      -- s-exponent
   46.72165060                                      -- s-exponent
   4.310564040                                      -- s-exponent
   1.966475840                                      -- s-exponent
   .4015383790                                      -- s-exponent
   .1453058790                                      -- s-exponent
   .004499703540                                    -- contr. coeff.
   .030157124800                                    -- contr. coeff.
   .089332590700                                    -- contr. coeff.
  -.288438151000                                    -- contr. coeff.
  -.279252515000                                    -- contr. coeff.
   .700286615000                                    -- contr. coeff.
   .482409523000                                    -- contr. coeff.
    6    1                                          -- 6p -> 1p
   78.08932440                                      -- p-exponent
   17.68304310                                      -- p-exponent
   4.966340810                                      -- p-exponent
   .5611646780                                      -- p-exponent
   .2130782690                                      -- p-exponent
   .8172415400E-01                                  -- p-exponent
  -.015853278200                                    -- contr. coeff.
  -.084808963800                                    -- contr. coeff.
  -.172934245000                                    -- contr. coeff.
   .420961662000                                    -- contr. coeff.
   .506647309000                                    -- contr. coeff.
   .200082121000                                    -- contr. coeff.
    1    1                                          -- 1d -> 1d
   .4210000000                                      -- d-exponent
  1.000000000000                                    -- contr. coeff.
*                                                   -- comment line
* Core AIMP: SQR-2P                                 -- comment line
*                                                   -- comment line
* Local Potential Parameters : (ECP convention)    -- comment line
*                            A(AIMP)=-Zeff*A(ECP)   -- comment line
M1                                                  -- M1 operator
    9                                               -- number of M1 terms
   237485.0100                                      -- M1 exponent
   24909.63500                                      -- M1 exponent
   4519.833100                                      -- M1 exponent
   1082.854700                                      -- M1 exponent
   310.5610000                                      -- M1 exponent
   96.91851000                                      -- M1 exponent
   26.63059000                                      -- M1 exponent
   9.762505000                                      -- M1 exponent
   4.014487500                                      -- M1 exponent
                                                    -- blank line
   .019335998333                                    -- M1 coeff.
   .031229360000                                    -- M1 coeff.
   .061638463333                                    -- M1 coeff.
   .114969451667                                    -- M1 coeff.
   .190198283333                                    -- M1 coeff.
   .211928633333                                    -- M1 coeff.
   .336340950000                                    -- M1 coeff.
   .538432350000                                    -- M1 coeff.
   .162593178333                                    -- M1 coeff.
M2                                                  -- M2 operator
    0                                               -- number of M2 terms
COREREP                                             -- CoreRep operator
   1.0                                              -- CoreRep constant
PROJOP                                              -- Projection operator
    1                                               -- highest ang. mom.
    8    2                                          -- 8s -> 2s
  184.666320      18.1126960                        -- 1s,2s proj. op. constants
   3459.000000                                      -- s-exponent
   620.3000000                                      -- s-exponent
   171.4000000                                      -- s-exponent
   58.53000000                                      -- s-exponent
   22.44000000                                      -- s-exponent
   6.553000000                                      -- s-exponent
   2.777000000                                      -- s-exponent
   1.155000000                                      -- s-exponent
   .018538249000   .005054826900                    -- contr. coeffs.
   .094569248000   .028197248000                    -- contr. coeffs.
   .283859290000   .088959130000                    -- contr. coeffs.
   .454711270000   .199724180000                    -- contr. coeffs.
   .279041370000   .158375340000                    -- contr. coeffs.
   .025985763000  -.381198090000                    -- contr. coeffs.
  -.005481472900  -.621887210000                    -- contr. coeffs.
   .001288714400  -.151789890000                    -- contr. coeffs.
    7    1                                          -- 7p -> 1p
  13.3703160                                        -- 2p proj. op. constant
   274.0000000                                      -- p-exponent
   70.57000000                                      -- p-exponent
   24.74000000                                      -- p-exponent
   9.995000000                                      -- p-exponent
   4.330000000                                      -- p-exponent
   1.946000000                                      -- p-exponent
   .8179000000                                      -- p-exponent
   .008300916100                                    -- cont. coeff.
   .048924254000                                    -- cont. coeff.
   .162411660000                                    -- cont. coeff.
   .327163550000                                    -- cont. coeff.
   .398615170000                                    -- cont. coeff.
   .232548200000                                    -- cont. coeff.
   .034091088000                                    -- cont. coeff.
*                                                   -- comment line
Spectral Representation Operator                    -- SR operator
```

```
Valence primitive basis                             -- SR basis specification
Exchange                                            -- Exchange operator
1stOrder Relativistic Correction                    -- mass-vel + Darwin oper.
SQR-2P                                              -- label in QRPLIB
End of Spectral Representation Operator             -- end of SR operator
```

Below is an example of an entry in the *ECP* library for a pseudo potential.

```
/Hg.ECP.Dolg.4s4p2d.2s2p1d.2e-MWB                   -- label (note the type ECP)
W. Kuechle, M. Dolg, H. Stoll, H. Preuss, Mol. Phys.-- ref. line 1
74, 1245 (1991); J. Chem. Phys. 94, 3011 (1991).    -- ref. line 2
    2.00000    2                                    -- eff. charge & highest ang.mom.
*s functions                                        -- comment line
  4  2                                              -- 4s -> 2s
  0.13548420E+01                                    -- s-exponent
  0.82889200E+00                                    -- s-exponent
  0.13393200E+00                                    -- s-exponent
  0.51017000E-01                                    -- s-exponent
  0.23649400E+00  0.00000000E+00                    -- contr. coeff.
 -0.59962800E+00  0.00000000E+00                    -- contr. coeff.
  0.84630500E+00  0.00000000E+00                    -- contr. coeff.
  0.00000000E+00  0.10000000E+01                    -- contr. coeff.
*p functions                                        -- comment line
  4  2                                              -- 4p -> 2p
  0.10001460E+01                                    -- p-exponent
  0.86645300E+00                                    -- p-exponent
  0.11820600E+00                                    -- p-exponent
  0.35155000E-01                                    -- p-exponent
  0.14495400E+00  0.00000000E+00                    -- contr. coeff.
 -0.20497100E+00  0.00000000E+00                    -- contr. coeff.
  0.49030100E+00  0.00000000E+00                    -- contr. coeff.
  0.00000000E+00  0.10000000E+01                    -- contr. coeff.
*d functions                                        -- comment line
  1  1                                              -- 1d -> 1d
  0.19000000E+00                                    -- d-exponent
  0.10000000E+01                                    -- contr. coeff.
*                                                   -- comment line
PP,Hg,78,5;                                         -- PP operator, label, # of core elec., L
1; ! H POTENTIAL                                    -- # number of exponents in the H potential
2, 1.00000000,.000000000;                           -- power, exponent and coeff.
3; ! S-H POTENTIAL                                  -- # number of exponents in the S-H potential
2,0.227210000,-.69617800;                           -- power, exponent and coeff.
2, 1.65753000,27.7581050;                           -- power, exponent and coeff.
2, 10.0002480,48.7804750;                           -- power, exponent and coeff.
2; ! P-H POTENTIAL                                  -- # number of exponents in the P-H potential
2,0.398377000,-2.7358110;                           -- power, exponent and coeff.
2,0.647307000,8.57563700;                           -- power, exponent and coeff.
2; ! D-H POTENTIAL                                  -- # number of exponents in the D-H potential
2,0.217999000,-.01311800;                           -- power, exponent and coeff.
2,0.386058000,2.79286200;                           -- power, exponent and coeff.
1; ! F-H POTENTIAL                                  -- # number of exponents in the F-H potential
2,0.500000000,-2.6351640;                           -- power, exponent and coeff.
1; ! G-H POTENTIAL                                  -- # number of exponents in the G-H potential
2,0.800756000,-13.393716;                           -- power, exponent and coeff.
```

```
*                                                    -- comment line
Spectral Representation Operator                     -- SR operator
End of Spectral Representation Operator              -- end of SR operator
```

Section 9

GUI

The manual for graphical viewer GV, and LUSCUS can be found online: http://www.molcas.org/GV/, https://sourceforge.net/projects/luscus/.

9.1 Writing LUSCUS/MOLDEN input

By default the GUESSORB, SCF, MBPT2, RASSCF, SLAPAF, LOCALISATION, and MCLR modules generate input in Molden format. The SCF, MBPT2, RASSCF, and LOCALISATION modules generate input for molecular orbital analysis, SLAPAF for geometry optimization analysis, minimum energy paths, Saddle optimization paths and IRC TS analysis, and the MCLR module generates input for analysis of harmonic frequencies.

The generic name of the input file and the actual name are different for the nodes as a reflection on the data generated by each module. Hence, the actual names (generic name) for the Molden files in each module are

- GUESSORB module: $Project.guessorb.molden (MD_GSS)
- SCF module: $Project.scf.molden (MD_SCF)
- MBPT2 module: $Project.mp2.molden (MD_MP2)
- RASSCF module: $Project.rasscf.molden (MD_CAS) for the state-averaged natural orbitals, and $Project.rasscf.x.molden (MD_CAS.x) for the state-specific natural spin orbitals, where x is the index of a CI root.
- SLAPAF module: $Project.geo.molden (MD_GEO) for geometry optimizations, $Project.mep.molden (MD_MEP) for minimum energy paths, $Project.irc.molden (MD_IRC) for IRC analysis of a TS, and $Project.saddle.molden (MD_SADDLE) for Saddle method TS optimizations.
- LOCALISATION module: $Project.local.molden (MD_LOC)
- MCLR module: $Project.freq.molden (MD_FREQ)

Files in MOLDEN format can be visualized by Molden program, or by MOLCAS GUI program `LUSCUS`.

Note, the MOLDEN code cannot handle cases in which some shells, with the same angular momentum, are Cartesian and some are real spherical harmonics. All shells for a give angular momentum have to be either Cartesian or real spherical harmonics, otherwise no MOLDEN file for orbitals will be generated.

For further details with respect to Molden consult http://www.caos.kun.nl/~shaft/molden/molden.html.

For further details with respect to LUSCUS consult https://sourceforge.net/projects/luscus/.

Part V

Advanced Examples and Annexes

Section 10

Examples

10.1 Computing high symmetry molecules.

MOLCAS makes intensive use of the symmetry properties of the molecular systems in all parts of the calculation. The highest symmetry point group available, however, is the D_{2h} point group, which makes things somewhat more complicated when the molecule has higher symmetry. One of such cases is the calculation of linear molecules. In this section we describe calculations on different electronic states of three diatomic molecules: NiH, a heteronuclear molecule which belongs to the $C_{\infty v}$ symmetry group and C_2 and Ni_2, two homonuclear molecules which belong to the $D_{\infty h}$ symmetry group. They must be computed in MOLCAS using the lower order symmetry groups C_{2v} and D_{2h}, respectively, and therefore some codes such RASSCF use specific tools to constrain the resulting wave functions to have the higher symmetry of the actual point group. It must be pointed out clearly that linear symmetry cannot always be fully obtained in MOLCAS because the tools to average over degenerate representations are not totally implemented presently in the RASSCF program. This is the case, for instance, for the δ orbitals in a C_{2v}-$C_{\infty v}$ situation, as will be shown below. (For problems related to accurate calculations of diatomic molecules and symmetry see Ref. [217] and [218], respectively.). In a final section we will briefly comment the situation of high symmetry systems other than linear.

10.1.1 A diatomic heteronuclear molecule: NiH

Chemical bonds involving transition-metal atoms are often complex in nature due to the common presence of several unpaired electrons resulting in many close-lying spectroscopic states and a number of different factors such spin-orbit coupling or the importance of relativistic effects. NiH was the first system containing a transition-metal atom to be studied with the CASSCF method [219]. The large dynamic correlation effects inherent in a $3d$ semi-occupied shell with many electrons is a most severe problem, which few methods have been able to compute. The calculated dipole moment of the system has become one measurement of the quality of many *ab initio* methods [79]. We are not going to analyze the effects in detail. Let us only say that an accurate treatment of the correlation effects requires high quality methods such as MRCI, ACPF or CASPT2, large basis sets, and an appropriate treatment of relativistic effects, basis set superposition errors, and core-valence correlation. A detailed CASPT2 calculation of the ground state of NiH can be found elsewhere [220].

The ^3F ($3d^84s^2$) and ^3D ($3d^94s^1$) states of the nickel atom are almost degenerate with a splitting of only 0.03 eV [221] and are characterized by quite different chemical behavior. In systems such as the $^2\Delta$ ground state of NiH molecule, where both states take part in the bonding, an accurate description of the low-lying Ni atomic states is required. The selection of the active space for NiH is not trivial. The smallest set of active orbitals for the $^2\Delta$ ground state which allows a proper dissociation and also takes into account the important $3d\sigma$ correlation comprises the singly occupied $3d_{xy}$ orbital and three σ orbitals ($3d_{z^2}, \sigma$, and σ^*). One cannot however expect to obtain accurate enough molecular properties just by including non-dynamical correlation effects. MRCI+Q calculations with the most important CASSCF configurations in the reference space proved that at least one additional $3d\delta$ ($3d_{x^2-y^2}$) and its correlating orbital were necessary to obtain spectroscopic constants in close agreement with the experimental values. It is, however, a larger active space comprising all the eleven valence electrons distributed in twelve active orbitals (σ, σ^*, d, d') that is the most consistent choice of active orbitals as evidenced in the calculation of other metal hydrides such as CuH [220] and in the electronic spectrum of the Ni atom [221]. This is the active space we are going to use in the following example. We will use the ANO-type basis set contracted to Ni [$5s4p3d1f$] / H [$3s2p$] for simplicity. In actual calculations g functions on the transition metal and d functions on the hydrogen atom are required to obtain accurate results.

First we need to know the behavior of each one of the basis functions within each one of the symmetries. Considering the molecule placed in the z axis the classification of the spherical harmonics into the $C_{\infty v}$ point group is:

Table 10.1: Classification of the spherical harmonics in the $C_{\infty v}$ group.

Symmetry	Spherical harmonics					
σ	s	p_z	d_{z^2}	f_{z^3}		
π	p_x	p_y	d_{xz}	d_{yz}	$f_{x(z^2-y^2)}$	$f_{y(z^2-x^2)}$
δ	$d_{x^2-y^2}$	d_{xy}	f_{xyz}	$f_{z(x^2-y^2)}$		
ϕ	f_{x^3}	f_{y^3}				

In C_{2v}, however, the functions are distributed into the four representations of the group and therefore different symmetry representations can be mixed. The next table lists the distribution of the functions in C_{2v} and the symmetry of the corresponding orbitals in $C_{\infty v}$.

Table 10.2: Classification of the spherical harmonics and $C_{\infty v}$ orbitals in the C_{2v} group.

Symm.[a]	Spherical harmonics (orbitals in $C_{\infty v}$)					
a$_1$ (1)	s (σ)	p_z (σ)	d_{z^2} (σ)	$d_{x^2-y^2}$ (δ)	f_{z^3} (σ)	$f_{z(x^2-y^2)}$ (δ)
b$_1$ (2)	p_x (π)	d_{xz} (π)	$f_{x(z^2-y^2)}$ (π)	f_{x^3} (ϕ)		
b$_2$ (3)	p_y (π)	d_{yz} (π)	$f_{y(z^2-x^2)}$ (π)	f_{y^3} (ϕ)		
a$_2$ (4)	d_{xy} (δ)	f_{xyz} (δ)				

[a] In parenthesis the number of the symmetry in MOLCAS. It depends on the generators used in `SEWARD`.

In symmetry a$_1$ we find both σ and δ orbitals. When the calculation is performed in C_{2v} symmetry all the orbitals of a$_1$ symmetry can mix because they belong to the same representation, but this is not correct for $C_{\infty v}$. The total symmetry must be kept $C_{\infty v}$ and therefore the δ orbitals should not be allowed to rotate and mix with the σ orbitals. The same is true in the b_1 and b_2 symmetries with the π and ϕ orbitals, while in a_2 symmetry this problem does not exist because it has only δ orbitals (with a basis set up to f functions).

The tool to restrict possible orbital rotations is the option SUPSYM in the RASSCF program. It is important to start with clean orbitals belonging to the actual symmetry, that is, without unwanted mixing.

But the problems with the symmetry are not solved with the SUPSYM option only. Orbitals belonging to different components of a degenerate representation should also be equivalent. For example: the π orbitals in b_1 and b_2 symmetries should have the same shape, and the same is true for the δ orbitals in a_1 and a_2 symmetries. This can only be partly achieved in the RASSCF code. The input option AVERAGE will average the density matrices for representations b_1 and b_2 (π and ϕ orbitals), thus producing equivalent orbitals. The present version does not, however, average the δ orbital densities in representations a_1 and a_2 (note that this problem does not occur for electronic states with an equal occupation of the two components of a degenerate set, for example Σ states). A safe way to obtain totally symmetric orbitals is to reduce the symmetry to C_1 (or C_s in the homonuclear case) and perform a state-average calculation for the degenerate components.

We need an equivalence table to know the correspondence of the symbols for the functions in MOLCAS to the spherical harmonics (SH):

Table 10.3: MOLCAS labeling of the spherical harmonics.

MOLCAS	SH	MOLCAS	SH	MOLCAS	SH
1s	s	3d2+	$d_{x^2-y^2}$	4f3+	f_{x^3}
2px	p_x	3d1+	d_{xz}	4f2+	$f_{z(x^2-y^2)}$
2pz	p_z	3d0	d_{z^2}	4f1+	$f_{x(z^2-y^2)}$
2py	p_y	3d1-	d_{yz}	4f0	f_{z^3}
		3d2-	d_{xy}	4f1-	$f_{y(z^2-x^2)}$
				4f2-	f_{xyz}
				4f3-	f_{y^3}

We begin by performing a SCF calculation and analyzing the resulting orbitals. The employed bond distance is close to the experimental equilibrium bond length for the ground state [220]. Observe in the following SEWARD input that the symmetry generators, planes yz and xz, lead to a C_{2v} representation. In the SCF input we have used the option OCCNUMBERS which allows specification of occupation numbers other than 0 or 2. It is still the closed shell SCF energy functional which is optimized, so the obtained SCF energy has no physical meaning. However, the computed orbitals are somewhat better for open shell cases as NiH. The energy of the virtual orbitals is set to zero due to the use of the IVO option. The order of the orbitals may change in different computers and versions of the code.

```
 &SEWARD
Title
 NiH G.S
Symmetry
X Y
Basis set
Ni.ANO-L...5s4p3d1f.
```

```
Ni    0.00000   0.00000   0.000000   Bohr
End of basis
Basis set
H.ANO-L...3s2p.
H     0.000000  0.000000  2.747000   Bohr
End of basis
End of Input
 &SCF
TITLE
 NiH G.S.
OCCUPIED
 8 3 3 1
OCCNumber
2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0
2.0 2.0 2.0
2.0 2.0 2.0
1.0
```

```
   SCF orbitals + arbitrary occupations

   Molecular orbitals for symmetry species 1

   ORBITAL        4         5         6         7         8         9        10
   ENERGY     -4.7208   -3.1159    -.5513    -.4963    -.3305     .0000     .0000
   OCC. NO.    2.0000    2.0000    2.0000    2.0000    2.0000     .0000     .0000

 1 NI  1s0      .0000     .0001     .0000    -.0009     .0019     .0112     .0000
 2 NI  1s0      .0002     .0006     .0000    -.0062     .0142     .0787     .0000
 3 NI  1s0     1.0005    -.0062     .0000    -.0326     .0758     .3565     .0000
 4 NI  1s0      .0053     .0098     .0000     .0531    -.4826     .7796     .0000
 5 NI  1s0     -.0043    -.0032     .0000     .0063    -.0102    -.0774     .0000
 6 NI  2pz      .0001     .0003     .0000    -.0015     .0029     .0113     .0000
 7 NI  2pz     -.0091    -.9974     .0000    -.0304     .0622     .1772     .0000
 8 NI  2pz      .0006     .0013     .0000     .0658    -.1219     .6544     .0000
 9 NI  2pz      .0016     .0060     .0000     .0077    -.0127    -.0646     .0000
10 NI  3d0     -.0034     .0089     .0000     .8730     .4270     .0838     .0000
11 NI  3d0      .0020     .0015     .0000     .0068     .0029     .8763     .0000
12 NI  3d0      .0002     .0003     .0000    -.0118    -.0029    -.7112     .0000
13 NI  3d2+     .0000     .0000    -.9986     .0000     .0000     .0000     .0175
14 NI  3d2+     .0000     .0000     .0482     .0000     .0000     .0000     .6872
15 NI  3d2+     .0000     .0000     .0215     .0000     .0000     .0000    -.7262
16 NI  4f0      .0002     .0050     .0000    -.0009    -.0061     .0988     .0000
17 NI  4f2+     .0000     .0000     .0047     .0000     .0000     .0000    -.0033
18 H   1s0     -.0012    -.0166     .0000     .3084    -.5437    -.9659     .0000
19 H   1s0     -.0008    -.0010     .0000    -.0284    -.0452    -.4191     .0000
20 H   1s0      .0014     .0007     .0000     .0057     .0208     .1416     .0000
21 H   2pz      .0001     .0050     .0000    -.0140     .0007     .5432     .0000
22 H   2pz      .0008    -.0006     .0000     .0060    -.0093     .2232     .0000

   ORBITAL       11        12        13        14        15        16        18
   ENERGY       .0000     .0000     .0000     .0000     .0000     .0000     .0000
   OCC. NO.     .0000     .0000     .0000     .0000     .0000     .0000     .0000

 1 NI  1s0     -.0117    -.0118     .0000     .0025     .0218    -.0294     .0000
 2 NI  1s0     -.0826    -.0839     .0000     .0178     .1557    -.2087     .0000
 3 NI  1s0     -.3696    -.3949     .0000     .0852     .7386    -.9544     .0000
 4 NI  1s0    -1.3543   -1.1537     .0000     .3672    2.3913   -2.8883     .0000
 5 NI  1s0     -.3125     .0849     .0000   -1.0844     .3670    -.0378     .0000
 6 NI  2pz     -.0097    -.0149     .0000     .0064     .0261    -.0296     .0000
 7 NI  2pz     -.1561    -.2525     .0000     .1176     .4515    -.4807     .0000
 8 NI  2pz     -.3655   -1.0681     .0000     .0096    1.7262   -2.9773     .0000
 9 NI  2pz    -1.1434    -.0140     .0000    -.1206     .2437    -.9573     .0000
10 NI  3d0     -.1209    -.2591     .0000     .2015     .5359    -.4113     .0000
11 NI  3d0     -.3992    -.3952     .0000     .1001     .3984    -.9939     .0000
12 NI  3d0     -.1546    -.1587     .0000    -.1676    -.2422    -.4852     .0000
```

```
 13 NI  3d2+     .0000     .0000    -.0048     .0000     .0000     .0000    -.0498
 14 NI  3d2+     .0000     .0000    -.0017     .0000     .0000     .0000    -.7248
 15 NI  3d2+     .0000     .0000     .0028     .0000     .0000     .0000    -.6871
 16 NI  4f0     -.1778   -1.0717     .0000    -.0233     .0928    -.0488     .0000
 17 NI  4f2+     .0000     .0000   -1.0000     .0000     .0000     .0000    -.0005
 18 H   1s0     1.2967    1.5873     .0000    -.3780   -2.7359    3.8753     .0000
 19 H   1s0     1.0032     .4861     .0000     .3969    -.9097    1.8227     .0000
 20 H   1s0     -.2224    -.2621     .0000     .1872     .0884    -.7173     .0000
 21 H   2pz     -.1164    -.4850     .0000     .3388    1.1689    -.4519     .0000
 22 H   2pz     -.1668    -.0359     .0000     .0047     .0925    -.3628     .0000

    Molecular orbitals for symmetry species 2

    ORBITAL          2         3         4         5         6         7
    ENERGY     -3.1244    -.5032     .0000     .0000     .0000     .0000
    OCC. NO.    2.0000    2.0000     .0000     .0000     .0000     .0000

  1 NI  2px     -.0001     .0001     .0015     .0018     .0012    -.0004
  2 NI  2px     -.9999     .0056     .0213     .0349     .0235    -.0054
  3 NI  2px     -.0062    -.0140     .1244    -.3887     .2021    -.0182
  4 NI  2px      .0042     .0037     .0893     .8855    -.0520     .0356
  5 NI  3d1+     .0053     .9993     .0268     .0329     .0586     .0005
  6 NI  3d1+    -.0002    -.0211    -.5975     .1616     .1313     .0044
  7 NI  3d1+    -.0012    -.0159     .7930     .0733     .0616     .0023
  8 NI  4f1+     .0013    -.0049     .0117     .1257    1.0211    -.0085
  9 NI  4f3+    -.0064     .0000    -.0003    -.0394     .0132     .9991
 10 H   2px     -.0008     .0024    -.0974    -.1614    -.2576    -.0029
 11 H   2px      .0003    -.0057    -.2060    -.2268    -.0768    -.0079

    Molecular orbitals for symmetry species 3

    ORBITAL          2         3         4         5         6         7
    ENERGY     -3.1244    -.5032     .0000     .0000     .0000     .0000
    OCC. NO.    2.0000    2.0000     .0000     .0000     .0000     .0000

  1 NI  2py     -.0001     .0001    -.0015     .0018     .0012     .0004
  2 NI  2py     -.9999     .0056    -.0213     .0349     .0235     .0054
  3 NI  2py     -.0062    -.0140    -.1244    -.3887     .2021     .0182
  4 NI  2py      .0042     .0037    -.0893     .8855    -.0520    -.0356
  5 NI  3d1-     .0053     .9993    -.0268     .0329     .0586    -.0005
  6 NI  3d1-    -.0002    -.0211     .5975     .1616     .1313    -.0044
  7 NI  3d1-    -.0012    -.0159    -.7930     .0733     .0616    -.0023
  8 NI  4f3-     .0064     .0000    -.0003     .0394    -.0132     .9991
  9 NI  4f1-     .0013    -.0049    -.0117     .1257    1.0211     .0085
 10 H   2py     -.0008     .0024     .0974    -.1614    -.2576     .0029
 11 H   2py      .0003    -.0057     .2060    -.2268    -.0768     .0079

    Molecular orbitals for symmetry species 4

    ORBITAL          1         2         3         4
    ENERGY      -.0799     .0000     .0000     .0000
    OCC. NO.    1.0000     .0000     .0000     .0000

  1 NI  3d2-    -.9877    -.0969     .0050    -.1226
  2 NI  3d2-    -.1527     .7651     .0019     .6255
  3 NI  3d2-    -.0332    -.6365    -.0043     .7705
  4 NI  4f2-     .0051    -.0037    1.0000     .0028
```

In difficult situations it can be useful to employ the AUFBAU option of the SCF program. Including this option, the subsequent classification of the orbitals in the different symmetry representations can be avoided. The program will look for the lowest-energy solution and will provide with a final occupation. This option must be used with caution. It is only expected to work in clear closed-shell situations.

We have only printed the orbitals most relevant to the following discussion. Starting with symmetry 1 (a_1) we observe that the orbitals are not mixed at all. Using a basis set contracted to Ni $5s4p3d1f$ / H $3s2p$ in symmetry a_1 we obtain 18 σ molecular orbitals (combinations from eight atomic s functions, six p_z functions, three d_{z^2} functions, and one f_{z^3} function) and four δ orbitals (from three $d_{x^2-y^2}$ functions and one $f_{z(x^2-y^2)}$ function). Orbitals 6, 10, 13, and 18 are formed by contributions from the three $d_{x^2-y^2}$ and one $f_{z(x^2-y^2)}$ δ functions, while the contributions of the remaining harmonics are zero. These orbitals are δ orbitals and should not mix with the remaining a_1 orbitals. The same situation occurs in symmetries b_1 and b_2 (2 and 3) but in this case we observe an important mixing among the orbitals. Orbitals $7b_1$ and $7b_2$ have main contributions from the harmonics 4f3+ (f_{x^3}) and 4f3- (f_{y^3}), respectively. They should be pure ϕ orbitals and not mix at all with the remaining π orbitals.

The first step is to evaluate the importance of the mixings for future calculations. Strictly, any kind of mixing should be avoided. If g functions are used, for instance, new contaminations show up. But, undoubtedly, not all mixings are going to be equally important. If the rotations occur among occupied or active orbitals the influence on the results is going to be larger than if they are high secondary orbitals. NiH is one of these cases. The ground state of the molecule is $^2\Delta$. It has two components and we can therefore compute it by placing the single electron in the d_{xy} orbital (leading to a state of a_2 symmetry in C_{2v}) or in the $d_{x^2-y^2}$ orbital of the a_1 symmetry. Both are δ orbitals and the resulting states will have the same energy provided that no mixing happens. In the a_2 symmetry no mixing is possible because it is only composed of δ orbitals but in a_1 symmetry the σ and δ orbitals can rotate. It is clear that this type of mixing will be more important for the calculation than the mixing of π and ϕ orbitals. However it might be necessary to prevent it. Because in the SCF calculation no high symmetry restriction was imposed on the orbitals, orbitals 2 and 4 of the b_1 and b_2 symmetries have erroneous contributions of the 4f3+ and 4f3- harmonics, and they are occupied or active orbitals in the following CASSCF calculation.

To use the supersymmetry (SUPSYM) option we must start with proper orbitals. In this case the a_1 orbitals are symmetry adapted (within the printed accuracy) but not the b_1 and b_2 orbitals. Orbitals $7b_1$ and $7b_2$ must have zero coefficients for all the harmonics except for 4f3+ and 4f3-, respectively. The remaining orbitals of these symmetries (even those not shown) must have zero in the coefficients corresponding to 4f3+ or 4f3-. To clean the orbitals the option CLEANUP of the `RASSCF` program can be used.

Once the orbitals are properly symmetrized we can perform CASSCF calculations on different electronic states. Deriving the types of the molecular electronic states resulting from the electron configurations is not simple in many cases. In general, for a given electronic configuration several electronic states of the molecule will result. Wigner and Witmer derived rules for determining what types of molecular states result from given states of the separated atoms. In chapter VI of reference [222] it is possible to find the tables of the resulting electronic states once the different couplings and the Pauli principle have been applied.

In the present CASSCF calculation we have chosen the active space ($3d$, $4d$, σ, σ^*) with all the 11 valence electrons active. If we consider $4d$ and σ^* as weakly occupied correlating orbitals, we are left with $3d$ and σ (six orbitals), which are to be occupied with 11 electrons. Since the bonding orbital σ (composed mainly of Ni $4s$ and H $1s$) will be doubly occupied in all low lying electronic states, we are left with nine electrons to occupy the $3d$ orbitals. There is thus one hole, and the possible electronic states are: $^2\Sigma^+$, $^2\Pi$, and $^2\Delta$, depending on the orbital where the hole is located. Taking Table 10.4 into account we observe that we have two low-lying electronic states in symmetry 1 (A_1): $^2\Sigma^+$ and $^2\Delta$, and one in each of the other three symmetries: $^2\Pi$ in symmetries 2 (B_1) and 3 (B_2), and $^2\Delta$ in symmetry 4 (A_2). It is not immediately obvious which of these states is the ground state as they are close in energy. It may therefore be necessary to study all of them. It has been found at different levels of theory that the NiH has a $^2\Delta$ ground state [220].

We continue by computing the $^2\Delta$ ground state. The previous SCF orbitals will be the initial orbitals for the CASSCF calculation. First we need to know in which C_{2v} symmetry or symmetries we can compute a Δ state. In the symmetry tables it is determined how the species of the linear molecules are resolved into those of lower symmetry (depends also on the orientation of the molecule). In Table 10.4 is listed the assignment of the different symmetries for the molecule placed on the z axis.

The Δ state has two degenerate components in symmetries a_1 and a_2. Two CASSCF calculations can be performed, one computing the first root of a_2 symmetry and the second for the first root of a_1 symmetry. The `RASSCF` input for the state of a_2 symmetry would be:

```
 &RASSCF &END
Title
 NiH 2Delta CAS s, s*, 3d, 3d'.
Symmetry
    4
Spin
    2
Nactel
   11    0    0
Inactive
    5    2    2    0
Ras2
    6    2    2    2
Thrs
1.0E-07,1.0E-05,1.0E-05
Cleanup
1
  4 6 10 13 18
 18 1 2 3 4 5 6 7 8 9 10 11 12 16 18 19 20 21 22
  4 13 14 15 17
1
  1 7
  10 1 2 3 4 5 6 7 8 10 11
  1 9
1
  1 7
  10 1 2 3 4 5 6 7 9 10 11
  1 8
0
Supsym
1
   4 6 10 13 18
1
   1 7
1
   1 7
0
*Average
*1 2 3
Iter
50,25
LumOrb
End of Input
```

The corresponding input for symmetry a_1 will be identical except for the SYMMETRY keyword

```
Symmetry
    1
```

Table 10.4: Resolution of the $C_{\infty v}$ species in the C_{2v} species.

State symmetry $C_{\infty v}$	State symmetry C_{2v}
Σ^+	A_1
Σ^-	A_2
Π	$B_1 + B_2$
Δ	$A_1 + A_2$
Φ	$B_1 + B_2$
Γ	$A_1 + A_2$

In the `RASSCF` inputs the CLEAnup option will take the initial orbitals (SCF here) and will place zeroes in all the coefficients of orbitals 6, 10, 13, and 18 in symmetry 1, except in coefficients 13, 14, 15, and 17. Likewise all coefficients 13, 14, 15, and 17 of the remaining a_1 orbitals will be set to zero. The same procedure is used in symmetries b_1 and b_2. Once cleaned, and because of the SUPSymmetry option, the δ orbitals 6, 10, 13, and 18 of a_1 symmetry will only rotate among themselves and they will not mix with the remaining a_1 σ orbitals. The same holds true for ϕ orbitals $7b_1$ and $7b_2$ in their respective symmetries.

Orbitals can change order during the calculation. Molcas incorporates a procedure to check the nature of the orbitals in each iteration. Therefore the right behavior of the SUPSym option is guaranteed during the calculation. The procedure can have problems if the initial orbitals are not symmetrized properly. Therefore, the output with the final results should be checked to compare the final order of the orbitals and the final labeling of the SUPSym matrix.

The AVERage option would average the density matrices of symmetries 2 and 3, corresponding to the Π and Φ symmetries in $C_{\infty v}$. In this case it is not necessary to use the option because the two components of the degenerate sets in symmetries b_1 and b_2 have the same occupation and therefore they will have the same shape. The use of the option in a situation like this ($^2\Delta$ and $^2\Sigma^+$ states) leads to convergence problems. The symmetry of the orbitals in symmetries 2 and 3 is retained even if the AVERage option is not used.

The output for the calculation on symmetry 4 (a_2) contains the following lines:

```
      Convergence after  29 iterations
       30   2    2    1 -1507.59605678    -.23E-11   3   9 1  -.68E-06  -.47E-05

                                 Wave function printout:
 occupation of active orbitals, and spin coupling of open shells (u,d: Spin up or down)

      printout of CI-coefficients larger than   .05 for root   1
      energy=  -1507.596057
      conf/sym  111111 22 33 44     Coeff  Weight
         15834  222000 20 20 u0    .97979  .95998
         15838  222000 ud ud u0    .05142  .00264
         15943  2u2d00 ud 20 u0   -.06511  .00424
         15945  2u2d00 20 ud u0    .06511  .00424
         16212  202200 20 20 u0   -.05279  .00279
         16483  u220d0 ud 20 u0   -.05047  .00255
         16485  u220d0 20 ud u0    .05047  .00255
```

```
Natural orbitals and occupation numbers for root  1
sym 1:   1.984969   1.977613   1.995456    .022289    .014882    .005049
sym 2:   1.983081    .016510
sym 3:   1.983081    .016510
sym 4:    .993674    .006884
```

The state is mainly (weight 96%) described by a single configuration (configuration number 15834) which placed one electron on the first active orbital of symmetry 4 (a_2) and the remaining electrons are paired. A close look to this orbital indicates that is has a coefficient -.9989 in the first 3d2- ($3d_{xy}$) function and small coefficients in the other functions. This results clearly indicate that we have computed the $^2\Delta$ state as the lowest root of that symmetry. The remaining configurations have negligible contributions. If the orbitals are properly symmetrized, all configurations will be compatible with a $^2\Delta$ electronic state.

The calculation of the first root of symmetry 1 (a_1) results:

```
      Convergence after  15 iterations
       16   2    3    1 -1507.59605678    -.20E-10   8  15 1   .35E-06  -.74E-05

                                  Wave function printout:
occupation of active orbitals, and spin coupling of open shells (u,d: Spin up or down)

      printout of CI-coefficients larger than   .05 for root   1
      energy=  -1507.596057
      conf/sym  111111 22 33 44     Coeff  Weight
         40800  u22000 20 20 20    -.97979  .95998
         42400  u02200 20 20 20     .05280  .00279

      Natural orbitals and occupation numbers for root  1
      sym 1:    .993674   1.977613   1.995456    .022289    .006884    .005049
      sym 2:   1.983081    .016510
      sym 3:   1.983081    .016510
      sym 4:   1.984969    .014882
```

We obtain the same energy as in the previous calculation. Here the dominant configuration places one electron on the first active orbital of symmetry 1 (a_1). It is important to remember that the orbitals are not ordered by energies or occupations into the active space. This orbital has also the coefficient -.9989 in the first 3d2$-$ ($3d_{x^2-y^2}$) function. We have then computed the other component of the $^2\Delta$ state. As the δ orbitals in different C_{2v} symmetries are not averaged by the program it could happen (not in the present case) that the two energies differ slightly from each other.

The consequences of not using the SUPSYM option are not extremely severe in the present example. If you perform a calculation without the option, the obtained energy is:

```
Convergence after  29 iterations
 30   2    2    1 -1507.59683719    -.20E-11   3   9 1  -.69E-06  -.48E-05
```

As it is a broken symmetry solution the energy is lower than in the other case. This is a typical behavior. If we were using an exact wave function it would have the right symmetry properties, but approximated wave functions do not necessarily fulfill this condition. So, more flexibility leads to lower energy solutions which have broken the orbital symmetry.

If in addition to the $^2\Delta$ state we want to compute the lowest $^2\Sigma^+$ state we can use the adapted orbitals from any of the $^2\Delta$ state calculations and use the previous `RASSCF` input without the CLEANUP option. The orbitals have not changed place in this example. If they do, one has to change the labels in the SUPSYM option. The simplest way to compute the lowest excited

$^2\Sigma^+$ state is having the unpaired electron in one of the σ orbitals because none of the other configurations, δ^3 or π^3, leads to the $^2\Sigma^+$ term. However, there are more possibilities such as the configuration $\sigma^1\sigma^1\sigma^1$; three nonequivalent electrons in three σ orbitals. In actuality the lowest $^2\Sigma^+$ state must be computed as a doublet state in symmetry A_1. Therefore, we set the symmetry in the RASSCF to 1 and compute the second root of the symmetry (the first was the $^2\Delta$ state):

```
CIRoot
1 2
2
```

Of course the SUPSym option must be maintained. The use of CIROot indicates that we are computing the second root of that symmetry. The obtained result:

```
      Convergence after  33 iterations
        9   2    3    2 -1507.58420263    -.44E-10   2  11 2  -.12E-05   .88E-05

                                 Wave function printout:
 occupation of active orbitals, and spin coupling of open shells (u,d: Spin up or down)

      printout of CI-coefficients larger than   .05 for root   1
      energy=  -1507.584813
      conf/sym  111111 22 33 44     Coeff  Weight
         40800  u22000 20 20 20   -.97917  .95877

      printout of CI-coefficients larger than   .05 for root   2
      energy=  -1507.584203
      conf/sym  111111 22 33 44     Coeff  Weight
         40700  2u2000 20 20 20    .98066  .96169

      Natural orbitals and occupation numbers for root  2
      sym 1:   1.983492    .992557   1.995106    .008720    .016204    .004920
      sym 2:   1.983461    .016192
      sym 3:   1.983451    .016192
      sym 4:   1.983492    .016204
```

As we have used two as the dimension of the CI matrix employed in the CI Davidson procedure we obtain the wave function of two roots, although the optimized root is the second. Root 1 places one electron in the first active orbital of symmetry one, which is a 3d2+ ($3d_{x^2-y^2}$) δ orbital. Root 2 places the electron in the second active orbital, which is a σ orbital with a large coefficient (.9639) in the first 3d0 ($3d_{z^2}$) function of the nickel atom. We have therefore computed the lowest $^2\Sigma^+$ state. The two $^2\Sigma^+$ states resulting from the configuration with the three unpaired σ electrons is higher in energy at the CASSCF level. If the second root of symmetry a_1 had not been a $^2\Sigma^+$ state we would have to study higher roots of the same symmetry.

It is important to remember that the active orbitals are not ordered at all within the active space. Therefore, their order might vary from calculation to calculation and, in addition, no conclusions about the orbital energy, occupation or any other information can be obtained from the order of the active orbitals.

We can compute also the lowest $^2\Pi$ excited state. The simplest possibility is having the configuration π^3, which only leads to one $^2\Pi$ state. The unpaired electron will be placed in either one b_1 or one b_2 orbital. That means that the state has two degenerate components and we can compute it equally in both symmetries. There are more possibilities, such as the configuration $\pi^3\sigma^1\sigma^1$ or the configuration $\pi^3\sigma^1\delta^1$. The resulting $^2\Pi$ state will always have two degenerate components in symmetries b_1 and b_2, and therefore it is the wave function analysis which gives us the information of which configuration leads to the lowest $^2\Pi$ state.

For NiH it turns out to be non trivial to compute the $^2\Pi$ state. Taking as initial orbitals the previous SCF orbitals and using any type of restriction such as the CLEAnup, SUPSym or AVERage options lead to severe convergence problems like these:

```
  45   9   17    1 -1507.42427683    -.65E-02   6  18 1  -.23E-01  -.15E+00
  46   5   19    1 -1507.41780710     .65E-02   8  15 1   .61E-01  -.15E+00
  47   9   17    1 -1507.42427683    -.65E-02   6  18 1  -.23E-01  -.15E+00
  48   5   19    1 -1507.41780710     .65E-02   8  15 1   .61E-01  -.15E+00
  49   9   17    1 -1507.42427683    -.65E-02   6  18 1  -.23E-01  -.15E+00
  50   5   19    1 -1507.41780710     .65E-02   8  15 1   .61E-01  -.15E+00

 No convergence after  50 iterations
  51   9   19    1 -1507.42427683    -.65E-02   6  18 1  -.23E-01  -.15E+00
```

The calculation, however, converges in an straightforward way if none of those tools are used:

```
       Convergence after  33 iterations
        34   2    2    1 -1507.58698677    -.23E-12   3   8 2  -.72E-06  -.65E-05

                                    Wave function printout:
 occupation of active orbitals, and spin coupling of open shells (u,d: Spin up or down)

       printout of CI-coefficients larger than   .05 for root   1
       energy=  -1507.586987
       conf/sym  111111 22 33 44     Coeff  Weight
          15845  222000 u0 20 20    .98026  .96091
          15957  2u2d00 u0 ud 20    .05712  .00326
          16513  u220d0 u0 20 ud   -.05131  .00263

       Natural orbitals and occupation numbers for root  1
       sym 1:   1.984111   1.980077   1.995482    .019865    .015666    .004660
       sym 2:    .993507    .007380
       sym 3:   1.982975    .016623
       sym 4:   1.983761    .015892
```

The π (and ϕ) orbitals, both in symmetries b_1 and b_2, are, however, differently occupied and therefore are not equal as they should be:

```
    Molecular orbitals for sym species 2     Molecular orbitals for symmetry species 3
    ORBITAL        3         4               ORBITAL        3         4
    ENERGY       .0000     .0000             ENERGY       .0000     .0000
    OCC. NO.     .9935     .0074             OCC. NO.    1.9830     .0166

  1 NI  2px      .0001     .0002           1 NI  2py      .0018    -.0001
  2 NI  2px      .0073     .0013           2 NI  2py      .0178    -.0002
  3 NI  2px     -.0155     .0229           3 NI  2py     -.0197    -.0329
  4 NI  2px      .0041     .0227           4 NI  2py      .0029    -.0254
  5 NI  3d1+     .9990    -.0199           5 NI  3d1-     .9998    -.0131
  6 NI  3d1+   -.0310    -.8964           6 NI  3d1-     .0128     .9235
  7 NI  3d1+   -.0105     .4304           7 NI  3d1-     .0009    -.3739
  8 NI  4f1+   -.0050     .0266           8 NI  4f3-     .0001    -.0003
  9 NI  4f3+    .0001     .0000           9 NI  4f1-    -.0050    -.0177
 10 H   2px      .0029    -.0149          10 H   2py      .0009     .0096
 11 H   2px     -.0056    -.0003          11 H   2py     -.0094    -.0052
```

Therefore what we have is a symmetry broken solution. To obtain a solution which is not of broken nature the π and ϕ orbitals must be equivalent. The tool to obtain equivalent orbitals is the AVERage option, which averages the density matrices of symmetries b_1 and

b_2. But starting with any of the preceding orbitals and using the AVERage option lead again to convergence problems. It is necessary to use better initial orbitals; orbitals which have already equal orbitals in symmetries b_1 and b_2. One possibility is to perform a SCF calculation on the NiH cation explicitly indicating occupation one in the two higher occupied π orbitals (symmetries 2 and 3):

```
 &SCF &END
TITLE
 NiH cation
OCCUPIED
 8 3 3 1
OCCNO
2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0
2.0 2.0 1.0                    <-- Note the extra occupation
2.0 2.0 1.0                    <-- Note the extra occupation
2.0
IVO
END OF INPUT
```

It can take some successive steps to obtain a converged calculation using the CLEAnup, SUPSym, and AVERage options. The calculation with a single root did not converge clearly. We obtained, however, a converged result for the lowest $^2\Pi$ state of NiH by computing two averaged CASSCF roots and setting a weight of 90% for the first root using the keyword:

```
CIROot
 2 2
 1 2
 9 1
```

```
                              Wave function printout:
 occupation of active orbitals, and spin coupling of open shells (u,d: Spin up or down)

      printout of CI-coefficients larger than   .05 for root   1
      energy=  -1507.566492
      conf/sym  111111 22 33 44     Coeff  Weight
          4913  222u00 20 d0 u0   -.05802  .00337
         15845  222000 u0 20 20    .97316  .94703
         15953  2u2d00 u0 20 20    .05763  .00332
         16459  2u20d0 u0 20 ud   -.05283  .00279

      Natural orbitals and occupation numbers for root  1
      sym 1:   1.972108   1.982895   1.998480    .028246    .016277    .007159
      sym 2:    .997773    .007847
      sym 3:   1.978019    .016453
      sym 4:   1.978377    .016366
```

The energy of the different states (only the first one shown above) is printed on the top of their configuration list. The converged energy is simply an average energy. The occupation numbers obtained in the section of the `RASSCF` output printed above are the occupation numbers of the natural orbitals of the corresponding root. They differ from the occupation numbers printed in the molecular orbital section where we have pseudonatural molecular orbitals and average occupation numbers. On top of each of the valence π orbitals an average occupation close to 1.5e will be printed; this is a consequence of the the averaging procedure.

The results obtained are only at the CASSCF level. Additional effects have to be considered and included. The most important of them is the dynamical correlation effect which can be added by computing, for instance, the CASPT2 energies. The reader can find a detailed explanation of the different approaches in ref. [220], and a careful discussion of their consequences and solutions in ref. [223].

We are going, however, to point out some details. In the first place the basis set must include up to *g* functions for the transition metal atom and up to *d* functions for the hydrogen. Relativistic effects must be taken into account, at least in a simple way as a first order correction. The keyword RELINT must be then included in the SEWARD input to compute the mass-velocity and one-electron Darwin contact term integrals and obtain a first-order correction to the energy with respect to relativistic effects at the CASSCF level in the RASSCF output. Scalar relativistic effects can be also included according the Douglas-Kroll or the Barysz-Sadlej-Snijders transformations, as it will be explained in section 10.7.

The CASPT2 input needed to compute the second-order correction to the energy will include the number of the CASSCF root to compute. For instance, for the first root of each symmetry:

```
 &CASPT2 &END
Title
 NiH
Frozen
5 2 2 0
Maxit
30
Lroot
1
End of input
```

The number of frozen orbitals taken by CASPT2 will be that specified in the RASSCF input except if this is changed in the CASPT2 input. In the perturbative step we have frozen all the occupied orbitals except the active ones. This is motivated by the desire to include exclusively the dynamical correlation related to the valence electrons. In this way we neglect correlation between core electrons, named core-core correlation, and between core and valence electrons, named core-valence correlation. This is not because the calculation is smaller but because of the inclusion of those type of correlation in a calculation designed to treat valence correlation is an inadequate approach. Core-core and core-valence correlation requires additional basis functions of the same spatial extent as the occupied orbitals being correlated, but with additional radial and angular nodes. Since the spatial extent of the core molecular orbitals is small, the exponents of these correlating functions must be much larger than those of the valence optimized basis sets. The consequence is that we must avoid the inclusion of the core electrons in the treatment in the first step. Afterwards, the amount of correlation introduced by the core electrons can be estimated in separated calculations for the different states and those effects added to the results with the valence electrons.

Core-valence correlation effects of the $3s$ and $3p$ nickel shells can be studied by increasing the basis set flexibility by uncontracting the basis set in the appropriate region. There are different possibilities. Here we show the increase of the basis set by four s, four p, and four d functions. f functions contribute less to the description of the $3s$ and $3p$ shells and can be excluded. The uncontracted exponents should correspond to the region where the $3s$ and $3p$ shells present their density maximum. Therefore, first we compute the absolute maxima of the radial distribution of the involved orbitals, then we determine the primitive gaussian functions which have their maxima in the same region as the orbitals and therefore which exponents should be uncontracted. The final basis set will be the valence basis set used before plus the new added functions. In the present example the SEWARD input can be:

```
 &SEWARD &END
Title
 NiH G.S.
Symmetry
X Y
*RelInt
Basis set
Ni.ANO-L...5s4p3d1f.
Ni    0.00000   0.00000   0.000000   Bohr
End of basis
Basis set
Ni....4s4p4d. / Inline
 0.  2
* Additional s functions
 4 4
3.918870 1.839853 0.804663 0.169846
 1. 0. 0. 0.
 0. 1. 0. 0.
 0. 0. 1. 0.
 0. 0. 0. 1.
* Additional p functions
 4 4
2.533837 1.135309 0.467891 0.187156
 1. 0. 0. 0.
 0. 1. 0. 0.
 0. 0. 1. 0.
 0. 0. 0. 1.
* Additional d functions
 4 4
2.551303 1.128060 0.475373 0.182128
 1. 0. 0. 0.
 0. 1. 0. 0.
 0. 0. 1. 0.
 0. 0. 0. 1.
Nix   0.00000   0.00000   0.000000   Bohr
End of basis
Basis set
H.ANO-L...3s2p.
H     0.000000  0.000000  2.747000   Bohr
End of basis
End of Input
```

We have used a special format to include the additional functions. We include the additional $4s4p4d$ functions for the nickel atom. The additional basis set input must use a dummy label (Nix here), the same coordinates of the original atom, and specify a CHARGE equal to zero, whether in an Inline basis set input as here or by specifically using keyword CHARGE. It is not necessary to include the basis set with the Inline format. A library can be created for this purpose. In this case the label for the additional functions could be:

```
Ni.Uncontracted...4s4p4d. / AUXLIB
Charge
0
```

and a proper link to AUXLIB should be included in the script (or in the input if one uses AUTO).

Now the CASPT2 is going to be different to include also the correlation related to the $3s, 3p$ shell of the nickel atom. Therefore, we only freeze the $1s, 2s, 2p$ shells:

```
 &CASPT2 &END
Title
 NiH. Core-valence.
```

```
Frozen
3 1 1 0
Maxit
30
Lroot
1
End of input
```

A final effect one should study is the basis set superposition error (BSSE). In many cases it is a minor effect but it is an everpresent phenomenon which should be investigated when high accuracy is required, especially in determining bond energies, and not only in cases with weakly interacting systems, as is frequently believed. The most common approach to estimate this effect is the counterpoise correction: the separated fragment energies are computed in the total basis set of the system. For a discussion of this issue see Refs. [223, 224]. In the present example we would compute the energy of the isolated nickel atom using a SEWARD input including the full nickel basis set plus the hydrogen basis set in the hydrogen position but with the charge set to zero. And then the opposite should be done to compute the energy of isolated hydrogen. The BSSE depends on the separation of the fragments and must be estimated at any computed geometry. For instance, the SEWARD input necessary to compute the isolated hydrogen atom at a given distance from the ghost nickel basis set including core uncontracted functions is:

```
>>UNIX mkdir AUXLIB
>>COPY $CurrDir/NiH.NewLib AUXLIB/UNCONTRACTED
 &SEWARD &END
Title
 NiH. 3s3p + H (BSSE)
Symmetry
X Y
RelInt
Basis set
Ni.ANO-L...5s4p3d1f.
Ni    0.00000    0.00000    0.000000    Bohr
Charge
0.0
End of basis
Basis set
Ni.Uncontracted...4s4p4d. / AUXLIB
Nix   0.00000    0.00000    0.000000    Bohr
Charge
0.0
End of basis
Basis set
H.ANO-L...3s2p.
H     0.000000   0.000000   2.747000    Bohr
End of basis
End of Input
```

Once the energy of each of the fragments with the corresponding ghost basis set of the other fragment is determined, the energies of the completely isolated fragments can be computed and subtracted from those which have the ghost basis sets. Other approaches used to estimate the BSSE effect are discussed in Ref. [223].

The results obtained at the CASPT2 level are close to those obtained by MRCI+Q and ACPF treatments but more accurate. They match well with experiment. The difference is that all the configuration functions (CSFs) of the active space can be included in CASPT2 in the zeroth-order references for the second-order perturbation calculation [220], while the other methods have to restrict the number of configurations.

Calculations of linear molecules become more and more complicated when the number of unpaired electrons increases. In the following sections we will discuss the more complicated situation occurring in the Ni_2 molecule.

10.1.2 A diatomic homonuclear molecule: C_2

C_2 is a classical example of a system where near-degeneracy effects have large amplitudes even near the equilibrium internuclear separation. The biradical character of the ground state of the molecule suggest that a single configurational treatment will not be appropriate for accurate descriptions of the spectroscopic constants [79]. There are two nearly degenerate states: $^1\Sigma_g^+$ and $^3\Pi_u$. The latter was earlier believed to be the ground state, an historical assignment which can be observed in the traditional labeling of the states.

As C_2 is a $D_{\infty h}$ molecule, we have to compute it in D_{2h} symmetry. We make a similar analysis as for the C_{2v} case. We begin by classifying the functions in $D_{\infty h}$ in Table 10.5. The molecule is placed on the z axis.

Table 10.5: Classification of the spherical harmonics in the $D_{\infty h}$ group[a].

Symmetry	Spherical harmonics			
σ_g	s	d_{z^2}		
σ_u	p_z	f_{z^3}		
π_g	d_{xz}	d_{yz}		
π_u	p_x	p_y	$f_{x(z^2-y^2)}$	$f_{y(z^2-x^2)}$
δ_g	$d_{x^2-y^2}$	d_{xy}		
δ_u	f_{xyz}	$f_{z(x^2-y^2)}$		
ϕ_u	f_{x^3}	f_{y^3}		

[a] Functions placed on the symmetry center.

Table 10.6 classifies the functions and orbitals into the symmetry representations of the D_{2h} symmetry. Note that in table 10.6 subindex b stands for bonding combination and a for antibonding combination.

The order of the symmetries, and therefore the number they have in MOLCAS, depends on the generators used in the `SEWARD` input. This must be carefully checked at the beginning of any calculation. In addition, the orientation of the molecule on the cartesian axis can change the labels of the symmetries. In Table 10.6 for instance we have used the order and numbering of a calculation performed with the three symmetry planes of the D_{2h} point group (X Y Z in the `SEWARD` input) and the z axis as the intermolecular axis (that is, x and y are equivalent in D_{2h}). Any change in the orientation of the molecule will affect the labels of the orbitals and states. In this case the π orbitals will belong to the b_{3u}, b_{2u}, b_{2g}, and b_{3g} symmetries. For instance, with x as the intermolecular axis b_{3u} and b_{3g} will be replaced by b_{1u} and b_{1g}, respectively, and finally with y as the intermolecular axis b_{1u}, b_{3u}, b_{3g}, and b_{1g} would be the π orbitals.

It is important to remember that MOLCAS works with symmetry adapted basis functions. Only the symmetry independent atoms are required in the `SEWARD` input. The remaining ones will be generated by the symmetry operators. This is also the case for the molecular orbitals. MOLCAS will only print the coefficients of the symmetry adapted basis functions.

Table 10.6: Classification of the spherical harmonics and $D_{\infty h}$ orbitals in the D_{2h} group[a].

Symm.[b]	Spherical harmonics (orbitals in $D_{\infty h}$)					
a_g (1)	s_b (σ_g)	p_{zb} (σ_g)	d_{z^2b} (σ_g)	$d_{x^2-y^2b}$ (δ_g)	f_{z^3b} (σ_g)	$f_{z(x^2-y^2)b}$ (δ_g)
b_{3u}(2)	p_{xb} (π_u)	d_{xzb} (π_u)	$f_{x(z^2-y^2)b}$ (π_u)	f_{x^3b} (ϕ_u)		
b_{2u}(3)	p_{yb} (π_u)	d_{yzb} (π_u)	$f_{y(z^2-x^2)b}$ (π_u)	f_{y^3b} (ϕ_u)		
b_{1g}(4)	d_{xyb} (δ_g)	f_{xyzb} (δ_g)				
b_{1u}(5)	s_a (σ_u)	p_{za} (σ_u)	d_{z^2a} (σ_u)	$d_{x^2-y^2a}$ (δ_u)	f_{z^3a} (σ_u)	$f_{z(x^2-y^2)a}$ (δ_u)
b_{2g}(6)	p_{ya} (π_g)	d_{yza} (π_g)	$f_{y(z^2-x^2)a}$ (π_g)	f_{y^3a} (ϕ_g)		
b_{3g}(7)	p_{xa} (π_g)	d_{xza} (π_g)	$f_{x(z^2-y^2)a}$ (π_g)	f_{x^3a} (ϕ_g)		
a_u (8)	d_{xya} (δ_u)	f_{xyza} (δ_u)				

[a] Subscripts a and b refer to the bonding and antibonding combination of the AO's, respectively.

[b] In parenthesis the number of the symmetry in MOLCAS. Note that the number and order of the symmetries depend on the generators and the orientation of the molecule.

The necessary information to obtain the complete set of orbitals is contained in the SEWARD output. Consider the case of the a_g symmetry:

```
                               **************************************************
                               ******** Symmetry adapted Basis Functions ********
                               **************************************************

                      Irreducible representation : ag
                      Basis function(s) of irrep:

            Basis Label        Type   Center Phase Center Phase
              1   C            1s0       1     1      2     1
              2   C            1s0       1     1      2     1
              3   C            1s0       1     1      2     1
              4   C            1s0       1     1      2     1
              5   C            2pz       1     1      2    -1
              6   C            2pz       1     1      2    -1
              7   C            2pz       1     1      2    -1
              8   C            3d0       1     1      2     1
              9   C            3d0       1     1      2     1
             10   C            3d2+      1     1      2     1
             11   C            3d2+      1     1      2     1
             12   C            4f0       1     1      2    -1
             13   C            4f2+      1     1      2    -1
```

The previous output indicates that symmetry adapted basis function 1, belonging to the a_g representation, is formed by the symmetric combination of a s type function centered on atom C and another s type function centered on the redundant center 2, the second carbon atom. Combination $s + s$ constitutes a bonding σ_g-type orbital. For the p_z function however the combination must be antisymmetric. It is the only way to make the p_z orbitals overlap and form a bonding orbital of a_g symmetry. Similar combinations are obtained for the remaining basis sets of the a_g and other symmetries.

The molecular orbitals will be combinations of these symmetry adapted functions. Consider the a_g orbitals:

```
   SCF orbitals

   Molecular orbitals for symmetry species 1

   ORBITAL         1           2           3           4           5           6
```

```
    ENERGY    -11.3932   -1.0151    -.1138     .1546     .2278     .2869
    OCC. NO.    2.0000    2.0000     .0098     .0000     .0000     .0000

  1 C   1s0     1.4139    -.0666    -.0696     .2599     .0626     .0000
  2 C   1s0      .0003    1.1076    -.6517    1.0224     .4459     .0000
  3 C   1s0      .0002    -.0880    -.2817     .9514     .0664     .0000
  4 C   1s0      .0000    -.0135    -.0655     .3448    -.0388     .0000
  5 C   2pz     -.0006    -.2581   -1.2543    1.1836     .8186     .0000
  6 C   2pz      .0000     .1345    -.0257    2.5126    1.8556     .0000
  7 C   2pz      .0005    -.0192    -.0240     .7025     .6639     .0000
  8 C   3d0      .0003     .0220    -.0005    -.9719     .2430     .0000
  9 C   3d0     -.0001    -.0382    -.0323    -.8577     .2345     .0000
 10 C   3d2+     .0000     .0000     .0000     .0000     .0000    -.7849
 11 C   3d2+     .0000     .0000     .0000     .0000     .0000    -.7428
 12 C   4f0     -.0002    -.0103    -.0165     .0743     .0081     .0000
 13 C   4f2+     .0000     .0000     .0000     .0000     .0000    -.0181
```

In MOLCAS outputs only 13 coefficients for orbital are going to be printed because they are the coefficients of the symmetry adapted basis functions. If the orbitals were not composed by symmetry adapted basis functions they would have, in this case, 26 coefficients, two for type of function (following the scheme observed above in the SEWARD output), symmetrically combined the s and d functions and antisymmetrically combined the p and f functions.

To compute $D_{\infty h}$ electronic states using the D_{2h} symmetry we need to go to the symmetry tables and determine how the species of the linear molecules are resolved into those of lower symmetry (this depends also on the orientation of the molecule [222]). Table 10.7 lists the case of a $D_{\infty h}$ linear molecule with z as the intermolecular axis.

Table 10.7: Resolution of the $D_{\infty h}$ species in the D_{2h} species.

State symmetry $D_{\infty h}$	State symmetry D_{2h}
Σ_g^+	A_g
Σ_u^+	B_{1u}
Σ_g^-	B_{1g}
Σ_u^-	A_u
Π_g	B_{2g} + B_{3g}
Π_u	B_{2u} + B_{3u}
Δ_g	A_g + B_{1g}
Δ_u	A_u + B_{1u}
Φ_g	B_{2g} + B_{3g}
Φ_u	B_{2u} + B_{3u}
Γ_g	A_g + B_{1g}
Γ_u	A_u + B_{1u}

To compute the ground state of C_2, a $^1\Sigma_g^+$ state, we will compute a singlet state of symmetry A_g (1 in this context). The input files for a CASSCF calculation on the C_2 ground state will be:

```
 &SEWARD &END
Title
 C2
Symmetry
  X  Y  Z
Basis set
C.ANO-L...4s3p2d1f.
```

```
C        .00000000    .00000000     1.4
End of basis
End of input
 &SCF &END
Title
 C2
ITERATIONS
 40
Occupied
  2  1  1  0  2  0  0  0
End of input
 &RASSCF &END
Title
 C2
Nactel
  4  0  0
Spin
  1
Symmetry
  1
Inactive
  2  0  0  0  2  0  0  0
Ras2
  1  1  1  0  1  1  1  0
*Average
*2 2 3 6 7
Supsymmetry
1
 3 6 9 11
1
 1 6
1
 1 6
0
1
 3 5 8 12
1
 1 6
1
 1 6
0
Iter
50,25
Lumorb
End of input
```

In this case the SCF orbitals are already clean symmetry adapted orbitals (within the printed accuracy). We can then directly use the SUPSym option. In symmetries a_g and b_{1u} we restrict the rotations among the σ and the δ orbitals, and in symmetries b_{3u}, b_{2u}, b_{2g}, and b_{3g} the rotations among π and ϕ orbitals. Additionally, symmetries b_{3u} and b_{2u} and symmetries b_{2g} and b_{3g} are averaged, respectively, by using the AVERage option. They belong to the Π_u and Π_g representations in $D_{\infty h}$, respectively.

A detailed explanation on different CASSCF calculations on the C_2 molecule and their states can be found elsewhere [79]. Instead we include here an example of how to combine the use of UNIX shell script commands with Molcas as a powerful tool.

The following example computes the transition dipole moment for the transition from the $^1\Sigma_g^+$ state to the $^1\Pi_u$ state in the C_2 molecule. This transition is known as the Phillips bands [222]. This is not a serious attempt to compute this property accurately, but serves as an example of how to set up an automatic calculation. The potential curves are computed using CASSCF wavefunctions along with the transition dipole moment.

Starting orbitals are generated by computing a CI wavefunction once and using the natural orbitals. We loop over a set of distances, compute the CASSCF wave functions for both states and use RASSI to compute the TDMs. Several UNIX commands are used to manipulate input and output files, such as grep, sed, and the *awk* language. For instance, an explicit 'sed' is used to insert the geometry into the seward input; the final CASSCF energy is extracted with an explicit 'grep', and the TDM is extracted from the RASSI output using an *awk* script. We are not going to include the *awk* scripts here. Other tools can be used to obtain and collect the data.

In the first script, when the loop over geometries is done, four files are available: geom.list (contains the distances), tdm.list (contains the TDMs), e1.list (contains the energy for the $^1\Sigma_g^+$ state), and e2.list (contains the energy for the $^1\Pi_u$ state). In the second script the vibrational wave functions for the two states and the vibrationally averaged TDMs are now computed using the VIBROT program. We will retain the RASSCF outputs in the scratch directory to check the wave function. It is always dangerous to assume that the wave functions will be correct in a CASSCF calculation. Different problems such as root flippings or incorrect orbitals rotating into the active space are not uncommon. Also, it is always necessary to control that the CASSCF calculation has converged. The first script (Korn shell) is:

```
#!/bin/ksh
#
# perform some initializations
#
export Project='C2'
export WorkDir=/temp/$LOGNAME/$Project
export Home=/u/$LOGNAME/$Project
echo "No log" > current.log
trap 'cat current.log ; exit 1' ERR
mkdir $WorkDir
cd $WorkDir
#
# Loop over the geometries and generate input for vibrot
#
list="1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 5.0 10.0"
scf='yes'
print "Sigma" > e1.list
print "Pi" > e2.list
for geom in $list
do
   #--- run seward
   print "Dist $geom" >> geom.list
   sed -e "s/#/$geom/" $Home/$Project.seward.input > seward.input
   molcas seward.input > current.log
   #--- optionally run scf, motra, guga and mrci to obtain good starting orbitals
   if [ "$scf" = 'yes' ]
   then
      scf='no'
      molcas    $Home/$Project.scf.input > current.log
      molcas    $Home/$Project.motra.input > current.log
      molcas    $Home/$Project.guga.input > current.log
      molcas    $Home/$Project.mrci.input > current.log
      cp $Project.CiOrb $Project.RasOrb1
      cp $Project.CiOrb $Project.RasOrb2
   fi
   #--- rasscf wavefunction for 1Sg+
   ln -fs $Project.Job001 JOBIPH
   ln -fs $Project.RasOrb1 INPORB
   molcas   $Home/$Project.rasscf1.input > current.log
   cat current.log >> rasscf1.log
   cat current.log | grep -i 'average ci' >> e1.list
   cp $Project.RasOrb $Project.RasOrb1
   rm -f JOBIPH INPORB
   #--- rasscf wavefunction for 1Pu
   ln -fs $Project.Job002 JOBIPH
   ln -fs $Project.RasOrb2 INPORB
   molcas   $Home/$Project.rasscf2.input > current.log
   cat current.log >> rasscf2.log
   cat current.log | grep -i 'average ci' >> e2.list
   cp $Project.RasOrb $Project.RasOrb2
   rm -f JOBIPH INPORB
   #--- rassi to obtain transition
   ln -fs $Project.Job001 JOB001
   ln -fs $Project.Job002 JOB002
   molcas   $Home/$Project.rassi.input > current.log
   awk -f $Home/tdm.awk current.log >> tdm.list
   rm -f JOB001 JOB002
   #---
done
#
# Finished so clean up the files.
#
print "Calculation finished" >&2
cd -
rm $WorkDir/molcas.temp*
#rm -r $WorkDir
exit 0
```

In a second script we will compute the vibrational wave functions

```
#!/bin/ksh
#
# perform some initializations
#
export Project='C2'
export WorkDir=/temp/$LOGNAME/$Project
export Home=/u/$LOGNAME/$Project
echo "No log" > current.log
trap 'cat current.log ; exit 1' ERR
mkdir $WorkDir
cd $WorkDir
#
# Build vibrot input
#
cp e1.list $Home
cp e2.list $Home
cp geom.list $Home
cp tdm.list $Home
#---
cat e1.list geom.list | awk -f $Home/wfn.awk > vibrot1.input
cat e2.list geom.list | awk -f $Home/wfn.awk > vibrot2.input
cat tdm.list geom.list | awk -f $Home/tmc.awk > vibrot3.input
#---
ln -fs $Project.VibWvs1 VIBWVS
molcas  vibrot1.input > current.log
cat current.log
rm -f VIBWVS
#---
ln -fs $Project.VibWvs2 VIBWVS
molcas  vibrot2.input > current.log
cat current.log
rm -f VIBWVS
#---
ln -fs $Project.VibWvs1 VIBWVS1
ln -fs $Project.VibWvs2 VIBWVS2
molcas  vibrot3.input > current.log
cat current.log
rm -f VIBWVS1 VIBWVS2
```

```
#
# Finished so clean up the files.
#
print "Calculation finished" >&2
cd -
rm $WorkDir/molcas.temp*
#rm -r $WorkDir
exit 0
```

The input for the first part of the calculations include the SEWARD, SCF, MOTRA, GUGA, and MRCI inputs:

```
 &SEWARD &END
Title
 C2
Pkthre
1.0D-11
Symmetry
  X  Y  Z
Basis set
C.ANO-S...3s2p.
C        .00000000    .00000000   #
End of basis
End of input
 &SCF &END
Title
 C2
ITERATIONS
 40
Occupied
  2  1  1  0  2  0  0  0
End of input
 &MOTRA &END
Title
 C2 molecule
Frozen
 1 0 0 0 1 0 0 0
LumOrb
End of input
 &GUGA &END
Title
 C2 molecule
Electrons
    8
Spin
    1
Inactive
    1    1    1    0    1    0    0    0
Active
    0    0    0    0    0    0    0    0
CiAll
    1
End of Input
 &MRCI &END
Title
 C2 molecule
SDCI
End of input
```

We are going to use a small ANO [$3s2p$] basis set because our purpose it is not to obtain an extreme accuracy. In the SEWARD input the sign '#' will be replaced by the right distance using the 'sed' command. In the MOTRA input we have frozen the two core orbitals in the molecule, which will be recognized by the MRCI program. The GUGA input defines the reference space of configurations for the subsequent MRCI or ACPF calculation. In this case the valence orbitals are doubly occupied and there is only one reference configuration (they are included as inactive). We thus use one single configuration to perform the SDCI calculation and obtain the initial set of orbitals for the CASSCF calculation.

The lowest $^1\Sigma_g^+$ state in C_2 is the result of the electronic configuration [core]$(2\sigma_g)^2$ $(2\sigma_u)^2$ $(1\pi_u)^4$. Only one electronic state is obtained from this configuration. The configuration $(1\pi_u)^3$ $(3\sigma_g)^1$ is close in energy and generates two possibilities, one $^3\Pi_u$ and one $^1\Pi_u$ state. The former is the lowest state of the Swan bands, and was thought to be the ground state of the molecule. Transitions to the $^1\Pi_u$ state are known as the Phillips band and this is the state we are going to compute. We have the possibility to compute the state in symmetry b_{3u} or b_{2u} (MOLCAS symmetry groups 2 and 3, respectively) in the D_{2h} group, because both represent the degenerate Π_u symmetry in $D_{\infty h}$.

The RASSCF input file to compute the two states are:

```
 &RASSCF &END
Title
 C2 1Sigmag+ state.
Nactel
  4  0  0
Spin
  1
Symmetry
  1
Inactive
  2  0  0  0  2  0  0  0
Ras2
  1  1  1  0  1  1  1  0
*Average
*2 2 3 6 7
OutOrbitals
 Natural
 1
Iter
50,25
Lumorb
End of input
```

```
 &RASSCF &END
Title
 C2 1Piu state.
Nactel
  4  0  0
Spin
  1
Symmetry
  2
Inactive
  2  0  0  0  2  0  0  0
Ras2
  1  1  1  0  1  1  1  0
Average
2 2 3 6 7
OutOrbitals
 Natural
 1
Iter
50,25
Lumorb
End of input
```

We can skip the SUPSYM option because our basis set contains only s, p functions and no undesired rotations can happen. Symmetries b_{3u} and b_{2u} on one hand and b_{2g} and b_{3g} on the other are averaged. Notice that to obtain natural orbitals we have used keyword OUTORBITALS instead of the old `RASREAD` program. In addition, we need the `RASSI` input:

```
 &RASSI &END
NrOfJobiphs
 2 1 1
 1
 1
End of input
```

The VIBROT inputs to compute the vibrational-rotational analysis and spectroscopic constants of the state should be:

```
 &VIBROT &END
RoVibrational spectrum
Title
 Vib-Rot spectrum for C2. 1Sigmag+
Atoms
0 C 0 C
Grid
400
Range
2.0 10.0
Vibrations
3
Rotations
0 4
Orbital
0
Potential
2.2 -75.42310136
...
End of input
```

Under the keyword POTENTIAL the bond distance and potential energy (both in au) of the corresponding state must be included. In this case we are going to compute three vibrational quanta and four rotational quantum numbers. For the $^1\Pi_u$ state, the keyword ORBITAL must be set to one, corresponding to the orbital angular momentum of the computed state. `VIBROT` fits the potential curve to an analytical curve using splines. The ro-vibrational Schrödinger equation is then solved numerically (using Numerov's method) for one vibrational state at a time and for the specified number of rotational quantum numbers. File *`VIBWVS`* will contain the corresponding wave function for further use.

Just to give some of the results obtained, the spectroscopic constants for the $^1\Sigma_g^+$ state were:

```
Re(a)                1.4461
De(ev)               3.1088
D0(ev)               3.0305
we(cm-1)         .126981E+04
wexe(cm-1)      -.130944E+02
weye(cm-1)      -.105159E+01
Be(cm-1)         .134383E+01
Alphae(cm-1)     .172923E-01
Gammae(cm-1)     .102756E-02
Dele(cm-1)       .583528E-05
Betae(cm-1)      .474317E-06
```

and for the $^1\Pi_u$ state:

```
Re(a)                 1.3683
De(ev)                2.6829
D0(ev)                2.5980
we(cm-1)          .137586E+04
wexe(cm-1)       -.144287E+02
weye(cm-1)        .292996E+01
Be(cm-1)          .149777E+01
Alphae(cm-1)      .328764E-01
Gammae(cm-1)      .186996E-02
Dele(cm-1)        .687090E-05
Betae(cm-1)      -.259311E-06
```

To compute vibrationally averaged TDMs the `VIBROT` input must be:

```
 &VIBROT &END
Transition moments
Observable
Transition dipole moment
2.2 0.412805
...
End of input
```

Keyword OBSErvable indicates the start of input for radial functions of observables other than the energy. In the present case the vibrational-rotational matrix elements of the transition dipole moment function will be generated. The values of the bond distance and the TDM at each distance must be then included in the input. VIBROT also requires the *`VIBWVS1`* and *`VIBWVS2`* files containing the vibrational wave functions of the involved electronic states. The results obtained contain matrix elements, transition moments over vibrational wave functions, and the lifetimes of the transition among all the computed vibrational-rotational states. The radiative lifetime of a vibrational level depends on the sum of the transition probabilities to all lower vibrational levels in all lower electronic states. If rotational effects are neglected, the lifetime (τ'_v) can be written as

$$\tau'_v = (\sum_{v''} A_{v'v''})^{-1} \tag{10.1}$$

where v' and v'' are the vibrational levels of the lower and upper electronic state and $A_{v'v''}$ is the Einstein A coefficient (ns^{-1}) computed as

$$A_{v'v''} = 21.419474\ (\Delta E_{v'v''})^3 (TDM_{v'v''})^2 \tag{10.2}$$

$\Delta E_{v'v''}$ is the energy difference (au) and $TDM_{v'v''}$ the transition dipole moment (au) of the transition.

For instance, for rotational states zero of the $^1\Sigma_g^+$ state and one of the $^1\Pi_u$ state:

```
Rotational quantum number for state 1:  0, for state 2:  1
----------------------------------------------------------------------------

Overlap matrix for vibrational wave functions for state number  1
1  1  .307535  2  1  .000000  2  2  .425936  3  1  .000000  3  2  .000000  3  3  .485199

Overlap matrix for vibrational wave functions for state number  2
1  1  .279631  2  1  .000000  2  2  .377566  3  1  .000000  3  2  .000000  3  3  .429572

Overlap matrix for state 1 and state 2 functions
```

```
    -.731192  -.617781  -.280533
     .547717  -.304345  -.650599
    -.342048   .502089  -.048727

  Transition moments over vibrational wave functions (atomic units)
    -.286286  -.236123  -.085294
     .218633  -.096088  -.240856
    -.125949   .183429   .005284

  Energy differences for vibrational wave functions(atomic units)
  1  1  .015897  2  1  .010246  2  2  .016427  3  1  .004758  3  2  .010939  3  3  .017108

  Contributions to inverse lifetimes (ns-1)
 No degeneracy factor is included in these values.
  1  1  .000007  2  1  .000001  2  2  .000001  3  1  .000000  3  2 .000001   3  3  .000000

  Lifetimes (in nano seconds)
    v       tau
    1 122090.44
    2  68160.26
    3  56017.08
```

Probably the most important caution when using the VIBROT program in diatomic molecules is that the number of vibrational states to compute and the accuracy obtained depends strongly on the computed surface. In the present case we compute all the curves to the dissociation limit. In other cases, the program will complain if we try to compute states which lie at energies above those obtained in the calculation of the curve.

10.1.3 A transition metal dimer: Ni_2

This section is a brief comment on a complex situation in a diatomic molecule such as Ni_2. Our purpose is to compute the ground state of this molecule. An explanation of how to calculate it accurately can be found in ref. [220]. However we will concentrate on computing the electronic states at the CASSCF level.

The nickel atom has two close low-lying configurations $3d^8 4s^2$ and $3d^9 4s^1$. The combination of two neutral Ni atoms leads to a Ni_2 dimer whose ground state has been somewhat controversial. For our purposes we commence with the assumption that it is one of the states derived from $3d^9 4s^1$ Ni atoms, with a single bond between the $4s$ orbitals, little $3d$ involvement, and the holes localized in the $3d\delta$ orbitals. Therefore, we compute the states resulting from two holes on δ orbitals: $\delta\delta$ states.

We shall not go through the procedure leading to the different electronic states that can arise from these electronic configurations, but refer to the Herzberg book on diatomic molecules [222] for details. In $D_{\infty h}$ we have three possible configurations with two holes, since the δ orbitals can be either *gerade* (g) or *ungerade* (u): $(\delta_g)^{-2}$, $(\delta_g)^{-1}(\delta_u)^{-1}$, or $(\delta_u)^{-2}$. The latter situation corresponds to nonequivalent electrons while the other two to equivalent electrons. Carrying through the analysis we obtain the following electronic states:

$(\delta_g)^{-2}$: ${}^1\Gamma_g$, ${}^3\Sigma_g^-$, ${}^1\Sigma_g^+$
$(\delta_u)^{-2}$: ${}^1\Gamma_g$, ${}^3\Sigma_g^-$, ${}^1\Sigma_g^+$
$(\delta_g)^{-1}(\delta_u)^{-1}$: ${}^3\Gamma_u$, ${}^1\Gamma_u$, ${}^3\Sigma_u^+$, ${}^3\Sigma_u^-$, ${}^1\Sigma_u^+$, ${}^1\Sigma_u^-$

In all there are thus 12 different electronic states.

Next, we need to classify these electronic states in the lower symmetry D_{2h}, in which MOLCAS works. This is done in Table 10.7, which relates the symmetry in $D_{\infty h}$ to that of D_{2h}. Since we have only Σ^+, Σ^-, and Γ states here, the D_{2h} symmetries will be only A_g, A_u, B_{1g}, and B_{1u}. The table above can now be rewritten in D_{2h}:

$(\delta_g)^{-2}$: $({}^1\mathrm{A}_g + {}^1\mathrm{B}_{1g})$, ${}^3\mathrm{B}_{1g}$, ${}^1\mathrm{A}_g$
$(\delta_u)^{-2}$: $({}^1\mathrm{A}_g + {}^1\mathrm{B}_{1g})$, ${}^3\mathrm{B}_{1g}$, ${}^1\mathrm{A}_g$
$(\delta_g)^{-1}(\delta_u)^{-1}$: $({}^3\mathrm{A}_u + {}^3\mathrm{B}_{1u})$, $({}^1\mathrm{A}_u + {}^1\mathrm{B}_{1u})$, ${}^3\mathrm{B}_{1u}$, ${}^3\mathrm{A}_u$, ${}^1\mathrm{B}_{1u}$, ${}^1\mathrm{A}_u$

or, if we rearrange the table after the D_{2h} symmetries:

${}^1\mathrm{A}_g$: ${}^1\Gamma_g(\delta_g)^{-2}$, ${}^1\Gamma_g(\delta_u)^{-2}$, ${}^1\Sigma_g^+(\delta_g)^{-2}$, ${}^1\Sigma_g^+(\delta_u)^{-2}$
${}^1\mathrm{B}_{1u}$: ${}^1\Gamma_u(\delta_g)^{-1}(\delta_u)^{-1}$, ${}^1\Sigma_u^+(\delta_g)^{-1}(\delta_u)^{-1}$
${}^1\mathrm{B}_{1g}$: ${}^1\Gamma_g(\delta_g)^{-2}$, ${}^1\Gamma_g(\delta_u)^{-2}$
${}^1\mathrm{A}_u$: ${}^1\Gamma_u(\delta_g)^{-1}(\delta_u)^{-1}$, ${}^1\Sigma_u^-(\delta_g)^{-1}(\delta_u)^{-1}$

${}^3\mathrm{B}_{1u}$: ${}^3\Gamma_u(\delta_g)^{-1}(\delta_u)^{-1}$, ${}^3\Sigma_u^+(\delta_g)^{-1}(\delta_u)^{-1}$
${}^3\mathrm{B}_{1g}$: ${}^3\Sigma_g^-(\delta_g)^{-2}$, ${}^3\Sigma_g^-(\delta_u)^{-2}$
${}^3\mathrm{A}_u$: ${}^3\Gamma_u(\delta_g)^{-1}(\delta_u)^{-1}$, ${}^3\Sigma_u^-(\delta_g)^{-1}(\delta_u)^{-1}$

It is not necessary to compute all the states because some of them (the Γ states) have degenerate components. It is both possible to make single state calculations looking for the lowest energy state of each symmetry or state-average calculations in each of the symmetries. The identification of the $D_{\infty h}$ states can be somewhat difficult. For instance, once we have computed one ${}^1\mathrm{A}_g$ state it can be a ${}^1\Gamma_g$ or a ${}^1\Sigma_g^+$ state. In this case the simplest solution is to compare the obtained energy to that of the ${}^1\Gamma_g$ degenerate component in B_{1g} symmetry, which must be equal to the energy of the ${}^1\Gamma_g$ state computed in A_g symmetry. Other situations can be more complicated and require a detailed analysis of the wave function.

It is important to have clean d-orbitals and the SUPSYM keyword may be needed to separate δ and σ (and γ if g-type functions are used in the basis set) orbitals in symmetry 1 (A_g). The AVERAGE keyword is not needed here because the π and ϕ orbitals have the same occupation for Σ and Γ states.

Finally, when states of different multiplicities are close in energy, the spin-orbit coupling which mix the different states should be included. The CASPT2 study of the Ni_2 molecule in reference [220], after considering all the mentioned effects determined that the ground state of the molecule is a 0_g^+ state, a mixture of the ${}^1\Sigma_g^+$ and ${}^3\Sigma_g^-$ electronic states. For a review of the spin-orbit coupling and other important coupling effects see reference [225].

10.1.4 High symmetry systems in Molcas

There are a large number of symmetry point groups in which MOLCAS cannot directly work. Although unusual in organic chemistry, some of them can be easily found in inorganic compounds. Systems belonging for instance to three-fold groups such as C_{3v}, D_{3h}, or D_{6h}, or to groups such O_h or D_{4h} must be computed using lower symmetry point groups. The consequence is, as in linear molecules, that orbitals and states belonging to different representations in the actual groups, belong to the same representation in the lower symmetry case, and *vice versa*. In the `RASSCF` program it is possible to prevent the orbital and configurational mixing caused by the first situation. The CLEANUP and SUPSYMMETRY keywords can be used in a careful, and somewhat tedious, way. The right symmetry behaviour of the RASSCF wave function is then assured. It is sometimes not a trivial task to identify the symmetry of the orbitals in the higher symmetry representation and which coefficients must vanish. In many situations the ground state wave function keeps the right symmetry (at least within the printing accuracy) and helps to identify the orbitals and coefficients. It is more frequent that the mixing happens for excited states.

The reverse situation, that is, that orbitals (normally degenerated) which belong to the same symmetry representation in the higher symmetry groups belong to different representations in the lower symmetry groups cannot be solved by the present implementation of the `RASSCF` program. The AVERage keyword, which performs this task in the linear molecules, is not prepared to do the same in non-linear systems. Provided that the symmetry problems mentioned in the previous paragraph are treated in the proper way and the trial orbitals have the right symmetry, the `RASSCF` code behaves properly.

There is a important final precaution concerning the high symmetry systems: the geometry of the molecule must be of the right symmetry. Any deviation will cause severe mixings. Figure 10.1 contains the `SEWARD` input for the magnesium porphirin molecule. This is a D_{4h} system which must be computed D_{2h} in Molcas.

For instance, the x and y coordinates of atoms C1 and C5 are interchanged with equal values in D_{4h} symmetry. Both atoms must appear in the `SEWARD` input because they are not independent by symmetry in the D_{2h} symmetry in which Molcas is going to work. Any deviation of the values, for instance to put the y coordinate to 0.681879 Å in C1 and the x to 0.681816 Å in C5 and similar deviations for the other coordinates, will lead to severe symmetry mixtures. This must be taken into account when geometry data are obtained from other program outputs or data bases.

Figure 10.1: Sample input of the SEWARD program for the magnesium porphirin molecule in the D_{2h} symmetry

```
 &SEWARD &END
Title
 Mg-Porphyrine D4h computed D2h
Symmetry
 X Y Z
Basis set
C.ANO-S...3s2p1d.
C1    4.254984     .681879     .000000  Angstrom
C2    2.873412    1.101185    0.000000  Angstrom
C3    2.426979    2.426979    0.000000  Angstrom
C4    1.101185    2.873412    0.000000  Angstrom
C5     .681879    4.254984    0.000000  Angstrom
End of basis
Basis set
N.ANO-S...3s2p1d.
N1    2.061400     .000000    0.000000  Angstrom
N2     .000000    2.061400    0.000000  Angstrom
End of basis
Basis set
H.ANO-S...2s0p.
H1    5.109145    1.348335    0.000000  Angstrom
H3    3.195605    3.195605    0.000000  Angstrom
H5    1.348335    5.109145    0.000000  Angstrom
End of basis
Basis set
Mg.ANO-S...4s3p1d.
Mg     .000000     .000000    0.000000  Angstrom
End of basis
End of Input
```

The situation can be more complex for some three-fold point groups such as D_{3h} or C_{3v}.

In these cases it is not possible to input in the exact cartesian geometry, which depends on trigonometric relations and relies on the numerical precision of the coordinates entry. It is necessary then to use in the SEWARD input as much precision as possible and check on the distance matrix of the SEWARD output if the symmetry of the system has been kept at least within the output printing criteria.

10.2 Geometry optimizations and Hessians.

To optimize a molecular geometry is probably one of the most frequent interests of a quantum chemist [226]. In the present section we examine some examples of obtaining stationary points on the energy surfaces. We will focus in this section in searching of minimal energy points, postponing the discussion on transition states to section 10.3. This type of calculations require the computation of molecular gradients, whether using analytical or numerical derivatives. We will also examine how to obtain the full geometrical Hessian for a molecular state, what will provide us with vibrational frequencies within the harmonic approximation and thermodynamic properties by the use of the proper partition functions.

The program ALASKA computes analytical gradients for optimized wave functions. In 8.2 the SCF, DFT, and CASSCF/RASSCF levels of calculation are available. The program ALASKA also computes numerical gradients from CASPT2 and MS-CASPT2 energies. Provided with the first order derivative matrix with respect to the nuclei and an approximate guess of the Hessian matrix, the program SLAPAF is then used to optimize molecular structures. From MOLCAS-5 it is not necessary to explicitly define the set of internal coordinates of the molecule in the SLAPAF input. Instead a redundant coordinates approach is used. If the definition is absent the program builds its own set of parameters based on curvature-weighted non-redundant internal coordinates and displacements [151]. As they depend on the symmetry of the system it might be somewhat difficult in some systems to define them. It is, therefore, strongly recommended to let the program define its own set of non-redundant internal coordinates. In certain situations such as bond dissociations the previous coordinates may not be appropriate and the code directs the user to use instead Cartesian coordinates, for instance.

10.2.1 Ground state optimizations and vibrational analysis

As an example we are going to work with the 1,3-cyclopentadiene molecule. This is a five-carbon system forming a ring which has two conjugated double bonds. Each carbon has one attached hydrogen atom except one which has two. We will use the CASSCF method and take advantage of the symmetry properties of the molecule to compute ground and excited states. To ensure the convergence of the results we will also perform Hessian calculations to compute the force fields at the optimized geometries.

In this section we will combine two types of procedures to perform calculations in MOLCAS. The user may then choose the most convenient for her/his taste. We can use an general script and perform an input-oriented calculation, when all the information relative to the calculation, including links for the files and control of iterations, are inserted in the input file. The other procedure is the classical script-oriented system used in previous examples and typically previous versions of MOLCAS. Let's start by making an input-oriented optimization.

A script is still needed to perform the basic definitions, although they can be mostly done within the input file. A suggested form for this general script could be:

```
#!/bin/sh
export MOLCAS=/home/molcas/molcashome
export MOLCAS_MEM=64
export Project=Cyclopentadiene1
export HomeDir=/home/somebody/somewhere
export WorkDir=$HomeDir/$Project
[ ! -d $WorkDir ] && mkdir $WorkDir
molcas $HomeDir/$Project.input >$HomeDir/$Project.out 2>$HomeDir/$Project.err
exit
```

We begin by defining the input for the initial calculation. In simple cases the optimization procedure is very efficient. We are going, however, to design a more complete procedure that may help in more complex situations. It is sometimes useful to start the optimization in a small size basis set and use the obtained approximate Hessian to continue the calculation with larger basis sets. Therefore, we will begin by using the minimal STO-3G basis set to optimize the ground state of 1,3-cyclopentadiene within C_{2v} symmetry.

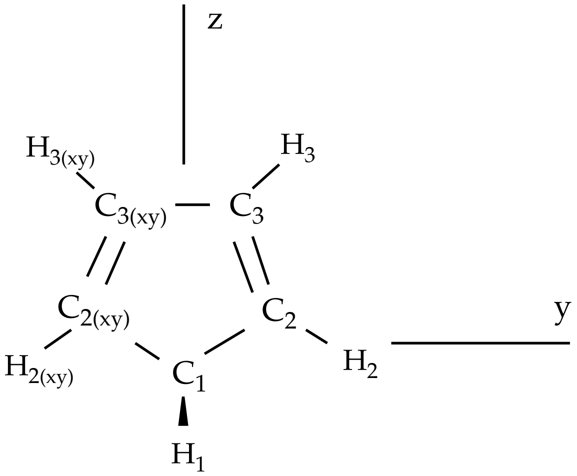


Figure 10.2: 1,3-cyclopentadiene

```
>>> EXPORT MOLCAS_MAXITER=50
 &GATEWAY; Title=1,3,-cyclopentadiene. STO-3G basis set.
   Symmetry= X XY
   Basis set
   C.STO-3G....
   C1    0.000000  0.000000  0.000000  Bohr
   C2    0.000000  2.222644  1.774314  Bohr
   C3    0.000000  1.384460  4.167793  Bohr
   End of basis
   Basis set
   H.STO-3G....
   H1    1.662033  0.000000 -1.245623  Bohr
   H2    0.000000  4.167844  1.149778  Bohr
   H3    0.000000  2.548637  5.849078  Bohr
   End of basis
>>> Do while <<<
 &SEWARD
>>> IF ( ITER = 1 ) THEN
 &SCF
    TITLE= cyclopentadiene molecule
    OCCUPIED=9 1 6 2
    ITERATIONS=40
>>> END IF
 &RASSCF
   TITLE=cyclopentadiene molecule 1A1
   SYMMETRY=1; SPIN=1
   NACTEL= 6    0    0
   INACTIVE= 9    0    6    0
   RAS2= 0    2    0    3            <--- All pi valence orbitals active
   ITER= 50,25; CIMX= 25
 &ALASKA
 &SLAPAF; Iterations=80; Thrs=0.5D-06 1.0D-03
>>> EndDo <<<
>>> COPY $Project.RunFile $CurrDir/$Project.ForceConstant.STO-3G
```

A copy of the ***RUNFILE*** has been made at the end of the input stream. This saves the file for use as (a) starting geometry and (b) a guess of the Hessian matrix in the following calculation. The link can be also done in the shell script.

The generators used to define the C_{2v} symmetry are X and XY, plane yz and axis z. They differ from those used in other examples as in section 10.1.1. The only consequence is that the order of the symmetries in SEWARD differs. In the present case the order is: a_1, a_2, b_1, and b_2, and consequently the classification by symmetries of the orbitals in the SCF and RASSCF inputs will differ. It is therefore recommended to initially use the option TEST in the GATEWAY input to check the symmetry option. This option, however, will stop the calculation after the GATEWAY input head is printed.

The calculation converges in four steps. We change now the input. We can choose between replacing by hand the geometry of the SEWARD input or use the same $WorkDir directory and let the program to take the last geometry stored into the ***RUNFILE*** file. In any case the new input can be:

```
>>COPY  $CurrDir/OPT.hessian.ForceConstant.STO-3G $Project.RunOld
 &GATEWAY; Title=1,3,-cyclopentadiene molecule
Symmetry=X XY
Basis set
C.ANO-L...4s3p1d.
C1              .0000000000         .0000000000       -2.3726116671
C2              .0000000000        2.2447443782        -.5623842095
C3              .0000000000        1.4008186026        1.8537195887
End of basis
Basis set
H.ANO-L...2s.
H1             1.6523486260         .0000000000       -3.6022531906
H2              .0000000000        4.1872267035       -1.1903003793
H3              .0000000000        2.5490335048        3.5419847446
End of basis
>>> Do while <<<
 &SEWARD
>>> IF ( ITER = 1 ) <<<<
 &SCF
   TITLE=cyclopentadiene molecule
   OCCUPIED= 9 1 6 2
   ITERATIONS= 40
>>> ENDIF <<<
 &RASSCF; TITLE cyclopentadiene molecule 1A1
   SYMMETRY=1; SPIN=1; NACTEL=6    0    0
   INACTIVE= 9    0    6    0
   RAS2    = 0    2    0    3
   ITER=50,25; CIMX= 25
 &SLAPAF; Iterations=80; Thrs=0.5D-06 1.0D-03
   OldForce Constant Matrix
 >>> EndDo <<<
```

The *RUNOLD* file will be used by SEWARD to pick up the molecular structure on the initial iteration and by SLAPAF as initial Hessian to carry out the relaxation. This use of the *RUNFILE* can be done between any different calculations provided they work in the same symmetry.

In the new basis set, the resulting optimized geometry at the CASSCF level in C_{2v} symmetry is:

```
  ******************************************
  * Values of internal coordinates         *
  ******************************************
  C2C1   2.851490 Bohr
  C3C2   2.545737 Bohr
  C3C3   2.790329 Bohr
  H1C1   2.064352 Bohr
  H2C2   2.031679 Bohr
  H3C3   2.032530 Bohr
  C1C2C3     109.71 Degrees
  C1C2H2     123.72 Degrees
  C2C3H3     126.36 Degrees
  H1C1H1     107.05 Degrees
```

Once we have the optimized geometry we can obtain the force field, to compute the force constant matrix and obtain an analysis of the harmonic frequency. This is done by computing the analytical Hessian at the optimized geometry. Notice that this is a single-shot calculation using the MCKINLEY, which will automatically start the MCLR module in case of a frequency calculation.

```
  &GATEWAY; Title=1,3,-cyclopentadiene molecule
    Symmetry= X XY
    Basis set
    C.ANO-L...4s3p1d.
      C1           0.0000000000        0.0000000000       -2.3483061484
      C2           0.0000000000        2.2245383122       -0.5643712787
      C3           0.0000000000        1.3951643642        1.8424767578
    End of basis
    Basis set
    H.ANO-L...2s.
      H1           1.6599988023        0.0000000000       -3.5754797471
      H2           0.0000000000        4.1615845660       -1.1772096132
      H3           0.0000000000        2.5501642966        3.5149458446
    End of basis
  &SEWARD
  &SCF; TITLE=cyclopentadiene molecule
    OCCUPIED= 9 1 6 2
    ITERATIONS= 40
  &RASSCF; TITLE=cyclopentadiene molecule 1A1
    SYMMETRY=1; SPIN=1; NACTEL= 6    0    0
    INACTIVE= 9    0    6    0
    RAS2    = 0    2    0    3
    ITER= 50,25; CIMX=25
  &MCKINLEY
```

Cyclopentadiene has 11 atoms, that mean 3N = 33 Cartesian degrees of freedom. Therefore the MCLR output will contain 33 frequencies. From those, we are just interested in the 3N-6 = 27 final degrees of freedom that correspond to the normal modes of the system. We will discard from the output the three translational (T_i) and three rotational (R_i) coordinates.

The table of characters gives us the classification of these six coordinates: a$_1$ (T_z), a$_2$ (R_z), b$_2$ (T_x,R_y), b$_1$ (T_y,R_x). This information is found in the Seward output:

```
Character Table for C2v
         E   s(yz) C2(z) s(xz)
a1       1     1     1     1  z
a2       1    -1     1    -1  xy, Rz, I
b2       1     1    -1    -1  y, yz, Rx
b1       1    -1    -1     1  x, xz, Ry
```

It is simply to distinguish these frequencies because they must be zero, although and because of numerical inaccuracies they will be simply close to zero. Note that the associated intensities are nonsense. In the present calculation the harmonic frequencies, the infrared intensities, and the corresponding normal modes printed below in Cartesian coordinates are the following:

```
   Symmetry a1
   ==============
              1          2          3          4          5          6
     Freq.      0.04     847.85     966.08    1044.69    1187.61    1492.42
     Intensity:   0.646E-08 0.125E-02 0.532E+01 0.416E+00 0.639E-01 0.393E+01
     C1         z    0.30151    0.35189   -0.21166   -0.11594    0.06874    0.03291
     C2         y    0.00000    0.31310    0.14169    0.12527   -0.01998   -0.08028
     C2         z    0.30151   -0.02858    0.06838   -0.00260    0.02502   -0.06133
     C3         y   -0.00000    0.04392   -0.07031    0.23891   -0.02473    0.16107
     C3         z    0.30151   -0.15907    0.00312    0.08851   -0.07733   -0.03146
     H1         x    0.00000   -0.02843   -0.00113   -0.01161    0.00294    0.04942
     H1         z    0.30151    0.31164   -0.21378   -0.13696    0.08233    0.11717
     H2         y    0.00000    0.24416    0.27642    0.12400    0.11727    0.07948
     H2         z    0.30151   -0.25054    0.46616   -0.05986    0.47744    0.46022
     H3         y   -0.00000   -0.29253   -0.28984    0.59698    0.34878   -0.34364
     H3         z    0.30151    0.07820    0.15644   -0.13576   -0.34625    0.33157
              7          8          9         10         11
     Freq.    1579.76    1633.36    3140.69    3315.46    3341.28
     Intensity:   0.474E+01 0.432E+00 0.255E+02 0.143E+02 0.572E+01
...
    Symmetry a2
   ==============
                     1          2          3          4          5
            Freq.      i9.26     492.62     663.74     872.47    1235.06
...
    Symmetry b2
   ==============
                  1          2          3          4          5          6
        Freq.     i10.61       0.04     858.72    1020.51    1173.33    1386.20
     Intensity:   0.249E-01 0.215E-07 0.259E+01 0.743E+01 0.629E-01 0.162E+00
...
                     7          8          9         10
             Freq.    1424.11    1699.07    3305.26    3334.09
       Intensity:   0.966E+00 0.426E+00 0.150E+00 0.302E+02
...
    Symmetry b1
   ==============
                     1          2          3          4          5          6
            Freq.     i11.31       0.11     349.15     662.98     881.19     980.54
       Intensity:   0.459E-01 0.202E-06 0.505E+01 0.896E+02 0.302E+00 0.169E+02
...
                     7
            Freq.    3159.81
      Intensity:   0.149E+02
...
```

Apart from the six mentioned translational and rotational coordinates There are no imaginary frequencies and therefore the geometry corresponds to a stationary point within the C_{2v} symmetry. The frequencies are expressed in reciprocal centimeters.

After the vibrational analysis the zero-point energy correction and the thermal corrections to the total energy, internal, entropy, and Gibbs free energy. The analysis uses the standard expressions for an ideal gas in the canonical ensemble which can be found in any standard statistical mechanics book. The analysis is performed at different temperatures, for instance:

```
 *****************************************************
  Temperature =   273.00 Kelvin, Pressure =   1.00 atm
  -----------------------------------------------------
  Molecular Partition Function and Molar Entropy:
                         q/V (M**-3)    S(kcal/mol*K)
  Electronic            0.100000D+01        0.000
  Translational         0.143889D+29       38.044
  Rotational            0.441593D+05       24.235
  Vibrational           0.111128D-47        3.002
  TOTAL                 0.706112D-15       65.281
  Thermal contributions to INTERNAL ENERGY:
  Electronic           0.000 kcal/mol      0.000000 au.
  Translational        0.814 kcal/mol      0.001297 au.
  Rotational           0.814 kcal/mol      0.001297 au.
  Vibrational         60.723 kcal/mol      0.096768 au.
  TOTAL               62.350 kcal/mol      0.099361 au.
  Thermal contributions to
  ENTHALPY            62.893 kcal/mol      0.100226 au.
  GIBBS FREE ENERGY   45.071 kcal/mol      0.071825 au.
  Sum of energy and thermal contributions
  INTERNAL ENERGY                       -192.786695 au.
  ENTHALPY                              -192.785831 au.
  GIBBS FREE ENERGY                     -192.814232 au.
```

Next, polarizabilities (see below) and isotope shifted frequencies are also displayed in the output.

```
  ************************************
  *                                  *
  *       Polarizabilities           *
  *                                  *
  ************************************
    34.76247619
    -0.00000000 51.86439359
    -0.00000000 -0.00000000 57.75391824
```

For a graphical representation of the harmonic frequencies one can also use the $Project.freq.molden file as an input to the MOLDEN program.

10.2.2 Excited state optimizations

The calculation of excited states using the `ALASKA` and `SLAPAF` codes has no special characteristic. The wave function is defined by the `SCF` or `RASSCF` programs. Therefore if we want to optimize an excited state the `RASSCF` input has to be defined accordingly. It is not, however, an easy task, normally because the excited states have lower symmetry than the ground state and one has to work in low order symmetries if the full optimization is pursued.

Take the example of the thiophene molecule (see fig. 10.10 in next section). The ground state has C_{2v} symmetry: $1\ ^1A_1$. The two lowest valence excited states are 2^1A_1 and 1^1B_2. If we optimize the geometries within the C_{2v} symmetry the calculations converge easily for the three states. They are the first, second, and first roots of their symmetry, respectively. But if we want to make a full optimization in C_1, or even a restricted one in C_s, all three states belong to the same symmetry representation. The higher the root more difficult is to converge it. A geometry optimization requires single-root optimized CASSCF wave-functions, but, unlike in previous MOLCAS versions, we can now carry out State-Average (SA) CASSCF calculations between different roots. The wave functions we have with this procedure are based on an averaged density matrix, and a further orbital relaxation is required. The MCLR program can perform such a task by means of a perturbational approach. Therefore, if we choose to carry out a SA-CASSCF calculations in the optimization procedure, the ALASKA module will automatically start up the MCLR module.

We are going to optimize the three states of thiophene in C_{2v} symmetry. The inputs are:

```
 &GATEWAY; Title=Thiophene molecule
   Symmetry= X XY
   Basis set
   S.ANO-S...4s3p2d.
   S1              .0000000000        .0000000000      -2.1793919255
   End of basis
   Basis set
   C.ANO-S...3s2p1d.
   C1              .0000000000       2.3420838459        .1014908659
   C2              .0000000000       1.3629012233       2.4874875281
   End of basis
   Basis set
   H.ANO-S...2s.
   H1              .0000000000       4.3076765963       -.4350463731
   H2              .0000000000       2.5065969281       4.1778544652
   End of basis
>>> Do while <<<
 &SEWARD
>>> IF ( ITER = 1 ) <<<
 &SCF; TITLE=Thiophene molecule
   OCCUPIED= 11 1 7 3
   ITERATIONS= 40
>>> ENDIF <<<
 &RASSCF; TITLE=Thiophene molecule 1 1A1
   SYMMETRY=1; SPIN=1; NACTEL= 6    0    0
   INACTIVE= 11    0    7    1
   RAS2     =  0    2    0    3
   ITER= 50,25
   &ALASKA
   &SLAPAF
End of Input
 >>> ENDDO <<<
```

for the ground state. For the two excited states we will replace the RASSCF inputs with

```
 &RASSCF; TITLE=Thiophene molecule 2 1A1
   SYMMETRY=1; SPIN=1; NACTEL= 6    0    0
   INACTIVE= 11    0    7    1
   RAS2     =  0    2    0    3
   ITER= 50,25
   CIROOT= 2 2; 1 2; 1 1
   LEVSHFT=1.0
   RLXRoot= 2
```

for the $2^1\mathrm{A}_1$ state. Notice that we are doing a SA-CASSCF calculation including two roots, therefore we must use the keyword RLXROOT within the RASSCF input to specify for which state we want the root. We have also

```
 &RASSCF; TITLE=Thiophene molecule 1 1B2
   SYMMETRY=2; SPIN=1; NACTEL= 6    0    0
   INACTIVE= 11    0    7    1
   RAS2    =  0    2    0    3
   ITER= 50,25
   LEVSHFT=1.0
```

for the 1^1B_2 state.

To help the program to converge we can include one or more initial RASSCF inputs in the input file. The following is an example for the calculation of the of the $3^1A'$ state of thiophene (C_s symmetry) with a previous calculation of the ground state to have better starting orbitals.

```
 &GATEWAY; Title= Thiophene molecule
   Symmetry=X
   Basis set
   S.ANO-S...4s3p2d.
   S1              .0000000000         .0000000000       -2.1174458547
   End of basis
   Basis set
   C.ANO-S...3s2p1d.
   C1              .0000000000        2.4102089951         .1119410701
   C1b             .0000000000       -2.4102089951         .1119410701
   C2              .0000000000        1.3751924147        2.7088559532
   C2b             .0000000000       -1.3751924147        2.7088559532
   End of basis
   Basis set
   H.ANO-S...2s.
   H1              .0000000000        4.3643321746        -.4429940876
   H1b             .0000000000       -4.3643321746        -.4429940876
   H2              .0000000000        2.5331491787        4.3818833166
   H2b             .0000000000       -2.5331491787        4.3818833166
   End of basis
>>> Do while <<<
 &SEWARD
>>> IF ( ITER = 1 ) <<<
 &SCF; TITLE= Thiophene molecule
   OCCUPIED= 18 4
   ITERATIONS = 40
 &RASSCF; TITLE= Thiophene molecule 1A'
   SYMMETRY=1; SPIN=1; NACTEL= 6    0    0
   INACTIVE= 18    1
   RAS2    =  0    5
   ITER= 50,25
>>> ENDIF <<<
 &RASSCF; TITLE= Thiophene molecule 3 1A'
   SYMMETRY=1; SPIN=1; NACTEL= 6    0    0
   INACTIVE= 18    1
   RAS2    =  0    5
   ITER= 50,25
   CIROOT=3 3 1
   RLXRoot= 3
 &ALASKA
 &SLAPAF &END
>>> ENDDO <<<
```

It should be remembered that geometry optimizations for excited states are difficult. Not only can it be difficult to converge the corresponding RASSCF calculation, but we must also be sure that the order of the states does not change during the optimization of the geometry. This is not uncommon and the optimization must be followed by the user.

10.2.3 Restrictions in symmetry or geometry.

Optimizing with geometrical constraints.

A common situation in geometry optimizations is to have one or several coordinates fixed or constrained and vary the remaining coordinates. As an example we will take the biphenyl molecule, two benzene moieties bridged by a single bond. The ground state of the molecule is not planar. One benzene group is twisted by 44^o degrees with respect to the other [227]. We can use this example to perform two types of restricted optimizations. The simplest way to introduce constraints is to give a coordinate a fixed value and let the other coordinates to be optimized. For instance, let's fix the dihedral angle between both benzenes to be fixed to 44^o degrees. Within this restriction, the remaining coordinates will be fully optimized. The CONSTRAINTS keyword in the program GATEWAY will take care of the restriction (note this keyword could also be placed in the program SLAPAF). The input could be:

```
 &GATEWAY; Title= Biphenyl twisted D2
   Symmetry= XY XZ
   Basis set
   C.ANO-S...3s2p1d.
   C1             1.4097582886         .0000000000         .0000000000
   C2             2.7703009377        2.1131321616         .8552434921
   C3             5.4130377085        2.1172148045         .8532344474
   C4             6.7468359904         .0000000000         .0000000000
   End of basis
   Basis set
   H.ANO-S...2s.
   H2             1.7692261798        3.7578798540        1.5134152112
   H3             6.4188773347        3.7589592975        1.5142479153
   H4             8.7821560635         .0000000000         .0000000000
   End of basis
   Constraints
      d1 = Dihedral C2 C1 C1(XY) C2(XY)
   Values
      d1 = -44.4 degrees
   End of Constraints
>>> Do while <<<
 &SEWARD
>>> IF ( ITER = 1 ) <<<
 &SCF; TITLE= Biphenyl twisted D2
   OCCUPIED= 12 9 9 11
   ITERATIONS= 50
>>> ENDIF <<<
 &RASSCF; TITLE= Biphenyl twisted D2
   SYMMETRY=1; SPIN=1; NACTEL= 12    0    0
   INACTIVE= 11    7    7   10
   RAS2    =  2    4    4    2
 &ALASKA
 &SLAPAF; Iterations=30; MaxStep=1.0
>>> ENDDO <<<
```

One important consideration about the constraint. You do not need to start at a geometry having the exact value for the coordinate you have selected (44.4 degrees for the dihedral angle here). The optimization will lead you to the right solution. On the other hand, if you start exactly with the dihedral being 44.4 deg the code does not necessarily will freeze this value in the first iterations, but will converge to it at the end. Therefore, it may happen that the value for the dihedral differs from the selected value in the initial iterations. You can follow the optimization steps in the $WorkDir directory using the MOLDEN files generated automatically by MOLCAS.

Now we will perform the opposite optimization: we want to optimize the dihedral angle relating both benzene units but keep all the other coordinates fixed. We could well use the same procedure as before adding constraints for all the remaining coordinates different from the interesting dihedral angle, but to build the input would be tedious. Therefore, instead of keyword CONSTRAINTS we will make use of the keywords VARY and FIX.

The input file should be:

```
  &GATEWAY; Title= Biphenyl twisted D2
   Symmetry=XY XZ
   Basis set
   C.ANO-S...3s2p1d.
   C1             1.4097582886         .0000000000         .0000000000
   C2             2.7703009377        2.1131321616         .8552434921
   C3             5.4130377085        2.1172148045         .8532344474
   C4             6.7468359904         .0000000000         .0000000000
   End of basis
   Basis set
   H.ANO-S...2s.
   H2             1.7692261798        3.7578798540        1.5134152112
   H3             6.4188773347        3.7589592975        1.5142479153
   H4             8.7821560635         .0000000000         .0000000000
   End of basis
>>> Do while <<<
 &SEWARD
>>> IF ( ITER = 1 ) <<<
 &SCF; TITLE= Biphenyl twisted D2
   OCCUPIED= 12 9 9 11
   ITERATIONS= 50
>>> ENDIF <<<
 &RASSCF; TITLE= Biphenyl twisted D2
   SYMMETRY=1; SPIN=1; NACTEL=12    0    0
   INACTIVE= 11    7    7   10
   RAS2    =  2    4    4    2
 &ALASKA
 &SLAPAF
Internal coordinates
b1 = Bond C1 C1(XY)
b2 = Bond C1 C2
b3 = Bond C2 C3
b4 = Bond C3 C4
h1 = Bond C2 H2
h2 = Bond C3 H3
h3 = Bond C4 H4
a1 = Angle C2 C1 C1(XY)
a2 = Angle C1 C2 C3
a3 = Angle C1 C2 H2
a4 = Angle C2 C3 H3
phi = Dihedral C2 C1 C1(XY) C2(XY)
d1 = Dihedral H2 C2 C1 C1(XY)
d2 = OutOfP C3 C1(XY) C1 C2
d3 = Dihedral H3 C3 C2 H2
Vary; phi
Fix; b1; b2; b3; b4; h1; h2; h3; a1; a2; a3; a4; d1; d2; d3
End of Internal
Iterations= 30
>>> ENDDO <<<
```

To be able to optimize the molecule in that way a D_2 symmetry has to be used. In the definition of the internal coordinates we can use an out-of-plane coordinate: C2 C2(xy) C1(xy) C1 or a dihedral angle C2 C1 C1(xy) C2(xy). In this case there is no major problem but in general one has to avoid as much as possible to define dihedral angles close to 180^o (trans conformation). The SLAPAF program will warn about this problem if necessary. In the present example, angle 'phi' is the angle to vary while the remaining coordinates are frozen. All this is only a problem in the user-defined internal approach, not in the non-redundant internal approach used by default in the program. In case we do not have the coordinates from a previous calculation we can always run a simple calculation with one iteration in the `SLAPAF` program.

It is not unusual to have problems in the relaxation step when one defines internal coordinates. Once the program has found that the definition is consistent with the molecule and the symmetry, it can happen that the selected coordinates are not the best choice to carry out the optimization, that the variation of some of the coordinates is too large or maybe some of the angles are close to their limiting values ($\pm 180^o$ for Dihedral angles and $\pm 90^o$ for Out of Plane angles). The SLAPAF program will inform about these problems. Most of the situations are solved by re-defining the coordinates, changing the basis set or the geometry if possible, or even freezing some of the coordinates. One easy solution is to froze this particular coordinate and optimize, at least partially, the other as an initial step to a full optimization. It can be recommended to change the definition of the coordinates from internal to Cartesian.

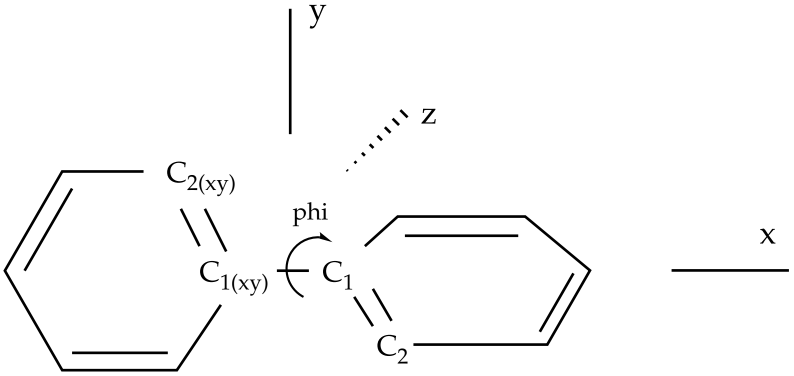


Figure 10.3: Twisted biphenyl molecule

Optimizing with symmetry restrictions.

Presently, MOLCAS is prepared to work in the point groups C_1, C_i, C_s, C_2, D_2, C_{2h}, C_{2v}, and D_{2h}. To have the wave functions or geometries in other symmetries we have to restrict orbital rotations or geometry relaxations specifically. We have shown how to in the RASSCF program by using the SUPSYM option. In a geometry optimization we may also want to restrict the geometry of the molecule to other symmetries. For instance, to optimize the benzene molecule which belongs to the D_{6h} point group we have to generate the integrals and wave function in D_{2h} symmetry, the highest group available, and then make the appropriate combinations of the coordinates chosen for the relaxation in the SLAPAF program, as is shown in the manual.

As an example we will take the ammonia molecule, NH_3. There is a planar transition state along the isomerization barrier between two pyramidal structures. We want to optimize the planar structure restricted to the D_{3h} point group. However, the electronic wave function will be computed in C_s symmetry (C_{2v} is also possible) and will not be restricted, although it is possible to do that in the `RASSCF` program.

The input for such a geometry optimization is:

```
 &GATEWAY; Title= NH3, planar
   Symmetry= Z
   Basis Set
   N.ANO-L...4s3p2d.
   N              .0000000000         .0000000000         .0000000000
   End of Basis
   Basis set
   H.ANO-L...3s2p.
   H1            1.9520879910         .0000000000         .0000000000
   H2            -.9760439955        1.6905577906         .0000000000
   H3            -.9760439955       -1.6905577906         .0000000000
   End of Basis
>>> Do while <<<
 &SEWARD
>>> IF ( ITER = 1 ) <<<
 &SCF; Title= NH3, planar
   Occupied= 4 1
   Iterations= 40
>>> ENDIF <<<
 &RASSCF; Title= NH3, planar
   Symmetry=1; Spin=1; Nactel=8  0  0
   INACTIVE=1 0
   RAS2    =6 2
 &ALASKA
 &SLAPAF
Internal coordinates
b1 = Bond N H1
b2 = Bond N H2
b3 = Bond N H3
a1 = Angle H1 N H2
a2 = Angle H1 N H3
Vary
  r1 = 1.0 b1 + 1.0 b2 + 1.0 b3
Fix
  r2 = 1.0 b1 - 1.0 b2
  r3 = 1.0 b1 - 1.0 b3
  a1 = 1.0 a1
  a2 = 1.0 a2
End of internal
>>> ENDDO <<<
```

All four atoms are in the same plane. Working in C_s, planar ammonia has five degrees of freedom. Therefore we must define five independent internal coordinates, in this case the three N-H bonds and two of the three angles H-N-H. The other is already defined knowing the two other angles. Now we must define the varying coordinates. The bond lengths will be optimized, but all three N-H distances must be equal. First we define (see definition in the previous input) coordinate $r1$ equal to the sum of all three bonds; then, we define coordinates $r2$ and $r3$ and keep them fixed. $r2$ will ensure that $bond1$ is equal to $bond2$ and $r3$ will assure that $bond3$ is equal to $bond1$. $r2$ and $r3$ will have a zero value. In this way all three bonds will have the same length. As we want the system constrained into the D_{3h} point group, the three angles must be equal with a value of 120 degrees. This is their initial value, therefore we simply keep coordinates $ang1$ and $ang2$ fixed. The result is a D_{3h} structure:

```
                 *******************************************
                 *    InterNuclear Distances / Angstrom    *
                 *******************************************
            1 N            2 H1           3 H2           4 H3
    1 N     0.000000
    2 H1    1.003163       0.000000
    3 H2    1.003163       1.737529       0.000000
    4 H3    1.003163       1.737529       1.737529       0.000000
                 **************************************
```

```
*    Valence Bond Angles / Degree    *
**************************************
      Atom centers                Phi
  2 H1       1 N        3 H2       120.00
  2 H1       1 N        4 H3       120.00
  3 H2       1 N        4 H3       120.00
```

In a simple case like this an optimization without restrictions would also end up in the same symmetry as the initial input.

10.2.4 Optimizing with Z-Matrix.

An alternative way to optimize a structure with geometrical and/or symmetrical constraints is to combine the Z-Matrix definition of the molecular structure used for the program SEWARD with a coherent definition for the INTERNAL COORDINATED used in the optimization by program SLAPAF.

Here is an examples of optimization of the methyl carbanion. Note that the wavefunction is calculated within the C_s symmetry but the geometry is optimized within the C_{3v} symmetry through the ZMAT and the INTERNAL COORDINATES definitions. Note that XBAS precedes ZMAT.

```
  &Gateway
    Symmetry=Y
    XBAS=Aug-cc-pVDZ
    ZMAT
    C1
    X2   1  1.00
    H3   1  1.09   2 105.
    H4   1  1.09   2 105.    3  120.
 >>>  export MOLCAS_MAXITER=500
 >>>  Do  While  <<<
  &SEWARD
  &SCF; Charge= -1
  &ALASKA
  &SLAPAF
    Internal Coordinates
     CX2  = Bond C1 X2
     CH3  = Bond C1 H3
     CH4  = Bond C1 H4
     XCH3 = Angle X2 C1 H3
     XCH4 = Angle X2 C1 H4
     DH4  = Dihedral H3 X2 C1 H4
    Vary
     SumCH34  = 1. CH3  +2. CH4
     SumXCH34 = 1. XCH3 +2. XCH4
    Fix
     rCX2  = 1.0 CX2
     DifCH34  = 2. CH3  -1. CH4
     DifXCH34 = 2. XCH3 -1. XCH4
     dDH4  = 1.0 DH4
    End of Internal
    PRFC
    Iterations= 10
 >>>  EndDo  <<<
```

Note that the *dummy* atom X2 is used to define the Z axis and the planar angles for the hydrogen atoms. The linear combinations of bond distances and planar angles in the expression in the VARY and FIX sections are used to impose the C_{3v} symmetry.

Another examples where the wavefunction and the geometry can be calculated within different symmetry groups is benzene. In this case, the former uses D_{2h} symmetry and the latter D_{6h} symmetry. Two special atoms are used: the *dummy* X1 atom defines the center of the molecule while the *ghost* Z2 atom is used to define the C_6 rotational axis (and the Z axis).

```
 &GATEWAY
   Symmetry=X Y Z
   XBAS
   H.ANO-S...2s.
   C.ANO-S...3s2p.
   End of basis
   ZMAT
   X1
   Z2   1  1.00
   C3   1  1.3915   2  90.
   C4   1  1.3915   2  90.    3  60.
   H5   1  2.4715   2  90.    3   0.
   H6   1  2.4715   2  90.    3  60.
>>>  export MOLCAS_MAXITER=500
>>>  Do  While  <<<
 &SEWARD; &SCF ; &ALASKA
 &SLAPAF
   Internal Coordinates
      XC3 = Bond X1 C3
      XC4 = Bond X1 C4
      XH5 = Bond X1 H5
      XH6 = Bond X1 H6
      CXC = Angle C3 X1 C4
      HXH = Angle H5 X1 H6
   Vary
      SumC = 1.0 XC3 + 2.0 XC4
      SumH = 1.0 XH5 + 2.0 XH6
   Fix
      DifC = 2.0 XC3 - 1.0 XC4
      DifH = 2.0 XH5 - 1.0 XH6
      aCXC = 1.0 CXC
      aHXH = 1.0 HXH
   End of Internal
   PRFC
>>> EndDo <<<
```

Note that the *ghost* atom Z2 is used to define the geometry within the Z-Matrix but it does not appear in the INTERNAL COORDINATES section. On the other hand, the *dummy* atom X1 represents the center of the molecule and it is used in the INTERNAL COORDINATES section.

10.2.5 CASPT2 optimizations

For systems showing a clear multiconfigurational nature, the CASSCF treatment on top of the HF results is of crucial importance in order to recover the large non dynamical correlation effects. On the other hand, ground-state geometry optimizations of closed shell systems are not exempt from non dynamical correlation effects. In general, molecules containing π-electrons suffer from significant effects of non dynamical correlation, even more in presence of conjugated groups. Several studies on systems with delocalized bonds have shown the effectiveness of the CASSCF approach in reproducing the main geometrical parameters with high accuracy [228, 229, 230].

However, pronounced effects of dynamical correlation often occur in systems with π-electrons, especially in combination with polarized bonds. An example is given by the C=O bond length, which is known to be very sensitive to an accurate description of the dynamical correlation effects [231]. We will show now that the inherent limitations of the CASSCF method can be successfully overcome by employing a CASPT2 geometry optimization, which uses a numerical gradient procedure of recent implementation. A suitable molecule for this investigation is acrolein. As many other conjugated aldehydes and ketones, offers an example of *s-cis/s-trans* isomerism (Figure 10.4). Due to the resonance between various structures involving π electrons, the bond order for the C-C bond is higher than the one for a non-conjugated C-C single bond. This partial double-bond character restricts the rotation about such a bond, giving rise to the possibility of geometrical isomerism, analogue to the *cis–trans* one observed for conventional double bonds.

A `CASPT2` geometry optimization can be performed in MOLCAS. A possible input for the CASPT2 geometry optimization of the *s-trans* isomer is displayed below. The procedure is invoking the resolution-of-identity approximation using the keyword RICD. This option will speed up the calculation, something which makes sense since we will compute the gradients numerically.

```
>>> Export MOLCAS_MAXITER=500
&GATEWAY
   Title= Acrolein Cs symmetry - transoid
   Coord
      8
    O      0.0000000    -1.4511781     -1.3744831
    C      0.0000000    -0.8224882     -0.1546649
    C      0.0000000     0.7589531     -0.0387200
    C      0.0000000     1.3465057      1.2841925
    H      0.0000000    -1.4247539      0.8878671
    H      0.0000000     1.3958142     -1.0393956
    H      0.0000000     0.6274048      2.2298215
    H      0.0000000     2.5287634      1.4123985
    Group=X
    Basis=ANO-RCC-VDZP
    RICD
>>>>>>>>>>>>> Do while <<<<<<<<<<<<
&SEWARD
>>>>>>>> IF ( ITER = 1 ) <<<<<<<<<<<
&SCF; Title= Acrolein Cs symmetry
*The symmetry species are a’  a’’
Occupied= 13 2
>>>>>>> ENDIF <<<<<<<<<<<<<<<<<<<<<<
&RASSCF; Title=Acrolein ground state
   nActEl= 4 0 0
   Inactive= 13 0
*  The symmetry species are a’  a’’
   Ras2= 0 4
&CASPT2
&SLAPAF
>>>>>>>>>>>>> ENDDO  <<<<<<<<<<<<<<
```

Experimental investigations assign a planar structure for both the isomers. We can take advantage of this result and use a C_s symmetry throughout the optimization procedure. Moreover, the choice of the active space is suggested by previous calculations on analogous systems. The active space contains 4 π MOs /4 π electrons, thus what we will call shortly a π-CASPT2 optimization.

The structure of the input follows the trends already explained in other geometry optimizations, that is, loops over the set of programs ending with SLAPAF. Notice that CASPT2 optimizations require obviously the CASPT2 input, but also the input for the ALASKA program, which computes the gradient numerically. Apart from that, a CASPT2 optimization input is identical to the corresponding CASSCF input. One should note that the numerical gradients are not as accurate as the analytic gradient. This can manifest itself in that there is no strict energy lowering the last few iterations, as displayed below:

```
 *************************************************************************************************************
 *                                   Energy Statistics for Geometry Optimization
 *************************************************************************************************************
                        Energy     Grad     Grad                 Step                 Estimated   Geom    Hessian
 Iter      Energy       Change     Norm     Max    Element    Max     Element     Final Energy Update Update In
   1   -191.38831696  0.00000000 0.208203-0.185586 nrc007  -0.285508* nrc007     -191.41950985 RS-
RFO  None    0
   2   -191.43810737 -0.04979041 0.117430-0.100908 nrc007  -0.190028* nrc007     -191.45424733 RS-
RFO  BFGS    0
   3   -191.45332692 -0.01521954 0.022751-0.021369 nrc007  -0.051028  nrc007     -191.45399070 RS-
RFO  BFGS    0
   4   -191.45414598 -0.00081906 0.012647 0.005657 nrc002  -0.013114  nrc007     -191.45421525 RS-
RFO  BFGS    0
   5   -191.45422730 -0.00008132 0.003630 0.001588 nrc002   0.004050  nrc002     -191.45423299 RS-
RFO  BFGS    0
   6   -191.45423140 -0.00000410 0.000744 0.000331 nrc006   0.000960  nrc013     -191.45423186 RS-
RFO  BFGS    0
   7   -191.45423123  0.00000017 0.000208-0.000098 nrc003  -0.001107  nrc013     -191.45423159 RS-
RFO  BFGS    0
   8   -191.45423116  0.00000007 0.000572 0.000184 nrc006   0.000422  nrc013     -191.45423131 RS-
RFO  BFGS    0
        +----------------------------------+----------------------------------+
        +    Cartesian Displacements       +    Gradient in internals         +
        +  Value      Threshold Converged? +  Value      Threshold Converged? +
  +-----+----------------------------------+----------------------------------+
  + RMS + 0.5275E-03  0.1200E-02     Yes   + 0.1652E-03  0.3000E-03     Yes   +
  +-----+----------------------------------+----------------------------------+
  + Max + 0.7738E-03  0.1800E-02     Yes   + 0.1842E-03  0.4500E-03     Yes   +
  +-----+----------------------------------+----------------------------------+
  Geometry is converged in   8 iterations to a Minimum Structure
 *************************************************************************************************************
 *************************************************************************************************************
```

The calculation converges in 8 iterations. At this point it is worth noticing how the convergence of CASPT2 energy is not chosen among the criteria for the convergence of the structure. The final structure is in fact decided by checking the Cartesian displacements and the gradient in non-redundant internal coordinates.

CASPT2 optimizations are expensive, however, the use for the resolution-of-identity options gives some relife. Notice that they are based on numerical gradients and many point-wise calculations are needed. In particular, the Cartesian gradients are computed using a two-point formula. Therefore, each macro-iteration in the optimization requires 2*N + 1 Seward/RASSCF/CASPT2 calculations, with N being the Cartesian degrees of freedom. In the present example, acrolein has eight atoms. From each atom, only two Cartesian coordinates are free to move (we are working within the C_s symmetry and the third coordinate is frozen), therefore the total number of Seward/RASSCF/CASPT2 iterations within each macro-iteration is 2*(8*2) + 1, that is, 33. In the current example a second trick has been used to speed up the numerical calculation. The explicit reference to ALASKA is excluded. This means that SLAPAF is called first without any gradients beeing computed explicitly. It does then abort automatically requesting an implicit calulation of the gradients, however, before doing so it compiles the internal coordinates and sets up a list of displayed geometries to be used in a numerical gradient procedure. In the present case this amounts to that the actuall number of micro iterations is reduced from 33 to 29.

The Table 10.8 displays the equilibrium geometrical parameters computed at the π-CASSCF and π-CASPT2 level of theory for the ground state of both isomers of acrolein. For sake of comparison, Table 10.8 includes experimental data obtained from microwave spectroscopy studies[232]. The computed parameters at π-CASPT2 level are in remarkable agreement with the experimental data. The predicted value of the C=C bond length is very close to the double bond length observed in ethylene. The other C-C bond has a length within the range expected for a C-C single bond: it appears shorter in the s-trans isomer as a consequence of the reduction of steric hindrance between the ethylenic and aldehydic moieties. CASSCF estimates a carbon-oxygen bond length shorter than the experimental value. For π-CASSCF optimization in conjugated systems this can be assumed as a general behavior [233, 231]. To explain such a discrepancy, one may invoke the fact that the C=O bond distance is particularly sensitive to electron correlation effects. The π electron correlation effects included at the π-CASSCF level tend to overestimate bond lengths. However, the lack of σ electron correlation, goes in the opposite direction, allowing shorter bond distances for double bonds. For the C-C double bonds, these contrasting behaviors compensate each other [230] resulting in quite an accurate value for the bond length at the π-CASSCF level. On the contrary, the extreme sensitivity of the C=O bond length to the electron correlation effects, leads to a general underestimation of the C-O double bond lengths, especially when such a bond is part of a conjugated system. It is indeed the effectiveness of the CASPT2 method in recovering dynamical correlation which leads to a substantial improvement in predicting the C-O double bond length.

Table 10.8: Geometrical parameters for the ground state of acrolein

Parameters[a]	π-CASSCF [04/4]		π-CASPT2		Expt.[b]
	s-cis	*s-trans*	*s-cis*	*s-trans*	
$C_1{=}O$	1.204	1.204	1.222	1.222	1.219
$C_1{-}C_2$	1.483	1.474	1.478	1.467	1.470
$C_2{=}C_3$	1.340	1.340	1.344	1.344	1.345
$\angle\ C_1C_2C_3$	123.0	121.7	121.9	120.5	119.8
$\angle\ C_2C_1O$	124.4	123.5	124.5	124.2	-

[a] Bond distances in Å and angles in degrees.

[b] Microwave spectroscopy data from ref. [232]. No difference between s-cis and s-trans isomers is reported

The use of numerical CASPT2 gradients can be extended to all the optimizations available in SLAPAF, for instance transition state searches. Use the following input for the water molecule to locate the linear transition state:

```
&GATEWAY; Title= Water, STO-3G Basis set
   Coord
   3
   H1   -0.761622       0.000000      -0.594478
   H2    0.761622       0.000000      -0.594478
   O     0.000000       0.000000       0.074915
   Basis set= STO-3G
   Group= NoSym
>>> EXPORT MOLCAS_MAXITER=500
>> DO WHILE
&SEWARD
```

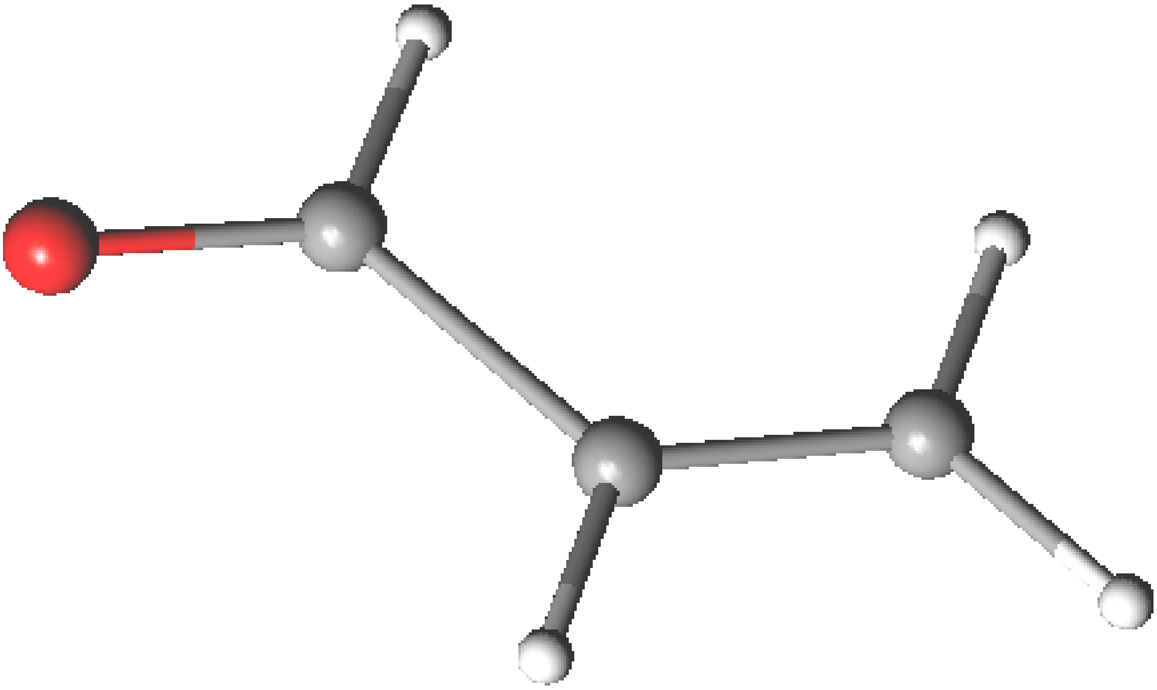


Figure 10.4: Acrolein geometrical isomers

```
>>> IF ( ITER = 1 ) <<<
&SCF; Title= water, STO-3g Basis set
Occupied= 5
>>> ENDIF <<<
&RASSCF
Nactel= 2 0 0
Inactive= 4
Ras2    = 2
&CASPT2
&SLAPAF; TS
>>> ENDDO <<<
```

After seventeen macro-iterations the linear water is reached:

```
***********************************************************************************************************
*                                  Energy Statistics for Geometry Optimization
***********************************************************************************************************
                       Energy     Grad     Grad              Step                 Estimated   Geom   Hess
Iter      Energy       Change     Norm     Max    Element    Max     Element     Final Energy Update Update
  1    -75.00567587  0.00000000 0.001456-0.001088 nrc003  -0.003312  nrc001      -75.00567822 RSIRFO  None
  2    -75.00567441  0.00000145 0.001471-0.001540 nrc003  -0.004162  nrc001      -75.00567851 RSIRFO  MSP
  3    -75.00566473  0.00000968 0.003484-0.002239 nrc003   0.008242  nrc003      -75.00567937 RSIRFO  MSP
  4    -75.00562159  0.00004314 0.006951-0.004476 nrc003   0.016392  nrc003      -75.00568012 RSIRFO  MSP
  5    -75.00544799  0.00017360 0.013935-0.008809 nrc003   0.033088  nrc003      -75.00568171 RSIRFO  MSP
  6    -75.00475385  0.00069414 0.027709-0.017269 nrc003   0.066565  nrc003      -75.00568219 RSIRFO  MSP
  7    -75.00201367  0.00274018 0.054556-0.032950 nrc003   0.084348* nrc003      -75.00430943 RSIRFO  MSP
  8    -74.99610698  0.00590669 0.086280-0.050499 nrc003   0.082995* nrc003      -74.99970484 RSIRFO  MSP
  9    -74.98774224  0.00836474 0.114866-0.065050 nrc003   0.080504* nrc003      -74.99249408 RSIRFO  MSP
 10    -74.97723219  0.01051005 0.139772 0.076893 nrc002   0.107680* nrc003      -74.98534124 RSIRFO  MSP
 11    -74.95944303  0.01778916 0.167230 0.096382 nrc002  -0.163238* nrc002      -74.97296260 RSIRFO  MSP
 12    -74.93101977  0.02842325 0.182451-0.114057 nrc002   0.185389* nrc002      -74.94544042 RSIRFO  MSP
 13    -74.90386636  0.02715341 0.157427-0.107779 nrc002   0.201775* nrc002      -74.91601550 RSIRFO  MSP
 14    -74.88449763  0.01936873 0.089073-0.064203 nrc002   0.240231  nrc002      -74.89232405 RSIRFO  MSP
 15    -74.87884197  0.00565566 0.032598-0.019326 nrc002   0.050486  nrc002      -74.87962885 RSIRFO  MSP
 16    -74.87855520  0.00028677 0.004934-0.004879 nrc003  -0.006591  nrc003      -74.87857157 RSIRFO  MSP
 17    -74.87857628 -0.00002108 0.000172-0.000120 nrc003   0.000262  nrc002      -74.87857630 RSIRFO  MSP
       +----------------------------------+----------------------------------+
       +    Cartesian Displacements       +    Gradient in internals         +
       +  Value      Threshold Converged? +  Value      Threshold Converged? +
 +-----+----------------------------------+----------------------------------+
 + RMS + 0.1458E-03  0.1200E-02     Yes   + 0.9925E-04  0.3000E-03     Yes   +
 +-----+----------------------------------+----------------------------------+
 + Max + 0.1552E-03  0.1800E-02     Yes   + 0.1196E-03  0.4500E-03     Yes   +
 +-----+----------------------------------+----------------------------------+
 Geometry is converged in  17 iterations to a Transition State Structure
***********************************************************************************************************
***********************************************************************************************************
```

We note that the optimization goes through three stages. The first one is while the structure still is very much ground-state-like. This is followed by the second stage in which the H-O-H angle is drastically changed at each iteration (iterations 7-13). The ”*” at ”Step Max” entry indicate that these steps were reduced because the steps were larger than allowed. Changing the default max step length from 0.3 to 0.6 (using keyword MaxStep) reduces the number of macro iterations by 2 iterations.

10.3 Computing a reaction path.

Chemists are familiarized with the description of a chemical reaction as a continuous motion on certain path of the potential energy hypersurfaces connecting reactants with products.

Those are considered minima in the hypersurface while an intermediate state known as the transition state would be a saddle point of higher energy. The height of the energy barrier separating reactants from products relates to the overall rate of reaction, the positions of the minima along the reaction coordinate give the equilibrium geometries of the species, and the relative energies relate to the thermodynamics of the process. All this is known as transition state theory.

The process to study a chemical reaction starts by obtaining proper geometries for reactants and products, follows by finding the position of the transition state, and finishes by computing as accurately as possible the relative energies relating the position of the species. To perform geometry optimizations searching for true minima in the potential energy surfaces (PES) is by now a well-established procedure (see section 10.2). An stationary point in the PES is characterized by having all the first derivatives of the energy with respect to each one of the independent coordinates equal to zero and the second derivatives larger than zero. First-order saddle points, on the contrary, have their second derivatives lower than zero for one coordinate, that is, they are maxima along this coordinate. A transition state is defined as a saddle point having only one negative second derivative along the specific coordinate known as the reaction coordinate. To simplify the treatment a special set of coordinates known as normal coordinates is defined in a way that the matrix of second derivatives is diagonal. A transition state will have one negative value in the diagonal of such a matrix.

Finally once the reactant, product and transition state geometries have been established one could perform a Intrinsic Reaction Coordinate (IRC) analysis. This to find the energy profile of the reaction and also to establish that the found transition state is connected to the reactant and the product.

10.3.1 Studying a reaction

The localization of the transition state of a reaction is of importance in both a qualitative and quantitative description of the reaction mechanism and the thermodynamics of a reaction. In the following example we will locate the transition state of the proton transfer reaction between the two species in Figs. 10.5 and 10.6. The example selected here is chosen to demonstrate the steps needed to find a transition state. For that sake we have limited our model to the SCF level of theory.

Reactant and product

The first step is to establish the two species in equilibrium. These calculations would constitute standard geometry optimizations with the input for the reactant

```
>>> Do while <<<
 &Seward
Basis set
C.cc-pVDZ....
C1            -1.9385577715        0.0976565175        0.4007212526
C2            -2.4151209200       -0.0592579424        2.8519334864
C3             0.7343463765        0.0088689871       -0.7477660837
End of Basis
Basis set
H.cc-pVDZ....
H1            -4.3244501026        0.0091320829        3.6086029352
H2            -0.8591520071       -0.2642180524        4.1663142585
```

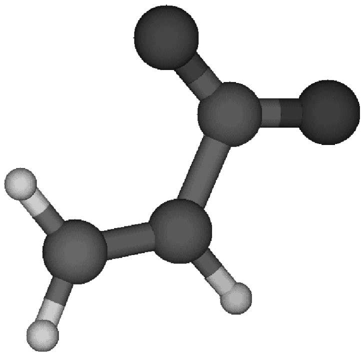


Figure 10.5: Reactant

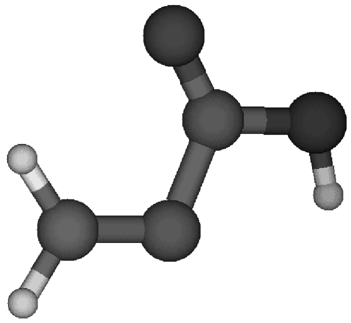


Figure 10.6: Product

```
H3           -3.4743702487        0.3026128386       -0.9501874771
End of Basis
Basis set
O.cc-pVDZ....
O1            0.7692102769        0.1847569555       -3.0700425345
O2            2.4916838932       -0.2232135341        0.7607580753
End of Basis
End of input
>>> IF ( ITER = 1 ) <<<
 &SCF
Core
Charge = -1
>>> ENDIF <<<
 &SCF &End
LUMORB
Charge = -1
 &Slapaf
Iterations = 20
>>> EndDo <<<
```

resulting in the following convergence pattern

```
                      Energy     Grad     Grad          Step          Estimated  Hessian Geom Hessian
Iter     Energy       Change     Norm      Max   Element  Max  Element   Final Energy Index Update Update
  1   -265.09033194 0.00000000 0.091418 0.044965 nrc003 0.069275  nrc003 -265.09529138  0  RF(S)   None
  2   -265.09646330-0.00613136 0.020358 0.008890 nrc003 0.040393  nrc008 -265.09684474  0  RF(S)   BFGS
  3   -265.09693242-0.00046912 0.011611-0.005191 nrc001 0.079285  nrc016 -265.09709856  0  RF(S)   BFGS
  4   -265.09655626 0.00037616 0.020775-0.010792 nrc016-0.070551  nrc016 -265.09706324  0  RF(S)   BFGS
  5   -265.09706308-0.00050682 0.003309-0.001628 nrc003-0.010263  nrc017 -265.09707265  0  RF(S)   BFGS
  6   -265.09707056-0.00000747 0.000958-0.000450 nrc011 0.017307  nrc017 -265.09707924  0  RF(S)   BFGS
  7   -265.09706612 0.00000444 0.002451 0.001148 nrc003-0.011228  nrc018 -265.09706837  0  RF(S)   BFGS
  8   -265.09707550-0.00000938 0.000516 0.000220 nrc001-0.004017  nrc014 -265.09707591  0  RF(S)   BFGS
  9   -265.09707586-0.00000036 0.000286 0.000104 nrc001 0.002132  nrc017 -265.09707604  0  RF(S)   BFGS
```

and for the product the input

```
>>> Do while <<<
&Seward
Basis set
C.cc-pVDZ....
C1           -2.0983667072        0.1000525724        0.5196668948
C2           -2.1177298783       -0.0920244467        3.0450747772
C3            0.5639781563        0.0024463770       -0.5245225314
End of Basis
Basis set
H.cc-pVDZ....
H1           -3.8870548756       -0.0558560582        4.1138131865
H2           -0.4133953535       -0.2946498869        4.2050068095
H3           -1.3495534119        0.3499572533       -3.3741881412
End of Basis
Basis set
O.cc-pVDZ....
O1            0.5100106099        0.2023808294       -3.0720173949
O2            2.5859515474       -0.2102046338        0.4795705925
End of Basis
End of input
>>> IF ( ITER = 1 ) <<<
 &SCF
Core
Charge = -1
>>> ENDIF <<<
 &SCF
LUMORB
Charge = -1
 &Slapaf
```

```
Iterations = 20
>>> EndDo <<<
```

resulting in the following convergence pattern

```
                        Energy      Grad      Grad              Step              Estimated   Hessian Geom Hessian
Iter      Energy        Change      Norm       Max  Element   Max  Element  Final Energy Index Update Update
 1    -265.02789209 0.00000000 0.062885-0.035740 nrc006-0.060778   nrc006 -265.02939600  0  RF(S)   None
 2    -265.02988181-0.00198972 0.018235-0.011496 nrc006-0.023664   nrc006 -265.03004886  0  RF(S)   BFGS
 3    -265.03005329-0.00017148 0.001631-0.000978 nrc009-0.015100   nrc017 -265.03006082  0  RF(S)   BFGS
 4    -265.03004953 0.00000376 0.002464-0.000896 nrc014 0.013752   nrc017 -265.03006022  0  RF(S)   BFGS
 5    -265.03006818-0.00001865 0.001059 0.000453 nrc013-0.007550   nrc014 -265.03007064  0  RF(S)   BFGS
 6    -265.03006524 0.00000294 0.001800 0.000778 nrc014 0.006710   nrc014 -265.03007032  0  RF(S)   BFGS
 7    -265.03006989-0.00000465 0.000381 0.000190 nrc005 0.003078   nrc016 -265.03007014  0  RF(S)   BFGS
 8    -265.03006997-0.00000008 0.000129-0.000094 nrc016-0.001305   nrc017 -265.03007003  0  RF(S)   BFGS
```

The computed reaction energy is estimated to about 42 kcal/mol at this level of theory.

Transition state optimization

To locate the transition state it is important to identify the reaction coordinate. In our case here we note that the significant reaction coordinates are the bond distances between C1 and H3, and O1 and H3. In the location of the transition state we will start from the geometry of the reactant for which the O1-H3 bond distance is 2.51 Ångström. We will conduct the search in a number of constrained geometry optimizations in which we step by step reduce the O1-H3 distance towards the distance in the product of 0.95 Ångström. The selected series is 2.0, 1.5, 1.3, and 1.0 Ångström. To constraint the O1-H3 bond distance we modify the input to the `GATEWAY` moduel by adding the following:

```
Constraint
R1 = Bond H3 O1
Value
R1 = 2.0 Angstrom
End of Constraint
```

The `SLAPAF` module's associated input looks like:

```
&Slapaf &End
Iterations
20
FindTS
PRFC
End of Input
```

This will correspond to the input for the first of the series of constraint geometry optimization. However, note the keyword FindTS. This keyword will make the SLAPAF module switch from a constrained geometry optimization to a transition state geometry optimization if the updated geometrical Hessian contains one negative eigenvalue. It is of course our hope that during the series of constrained geometry optimizations that we will run into this situation and find the transition state. The convergence pattern for the first constrained optimization is

```
                        Energy      Grad      Grad              Step              Estimated   Hessian Geom Hessian
Iter      Energy        Change      Norm       Max  Element   Max  Element  Final Energy Index Update Update
 1    -265.09707600 0.00000000 0.965614 0.965614 Cns001 0.230366* nrc009 -265.07671229  0  MFRFS   None
 2   -265.08759913 0.00947687 0.216939 0.214768 Cns001 0.081441  nrc012 -265.08946379  0  MFRFS   MSP
 3   -265.08218288 0.00541624 0.014770 0.007032 nrc010 0.019690  nrc010 -265.08242668  0  MFRFS   MSP
 4   -265.08251826-0.00033537 0.003644-0.001560 nrc003 0.005075  nrc002 -265.08254163  0  MFRFS   MSP
 5   -265.08254834-0.00003008 0.001274-0.000907 nrc012 0.026237! nrc016 -265.08257455  0  MFRFS   MSP
 6   -265.08251413 0.00003421 0.003036-0.002420 nrc016-0.024325  nrc016 -265.08254699  0  MFRFS   MSP
 7   -265.08254682-0.00003269 0.000837-0.000426 nrc012 0.012351  nrc017 -265.08255083  0  MFRFS   MSP
 8   -265.08255298-0.00000616 0.000470 0.000238 nrc016-0.005376  nrc017 -265.08255421  0  MFRFS   MSP
 9   -265.08255337-0.00000038 0.000329-0.000154 nrc012-0.004581  nrc014 -265.08255409  0  MFRFS   MSP
10   -265.08255418-0.00000081 0.000206-0.000148 nrc012-0.000886  nrc014 -265.08255425  0  MFRFS   MSP
11   -265.08255430-0.00000013 0.000123-0.000097 nrc012-0.001131  nrc014 -265.08255436  0  MFRFS   MSP
```

Here we note that the Hessian index is zero, i.e. the optimization is a constrained geometry optimization. The final structure is used as the starting geometry for the 2nd constrained optimization at 1.5 Ångström. This optimization did not find a negative eigenvalue either. However, starting the 3rd constrained optimization from the final structure of the 2nd constrained optimization resulted in the convergence pattern

```
                      Energy     Grad     Grad          Step          Estimated  Hessian Geom Hessian
Iter     Energy       Change     Norm      Max  Element  Max  Element  Final Energy Index Update Update
 1   -265.03250948 0.00000000 0.384120 0.377945 Cns001-0.209028* nrc007 -264.99837542 0  MFRFS   None
 2   -265.01103140 0.02147809 0.120709 0.116546 Cns001-0.135181  nrc007 -265.01209656 0  MFRFS   MSP
 3   -265.00341440 0.00761699 0.121043-0.055983 nrc005-0.212301* nrc007 -264.98788416 1  MFRFS   MSP
 4   -264.99451339 0.00890101 0.089986 0.045423 nrc007 0.123178* nrc002 -264.99582814 1  MFRFS   MSP
 5   -264.99707885-0.00256546 0.044095-0.015003 nrc009 0.159069* nrc015 -265.00090995 1  MFRFS   MSP
 6   -264.99892919-0.00185034 0.033489-0.013653 nrc015-0.124146  nrc015 -265.00050567 1  MFRFS   MSP
 7   -265.00031159-0.00138240 0.009416-0.004916 nrc018-0.156924  nrc018 -265.00070286 1  MFRFS   MSP
 8   -265.00019076 0.00012083 0.009057 0.005870 nrc018 0.081240  nrc018 -265.00049408 1  MFRFS   MSP
 9   -265.00049567-0.00030490 0.003380 0.001481 nrc011-0.070124  nrc015 -265.00056966 1  MFRFS   MSP
10   -265.00030276 0.00019291 0.159266-0.159144 Cns001 0.114927! nrc015 -264.99874954 0  MFRFS   MSP
11   -265.00098377-0.00068101 0.031621-0.008700 nrc005-0.101187  nrc007 -265.00046906 1  MFRFS   MSP
12   -265.00050857 0.00047520 0.003360 0.001719 nrc015 0.012580  nrc015 -265.00052069 1  MFRFS   MSP
13   -265.00052089-0.00001233 0.001243-0.000590 nrc017-0.006069  nrc017 -265.00052323 1  MFRFS   MSP
14   -265.00052429-0.00000340 0.000753 0.000259 nrc011-0.002449  nrc018 -265.00052458 1  MFRFS   MSP
15   -265.00052441-0.00000011 0.000442-0.000136 nrc007 0.003334  nrc018 -265.00052464 1  MFRFS   MSP
16   -265.00052435 0.00000006 0.000397 0.000145 nrc017 0.001628  nrc010 -265.00052459 1  MFRFS   MSP
```

Here a negative Hessian eigenvalue was found at iteration 3. At this point the optimization turn to a normal quasi-Newton Raphson optimization without any constraints. We note that the procedure flips back to a constrained optimization at iteration 10 but is finished as an optimization for a transition state. The predicted activation energy is estimated to 60.6 kcal/mol (excluding vibrational corrections). The computed transition state is depicted in Fig. 10.7.

The remaining issue is if this is a true transition state. This issue can only be resolved by doing a calculation of the analytical Hessian using the MCKINLEY module (execution of the MCLR module is automatic). The corresponding input is

```
 &Seward
Basis set
C.cc-pVDZ....
C1            -1.8937541206       0.0797525492       0.5330826031
C2            -2.3239194706      -0.0748842444       3.0012862573
C3             0.7556108398      -0.0065134659      -0.5801137465
End of Basis
Basis set
H.cc-pVDZ....
H1            -4.2196708766      -0.0106202053       3.8051971560
H2            -0.7745261239      -0.2775291936       4.3506967746
H3            -1.9256618348       0.2927045555      -2.1370156139
End of Basis
```

```
Basis set
O.cc-pVDZ....
O1            0.2162486684       0.2196587542       -2.9675781183
O2            2.8171388123      -0.2187115071        0.3719375423
End of Basis
End of input
 &SCF
Charge = -1
 &McKinley
Perturbation
Hessian
```

From the output of the MCLR code

```
      ***********************************
      *                                 *
      * Harmonic frequencies in cm-1    *
      * Intensities in km/mole          *
      *                                 *
      * No correction due to curvlinear *
      * representations has been done   *
      *                                 *
      ***********************************
    Symmetry a
   ==============
                         1         2         3         4         5         6
            Freq.   i2027.40     i2.00     i0.07      0.05      0.07      2.02
...
...
                         7         8         9        10        11        12
            Freq.       3.57    145.36    278.41    574.44    675.27    759.94
...
...
                        13        14        15        16        17        18
            Freq.     927.78    943.60   1000.07   1225.34   1265.63   1442.57
...
...
                        19        20        21        22        23        24
            Freq.    1517.91   1800.86   1878.11   2294.83   3198.94   3262.66
```

we can conclude that we have one imaginary eigenvalue (modes 2-7 corresponds to the translational and rotational zero frequency modes) and that the structure found with this procedure indeed is a transition state. A post calculation analysis of the vibrational modes using the MOLDEN package confirm that the vibrational mode with the imaginary frequency is a mode which moves the proton from the oxygen to the carbon.

10.3.2 Finding the reaction path - an IRC study

A minimum energy path (MEP) is defined as the path defined by a sequence of geometries obtained by a series of optimizations on a hypersphere. The series of constrained optimization starts from some starting structure and the optimized structure at each step is taken as the start for the next step. The constraint in these optimizations is the radius (in mass weighted coordinates) of the hyper sphere with the origin defined by the starting geometry. If the starting structure is a transition state the path is called an Intrinsic Reaction Coordinate (IRC) path. Since the transition structure (TS) has a negative index of the Hessian we have two paths away from the TS. One leading us to the product(s) and one going to the reactant(s). The IRC analysis is used to verify whether the TS is really connecting the expected reactant(s) and product(s) and it is performed in forward and backward directions starting from the TS. This analysis is obtained using the keyword IRC with the SLAPAF specifying the number of points and, if different from the default value (0.10 au), the radius of the hypersphere with the keywords nIRC and IRCStep, respectively. The reaction vector can be found on RUNOLD or it can be specified explicitly (see keyword "REACtion vector"). In the latter case, the vector can be find at the end of the optimization job in the "The Cartesian Reaction vector" section of the SLAPAF output. A file named *$Project.irc.molden* (read by MOLDEN) will be generated in $WorkDir containing only those points belonging to the IRC. Here an example for an IRC analysis with 20 points back and forth and with 0.05 au as step. The reaction vector will be read on RUNOLD.

```
>>> EXPORT MOLCAS_MAXITER=500
>>> Do while <<<
...
 &Slapaf &End
IRC
nIRC
 20
IRCStep
 0.05
Iterations
200
End of Input
>>> EndDo <<<
```

If the file *RUNFILE* is not available, the reaction vector must be specified in the input.

```
>>> EXPORT MOLCAS_MAXITER=500
>>> Do while <<<
...
 &Slapaf &End
IRC
nIRC
 20
IRCStep
 0.05
 REACtion vector
   0.140262        0.000000        0.179838
   0.321829        0.000000       -0.375102
  -0.006582        0.000000       -0.048402
  -0.032042       -0.018981       -0.003859
  -0.423466        0.000000        0.247525
Iterations
200
End of Input
>>> EndDo <<<
```

10.4 High quality wave functions at optimized structures

Here we will give an example of how geometrical structures obtained at one level of theory can be used in an analysis at high quality wave functions. Table 10.9 compiles the obtained CASSCF geometries for the dimethylcarbene to propene reaction (see Fig 10.8). They can be compared to the MP2 geometries [234]. The overall agreement is good.

The wave function at each of the geometries was proved to be almost a single configuration. The second configuration in all the cases contributed by less than 5% to the weight of the wave function. It is a double excited replacement. Therefore, although MP2 is not generally expected to describe properly a bond formation in this case its behavior seems to be validated.

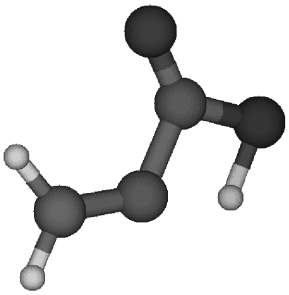
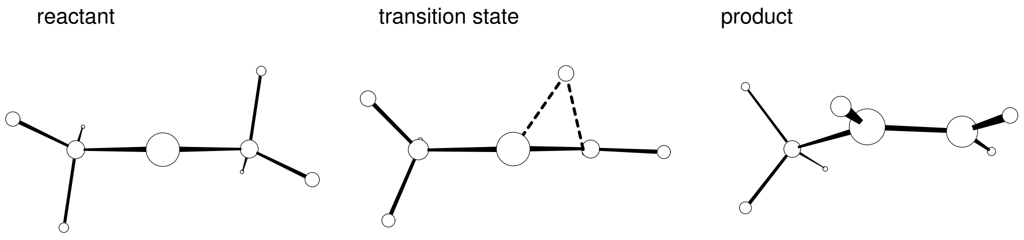


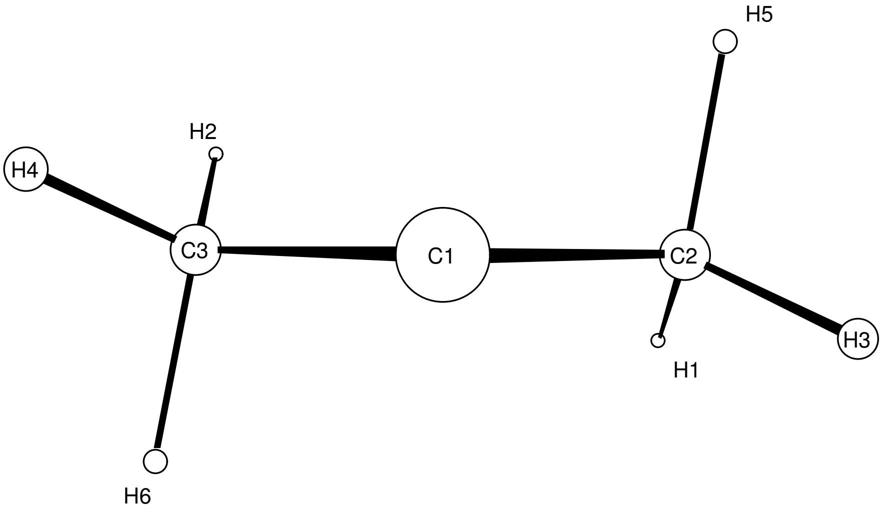
Figure 10.7: Transition state

Figure 10.8: Dimethylcarbene to propene reaction path



The larger discrepancies appear in the carbon-carbon distances in the dimethylcarbene and in the transition state. On one hand the basis set used in the present example were small; on the other hand there are indications that the MP2 method overestimates the hyper conjugation effects present in the dimethylcarbene [234]. Figure 10.9 displays the dimethylcarbene with indication of the employed labeling.

Figure 10.9: Dimethylcarbene atom labeling



The main structural effects occurring during the reaction can be observed displayed in Table 10.9. As the rearrangement starts out one hydrogen atom (H_5) moves in a plane almost perpendicular to the plane formed by the three carbon atoms while the remaining two hydrogen atoms on the same methyl group swing very rapidly into a nearly planar position (see Figure 10.8 on page 477). As the π bond is formed we observe a contraction of the C_1-C_2 distance. In contrast, the spectator methyl group behaves as a rigid body. Their parameters were not compiled here but it rotates and bends slightly [234]. Focusing on the second half reaction, the moving hydrogen atom rotates into the plane of the carbon atoms to form the new C_1-H_5 bond. This movement is followed by a further shortening of the preformed C_1-C_2 bond, which acquires the bond distance of a typical double carbon bond, and smaller adjustments in the positions of the other atoms. The structures of the reactant, transition state, and product are shown in Figure 10.8.

As was already mentioned we will apply now higher-correlated methods for the reactant, product, and transition state system at the CASSCF optimized geometries to account for more accurate relative energies. In any case a small basis set has been used and therefore the goal is not to be extremely accurate. For more complete results see Ref. [234]. We are going to perform calculations with the MP2, MRCI, ACPF, CASPT2, CCSD, and CCSD(T) methods.

Starting with dimethylcarbene, we will use the following input file:

```
  &SEWARD &END
 Title
  Dimethylcarbene singlet C2-sym
  CASSCF(ANO-VDZP) opt geometry
 Symmetry
  XY
```

Table 10.9: Bond distances (Å) and bond angles (deg) of dimethylcarbene, propene, and their transition state[a]

	C_1C_3	C_1C_2	$C_2C_1C_3$	$C_1C_3H_6$	$C_2C_1C_3H_6$	C_2H_5	C_1H_5	$C_1C_2H_5$	$C_3C_1C_2H_5$
					Dimethylcarbene				
CAS[b]	1.497	1.497	110.9	102.9	88.9	1.099		102.9	88.9
MP2[c]	1.480	1.480	110.3	98.0	85.5	1.106		98.0	85.5
					Transition structure				
CAS[b]	1.512	1.394	114.6	106.1	68.6	1.287	1.315	58.6	76.6
MP2[c]	1.509	1.402	112.3	105.1	69.2	1.251	1.326	59.6	77.7
					Propene				
CAS[b]	1.505	1.344	124.9	110.7	59.4				
MP2[c]	1.501	1.338	124.4	111.1	59.4				

[a] C_1, carbenoid center; C_2, carbon which looses the hydrogen H_5. See Figure 10.9.

[b] Present results. CASSCF, ANO-S C 3*s*2*p*1*d*, H 2*d*1*p*. Two electrons in two orbitals.

[c] MP2 6-31G(2p,d), Ref. [234].

```
Basis set
C.ANO-S...3s2p1d.
C1              .0000000000        .0000000000       1.2019871414
C2              .0369055124       2.3301037548       -.4006974719
End of basis
Basis set
H.ANO-S...2s1p.
H1             -.8322309260       2.1305589948      -2.2666729831
H2             -.7079699536       3.9796589218        .5772009623
H3             2.0671154914       2.6585385786       -.6954193494
End of basis
PkThrs
 1.0E-10
End of input
 &SCF &END
Title
Dmc
Occupied
7 5
End of input
 &RASSCF &END
Title
Dmc
Symmetry
 1
Spin
 1
Nactel
 2  0  0
Inactive
 6  5
Ras2
 1  1
Thrs
```

```
1.0E-05,1.0E-03,1.0E-03
Iteration
50,25
LumOrb
End of input
 &CASPT2 &END
Title
Dmc
LRoot
1
Frozen
 2  1
End of input
 &MOTRA &END
Title
Dmc
Frozen
 2  1
JobIph
End of input
 &GUGA &END
Title
Dmc
Electrons
18
Spin
 1
Inactive
 4  4
Active
 1  1
Ciall
 1
Print
 5
End of input
 &MRCI &END
Title
Dimethylcarbene
SDCI
End of input
 &MRCI &END
Title
Dimethylcarbene
ACPF
End of input
* Now we generate the single ref. function
* for coupled-cluster calculations
 &RASSCF &END
Title
Dmc
Symmetry
 1
Spin
 1
Nactel
 0  0  0
Inactive
 7  5
Ras2
 0  0
Thrs
1.0E-05,1.0E-03,1.0E-03
Iteration
50,25
LumOrb
```

```
OutOrbitals
 Canonical
End of input
 &MOTRA &END
Title
Dmc
Frozen
 2  1
JobIph
End of input
 &CCSDT &END
Title
 Dmc
CCT
Iterations
 40
Triples
 2
End of input
```

Observe in the previous input that we have generated a multiconfigurational wave function for CASPT2, MRCI, and ACPF wave functions but a single configuration reference wave function (using RASSCF program with the options OUTOrbitals and CANOnical) for the CCSD and CCSD(T) wave functions. Notice also that to compute a multiconfigurational ACPF wave function we have to use the MRCI program, not the CPF module which does not accept more than one single reference. In all the highly correlated methods we have frozen the three carbon core orbitals because of the reasons already explained in section 10.1. For MRCI, ACPF, CCSD, and CCSD(T) the freezing is performed in the MOTRA step.

One question that can be addressed is which is the proper reference space for the multiconfigurational calculations. As was explained when we selected the active space for the geometry optimizations, we performed several tests at different stages in the reaction path and observed that the smallest meaningful active space, two electrons in two orbitals, was sufficient in all the cases. We can come back to this problem here to select the reference for CASPT2, MRCI, and ACPF methods. The simple analysis of the SCF orbital energies shows that in dimethylcarbene, for instance, the orbital energies of the C-H bonds are close to those of the C-C σ bonds and additionally those orbitals are strongly mixed along the reaction path. A balanced active space including all orbitals necessary to describe the shifting H-atom properly would require a full valence space of 18 electrons in 18 orbitals. This is not a feasible space, therefore we proceed with the minimal active space and analyze later the quality of the results. The CASSCF wave function will then include for dimethylcarbene and the transition state structure the $(\sigma)^2(\pi)^0$ and $(\sigma)^0(\pi)^2$ configurations correlating the non-bonded electrons localized at the carbenoid center where as for propene the active space include the equivalent valence π space.

The GUGA input must be built carefully. There are several ways to specify the reference configurations for the following methods. First, the keyword ELECtrons refers to the total number of electrons that are going to be correlated, that is, all except those frozen in the previous MOTRA step. Keywords INACtive and ACTIve are optional and describe the number of inactive (occupation two in all the reference configurations) and active (varying occupation number in the reference configurations) orbitals of the space. Here ACTIve indicates one orbital of each of the symmetries. The following keyword CIALl indicates that the reference space will be the full CI within the subspace of active orbitals. It must be always followed by symmetry index (number of the irrep) for the resulting wave function, one here.

For the transition state structure we do not impose any symmetry restriction, therefore the calculations are performed in the C_1 group with the input file:

```
 &SEWARD &END
Title
 Dimethylcarbene to propene
 Transition State C1 symmetry
 CASSCF (ANO-VDZP) opt geometry
Basis set
C.ANO-S...3s2p1d.
End of basis
Basis set
H.ANO-S...2s1p.
End of basis
PkThrs
 1.0E-10
End of input
 &SCF &END
Title
 Ts
Occupied
 12
End of input
 &MBPT2 &END
Title
 Ts
Frozen
 3
End of input
 &RASSCF &END
Title
 Ts
Symmetry
 1
Spin
 1
Nactel
 2  0  0
Inactive
 11
Ras2
 2
Iteration
50,25
LumOrb
End of input
 &CASPT2 &END
Title
 Ts
LRoot
 1
Frozen
 3
End of input
 &MOTRA &END
Title
 Ts
Frozen
 3
JobIph
End of input
 &GUGA &END
Title
 Ts
Electrons
 18
```

```
Spin
 1
Inactive
 8
Active
 2
Ciall
 1
Print
 5
End of input
 &MRCI &END
Title
 Ts
SDCI
End of input
 &MRCI &END
Title
 Ts
ACPF
End of input
 &RASSCF &END
Title
 Ts
Symmetry
 1
Spin
 1
Nactel
 0  0  0
Inactive
 12
Ras2
 0
Iteration
50,25
LumOrb
OutOrbitals
 Canonical
End of input
 &MOTRA &END
Title
 Ts
Frozen
 3
JobIph
End of input
 &CCSDT &END
Title
 Ts
CCT
Iterations
 40
Triples
 2
End of input
```

Finally we compute the wave functions for the product, propene, in the C$_s$ symmetry group with the input:

```
 &SEWARD &END
Title
 Propene singlet Cs-sym
 CASSCF(ANO-VDZP) opt geometry
Symmetry
```

```
 Z
Basis set
C.ANO-S...3s2p1d.
C1           -2.4150580342         .2276105054         .0000000000
C2             .0418519070         .8733601069         .0000000000
C3            2.2070668305       -.9719171861         .0000000000
End of basis
Basis set
H.ANO-S...2s1p.
H1           -3.0022907382       -1.7332097498         .0000000000
H2           -3.8884900111        1.6454331428         .0000000000
H3             .5407865292        2.8637419734         .0000000000
H4            1.5296107561       -2.9154199848         .0000000000
H5            3.3992878183       -.6985812202        1.6621549148
End of basis
PkThrs
 1.0E-10
End of input
 &SCF &END
Title
Propene
Occupied
10 2
End of input
 &MBPT2 &END
Title
 Propene
Frozen
 3 0
End of input
 &RASSCF &END
Title
Propene
Symmetry
1
Spin
1
Nactel
 2  0  0
Inactive
10  1
Ras2
 0  2
Thrs
1.0E-05,1.0E-03,1.0E-03
Iteration
50,25
LumOrb
End of input
 &CASPT2 &END
Title
Propene
LRoot
1
Frozen
 3  0
End of input
 &MOTRA &END
Title
Propene
Frozen
 3  0
JobIph
End of input
 &GUGA &END
Title
```

```
Propene
Electrons
18
Spin
 1
Inactive
 7  1
Active
 0  2
Ciall
 1
Print
 5
End of input
 &MRCI &END
Title
Propene
SDCI
End of input
 &MRCI &END
Title
Propene
ACPF
End of input
 &RASSCF &END
Title
Propene
Symmetry
1
Spin
1
Nactel
 0  0  0
Inactive
10  2
Ras2
 0  0
Thrs
1.0E-05,1.0E-03,1.0E-03
Iteration
50,25
LumOrb
OutOrbitals
 Canonical
End of input
 &MOTRA &END
Title
Propene
Frozen
 3  0
JobIph
End of input
 &CCSDT &END
Title
 Propene
CCT
Iterations
 40
Triples
 2
End of input
```

Table 10.10 compiles the total and relative energies obtained for the studied reaction at the different levels of theory employed.

Table 10.10: Total (au) and relative (Kcal/mol, in braces) energies obtained at the different theory levels for the reaction path from dimethylcarbene to propene

Single configurational methods

method	RHF	MP2	CCSD	CCSD(T)
Dimethylcarbene				
	-117.001170	-117.392130	-117.442422	-117.455788
Transition state structure				
	-116.972670	-117.381342	-117.424088	-117.439239
BH[a]	(17.88)	(6.77)	(11.50)	(10.38)
Propene				
	-117.094700	-117.504053	-117.545133	-117.559729
EX[b]	(-58.69)	(-70.23)	(-64.45)	(-65.22)

Multiconfigurational methods

method	CASSCF	CASPT2	SD-MRCI+Q	ACPF
Dimethylcarbene				
	-117.020462	-117.398025	-117.447395	-117.448813
Transition state structure				
	-116.988419	-117.383017	-117.430951	-117.432554
BH[a]	(20.11)	(9.42)	(10.32)	(10.20)
Propene				
	-117.122264	-117.506315	-117.554048	-117.554874
EX[b]	(-63.88)	(-67.95)	(-66.93)	(-66.55)

[a] Barrier height. Needs to be corrected with the zero point vibrational correction.

[b] Exothermicity. Needs to be corrected with the zero point vibrational correction.

We can discuss now the quality of the results obtained and their reliability (for a more careful discussion of the accuracy of quantum chemical calculations see Ref. [223]). In first place we have to consider that a valence double-zeta plus polarization basis set is somewhat small to obtain accurate results. At least a triple-zeta quality would be required. The present results have, however, the goal to serve as an example. We already pointed out that the CASSCF geometries were very similar to the MP2 reported geometries [234]. This fact validates both methods. MP2 provides remarkably accurate geometries using basis sets of triple-zeta quality, as in Ref. [234], in situations were the systems can be described as singly configurational, as the CASSCF calculations show. The Hartree-Fock configuration has a contribution of more than 95% in all three structures, while the largest weight for another configuration appears in propene for $(\pi)^0(\pi^*)^2$ (4.2%).

The MRCI calculations provide also one test of the validity of the reference wave function. For instance, the MRCI output for propene is:

```
               FINAL RESULTS FOR STATE NR   1
 CORRESPONDING ROOT OF REFERENCE CI IS NR:  1
            REFERENCE CI ENERGY: -117.12226386
         EXTRA-REFERENCE WEIGHT:      .11847074
          CI CORRELATION ENERGY:     -.38063043
                      CI ENERGY: -117.50289429
            DAVIDSON CORRECTION:     -.05115380
               CORRECTED ENERGY: -117.55404809
                ACPF CORRECTION:     -.04480105
               CORRECTED ENERGY: -117.54769535
      CI-COEFFICIENTS LARGER THAN  .050
  NOTE: THE FOLLOWING ORBITALS WERE FROZEN
  ALREADY AT THE INTEGRAL TRANSFORMATION STEP
  AND DO NOT EXPLICITLY APPEAR:
        SYMMETRY:    1    2
      PRE-FROZEN:    3    0
  ORDER OF SPIN-COUPLING: (PRE-FROZEN, NOT SHOWN)
                          (FROZEN, NOT SHOWN)
                           VIRTUAL
                           ADDED VALENCE
                           INACTIVE
                           ACTIVE
  ORBITALS ARE NUMBERED WITHIN EACH SEPARATE SYMMETRY.
      CONFIGURATION     32   COEFFICIENT  -.165909   REFERENCE
 SYMMETRY             1  1  1  1  1  1  1  2  2  2
 ORBITALS             4  5  6  7  8  9 10  1  2  3
 OCCUPATION           2  2  2  2  2  2  2  2  0  2
 SPIN-COUPLING        3  3  3  3  3  3  3  3  0  3
      CONFIGURATION     33   COEFFICIENT  -.000370   REFERENCE
 SYMMETRY             1  1  1  1  1  1  1  2  2  2
 ORBITALS             4  5  6  7  8  9 10  1  2  3
 OCCUPATION           2  2  2  2  2  2  2  2  1  1
 SPIN-COUPLING        3  3  3  3  3  3  3  3  1  2
      CONFIGURATION     34   COEFFICIENT   .924123   REFERENCE
 SYMMETRY             1  1  1  1  1  1  1  2  2  2
 ORBITALS             4  5  6  7  8  9 10  1  2  3
 OCCUPATION           2  2  2  2  2  2  2  2  2  0
 SPIN-COUPLING        3  3  3  3  3  3  3  3  3  0
**************************************************************
```

The Hartree-Fock configuration contributes to the MRCI configuration with a weight of 85.4%, while the next configuration contributes by 2.8%. Similar conclusions can be obtained analyzing the ACPF results and for the other structures. We will keep the MRCI results including the Davidson correction (MRCI+Q) which corrects for the size-inconsistency of the truncated CI expansion [223].

For CASPT2 the evaluation criteria were already commented in section 10.5. The portion of the CASPT2 output for propene is:

```
      Reference energy:        -117.1222638304
      E2 (Non-variational):       -.3851719971
      E2 (Variational):           -.3840516039
      Total energy:            -117.5063154343
      Residual norm:               .0000000000
      Reference weight:            .87905
      Contributions to the CASPT2 correlation energy
      Active & Virtual Only:          -.0057016698
      One Inactive Excited:           -.0828133881
      Two Inactive Excited:           -.2966569393
----------------------------------------------------------------------------
Report on small energy denominators, large components, and large energy contributions.
The ACTIVE-MIX index denotes linear combinations which gives ON expansion functions
  and makes H0 diagonal within type.
DENOMINATOR: The (H0_ii - E0) value from the above-mentioned diagonal approximation.
RHS value: Right-Hand Side of CASPT2 Eqs.
COEFFICIENT: Multiplies each of the above ON terms in the first-order wave function.
Thresholds used:
        Denominators:  .3000
          Components:  .0250
Energy contributions:  .0050
CASE SYMM ACTIVE   NON-ACT IND    DENOMINATOR  RHS VALUE   COEFFICIENT CONTRIBUTION
AIVX   1  Mu1.0003 In1.004 Se1.022  2.28926570 .05988708  -.02615995  -.00156664
```

The weight of the CASSCF reference to the first-order wave function is here 87.9%, very close to the weights obtained for the dimethylcarbene and the transition state structure, and there is only a small contribution to the wave function and energy which is larger than the selected thresholds. This should not be considered as a intruder state, but as a contribution from the fourth inactive orbital which could be, eventually, included in the active space. The contribution to the second-order energy in this case is smaller than 1 Kcal/mol. It can be observed that the same contribution shows up for the transition state structure but not for the dimethylcarbene. In principle this could be an indication that a larger active space, that is, four electrons in four orbitals, would give a slightly more accurate CASPT2 energy. The present results will probably overestimate the second-order energies for the transition state structure and the propene, leading to a slightly smaller activation barrier and a slightly larger exothermicity, as can be observed in Table 10.10. The orbitals pointed out as responsible for the large contributions in propene are the fourth inactive and 22nd secondary orbitals of the first symmetry. They are too deep and too high, respectively, to expect that an increase in the active space could in fact represent a great improvement in the CASPT2 result. In any case we tested for four orbitals-four electrons CASSCF/CASPT2 calculations and the results were very similar to those presented here.

Finally we can analyze the so-called τ_1-diagnostic [235] for the coupled-cluster wave functions. τ_1 is defined for closed-shell coupled-cluster methods as the Euclidean norm of the vector of T_1 amplitudes normalized by the number of electrons correlated: $\tau_1 = ||T_1||/N_{el}^{1/2}$. In the output of the CCSD program we have:

```
Convergence after  17  Iterations
Total energy (diff) :    -117.54513288       -.00000061
Correlation energy  :       -.45043295
E1aa   contribution :        .00000000
E1bb   contribution :        .00000000
E2aaaa contribution :       -.04300448
E2bbbb contribution :       -.04300448
E2abab contribution :       -.36442400
```

```
 Five largest amplitudes of :T1aa
  SYMA   SYMB   SYMI   SYMJ     A      B      I      J     VALUE
    2      0      2      0      4      0      2      0     -.0149364994
    2      0      2      0      2      0      2      0      .0132231037
    2      0      2      0      8      0      2      0     -.0104167047
    2      0      2      0      7      0      2      0     -.0103366543
    2      0      2      0      1      0      2      0      .0077537734
 Euclidean norm is :      .0403635306
 Five largest amplitudes of :T1bb
  SYMA   SYMB   SYMI   SYMJ     A      B      I      J     VALUE
    2      0      2      0      4      0      2      0     -.0149364994
    2      0      2      0      2      0      2      0      .0132231037
    2      0      2      0      8      0      2      0     -.0104167047
    2      0      2      0      7      0      2      0     -.0103366543
    2      0      2      0      1      0      2      0      .0077537734
 Euclidean norm is :      .0403635306
```

In this case T1aa and T1bb are identical because we are computing a closed-shell singlet state. The five largest T_1 amplitudes are printed, as well as the Euclidean norm. Here the number of correlated electrons is 18, therefore the value for the τ_1 diagnostic is 0.01. This value can be considered acceptable as evaluation of the quality of the calculation. The use of τ_1 as a diagnostic is based on an observed empirical correlation: larger values give poor CCSD results for molecular structures, binding energies, and vibrational frequencies [236]. It was considered that values larger than 0.02 indicated that results from single-reference electron correlation methods limited to single and double excitations should be viewed with caution.

There are several considerations concerning the τ_1 diagnostic [235]. First, it is only valid within the frozen core approximation and it was defined for coupled-cluster procedures using SCF molecular orbitals in the reference function. Second, it is a measure of the importance of non-dynamical electron correlation effects and not of the degree of the multireference effects. Sometimes the two effects are related, but not always (see discussion in Ref. [236]). Finally, the performance of the CCSD(T) method is reasonably good even in situations where τ_1 has a value as large as 0.08. In conclusion, the use of τ_1 together with other wave function analysis, such as explicitly examining the largest T_1 and T_2 amplitudes, is the best approach to evaluate the quality of the calculations but this must be done with extreme caution.

As the present systems are reasonably well described by a single determinant reference function there is no doubt that the CCSD(T) method provides the most accurate results. Here CASPT2, MRCI+Q, ACPF, and CCSD(T) predict the barrier height from the reactant to the transition state with an accuracy better than 1 Kcal/mol. The correspondence is somewhat worse, about 3 Kcal/mol, for the exothermicity. As the difference is largest for the CCSD(T) method we may conclude than triple and higher order excitations are of importance to achieve a balanced correlation treatment, in particular with respect to the partially occupied π^* orbital at the carbenoid center. It is also noticeable that the relative MP2 energies appear to be shifted about 3-4 Kcal/mol towards lower values. This effect may be due to the overestimation of the hyper-conjugation effect which appears to be strongest in dimethylcarbene [237, 234].

Additional factors affecting the accuracy of the results obtained are the zero point vibrational energy correction and, of course, the saturation of the one particle basis sets. The zero point vibrational correction could be computed by performing a numerical harmonic vibrational analysis at the CASSCF level using MOLCAS. At the MP2 level [234] the obtained values were -1.1 Kcal/mol and 2.4 Kcal/mol for the activation barrier height and exothermicity, respectively. Therefore, if we take as our best values the CCSD(T) results of 10.4 and

-65.2 Kcal/mol, respectively, our prediction would be an activation barrier height of 9.3 Kcal/mol and an exothermicity of -62.8 Kcal/mol. Calculations with larger basis sets and MP2 geometries gave 7.4 and -66.2 Kcal/mol, respectively [234]. The experimental estimation gives a lower limit to the activation barrier of 3.3 Kcal/mol [234].

MOLCAS provides also a number of one-electron properties which can be useful to analyze the chemical behavior of the systems. For instance, the Mulliken population analysis is available for the RHF, CASSCF, CASPT2, MRCI, and ACPF wave functions. Mulliken charges are known to be strongly biased by the choice of the basis sets, nevertheless one can restrict the analysis to the relative charge differences during the course of the reaction to obtain a qualitative picture. We can use, for instance, the charge distribution obtained for the MRCI wave function, which is listed in Table 10.11. Take into account that the absolute values of the charges can vary with the change of basis set.

Table 10.11: Mulliken's population analysis (partial charges) for the reaction path from dimethylcarbene to propene. MRCI wave functions.

C_2^a	C_1^b	H_5^c	Σ^d	$H_1+H_3^e$	Me^f
Dimethylcarbene					
-0.12	-0.13	0.05	-0.20	0.14	0.07
Transition state structure					
-0.02	-0.23	0.05	-0.20	0.17	0.02
Propene					
-0.18	-0.02	0.05	-0.15	0.18	-0.02

[a] Carbon from which the hydrogen is withdrawn.

[b] Central carbenoid carbon.

[c] Migrating hydrogen.

[d] Sum of charges for centers C_2, C_1, and H_5.

[e] Sum of charges for the remaining hydrogens attached to C_2.

[f] Sum of charges for the spectator methyl group.

In dimethylcarbene both the medium and terminal carbons appear equally charged. During the migration of hydrogen H_5 charge flows from the hydrogen donating carbon, C_2, to the carbenoid center. For the second half of the reaction the charge flows back to the terminal carbon from the centered carbon, probably due to the effect of the π delocalization.

10.5 Excited states.

The accurate calculation of excited electronic states has been a challenge for quantum chemistry. The possibility for accurate calculations of such states in molecules has only recently been made possible through the development of new quantum chemical techniques. CASPT2 is currently one of the more successful methods to compute excited states due to its balance between accuracy and cost. In addition to the intrinsic limitations of the method, photochemistry and photophysics involves a large number of situations and mechanisms which complicate the problems enormously. In the present section we are going to show a systematic way to deal with a large number of states in a molecule. We have selected the thiophene molecule and our goal will be to compute the lowest valence and Rydberg singlet states at the ground state geometry. This can be considered to be the gas-phase absorption spectrum of the molecule. The calculations comprise an extensive use of the RASSCF, CASPT2, and RASSI programs. Selection of proper active spaces, building of appropriate diffuse basis functions, calculation of transition dipole moments, and use of the level-shift technique in CASPT2 will be some of the topics covered.

10.5.1 The vertical spectrum of thiophene.

Besides the usual limitation typical of any *ab initio* procedure due to the size of the system and the calculation of the integrals, the CASPT2 method has the basic limitation of the size and selection of the active space in the preliminary CASSCF step, not only because the space cannot be too large but because the active space defines the type and number of configurations (read excitations) to be included in the multiconfigurational wave functions. The near-degenerate configurations describing all states must be present in the reference wave function. Therefore, certain knowledge of the system is necessary to design the calculation and, for excited states, this will limit the number of states we are able to study.

Planning the calculations.

Thiophene is a planar five membered ring molecule containing one sulfur and four carbon atoms. The π structure of the system contains two conjugated double bonds between carbon atoms. Therefore, the orbital π valence structure is composed by two π bonding, two π^* antibonding orbitals, and one π nonbonding orbital placed on the sulfur atom. The π orbitals are the highest occupied ones in this type of systems and excitations from them form the UV spectrum in gas phase and solution. Also, typical orbitals involved in low-lying excited states are the lone-pair orbitals such as the sulfur n orbital co-planar with the σ skeleton of the molecule. On the other hand, σ orbitals forming C–H and C–C bonds do not participate in the low-lying excited electronic states. One has, however to be careful here. In thiophene there are low-lying virtual σ that give rise to excited states in the region around 6 eV [238].

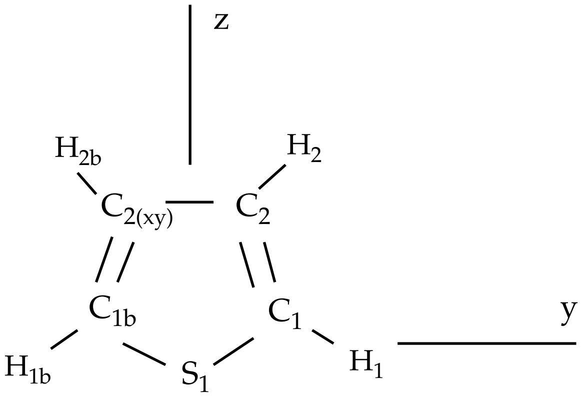


Figure 10.10: Thiophene

With this in mind we have to include at least the three π and two π^* valence orbitals and the valence σ lone-pair on the sulfur in the active space. The molecule belongs to the C_{2v} point group, therefore we have three b_1 and two a_2 π, π^* orbitals and one a_1 n orbital. That is, our minimal valence active space can be labeled (1302), where each number corresponds to the number of a_1,b_1, b_2, and a_2 orbitals, respectively.

But the valence states are not the only states present at low energies. In a gas-phase spectrum of a neutral molecule the Rydberg states start to appear at energies above 5 eV. Therefore, they must be simultaneously included in the calculations. The Rydberg orbitals are large compared to the molecular dimension and therefore have quasi atomic shapes. Rydberg states are commonly labeled as excited states of atoms with a principal quantum number n and the usual angular quantum numbers l and m. For molecules containing only first row atoms n conventionally starts with 3. This convention is actually used also in a molecule like thiophene, although in sulfur the valence electrons are in the third shell. Increasing the value of n will lead to more and more diffuse orbitals, eventually converging to an ionized state of the molecule. The lowest Rydberg state corresponds to the excitation HOMO→$3s$. The next components will be $3p_x$, $3p_y$, and $3p_z$, followed by the five components of $3d$.

The Rydberg orbitals classify into the point group like their corresponding atomic orbitals. Therefore, a look at the character table (see Table 10.2) indicates that in C_{2v} the s,p_z,d_{z^2}, and $d_{x^2-y^2}$ Rydberg orbitals belong to symmetry a_1, p_x and d_{xz} to symmetry b_1, p_y and d_{yz} to symmetry b_2 and, finally, d_{xy} to symmetry a_2. According to the labeling defined above the nine lowest Rydberg orbitals classify to (4221). It is obvious that we cannot normally afford to have simultaneously the whole valence plus Rydberg space (15 active orbitals in the present example). Therefore we are going to exploit the symmetry properties to select different active spaces.

By inspection of the SCF orbital energies or the ionization potentials of the molecule we observe that the highest occupied orbitals HOMO ($1a_2$) and HOMO-1 ($2b_1$) are reasonably close in energy (around 0.6 eV). Therefore, two Rydberg series close in energy can be expected at low energies, the first one arising from the HOMO orbital and the second from the HOMO-1 orbital. By exciting one electron from each of those orbitals to each one of the Rydberg orbitals we know the symmetry of the resulting state. For instance, the excitation HOMO (a_2) → 3s (a_1) leads to a A_2 by direct product of the symmetry representations. Table 10.12 contains the analysis for the Rydberg states arising both from HOMO and HOMO-1 orbitals to the n=3 Rydberg orbitals. They form the two lowest Rydberg series. We want also to locate the state from the lone-pair HOMO-2 ($11a_1$) to 3s.

The computed states will use different partitionings of the active space. The basic valence space (1302) must be included in all the cases. The valence $\pi \rightarrow \pi^*$ states only involve excitations into the π and π^* orbitals. Therefore they belong to the A_1 and B_2 symmetries. In addition we can have single excitations (Rydberg states) from the occupied π orbitals to the Rydberg orbitals of b_1 and a_2 symmetries. The number of Rydberg orbitals belonging to those symmetries is (0201). Thus, the final space to compute simultaneously valence and Rydberg $\pi \rightarrow \pi^*$ states is (1302) + (0201): (1503). The same space can be used to compute $n \rightarrow \pi^*$ states because the n orbital and the π^* orbitals are included into the active space. The symmetries of these states, however, will be A_2 and B_1. In the table we also have another division for the A_2 and B_1, $\pi \rightarrow$R(σ), and A_1, $n \rightarrow$R(σ), (only the $n \rightarrow 3s$) Rydberg states, using an active space (5322). We have, therefore, divided the excited states to be computed by symmetries and active space. State-average CASSCF calculations for each one of the cases have to be performed. The only question which remains is how many roots we have to include in each of the cases. This is also determined by the symmetry and active space available. For instance, for the $\pi \rightarrow \pi^*$ A_1 states, we want to compute the ground state plus three Rydberg states (see Table 10.12 in both HOMO and HOMO-1 → n=3 series)

Table 10.12: Selection of active spaces in thiophene.

	Symmetries			
	a_1	b_1	b_2	a_2
Frozen orb.	5	1	3	0
Inactive orb.	6	0	4	0
Valence active orb.	1	3	0	2

Rydberg states

HOMO→n=3	State	HOMO-1→n=3	State	HOMO-2→n=3	State[a]
(π) $a_2 \rightarrow$ $3sa_1$	A_2	(π) $b_1 \rightarrow$ $3sa_1$	B_1	(n) $a_1 \rightarrow$ $3sa_1$	A_1
$3pa_1$	A_2	$3pa_1$	B_1		
$3pb_1$	B_2	$3pb_1$	A_1		
$3pb_2$	B_1	$3pb_2$	A_2		
$3da_1$	A_2	$3da_1$	B_1		
$3da_1$	A_2	$3da_1$	B_1		
$3db_1$	B_2	$3db_1$	A_1		
$3db_2$	B_1	$3db_2$	A_2		
$3da_2$	A_1	$3da_2$	B_2		

Total active space

A_1, B_2 states ($\pi \rightarrow \pi^*$) $_1$, B_2 states ($\pi \rightarrow$R(π^*)) A_2, B_1 states ($n \rightarrow \pi^*$)	Valence (1302) + Rydberg (0201) = (1503)
A_2, B_1 states ($\pi \rightarrow$R(σ)) A_1 states ($n \rightarrow$R(σ))	Valence (1302) + Rydberg (4020) = (5322)

[a] Only considered up to the A_1 (3s) state because the remaining are expected at higher energy.

plus a certain number of valence states. If we do not have any previous experience we may think of three or four possible valence states but we know that the usual number of low-lying valence states is close to the number of valence singly excited states, in this case two of A_1 symmetry. This does not mean that the states are going to be described by one single configuration; it is simply an estimation of the number of relevant states based on experience. In summary, we expect to compute six A_1 states and therefore we include six roots in the CASSCF state-average input.

It is not uncommon that one or more valence states do not appear in the initial CASSCF calculation including the desired roots and other higher Rydberg states. This is due to the fact that valence states usually require larger dynamical correlation corrections than the Rydberg states. Therefore in a CASSCF calculation the Rydberg states are, in general, lower in energy than the valence states. The dynamical correlation included by the CASPT2 method will place the states correctly. However this is only possible if the states are present in the CASSCF calculation. It is then necessary to be sure that the states are located at the CASSCF level. Maybe it is necessary to increase the number of roots and in special cases like those with low symmetry even to delete Rydberg orbitals from the active space [22, 23, 229, 239].

In the following we will describe briefly the calculations [240]. A detailed report of the vertical excited spectrum of thiophene can be found in references [240, 241]. The selection of the active spaces in that work included additional orbitals to minimize the effect of intruder states. The availability of the level-shift technique in later versions of MOLCAS allow us to use a smaller active space.

Generating Rydberg basis functions

First we describe a method for generating Rydberg basis functions for molecules. Such Rydberg orbitals are diffuse and thus require diffuse basis functions. Due to this diffuseness they are not "localized" to atoms in the sense that valence orbitals are, but should be considered to be spread out over the entire molecule.

The basis of the method lies in the fact that if we add an electron into a virtual orbital, the energy for the system is increased by the orbital energy, according to Koopmanns' theorem. The reorganizational effects are very minor for the diffuse virtual orbitals. Thus adding an electron into a virtual orbital for a cation is an reasonable approximation to the proper Rydberg state. A more extensive discussion of the method outlined below can be found in [23].

The method can be broken down into a few steps (see Ref. [23] for details):

1. Perform a RHF or valence CASSCF calculation of the system with one electron removed, using the `RASSCF` program. This will determine the center of charge which is a suitable choice to center the Rydberg basis function expansion. The result is rather insensitive to this choice.

2. Add a suitable diffuse primitive basis set at the center of charge. We use as universal exponents those optimized by Kaufmann *et al.* [242] for Rydberg wave functions.

3. Repeat the RHF or CASSCF calculation in the new basis.

4. Construct the basis set using the program GENANO and use the lowest virtual function to define the basis set.

It is better not to use an extremely large valence basis set to perform these calculations. The best choice is a double-zeta or double-zeta plus polarization basis set. In this example we will use benzene which have a natural origin in the center of the ring. Thus we have eliminated the step of determining the center of charge. Also we have made the simplification of only considering s-functions.

The procedure we will follow is

1. Create inputs for `SEWARD`, `SCF`, `RASSCF`, and `GENANO`.
2. Create a shell script to run `SEWARD`, `SCF`, and `RASSCF`, and run the job.
3. Hand edit the resulting formated orbital file, *`C6H6.RasOrb`*. Set the occupation numbers for the occupied space to zero, while the first three virtual orbitals in the first irreducible representation get the occupation numbers 10^{-1}, 10^{-2} and 10^{-3} respectively. These occupation numbers are quite arbitrary as long as they form a decreasing sequence.
4. Create a shell script to run `GENANO` and run the job.
5. The resulting file *`C6H6.Ano`* now contains the contraction coefficients. Merge this file with the exponents in the `SEWARD` input to obtain the final contracted basis set. We normally use only one function of each type.

The radial extent of the resulting basis functions is shown in figure 10.11.

Here are the inputs used for this example. First the SEWARD input using the uncontracted Rydberg functions (note that only the s-type Rydberg basis is shown).

```
 &SEWARD &END
Title
 Benzene molecule.
Symmetry
X Y Z
*OneOnly
Basis set
C.ano-s...3s2p1d.
C1    2.636169     .000000     .000000
C2    1.318084    2.282990     .000000
End of basis
Basis set
H.ano-s...2s1p.
H1    4.684633     .000000     .000000
H2    2.342316    4.057011     .000000
End of basis
Basis set
X....8s8p8d. / Inline
  0.0 0
8 8
.02462393 .01125334 .00585838 .00334597 .00204842 .00132364 .00089310 .00062431
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0
```

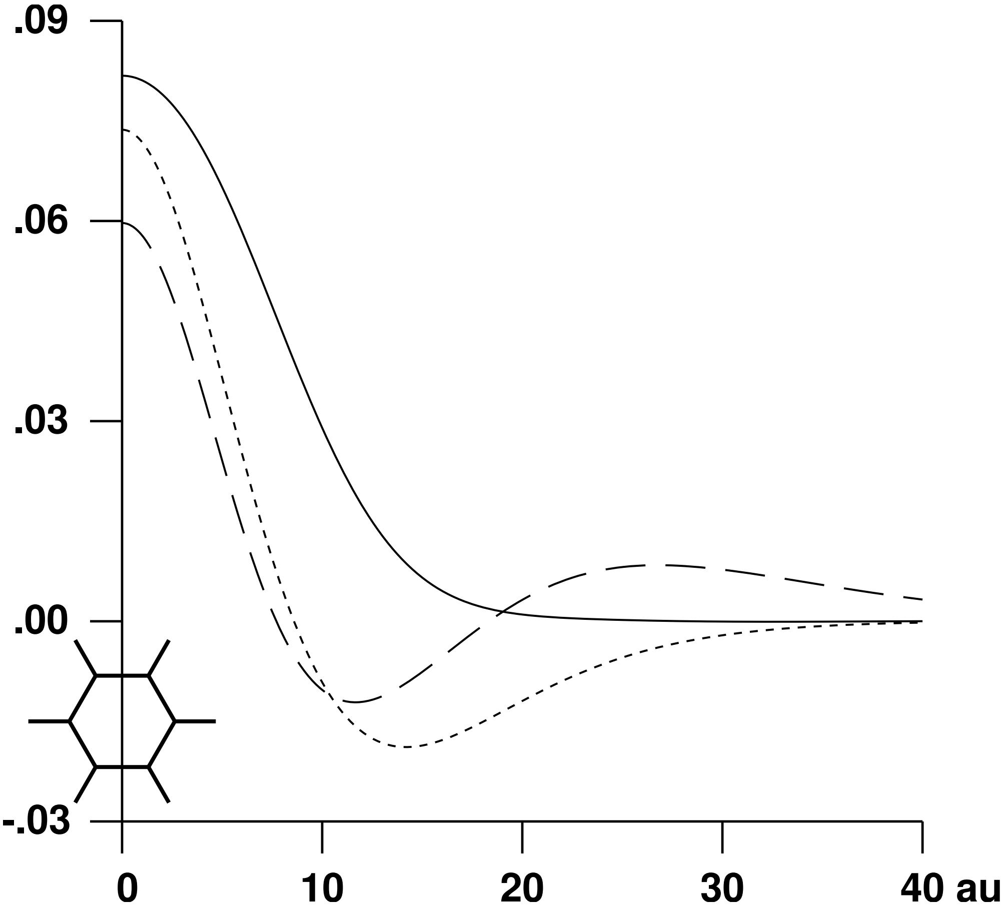


Figure 10.11: Radial extent of the Rydberg orbitals

```
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0
X     0.000000    0.000000     .000000
End of basis
End of input
```

Once computed, the contracted functions will replace the uncontracted ones. In the usual calculations we are going to use one function of each type, $1s1p1d$, but we can keep three of them if we want to increase the Rydberg basis for some particular use. Here is the input listing for the generation of the ANO. Note that in newer versions of MOLCAS the sequence of calculations is driven by the input list. You can skip parts of the calculation by commenting out (with a *) the corresponding namelist input (for example * &SEWARD &END skips the integral calculation).

```
 &SEWARD &END
Title
 Benzene molecule.
Symmetry
X Y Z
*OneOnly
Basis set
C.ano-s...3s2p1d.
C1    2.636169     .000000     .000000
C2    1.318084    2.282990     .000000
End of basis
Basis set
H.ano-s...2s1p.
H1    4.684633     .000000     .000000
H2    2.342316    4.057011     .000000
End of basis
Basis set
X....1s1p1d. / Inline
  0.0 0
8 1
.02462393 .01125334 .00585838 .00334597 .00204842 .00132364 .00089310 .00062431
    .15531366  -.26126804   .38654527
 -1.53362747 -1.27182240   .94560891
  1.10186802   .95250581 -1.24269525
 -1.70918216   .49632170 -2.22724281
  2.03031830   .68292933  1.94719179
 -1.73187442  -.56245782   .68883478
    .92694465   .30675927   .15138171
  -.22934028  -.07852136  -.02092438
X     0.000000    0.000000     .000000
End of basis
 &SCF &END
Title
 Benzene molecule.
Occupied
 6 5 4 3 1 1 1 0
End of input
 &RASSCF &END
Title
 Benzene molecule
Symmetry
 7
Spin
 2
nActEl
  1 0 0
Inactive
 6 5 4 3 1 1 0 0
Ras2
 0 0 0 0 0 0 1 0
```

```
LumOrb
Thrshld
0.5d-8 0.5d-4 1.0d-4
Iterations
 50 25
End of input
>>COPY $Project.RasOrb  NAT001
>>COPY $Project.OneInt  ONE001
>>COPY $Project.RunFile RUN001
 &GENANO &END
Title
 Rydberg basis set for benzene.
sets
 1
Center
X
Weights
 1.0
end of input
```

Here is the shell script used for this example. It is written in Korn shell, but no exotic features of Korn shell are used, so rewriting them into C shell, or whatever your favorite shell is, is a straightforward matter.

```
#!/bin/ksh
Project='C6H6'
Home=$PWD
WorkDir=/temp1/$LOGNAME/$Project
export Project WorkDir
print 'Start of job:' $Project
print 'Current directory:' $Home
print 'Scratch directory:' $WorkDir
#
trap 'exit' ERR
rm -fr $WorkDir
#
molcas  $Home/$Project.input >$Project.output
#
rm -r $WorkDir
```

For thiophene one can proceed in the same way. The only difference (apart from the fact that we generate s, p, d functions) is that two states of the cation are going to be computed and therefore the final step using the GENANO program will involve two files and have the following input:

```
!ln -s $Home/Thiophene.Ano     ANO
!ln -s $Home/Thiophene.RasOrb1  NAT001
!ln -s $Home/Thiophene.RasOrb2  NAT002
!ln -s Thiophene.OneInt        ONE001
!ln -s Thiophene.OneInt        ONE002
 &GENANO &END
Title
 Rydberg basis set for thiophene.
sets
 2
Center
X
Weights
 0.5 0.5
End of input
```

The charge centroid is chosen as an average of the charge centroids of the two cations.

SEWARD and CASSCF calculations.

Once we have built the diffuse basis set we can proceed with the SEWARD and CASSCF calculations of the different states. Remember that no quantitative result can be expected for calculations which use less than a DZP basis set. Additionally, as we are using methods which include large amounts of correlation, it is also recommended to use basis sets designed to include the correlation, such as the Dunning correlation-consistent basis sets or the Atomic Natural Orbital-type basis sets. Several tests of the accuracy of the ANO-type basis sets for calculations on excited states can be found elsewhere [243]. It was found that the minimum basis set suitable for calculations on excited states is the ANO $3s2p1d$ basis set for the first row atoms, with $2s$ functions for the hydrogen. The recommended basis however is an ANO $4s3p1d$ basis set.

We proceed with the calculations on thiophene. The inputs for the programs SEWARD, SCF, and RASSCF ($^1A_1$ states) are:

```
 &SEWARD &END
Title
Thiophene molecule. Experimental gas-phase geometry.
Symmetry
 X Y
Basis set
S.ANO-L...5s4p2d.
S1    0.000000  0.000000  0.000000  Bohr
End of basis
Basis set
C.ANO-L...4s3p1d.
C1    0.000000  2.333062  2.246725  Bohr
C2    0.000000  1.344416  4.639431  Bohr
End of basis
Basis set
H.ANO-L...2s1p.
H1    0.000000  4.288992  1.677364  Bohr
H2    0.000000  2.494694  6.327573  Bohr
End of basis
Basis set
X....1s1p1d / Inline
 0.0000000 2
*  s-type diffuse functions
    8    1
 .024624 .011253 .005858 .003346 .002048 .001324 .000893 .000624
  .38826283
-1.91720062
 1.70115553
-2.69265935
 3.15654806
-2.69329518
 1.44320084
 -.35712479
*  p-type diffuse functions
    8    1
 .042335 .019254 .009988 .005689 .003476 .002242 .001511 .001055
  .14713386
 -.64370136
 -.17112583
 -.62433766
  .58193247
 -.53426167
  .30777301
 -.08250038
*  d-type diffuse functions
    8    1
 .060540 .027446 .014204 .008077 .004927 .003175 .002137 .001491
  .24501363
  .04635428
  .66592833
 -.08963981
  .52211247
 -.32807746
  .18219220
 -.04616325
X               .0000000000         .0000000000         .1609268500
End of Basis
End of Input
 &SCF &END
Title
 Thiophene molecule
Occupied
11 1 7 3
Iterations
40
End of Input
 &RASSCF &END
Title
 Thiophene. pipi  1A1 states
Symmetry
    1
Spin
    1
Nactel
    8    0    0
Frozen
    4    1    3    0
Inactive
    6    0    4    0
Ras2
    1    5    0    3
CiRoot
6 6
1 2 3 4 5 6
1 1 1 1 1 1
Iter
50,25
LumOrb
End of Input
>> COPY $Project.JobIph $CurrDir/$Project.1A1.JobIph
*$
```

The last line will copy the current JOBIPH file to a file in the directory where the job was submitted.

The wave function and natural occupation numbers obtained for the $^1A_1$ states are:

```
                                 Wave function printout:
 occupation of active orbitals, and spin coupling of open shells (u,d: Spin up or down)

 printout of CI-coefficients larger than  0.38 for root  1
       energy= -551.412548
 conf/sym 1 22222 444 Coeff  Weight
            11 2 22000 200   0.95720 0.91624

 printout of CI-coefficients larger than  0.38 for root  2
 energy= -551.192455
 conf/sym 1 22222 444 Coeff  Weight
            14 2 22000 u0d   0.38522 0.14839
            20 2 2ud00 200   0.68777 0.47302
```

```
printout of CI-coefficients larger than  0.38 for root  3
energy= -551.178212
conf/sym 1 22222 444 Coeff  Weight
            85 2 2u0d0 200   0.74016 0.54783
    86 2 2u00d 200   0.46282 0.21421

printout of CI-coefficients larger than  0.38 for root  4
energy= -551.155996
conf/sym 1 22222 444 Coeff  Weight
            12 2 22000 ud0   0.49009 0.24019
            14 2 22000 u0d   0.72977 0.53257

printout of CI-coefficients larger than  0.38 for root  5
energy= -551.151801
conf/sym 1 22222 444 Coeff  Weight
    85 2 2u0d0 200  -0.48463 0.23486
    86 2 2u00d 200   0.78218 0.61180

printout of CI-coefficients larger than  0.38 for root  6
energy= -551.106218
conf/sym 1 22222 444 Coeff  Weight
     1 2 22200 000  -0.50027 0.25027
    20 2 2ud00 200  -0.49511 0.24514
    29 2 u2d00 200   0.46904 0.22000

Natural orbitals and occupation numbers for root 1
sym 1:   1.999604
sym 2:   1.991918  1.943992  0.097398  0.000219  0.000640
sym 4:   1.904095  0.061524  0.000611
Natural orbitals and occupation numbers for root 2
sym 1:   1.999436
sym 2:   1.947529  1.248261  0.788864  0.028171  0.000731
sym 4:   1.617765  0.032985  0.336259
Natural orbitals and occupation numbers for root 3
sym 1:   1.999273
sym 2:   1.926567  1.085938  0.128802  0.904415  0.000774
sym 4:   1.805386  0.141116  0.007730
Natural orbitals and occupation numbers for root 4
sym 1:   1.999591
sym 2:   1.938931  1.828828  0.185815  0.001667  0.027931
sym 4:   1.100050  0.074750  0.842438
Natural orbitals and occupation numbers for root 5
sym 1:   1.999251
sym 2:   1.935074  1.086440  0.103317  0.001139  0.911640
sym 4:   1.854839  0.074961  0.033340
Natural orbitals and occupation numbers for root 6
sym 1:   1.999766
sym 2:   1.874358  1.484874  1.099307  0.004906  0.008790
sym 4:   1.285113  0.235809  0.007076
```

We have only included the configurations with weights larger than 10%. Root one corresponds to the closed-shell ground state. To understand the character of the states one must also analyze the orbitals, remembering that the active orbitals are not ordered within the active space.

The following output shows the coefficients of the diffuse functions (center X) which appear in the MOLCAS output. Active orbitals two, three, and six in symmetry 2 are valence orbitals (they have main contributions from the other functions not printed here) and orbitals four and five are Rydberg orbitals. It is usual that they appear as mixed orbitals ($3p$-$3d$ here) but this mixing has no consequences on the excitation energies. This is also the reason why the Rydberg states appear not as clearly singly configurational states but mixed as in root 5 (see above).

```
    Molecular orbitals for symmetry species 2

    ORBITAL       2        3        4        5        6
    ENERGY      .0000    .0000    .0000    .0000    .0000
    OCC. NO.   1.8923   1.4570    .4122    .1674    .1689

 19 X  2px     -.0203    .0055   -.0082   .8091    .4535
 20 X  3d1+     .0064   -.0037  .0369    .4430  -1.0132

    Molecular orbitals for symmetry species 4

    ORBITAL       1        2        3
    ENERGY      .0000    .0000    .0000
    OCC. NO.   1.5865    .1722    .1439

 15 X  3d2-     .0032    .5171    .9600
```

Both by looking at the configurations and the occupation numbers we can identify the states. Root two has a main configuration described by an excitation 3b$_1$ → 4b$_1$. As 4b$_1$ is a valence orbital, the resulting state will also be a valence state. Root three, on the contrary, has a main configuration 3b$_1$ → 5b$_1$, and 5b$_1$ is a Rydberg orbital. 3b$_1$ is the HOMO-1 orbital, therefore we can expect the state represented by root three to be the HOMO-1→3p_x Rydberg state. So, why does configuration 3b$_1$ → 5b$_1$ contribute 21% to this wave function if a Rydberg state is just a singly excited state?. The answer is in the composition of the orbitals. Orbitals four and five are a mixture of p_x and d_{xz}, and the configurational description must reflect that.

In summary we can make a initial classification of the states:

Root 1: Ground state
Root 2: Valence $\pi \rightarrow \pi^*$ state
Root 3: Rydberg 3b$_1$→3p_x state
Root 4: Rydberg 3a$_2$→3d_{xy} state
Root 5: Rydberg 3b$_1$→3d_{xz} state
Root 6: Valence $\pi \rightarrow \pi^*$ state

Orbital two of symmetry 4 also deserves attention. It has large contributions from the diffuse functions, although the remaining non-printed coefficients are even larger. It is an orbital of mixed valence-Rydberg character. This can affect the description of the valence states. In the present system the problem is minor because the orbital does not strongly participate in the description of the valence states as it is shown by the configurations and the occupation numbers, but in other systems the effect is going to be larger as we shall show later.

One important difference between valence and Rydberg states is the diffuse character of the latter. We can analyze the orbital extension of the states. Valence states have an orbital extension (second Cartesian moment) similar to the ground state extension. Rydberg states, on the contrary, should have a diffuse character. Additionally we can also study the Mulliken population analysis. Both appear in the RASSCF output.

```
        Mulliken population Analysis for root number: 1

        Gross atomic populations per centre and basis function type

                  S1      C1      C2      H1      H2      X
        Total 15.8153 12.3470 12.2660  1.6887  1.8021   .0809

        Expectation values of various properties for root number: 1
```

```
 2-nd Cartesian moments: origin at (   .00000000,   .00000000,  2.15947162)
----------------------------------------------------------------------------
Component                           XX              YY              ZZ
Total                         -30.24626427    -21.54920631    -24.73702724

     Mulliken population Analysis for root number: 2

     Gross atomic populations per centre and basis function type

              S1      C1      C2      H1      H2      X
     Total 15.6548 12.3730 12.1962  1.6914  1.8015   .2831
     Expectation values of various properties for root number: 2

 2-nd cartesian moments: origin at (   .00000000,   .00000000,  2.15947162)
----------------------------------------------------------------------------
Component                           XX              YY              ZZ
Total                         -42.75835009    -28.13902538    -28.72863222

     Mulliken population Analysis for root number: 4

     Gross atomic populations per centre and basis function type

              S1      C1      C2      H1      H2      X
     3d2-    .0334   .0306   .0413   .0000   .0000   .9662
     Total 15.5924 11.8522 12.0083  1.6814  1.7986  1.0671
     Expectation values of various properties for root number: 4

 2-nd cartesian moments: origin at (   .00000000,   .00000000,  2.15947162)
----------------------------------------------------------------------------
Component                           XX              YY              ZZ
Total                         -89.85913318    -76.33249740    -44.45493589

     Mulliken population Analysis for root number: 6

     Gross atomic populations per centre and basis function type

              S1      C1      C2      H1      H2      X
     Total 15.6154 12.4779 12.3182  1.6946  1.8028   .0911

     Expectation values of various properties for root number: 6

 2-nd cartesian moments: origin at (   .00000000,   .00000000,  2.15947162)
----------------------------------------------------------------------------
Component                           XX              YY              ZZ
Total                         -31.85163136    -24.13169375    -26.69322385
```

The Mulliken analysis provides us with the charge distribution per atom and basis function. If we have used for the Rydberg states singly centered Rydberg functions we can observe a population close to one on the X center. This is what happened in root four (see above). In addition we can see that the electron is placed in the 3d2- ($3d_{xy}$) Rydberg orbital, confirming the character of the state. The orbital extension is undoubtedly much larger in the fourth root than in the ground state. The second and sixth roots however have a much more compact description, especially the sixth, and they have low populations on center X. The second root is somewhat more diffuse but it can be still considered a clear valence state with minor Rydberg mixing.

It is very important to ensure that the relevant states of the symmetry are included in the CASSCF calculation. This may mean performing different experiments by increasing the number of roots and analyzing the results. Valence states are specially sensitive to this because they are high roots at the CASSCF level. Take for instance the sixth root. At the

CASSCF level, it is 1.35 eV higher in energy than its preceding root. It could happen that other close Rydberg states or even valence states (such as mainly doubly excited states) were lower at this level of calculation. It can be also helpful to analyze the transition moment to be sure that the intense valence states are present in the set of computed states.

The RASSCF inputs for the remaining states replace the following keywords:

```
 &RASSCF
Title
 Thiophene. pipi  1B2 states
Symmetry
    3
CiRoot
5 5
1 2 3 4 5
1 1 1 1 1
...
End of Input
>> COPY $Project.JobIph $CurrDir/$Project.1B2.JobIph
*$


 &RASSCF
Title
 Thiophene. npi  1B1 states
Symmetry
    2
CiRoot
1 1
1
...
End of Input
>> COPY $Project.JobIph $CurrDir/$Project.1B1n.JobIph
*$


 &RASSCF &END
Title
 Thiophene. npi  1A2 states
Symmetry
    4
CiRoot
2 2
1 2
1 1
...
End of Input
>> COPY $Project.JobIph $CurrDir/$Project.1A2n.JobIph
*$


 &RASSCF &END
Title
 Thiophene. pisigma  1B1 states
Symmetry
    2
Ras2
    5    3    2    2
CiRoot
6 6
1 2 3 4 5 6
1 1 1 1 1 1
...
End of Input
>> COPY $Project.JobIph $CurrDir/$Project.1B1.JobIph
```

```
*$


 &RASSCF &END
Title
 Thiophene. pisigma  1A2 states
Symmetry
    4
Ras2
    5    3    2    2
CiRoot
6 6
1 2 3 4 5 6
1 1 1 1 1 1
...
End of Input
>> COPY $Project.JobIph $CurrDir/$Project.1A2.JobIph
*$


 &RASSCF &END
Title
 Thiophene. nsigma  1A1 states
Symmetry
    1
Ras2
    5    3    2    2
CiRoot
4 4
1 2 3 4
1 1 1 1
...
End of Input
>> COPY $Project.JobIph $CurrDir/$Project.1A1n.JobIph
*$
```

and use the saved *JOBIPH* files subsequently.

We must ensure that the right orbitals are included into the active space. For instance, computing the ^{1}A$_2$ and ^{1}B$_1$ Rydberg states with the active space (5322) we observe that one Rydberg orbital is absent from the active space in both cases. For the ^{1}A$_2$ state it was orbital $3d_{yz}$. Instead, an extra-valence σ^* orbital took its place and therefore the sixth root of symmetry ^{1}A$_2$ was not the expected 2b$_1$ → $3d_{yz}$ Rydberg state. In this case we can reorder the orbitals including the Rydberg state in the active space and excluding the other orbital and make the calculation again. Hopefully the new calculation will include the Rydberg state into the selected roots. If not we can always increase the number of roots or increase the active space to have both orbitals included.

It is very important to remember that to compute energy differences one must always use states computed using the same active space. Therefore, if we are computing vertical excitation energies we must have the ground state energy computed in all the different active spaces employed. One can make the comparison using a ground state computed in the average procedure or as a single root. They do not differ significantly. For consistency, we will use a ground state computed as a single root. Therefore we have to perform two CASSCF calculations using the inputs where we replace:

```
>> COPY $CurrDir/$Project.11A1.JobIph JOBIPH
 &RASSCF &END
Title
 Thiophene. Ground state (1503)
```

```
Symmetry
    1
Ras2
    1    5    0    3
CiRoot
1 1
1
>> COPY $CurrDir/$Project.11Ar.JobIph JOBIPH
 &RASSCF &END
Title
 Thiophene. Ground state (5322)
Symmetry
    1
Ras2
    5    3    2    2
CiRoot
1 1
1
```

CASPT2 calculations.

Once the reference wave functions have been computed at the CASSCF level we can perform the CASPT2 calculations. The *JOBIPH* file from each CASSCF calculation contains data that describes the state(s). If several CASSCF states are present on a *JOBIPH* file, then any of this may act as root function for the CASPT2. The input to the CASPT2 must then tell which one of the states we want. In previous MOLCAS version the keyword LROOT was used. Although it will still work, it has been substituted by the more convenient keyword MULTISTATE, which allows now to perform Multi-State CASPT2 calculations. We will start by discussing single state CASPT2 calculations:

```
 &CASPT2 &END
Title
 caspt2 input
MultiState
1 1
End of input
```

The CASPT2 calculation will be performed on the ground state with the active space (1305), stored on the *JOBIPH* file that we named *$Project.11A1.JobIph*. The final full CASPT2 result is:

```
      Reference energy:        -551.4423376617
      E2 (Non-variational):       -.6341237973
      E2 (Variational):           -.6341237319
      Total energy:            -552.0764613935
      Residual norm:               .0000008080
      Reference weight:           .80657
```

For a perfectly converged result, the two formulae used to compute E2 are equivalent, but if there are (as is usually the case) a small residual error in the CASPT2 equation system, then the variational result is much more accurate. In particular, for numerical differentiation the variational energy should always be used. If a level shift has been used, in order to avoid singularities (see below), then the non-variational energy and the variational one will differ. The former is the conventional E2 as obtained with the modified (shifted) $\hat{H}_0$ operator, while the latter is a corrected value very close to what would have been obtained with the unshifted operator if the near-singular term had been removed. The latter energy is the one that should normally be used.

For the ground state with a reasonable active space, all coefficients in the first order wave function and all contributions to the second-order energy will be small. For excited states, large contributions may occur, and then the second-order perturbation treatment may be invalid. One criterion for a good calculation is that the reference weight should be close to that of the ground state. When this is not true, special remedies may be considered. For example, we compute the CASPT2 correction for the sixth root of symmetry one, using the *JOBIPH* file called $Project.1A1.JobIph. The input is:

```
 &CASPT2 &END
Title
 caspt2 input
MultiState
1 6
End of input
```

and the result (always full CASPT2 results):

```
      Reference energy:        -551.1062184006
      E2 (Non-variational):       -.7460718503
      E2 (Variational):           -.7460719607
      Total energy:            -551.8520232128
      Residual norm:               .0000009146
      Reference weight:            .29470
```

We observe a low weight of 0.295 for the CASSCF reference, compared to the value 0.807 in the ground state. The low weight for the excited state is a warning sign: the second order treatment may be invalid. However, if so, the problem is due to one or a few specific terms in the first-order wave function.

In the output, there is a section with warnings for large contributions to the energy, low denominator values, or large coefficients.

```
CASE  SYM   ACT IND   NON-ACT INDICES  DENOMINATOR  RHS value  COEFFICIENT CONTRIBUTION

ATVX   2   Mu2.0001   Se2.007            .01778941  -.00706261   .72136097  -.00509469
ATVX   2   Mu2.0001   Se2.009            .20859986   .03118841  -.14372642  -.00448260
ATVX   4   Mu4.0001   Se4.004            .02156184  -.01357269  1.20409651  -.01634282
AIVX   1   Mu1.0001   In1.010 Se1.014    .08105563   .00023689  -.00197645  -.00000047
AIVX   1   Mu1.0001   In3.007 Se3.012    .28275882  -.02231776   .08282960  -.00184857
```

In CASPT2, the wave operator is a sum of two-electron excitations, $\sum C_{pqrs}\hat{E}_{pqrs}$, where the singlet excitation operator $\hat{E}_{pqrs}$ is normal-ordered and summed over spin. The electrons are transferred from s to r and from q to p.

No one-electron excitations are used. This is not due to any approximation; it is simply because, for a RASSCF root function with active electrons, the single excitations are exact linear combinations of the double excitations.

The non-orthogonality, as well as the non-diagonal terms of the $\hat{H}_0$, makes it difficult (and to some extent irrelevant) to obtain a label that partitions the wave function and correlation energy in terms of orbital indices of elementary excitations. However, the CASPT2 program uses internally an orbital system that diagonalizes part of the Fock matrix: the block diagonal part which does not include coupling between inactive, active and virtual orbitals. The first-order wave function, or equivalently the first-order wave operator, can be subdivided into terms that are grouped into eight different cases. These are named by four-letter combinations as follows. The letters A, B,C or D are used for secondary (virtual) orbitals; T, U, V, or X for active ones, and I, J, K or L for inactive orbitals. A case such as ATVX contains wave operator terms that can be written as $\hat{E}_{atvx}$, where a is a virtual orbital and t, v, and x are active.

The first-order wave function can be subdivided into individual terms labeled by the case (e.g. ATVX), the individual non-active orbital indices, and an active superindex that labels a linear combination of terms with different active orbital indices. The linear combination will 'mix' all active indices or index combinations within the case (with symmetry restrictions, if any) in such a way that *the individual terms that are used internally in the CASPT2 programs are orthogonal, and they diagonalize the block-diagonal part of* $\hat{H}_0$.

Of course, the complete $\hat{H}_0$ is used to solve the CASPT2 equations, which is why an iterative procedure is needed. However, in the diagnostic output above, the "DENOMINATOR" value is that of the resolvent of the block-diagonal part of $\hat{H}_0$. However, for diagnostics, this is a good approximation. (That it is not exact only shows by the fact that singularities in the energy do not occur exactly when the "DENOMINATOR" reported is equal to 0.)

The orbitals are labeled by the symmetry type, a period, and then the ordering number within that symmetry type. However, for clarity, it also is prefixed by the letters "Fr", "In", "Ac", "Se" or "De" for frozen (uncorrelated), inactive, active, secondary, and deleted orbitals. In the wave operator, the only possible orbital labels are "In" and "Se". The active superindex is given in formulae as μ, ν, etc so it is given a prefix "Mu".

Most of the cases are further subdivided into a plus and a minus linear combination making altogether 13 cases. Thus, the BVAT case is subdivided into BVATP and BVATM, containing terms of the type $\hat{E}_{bvat} \pm \hat{E}_{avbt}$, respectively. This has nothing to do with spin. It offers some technical advantages in the equation solution.

Table 10.13: Labeling for the configurations in caspt2.

Config.	Excitation 1	Excitation 2
VJTU	Inactive (J) → Active (V)	Active (U) → Active (T)
VJTIP	Inactive (J) → Active (V)	Inactive (I) → Active (T)
VJTIM	Inactive (J) → Active (V)	Inactive (I) → Active (T)
ATVX	Active (T) → Secondary (A)	Active (X) → Active (V)
AIVX	Inactive (I) → Secondary (A)	Active (X) → Active (V)
or:	Active (X) → Secondary (A)	Inactive (I) → Active (V)
VJAIP	Inactive (J) → Active (V)	Inactive (I) → Secondary (A)
VJAIM	Inactive (J) → Active (V)	Inactive (I) → Secondary (A)
BVATP	Active (V) → Secondary (B)	Active (T) → Secondary (A)
BVATM	Active (V) → Secondary (B)	Active (T) → Secondary (A)
BJATP	Inactive (J) → Secondary (B)	Active (T) → Secondary (A)
BJATM	Inactive (J) → Secondary (B)	Active (T) → Secondary (A)
BJAIP	Inactive (J) → Secondary (B)	Inactive (I) → Secondary (A)
BJAIM	Inactive (J) → Secondary (B)	Inactive (I) → Secondary (A)

For more details see Refs. [18, 19, 244]

The first configuration shown in the thiophene output involves the excitation from the active space to the secondary orbital, which is orbital nr seven of symmetry two (Se2.007). The denominator value for this configuration is close to zero (0.01778941). This is an energy difference, in the $\hat{H}_0$ approximation. Thus the root state, and some eigenstate of $\hat{H}_0$ in the interacting space, have almost the same energy value.

Such states, that were not included in the CASSCF configuration interaction but have energies within the range of the lowest CAS states, cause frequent problems in excited state calculations, since they often give small denominators and even, at particular geometries, singularities. We call these states intruders, by analogy to a similar phenomenon in multistate perturbation theory. A calculation of excited states by means of a perturbation theory based on an active space has to deal with the problem of intruder states. This is especially common when large and diffuse basis sets, such as the Rydberg functions, are included in the calculations.

In this example, the coefficient to the first order wave function is large (0.72136094). So is the contribution to the second order energy (-0.00509469 H), -0.14 eV. Even worse is the situation for the third term printed involving the fourth orbital (secondary) of symmetry four with an energy contribution of 0.44 eV. The analysis of the secondary orbitals $7b_1$ and $4a_2$ (they are the first virtual orbital of their symmetry) indicates that they are extremely diffuse orbitals with large Rydberg character. Remember that the subspaces we are using are: frozen (4130), inactive (6040), and active (1503).

This is not the case in the other configurations shown. First we have other ATVX terms including the excitation to the secondary orbital Se2.009. Also we have an AIVX term, involving the excitation from inactive In3.007 to secondary Se3.012. Their contributions to the second order energy, -0.00448260 and -0.00184857, respectively, are not caused by accidental near degeneracies in the value of the denominator. The orbitals involved are not of Rydberg character either. We have finally included as an example the excitation AIVX involving the excitation from In1.010 to Se1.014. Although it has a small value for the denominator, its contribution to the second order energy is very small and therefore it does not represent an important problem.

Intruders can be eliminated by including sufficiently many orbitals in the active space. When this is a reasonable alternative, it is the preferred solution. Limitations in the number of active orbitals can make this approach impractical. However, especially when intruders have clear Rydberg character, their effect on the second-order energy is often small, except perhaps in a small range of geometries around a singularity due to accidental degeneracy. In this common situation, two other remedies are available: shifting the $\hat{H}_0$ Hamiltonian, or deleting virtual orbitals. These remedies will be described in some detail in the following.

In order to obtain continuous potential energy functions, one cannot use a case-by-case approach, such as deleting an orbital. However, the $\hat{H}_0$ can be modified in such a way as to eliminate weak singularities. A well-tested method is a level-shift technique called LS-CASPT2[23, 27]. A constant parameter is added to the external part of the zeroth-order Hamiltonian. Any denominator close to zero is thus shifted away from zero, and does not produce any singular term. Of course, in a worst-case scenario, it might happen that some other denominator, previously non-zero, is shifted to come close to zero. In general, it is the higher excited states, in combination with large diffuse basis sets and exploration of a large range of geometries, that is the greatest risk for troublesome intruders.

There is also a new, less tried technique, called the imaginary shift method [29]. Here, the use of an imaginary shift value (but taking the real part of the computed correlation energy) offers some advantage, since an imaginary shift cannot introduce new singularities.

With either of the level shift methods, the (2nd order) correlation energy E_2 and the (1st order) wave function will depend on the level shift used. A correction of therefore applied, whereby in practice this dependence is made small, except of course for the spurious term that has disappeared. The corrected energy is in fact computed by using Hylleraas' 2nd-order variational formula to evaluate E_2, with the *unshifted* $\hat{H}_0$,

$$E_2 = 2\langle\Psi_1|\hat{H}|\Psi_0\rangle + \langle\Psi_1|\hat{H}_0|\Psi_1\rangle \tag{10.3}$$

which we call the *variational* E_2 in the output listing.

To minimize the effect on relative energies, we recommend that the same level shift is used for all states and geometries, if possible. This may require some experimenting. A criterion on absence of disturbing intruders is that the weight of the reference wave function should be roughly the same in all calculations. Without shift, a difference of up to 10% between the weights of the ground and an excited state can be acceptable (that is, the excitation energy is accurate enough) in a CASPT2 calculation without level shift. Using level shift, this should be adjusted to find a better match of reference weights. A detailed explanation of how to use the level-shift technique has been published [28]. Here we will simply summarize the main aspects.

Using the same *JOBIPH* file as before we perform a new CASPT2 calculation using the input:

```
 &CASPT2 &END
Title
 caspt2 input
MultiState
1 6
Shift
0.1
End of input
```

A level-shift of 0.1 Hartree has been introduced as a separation of the eigenvalues of the zeroth-order Hamiltonian. The final energy is then corrected, and the result is:

```
      Reference energy:        -551.1062184006
      E2 (Non-variational):       -.6921992859
      Shift correction:           -.0334372801
      E2 (Variational):           -.7256365659
      Total energy:            -551.8315878181
      Residual norm:               .0000003986
      Reference weight:            .74942

CASE  SYM   ACT IND   NON-ACT INDICES  DENOMINATOR  RHS value  COEFFICIENT CONTRIBUTION

ATVX   2   Mu2.0001   Se2.007            .01778941  -.00706261   .06072347  -.00042887
ATVX   2   Mu2.0001   Se2.009            .20859986   .03118841  -.09700134  -.00302532
ATVX   4   Mu4.0001   Se4.004            .02156184  -.01357269   .11838970  -.00160687
AIVX   1   Mu1.0001   In3.007 Se3.012    .28275882  -.02231776   .05918658  -.00132091
```

Several details come to our attention. Firstly, the final CASPT2 energy is higher than the result with level-shift 0.0. This is because the introduction of the parameter decreases the amount of dynamical correlation included. Secondly, the weight of the reference function has increased greatly, from 0.29 to 0.74, meaning that the most important intruder states have been removed from the treatment. Finally, we can observe the new contributions of the printed configurations to the second order energy. Configurations involving excitations to the $7b_1$ and $4a_2$ orbitals have drastically decreased their contributions, proving that the previous contributions were due to degeneracies in the denominators. However, the other two configurations remain almost as they were before, only slightly decreasing their contributions.

Now we use a value for the level-shift parameter of 0.2 Hartree:

```
      Reference energy:        -551.1062184006
      E2 (Non-variational):       -.6619040669
      Shift correction:           -.0557159229
      E2 (Variational):           -.7176199898
      Total energy:            -551.8235712419
      Residual norm:               .0000009298
      Reference weight:            .78212

 CASE  SYM   ACT IND   NON-ACT INDICES  DENOMINATOR  RHS value  COEFFICIENT CONTRIBUTION

 ATVX   2   Mu2.0001   Se2.007            .01778941  -.00706261   .03193515  -.00022555
 ATVX   2   Mu2.0001   Se2.009            .20859986   .03118841  -.07304944  -.00227830
 ATVX   4   Mu4.0001   Se4.004            .02156184  -.01357269   .06238180  -.00084669
 AIVX   1   Mu1.0001   In3.007 Se3.012    .28275882  -.02231776   .04673419  -.00104300
```

The observed tendencies are maintained. Finally, a value of 0.3 Hartree:

```
      Reference energy:           -551.1062184006
      E2 (Non-variational):          -.6347955450
      Shift correction:              -.0735679820
      E2 (Variational):              -.7083635270
      Total energy:               -551.8145819276
      Residual norm:                  .0000006328
      Reference weight:               .80307

 CASE  SYM   ACT IND   NON-ACT INDICES  DENOMINATOR  RHS value  COEFFICIENT CONTRIBUTION

 ATVX   2   Mu2.0001   Se2.007            .01778941  -.00706261   .02173413  -.00015350
 ATVX   2   Mu2.0001   Se2.009            .20859986   .03118841  -.05865340  -.00182931
 ATVX   4   Mu4.0001   Se4.004            .02156184  -.01357269   .04240583  -.00057556
 AIVX   1   Mu1.0001   In3.007 Se3.012    .28275882  -.02231776   .03862959  -.00086213
```

The contributions to the energy are much lower for each increase of the parameter, but we must never forget that we are loosing dynamical correlation with the increase of the level-shift factor. In a calculation of excitation energies that means that the resulting excitation energies become larger each time (dynamical correlation is larger in the excited state). Therefore, the level-shift parameter must be set to the lowest possible value which solves the intruder state problems. In practice it is then convenient to scan all the valence states for several values of the parameter and look for two factors:

- Reference weight as close as possible to the ground state reference weight with the same level shift parameter (LS).
- Excitation energies (ES) as stable as possible with the increment of the level-shift parameter (LS).

We now compute the ground state (GS) also for the level-shift values of 0.1, 0.2, and 0.3, and compare the excitation energies ΔE (always between states computed with the same parameter):

Table 10.14: Excitation energies and reference weights of thiophene for different level shift values.

LS (H)	ΔE (eV)	weight GS	weight ES
0.0	6.11	0.81	0.29
0.1	6.64	0.82	0.75
0.2	6.79	0.83	0.78
0.3	6.89	0.84	0.80

After checking the remaining states we conclude that a level shift of 0.1 Hartree is enough for our purposes. However the results seem to be too unstable with respect to the increase of the level-shift parameter. As our active space only comprises nine orbitals, we can consider the possibility of increasing it by including two more active orbitals in symmetries b_1 and a_2. In this way we minimize the intruder states problems in the best way, by introducing extra (not diffuse hopefully) orbitals. This will increase the accuracy.

The introduction of a (real) level-shift parameter does not automatically remove intruder state problems. It happens that a shift leads to more severe problems that those observed without level-shift. Examples and further explanations are given in e.g. ref. [28]. In such a case is may be possible to find a range of level-shift values where none of the computed states present intruder state problems. In a few cases we have found it necessary to use a shift larger than 0.3 Hartree. Another solution is to try an imaginary shift. This option has not been extensively investigated yet.

Consider a situation like the following:

```
CASE  SYM   ACT IND    NON-ACT INDICES  DENOMINATOR  RHS value  COEFFICIENT CONTRIBUTION

ATVX   2   Mu2.0001    Se2.004           -.30281661  -.00194108  -.37224517   .00072256
```

This is a calculation performed using level shift of 0.3 H. (The approximate denominator printed in the listing is that *without* the added shift). We have added the level shift to solve intruder states problem in other states, but we should use the same technique for all the computed states for consistency reasons (of course always using a ground state computed with the same level shift value). We find, however, that the weight of the CASSCF reference function is lower in the case with level shift 0.3 H (0.61) than in the case without level shift (0.69). In this state we have a denominator with a value close to -0.3 H. As the level shift we apply is a positive quantity (0.3 H) added to this denominator, we have created a problem by decreasing the denominator to a value close to zero. The coefficient of the configuration increases, which is reflected in the contributions to the second-order energy. Therefore, before applying any level shift, it is wise to check the values of the most important denominators to see if any of them is going to be close to the value of the applied level shift. In those situations we should set the level shift to another value. Sometimes the consequences for the final energy are small (here for instance) but this is not always the case (see ref. [28]).

It is also possible to delete virtual orbitals. This is occasionally used, e.g. when using other types of basis sets than ANO's, in order to delete virtual orbitals that are core-correlating. The procedure to do that is to take an orbital file, such as that produced by SCF or RASSCF, and edit it by hand and then using it as *INPORB* file in the RASSCF step. The orbitals one wants to delete are placed at the end of their symmetry group, and the keyword DELETED in used the RASSCF input, indicating how many orbitals are going to be deleted by symmetry. The program will ignore the deleted orbitals, both in RASSCF and the subsequent CASPT2 steps. To obtain accurate energy differences it is necessary to use the same set of initial orbitals and recompute the ground state (or the state one is comparing with) with the same number of deleted orbitals.

When the above scheme is used in order to try to eliminate intruders in CASPT2, the best way is if the *INPORB* can be prepared from the CASPT2 calculation where the intruder problem occurred.

For that calculation, the natural orbital analysis that follows the CASPT2 calculation shows up a virtual orbital with abnormally large occupation number and diffuse character. Use an editor to move this orbital to the end of the orbital file, and use it as *INPORB*. When the calculation is repeated, intruders with this orbital heavily populated have been eliminated. Occasionally, several orbitals need to be removed.

The deletion of virtual orbitals works best at single-geometry calculations, such as obtaining the vertical electronic spectrum.

Let us focus on the Multi-State CASPT2 type of calculations. The original reference [12] should be carefully read before using the method. This multidimensional perturbative approach considers the coupling of a number of CASPT2 states, a condition which is crucial to solve certain problems such as adiabatic crossing among states, strong valence-Rydberg situations, etc. The treatment is performed for a number of roots of the same symmetry provided they originate from a previous State-Average CASSCF calculation, that is, the CASPT2 program will use the binary *JOBIPH* file from a previous SA-CASSCF calculation, for instance, the six roots $^1A_1$ CASSCF calculation in thiophene. The corresponding CASPT2 input to treat simultaneously the six states will be:

```
 &CASPT2 &END
Title
 mscaspt2 input
MultiState
6 1 2 3 4 5 6
Shift
0.3
End of input
```

A level shift parameter of 0.3 au has been selected for comparison with the previous calculations. The program creates a new binary file, *JOBMIX*, which contains the newly generated Perturbatively Modified (PM) CASSCF wave function.

Using the previous input, the CASPT2 module will perform in a single run six consecutive single-root CASPT2 calculations for each one of the CASSCF states. At the end of each of the calculations the contributions to the Hamiltonian coupling elements between the computed and the remaining states will be printed. After computing the six CASPT2 roots, the MS-CASPT2 treatment will be performed. First, the effective Hamiltonian matrix, asymmetric and symmetric, is printed.

```
Effective Hamiltonian matrix (Symmetric):
              1  2         3 4         5
1    -.07013926
2    -.01263691     .12976380
3     .00071175     .01001560     .18051855
4     .00509735     .00990244    -.00321669     .19922802
5     .00607124     .00070650    -.00129815    -.00225583     .21601193
```

```
 6     .01998132     .02350235    -.00771000    -.01037132    -.00264941
              6
 1     .18541807
```

Notice that the diagonal elements of the matrix correspond to the single root CASPT2 state energies, where some quantity, 551.0 au here, has been added to get a better print of the output. Following, the eigenvalues and eigenvectors of the diagonalized matrix are obtained:

```
Energies and eigenvectors:
  -552.07305076  -551.88140802  -551.81866833  -551.80756578  -551.79500203
      .99308520     -.10131857      .01038991      .05207094     -.02055799
      .07343489      .90295279      .31190606      .28061095     -.05245262
     -.00869768     -.19493901      .90626880     -.37241673      .03796203
     -.02478279     -.15572120      .13596794      .50373403      .83205915
     -.02204833     -.01553573      .05330075      .08679334      .05789830
     -.08492920     -.33454317      .24485766      .72011863     -.54745806
  -551.78350398
      .01655899
     -.02245882
     -.02155609
     -.10285444
      .99274682
     -.05129770
```

The eigenvalues correspond to the final MS-CASPT2 energies, while the eigenvectors describe the combination of the coupled CASPT2 state which give rise to the final MS-CASPT2 states. **Important:** Notice that the states are written in an increasing energy order, and therefore they do not, in general, correspond to the order obtained in the previous SA-CASSCF calculation. For instance, the MS-CASPT2 state number six, energy -551.78350398 au, mainly correspond to the fifth state of the previous calculation. It is very important to remember that the final states are linear combinations of the preceding ones, and therefore a one to one correspondence is hardly possible. In the present example most of the MS-CASPT2 states have a strong weight in just one of the preceding states, but this is not the case in many situations. Following in the output, a printing of the new wave function is obtained. It corresponds to linear combinations of the SA-CASSCF CI wave functions, obtained in the basis of the previous CASSCF averaged orbitals.

```
 The CI coefficients for the MIXED state nr.  1
----------------------------------------------------------------------------
CI COEFFICIENTS LARGER THAN 0.36
 Occupation of active orbitals, and spin coupling
 of open shells. (u,d: Spin up or down).
  Conf Occupation   Coef Weight
    11  2 22000 200 .960835       .923204
 The CI coefficients for the MIXED state nr.  2
----------------------------------------------------------------------------
CI COEFFICIENTS LARGER THAN 0.36
 Occupation of active orbitals, and spin coupling
 of open shells. (u,d: Spin up or down).
  Conf Occupation   Coef Weight
    20  2 2ud00 200 .856751       .734023
 The CI coefficients for the MIXED state nr.  3
----------------------------------------------------------------------------
CI COEFFICIENTS LARGER THAN 0.36
 Occupation of active orbitals, and spin coupling
 of open shells. (u,d: Spin up or down).
  Conf Occupation   Coef Weight
    85  2 2u0d0 200 .764848       .584993
    86  2 2u00d 200 .507350       .257404
 The CI coefficients for the MIXED state nr.  4
```

```
--------------------------------------------------------------------------
CI COEFFICIENTS LARGER THAN 0.36
 Occupation of active orbitals, and spin coupling
 of open shells. (u,d: Spin up or down).
  Conf Occupation   Coef Weight
     1  2 22200 000   -.368003        .135427
    14  2 22000 u0d .732276        .536229
 The CI coefficients for the MIXED state nr.  5
--------------------------------------------------------------------------
CI COEFFICIENTS LARGER THAN 0.36
 Occupation of active orbitals, and spin coupling
 of open shells. (u,d: Spin up or down).
  Conf Occupation   Coef Weight
     1  2 22200 000 .416925        .173826
    12  2 22000 ud0 .549793        .302272
    14  2 22000 u0d .455052        .207072
 The CI coefficients for the MIXED state nr.  6
--------------------------------------------------------------------------
CI COEFFICIENTS LARGER THAN 0.36
 Occupation of active orbitals, and spin coupling
 of open shells. (u,d: Spin up or down).
  Conf Occupation   Coef Weight
    85  2 2u0d0 200    -.517972       .268295
    86  2 2u00d 200     .776117       .602358
```

The comparison of the present wave functions, that will be hereafter called Perturbatively Modified (PM) CASSCF wave functions, and the previous CASSCF wave functions leads to several conclusions. Remember that the orbital basis has not changed, therefore those mixing related to the orbitals are not going to disappear. For instance, state number three will still be formed by two configurations, because the Rydberg 3px character is still delocalized between orbitals 5 and 6 or symmetry b$_1$. However the character of the second root has changed dramatically. Now one single configuration describes the state, which has acquired a very clear valence character. The previous mixing with a Rydberg-like configuration has disappeared. It is illustrative to carry out an additional analysis of the obtained states using the generated file *JOBMIX* as input file to perform a RASSI calculation, in which new PM-CASSCF properties for the states will be obtained. Even when the changes in energies are small, changes in the properties can be considerable. RASSI provides different types of matrix elements (see next section), and dipole moments, transition dipole moments and their directions, and orbital extensions (all of them available from the RASSI output) will be crucial for our purposes in the study of excited states.

Finally, it is necessary to remember that the extent of the MS interaction relies on the mixing of the previous states. This depends on different factors. The basis sets is one of them. The use of one or other atomic basis set to describe the diffuse functions may lead to different answers. It is not uncommon that CASPT2 results with different diffuse basis sets give different answers due to different extents of the valence-Rydberg mixing. It will be necessary to perform final MS-CASPT2 calculations. Those will change the CASPT2 result in some cases, but it will be unaffected in other cases. Another effect comes from the use of the level shift. The use of MS-CASPT2 does not prevent or affect the extent of the intruder effects. Remember that this effect is already included both in the diagonal terms of the effective Hamiltonian as in the non-diagonal coupling terms. Still a careful checking of different LS values and how they affect the CASPT2 values must be performed, and the final MS-CASPT2 results should be those in which the effect of the intruder states is small, always trying to use as low level shift values as possible. An alternative is to use an imaginary level shift. Finally, the extent of the off-diagonal coupling elements and its asymmetric character introduce further inaccuracies in the treatment. In most cases the proper enlargement of the active space diminishes most of the spurious effects and increases the accuracy.

Transition dipole moment calculations.

One powerful tool included in the MOLCAS package is the RASSI program. RASSI (RAS State Interaction) forms matrix elements of the Hamiltonian and other operators in a wave function basis which consists of individually optimized CI expansions from the RASSCF program. It also solves the Schrödinger equation within the space of these wave functions. In spectroscopy we need to compute the matrix elements of a one-electron operator such as the dipole transition moment to obtain the intensity of the transitions. In an absorption process this means computing the interaction of the ground state with the excited states. RASSI will compute all matrix elements among the states provided they have been computed with the number of inactive and active orbitals, and using the same basis set. The transition dipole moments are computed using the length representation.

In our example we have used two different active spaces. We therefore need to perform at least two RASSI calculations. First we will compute the interaction of the ground state 1^1A_1 (computed as single root), with the $\pi \rightarrow \pi^*$ $^1A_1$ and $^1B_2$ excited states. We should link the corresponding *JOBIPH* files:

```
ln -fs $Project.11A1.JobIph JOB001
ln -fs $Project.1A1.JobIph JOB002
ln -fs $Project.1B2.JobIph JOB003
```

and use the RASSI input file:

```
 &RASSI &END
Nrofjobiphs
 3 1 5 5
  1
  2 3 4 5 6
  1 2 3 4 5
End of input
```

As we are using states that are not orthogonal (this is the case among the 1^1A_1 ground state computed as a single root and the other $^1A_1$ states) we must take the matrix elements of the transition dipole moment computed after the transformation to the eigenbasis; the second time they appear in the output:

```
 PROPERTY: MLTPL  1   COMPONENT:   2
 ORIGIN    :  .00000000D+00  .00000000D+00  .00000000D+00
 STATE     :       1              2              3              4

     1         .00000000D+00  .00000000D+00 -.43587844D+00  .00000000D+00
     2         .00000000D+00  .00000000D+00 -.10019699D+01  .00000000D+00
     3        -.43587844D+00 -.10019699D+01  .00000000D+00 -.46859879D+00
     4         .00000000D+00  .00000000D+00 -.46859879D+00  .00000000D+00
     5         .90773544D-01  .75718497D-01  .00000000D+00  .27645327D+00
     6         .00000000D+00  .00000000D+00  .41227462D+01  .00000000D+00
     7         .00000000D+00  .00000000D+00  .89741299D+00  .00000000D+00
     8        -.16935368D+00  .15487793D+01  .00000000D+00 -.41013917D+01
     9         .81381108D+00  .79559359D+00  .00000000D+00 -.88184724D-01
    10         .00000000D+00  .00000000D+00 -.43659784D+00  .00000000D+00
    11         .13520301D+01  .50454715D+00  .00000000D+00  .56986607D-01

...
```

```
PROPERTY: MLTPL  1   COMPONENT:   3
ORIGIN    :  .00000000D+00  .00000000D+00  .22419033D+01
STATE     :       1              2              3              4
    1        .28126942D+00 -.92709234D+00  .00000000D+00  .11876829D+00
    2       -.92709234D+00  .26218513D+00  .00000000D+00  .14100968D+00
    3        .00000000D+00  .00000000D+00  .52558493D-01  .00000000D+00
    4        .11876829D+00  .14100968D+00  .00000000D+00  .36996295D+00
    5        .00000000D+00  .00000000D+00 -.43197968D+01  .00000000D+00
    6       -.15470487D+00 -.42660550D+00  .00000000D+00  .94593876D+00
    7       -.18676753D-01  .18738780D+01  .00000000D+00 -.37737952D+01
    8        .00000000D+00  .00000000D+00 -.28182178D+00  .00000000D+00
    9        .00000000D+00  .00000000D+00  .38253559D+00  .00000000D+00
   10        .12859613D+01  .48476356D+00  .00000000D+00  .35525361D+00
   11        .00000000D+00  .00000000D+00 -.39325294D-01  .00000000D+00
```

We have a symmetric matrix containing the results. The matrix elements corresponding to the interaction of the first state in the input (ground state) and the remaining states appear both in the first column and in the first row (only partially printed here). Remember that the transition dipole moment (TDM) matrix elements are determined by the symmetry. The matrix element $\langle {}^1\mathrm{A}_1|\mathrm{TDM}|{}^1\mathrm{A}_1\rangle$ will be zero for the x and y components of TDM, and non-zero otherwise. The matrix element $\langle {}^1\mathrm{A}_1|\mathrm{TDM}|{}^1\mathrm{B}_2\rangle$ will be non-zero only for the y component of TDM. This is because the product (wave function 1 × dipole moment component × wave function 2), if decomposed into irreducible representations, must contain the totally symmetric representation to have an allowed transition. In this simple case, we can use a multiplication table for the irreps. Thus, for instance, (${}^1\mathrm{A}_1$ (z) × TDM_y × ${}^1\mathrm{A}_1$ (z)) gives y, which does not belong to the totally symmetric representation. A look at the character table and the behavior of the x, y, z functions will give us the information we need.

Therefore, in the component two (y) of the transition dipole moment matrix elements we have zero values for the interaction among ${}^1\mathrm{A}_1$ states and non-zero values for the interaction among ${}^1\mathrm{A}_1$ and ${}^1\mathrm{B}_2$ states.

The RASSI program in 6.0 and later versions of MOLCAS will print the oscillator strengths and the Einstein A coefficients for all transitions. Also the angles of the transition moment vectors to the coordinate axes will be printed. In the calculation RASSI will use the energies given as input, so be careful to use the keywords HDIAG or EJOB to use energies which include dynamic correlation.

We illustrate how the oscillator strengths are computed. The 11 states are ordered by CASSCF energies. We focus on the valence states; firstly the fourth and fifth ${}^1\mathrm{B}_2$ states. Their transition dipole moment values in atomic units are 0.81381108 and 0.13520301D+01, respectively. The oscillator strength is defined as:

$$f = \frac{2}{3}(TDM)^2\Delta E \tag{10.4}$$

The energy difference ΔE is the excitation energy expressed in atomic units. The transition moments were computed by CASSCF. It is usually not practically possible to compute them with dynamic correlation included, except if a common set of orbitals are used. However, the CASSCF values are usually good enough. (Exceptions occur, e.g. close to narrowly avoided crossings or conical intersections). The excitation energies, on the other hand, are quite sensitive to dynamic correlation. Thus, it is a good approach to use CASSCF TDMs and CASPT2 excitation energies. The values for the oscillator strengths of the two

$^1B_2$ valence states are 0.086 and 0.324, respectively. The excitation energies are 5.31 and 7.23 eV, respectively. All data corresponds to results obtained using the 0.1 Hartree value for the level-shift parameter.

Remember that in other symmetries like C_{2h} the $^1B_2$ states have two components of TDM, x and y, for which the matrix elements with respect to the ground state are non-zero. In this case the TDM^2 value is computed as TDM$_x^2$ + TMD$_y^2$. In those cases is is also possible to compute the direction of the total TDM vector by taking their components and compute the angle respect to any of the axis.

You will find the complete calculation of the absorption spectrum of thiophene in reference [228]. You can observe that, despite there being no level-shift technique used, the final results on the excitation energies agree to within 0.1 eV to those shown here.

10.5.2 Influence of the Rydberg orbitals and states. One example: guanine.

Thiophene has a valence π,π^* orbital space small enough to allow the simultaneous inclusion of all the corresponding Rydberg orbitals into the active space (remember valence space (1302) + Rydberg spaces (0201) or (4020)), but this is not always the case. In addition, the valence-Rydberg mixing is not severe. This mixing is reflected in the orbital extension or the population analysis. In difficult cases valence and Rydberg orbitals mix, and then the configurations also mix. Valence states become more diffuse and Rydberg states more compact. Energetically this has minor consequences for the Rydberg states, which can be computed using these CASSCF mixed wave functions. This is not the case for the valence states. They are extremely sensitive to the mixing. Therefore, if we do not observe clear and compact valence states some mixing has occurred.

We consider the example of the guanine molecule, the nucleic acid base monomer. It is a system with 11 valence π,π^* orbitals which should be included into the active space. It is a planar system in the C_s point group. Focusing only in the $\pi \rightarrow \pi^*$ states we can label the active orbital space (0,11) where 0 is the number of a$'$ orbitals and 11 the number of a$''$ orbitals. In C_s symmetry the Rydberg orbitals are distributed as (6,3), using the same labeling. Therefore the calculation of the corresponding A$'$ states should use the space (0,14) with 14 active electrons and a large number of roots. This is a large calculation that one might want to avoid. One can perform several test calculations (maybe even RASSCF calculations) and find if any orbitals can be excluded. The lowest occupied π orbital is a deep orbital which does not participate in the lowest valence excited states and can be excluded from the active space. Despite this exclusion, a (0,13) orbitals calculation is still expensive. We can proceed in another way. Consider the new valence space (0,10), and add only one more orbital designed to include the first Rydberg orbital. With this space of (0,11) orbitals and 12 active electrons we perform a CASSCF including 6 roots.

Our basis set is of the ANO-L type contracted to C,N,O $4s3p1d$ / H $2s$, plus $1s1p1d$ optimized diffuse functions placed in the cation charge centroid. The results are collected in Table 10.15.

There are important discrepancies between theoretical and experimental results, more important in the properties such as the intensities and the transition dipole moments than in the excitation energies. If we analyze the CASSCF output everything is apparently correct: six converged roots, all of them clear valence states, and no Rydberg orbital into the active space. This is the problem. At least one of the Rydberg orbitals should have been introduced

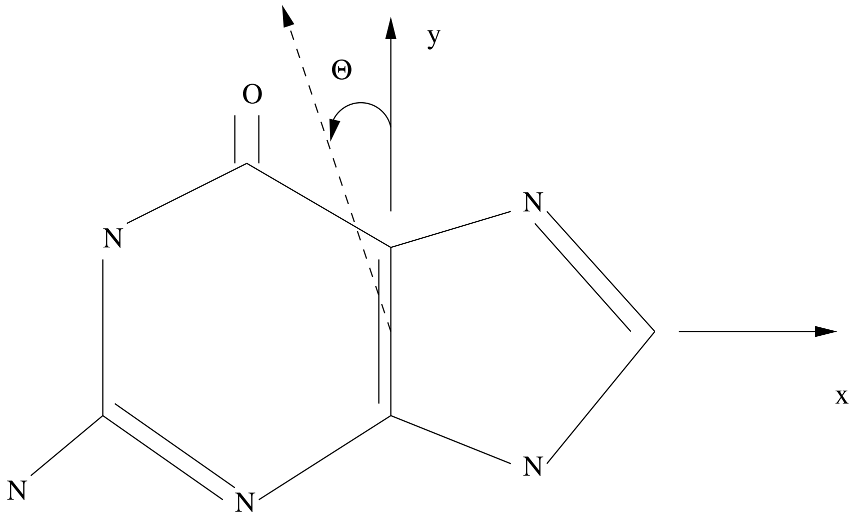


Figure 10.12: Guanine

Table 10.15: CASSCF and CASPT2 excitation energies (eV), oscillator strengths (f), dipole moments (μ(D)), and transition moment directions (Θ) of singlet valence excited states of guanine[a]. The Rydberg orbitals have not been included in the active space.

State	Theoretical					Experiment[b]		
	CAS	$PT2$	f	Θ	μ	ΔE	f	Θ
π–π^* transitions								
2^1A$'$	5.72	4.47	.20	-64^o	1.07	4.4-4.5	.16	(-4^o,35^o)
3^1A$'$	6.74	5.30	.09	+52^o	2.72	4.9-5.0	.25	(-75^o)
4^1A$'$	7.18	5.63	.05	-90^o	3.10	5.7-5.8	<.05[c]	
5^1A$'$	8.45	6.83	.26	0^o	3.20	6.1-6.3	.41	(-71^o,-79^o)

[a] See ref. [245] for details.

Table 10.16: CASSCF and CASPT2 excitation energies (eV), oscillator strengths (f), dipole moments (μ(D)), and transition moment directions (Θ) of singlet valence excited states of guanine[a,b]. The Rydberg orbitals have been first included in the active space and then deleted.

State	Theoretical					Experiment		
	CAS	*PT2*	f	Θ	μ	ΔE	f	Θ
π–π^* transitions								
2^1A$'$	6.08	4.76	.133	-15^o	7.72	4.4-4.5	.16	(-4^o,35^o)
3^1A$'$	6.99	5.09	.231	+73^o	6.03	4.9-5.0	.25	(-75^o)
4^1A$'$	7.89	5.96	.023	+7^o	5.54	5.7-5.8	<.05[c]	
5^1A$'$	8.60	6.65	.161	-80^o	10.17	6.1-6.3	.41	(-71^o,-79^o)
6^1A$'$	9.76	6.55	.225	-41^o	6.11	6.6-6.7	.48	(-9^o,41^o)
7^1A$'$	8.69	6.66	.479	+43^o	6.57			
8^1A$'$	9.43	6.77	.098	+52^o	7.17			

[a] See ref. [245] for details.

[b] A better match with the experimental values is obtained by considering solvent effects.

into the active space. Rydberg and valence orbitals must be treated simultaneously and this is not possible if there is no Rydberg orbital in the active space.

The correct way to proceed is to take the first Rydberg orbital ($3p_z$) and place it as the 11th active orbital of a$''$ symmetry. Then the CASSCF calculation will retain it in the space. Once the calculation has converged we observe than at least one of the computed states is of Rydberg character. It can also happen that some mixing appears in the valence states due to the presence of the diffuse orbital in the active space. The Rydberg orbital is then removed (placed in the last position of its symmetry and the DELETE option used) from the active space and the calculation repeated. This time the next Rydberg orbital ($3d_{xz}$ or $3d_{yz}$) will take its place. The process is repeated once again until the three Rydberg orbitals have been first included in the active space and then deleted (option DELETED of the RASSCF program). Now we can reduce the active space to (0,10), only including valence orbitals and valence excited states.

We can repeat the calculation including even more roots. The results are in Table 10.16.

The results are quite different from those obtained previously, especially regarding the oscillator strengths and transition dipole moment directions. What we have before was a set of states with valence-Rydberg character, although it was not reflected in the orbital extension or population analysis because the orbitals in the active space were too compact to be able to reflect it. The states we have now are also of clear valence character but the difference is that we have first included the Rydberg orbitals in the active space, allowed the flexibility to describe the Rydberg state, and then removed them from the space to finish with a set of compact valence orbitals which cannot represent the Rydberg states. Then, the latter are removed from the computed spectrum of states.

The experience of this type of treatment in different molecules [23, 27, 245] points out that if the valence states of a molecule are computed without considering the Rydberg states and functions (whether by excluding them from the basis set or from the active space) can result in an additional CASPT2 error as large as 0.3-0.4 eV. The errors are more severe for other transitions properties. One example of this can be found for two different CASPT2 treatments of the formamide molecule, one including diffuse functions and other excluding them (see ref. [246] for details). Notice, however, that this approach cannot describe a true valence-Rydberg mixing. An alternative to such an approach is to use the Multi-State CASPT2 treatment that, although computationally expensive, might properly treats the valence-Rydberg mixing. It must be remembered, however, that the performance of the MS-CASPT2 method relies on the previous mixing of the wave functions, and therefore it will not be unusual, depending on the employed basis set, to obtain CASPT2 results that already give the same answer as MS-CASPT2 results when the initial basis sets are changed.

10.5.3 Other cases.

The calculations become increasingly difficult with increased size of the system or in low symmetry cases. Common problems one has to solve are the selection of the active space when it is not possible to include all orbitals expected to be important and the presence of artificial valence-Rydberg mixing in the description of the states. Specific problems appear in systems containing transition metals, where there are a large amount of states close in energy.

To include all the required orbitals into the active space is sometimes impossible. This is one of the important limitations of the methodology. But some solutions are available if one is aware of the limitations. References [247] and [248] report studies on the porphin and indigo molecules, respectively. Porphin and indigo have 24 and 20 π, π^* orbitals, respectively. It is obviously impossible to include all of them in the active spaces. The analysis of the configurations and occupation numbers of the orbitals in a restricted number of excited states by means of the RASSCF method has been found to be a useful procedure to find a proper active space to study different states of the systems. The RASSCF method is able to deal with a larger number of configurations making possible to include all the π orbitals in the active space and analyze the role of the different orbitals. Our goal in this case is to be able to discard some of the deepest or highest orbitals if they become less important in the description of the desired states.

One possibility is to perform a SDTQ calculation involving all the presumably important active space (occupied orbitals in RAS1, empty orbitals in RAS3, no orbitals in RAS2, and four holes/electrons allowed in RAS1/RAS3). The occupation numbers for the active orbitals obtained for such calculation are usually similar to those of a full CASSCF treatment. Another possibility is to place in the CAS space (RAS2) the most important orbitals and the corresponding electrons and only allow singles and doubles excitations from RAS1 (occupied orbitals) to RAS3 (empty orbitals). In all these cases we will study the configurations and occupation numbers of the orbitals to find if some of them are or minor importance for the description of the states we are considering and then reduce the active space for the CASSCF/CASPT2 calculation [247, 248].

Calculation on the excited states of transition metal compounds have to deal with another set of problems. For instance, the known $3d$ double-shell effect: two sets of d orbitals ($3d$ and $4d$) must be included in the reference space in order to obtain accurate results [23] in molecules containing metal atoms of the first transition row with many d-electrons (Fe-Zn). This is a severe limitation when more ligands are included together with the metal atom. Illustrations of such problems are the calculation of the cyanide and carbonyl transition metal compounds [23, 249] and metal-protein models [250]. Core-valence [251] and relativistic effects [28] have been shown to be important for obtaining accurate results. Finally, the problem of the high multiplicity states in the standard CASPT2 formulation has to be considered. The zeroth-order Hamiltonian is defined as a Fock-type one-electron operator. Apart from the originally proposed Fock matrix [18, 19], a correction, denoted g_1 [26], has been designed so that CASSCF wave functions dominated by a closed-shell configuration, on the one hand, and an open-shell configuration, on the other hand, are treated in similar and balanced ways in the perturbation calculation. This correction was shown to be essential in order to obtain reliable results for the Cr_2 molecule with the CASSCF/CASPT2 method [27].

Each type of system and situation has its own specific problems. Size and convergence problems in systems without any symmetry [252, 253], symmetry breaking and localization problems in high symmetry cases [254], excited states in radical cations [255] and anions [256], etc. In addition, there are situations such as the crossing regions which require the simultaneous treatment of more than one state at the CASPT2 level, which can only be solved using the multi-state option in CASPT2.

10.6 Solvent models.

For isolated molecules of modest size the *ab initio* methods have reached great accuracy at present both for ground and excited states. Theoretical studies on isolated molecules, however, may have limited value to bench chemists since most of the actual chemistry takes place in a solvent. If solute-solvent interactions are strong they may have a large impact on the electronic structure of a system and then on its excitation spectrum, reactivity, and properties. For these reasons, numerous models have been developed to deal with solute-solvent interactions in *ab initio* quantum chemical calculations. A microscopic description of solvation effects can be obtained by a supermolecule approach or by combining statistical mechanical simulation techniques with quantum chemical methods. Such methods, however, demand expensive computations. By contrast, at the phenomenological level, the solvent can be regarded as a dielectric continuum, and there are a number of approaches [257, 258, 259, 260, 261] based on the classical reaction field concept.

MOLCAS can model the solvent within the framework of SCF, RASSCF and CASPT2 programs, for the calculation of energies and properties and also for geometry optimizations. The reaction field formalism is based on a sharp partition of the system: the solute molecule (possibly supplemented by some explicit solvent molecules) is placed in a cavity surrounded by a polarizable dielectric. The surrounding is characterized mainly by its dielectric constant and density: an important parameter of the method is the size of the cavity; the dielectric medium is polarized by the solute, and this polarization creates a reaction field which perturbs the solute itself.

Two versions of the model are presently available: one is based on the Kirkwood model [259, 260] and uses only spherical cavities; the other is called PCM (polarizable continuum model) [257, 258] and can use cavities of general shape, modeled on the actual solute molecule. In the former case, the reaction field is computed as a truncated multipolar expansion and added as a perturbation to the one-electron Hamiltonian; in the latter case the reaction field is expressed in terms of a collection of apparent charges (solvation charges) spread on the cavity surface: the PCM reaction field perturbs both one- and two-electron Hamiltonian operators. In both cases, the solvent effects can be added to the Hamiltonian at any level of theory, including MRCI and CASPT2.

10.6.1 Kirkwood model.

This version of the model only uses spherical cavities. In addition, it includes Pauli repulsion due to the medium by introducing a repulsive potential representing the exchange repulsion between the solute and the solvent. This is done by defining a penalty function of Gaussian type, generating the corresponding spherical well integrals, and adding them to the one-electron Hamiltonian. When the repulsion potential is used, the size of the cavity should be optimized for the ground state of the molecule (see below). If the repulsive potential is not used and the cavity size is chosen to be smaller (molecular size plus van der Waals radius as is the usual choice in the literature) one must be aware of the consequences: larger solvent effects but also an unknown presence of molecular charge outside the boundaries of the cavity. This is not a consequence of the present model but it is a general feature of cavity models [260].

10.6.2 PCM

The cavities are defined as the envelope of spheres centered on solute atoms or atomic groups (usually hydrogens are included in the same sphere of the atoms they are bonded to). Two selection of radii are presently available, i. e. Pauling radii, and the so-called UATM (united atom topological model) radii: the latter is the default for PCM calculations; sphere radii can also be provided by the user in the input file. The solvation charges are placed in the middle of small tiles (*tesserae*) drawn on the surface; the number of solvation charges can be gauged by changing the average area of tesserae (keyword AAre in `SEWARD`).

The program prints some information related to the cavity, where one should always check carefully the magnitude of sphere radii: the program adjusts them automatically to the solute topology (each radius depends on hybridization, bonds, etc.), and sometimes this causes some problems (for instance, discontinuities could appear during the scan of a potential energy surface): if this happens, it is preferable to provide the desired radii in the input file, so that they will be kept at all geometries.

When doing state-average RASSCF calculations, one has to specify which root is to be used to generate the solvation charges: this means that the PCM reaction field will be in equilibrium with a specific electronic state, while it perturbs all the states included in the calculation.

In electronic transitions (e. g. photon absorption or emission) one has to include non-equilibrium effects, due to the finite relaxation time of solvent molecules following a sudden change in electronic distribution. This is done by partitioning the reaction field in two components (fast and slow, the former always equilibrated, the latter delayed), whose magnitude is determined by the static dielectric constant and by a “fast” dielectric constant [262] (for very fast processes, like photon absorption, the fast constant is equal to the square of the refraction index). To perform a non-equilibrium calculation, for example to study a ground-to-excited state transition, one has to perform a regular calculation at equilibrium for the ground state, followed by a calculation for the excited state specifying the keyword NONEQ in the `RASSCF` program. Failing to include the keyword NONEQ will cause the program to compute equilibrium solvation also for the excited state, what would be appropriate for an adiabatic, instead of a vertical, transition.

CASPT2 calculations can be performed as usual for isolated molecules, specifying the keyword RFPERT. Geometry optimizations can be performed as usual: note that the arrangement of solvation charges around the solute molecule is likely to break the molecular symmetry. If the symmetry was explicitly requested in SEWARD, the system will keep it through the optimization even in the presence of the solvent, otherwise the convergence could be more difficult, and the final geometry could result of a lower symmetry.

10.6.3 Calculation of solvent effects: Kirkwood model.

We begin by performing a CASSCF/CASPT2 reaction field calculation on the ground state of a molecule.

To use the Kirkwood model, the keyword

```
REACtion field
```

is needed; if no repulsive potential is going to be used the input simply consists in adding the appropriate data (dielectric constant of the medium, cavity size, and angular quantum number of the highest multipole moment of the charge distribution) into the SEWARD input:

```
 &SEWARD &END
...
...
RF-Input
Reaction field
80 8.0 4
End of RF-Input
...
...
End of Input
```

This will compute the reaction field at those levels. The dielectric constant 80.0 correspond to water as solvent. The radius of the cavity is 8.0 in atomic units. Finally 4 is the maximum angular moment number used in the multipole expansion. The cavity origin is the coordinate origin, thus the molecule must be placed accordingly.

If we want to include the reaction field (either PCM or Kirkwood model) at other levels of theory the keyword RFPErt must be added to the MOTRA or CASPT2 inputs.

We are, however, going to explain the more complicated situation where a repulsive well potential has to be added to the model. In this case it is convenient to optimize the size of the cavity, although in so doing we obtain large cavity sizes and therefore smaller solvent effects. More realistic results can be obtained if additional and specific solvent molecules are added inside the cavity.

To define the well potential we have to add the keyword WELL Integrals to the SEWARD input to compute and add the Pauli repulsion integrals to the bare Hamiltonian.

The requirements considered to build this potential are that it shall reproduce solvation energies for spherical particles, ions, and that it must be wide enough so that the electrons in the excited state of the molecules are also confined to the cavity. Negative ions have the property that their electrons are loosely bound and they are thus suited for parametrizing the repulsive potential. The final result of different calibration calculations [263, 260] is a penalty function which includes four Gaussians. If a is the radius of the cavity the Gaussians are placed at distances $a + 2.0$, $a + 3.0$, $a + 5.0$ and $a + 7.0$ a.u. from the cavity's center with exponents 5.0, 3.5, 2.0 and 1.4, respectively.

As an example we will use the N,N-dimethylaminobenzonitrile (DMABN) molecule (see Figure 10.13). This is a well known system with large dipole moments both in ground and excited states which suffer important effects due to the polar environment.

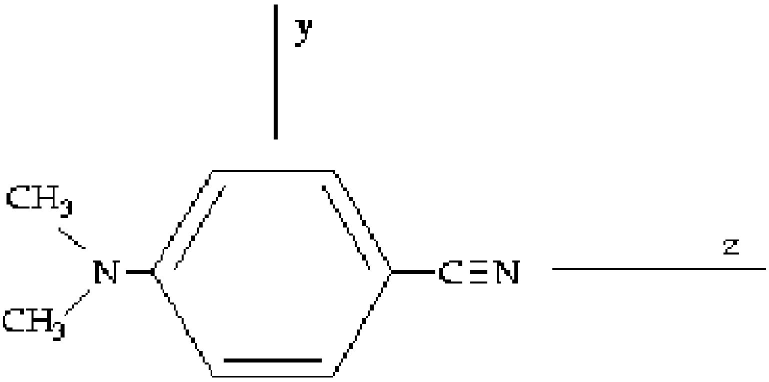


Figure 10.13: N,N-dimethylaminobenzonitrile (DMABN)

```
 &SEWARD &END
Title
para-DMABN molecule. Cavity size: 10 au.
Symmetry
 X XY
Basis set
N.ANO-S...3s2p1d.
N1             0.0000000000        0.0000000000        4.7847613288
N2             0.0000000000        0.0000000000       -8.1106617786
End of basis
Basis set
C.ANO-S...3s2p1d.
C1             0.0000000000        0.0000000000        2.1618352923
C2             0.0000000000        2.2430930886        0.7747833630
C3             0.0000000000        2.2317547910       -1.8500321252
C4             0.0000000000        0.0000000000       -3.1917306021
C5             0.0000000000        0.0000000000       -5.9242601761
C6             0.0000000000        2.4377336900        6.0640991723
End of basis
Basis set
H.ANO-S...2s.
H1             0.0000000000        4.0043085530       -2.8534714086
H2             0.0000000000        4.0326542950        1.7215314260
H3             0.0000000000        2.1467175630        8.0879851846
H4             1.5779129980        3.6622699270        5.5104123361
End of basis
RF-Input
reaction field
38.8 10.0 4
End of RF-Input
Well Int
4
1.0 5.0 12.0
1.0 3.5 13.0
1.0 2.0 15.0
```

```
1.0 1.4 17.0
End of Input
 &SCF &END
TITLE
 DMABN molecule
OCCUPIED
20 2 12 5
ITERATIONS
50
END OF INPUT
 &RASSCF &END
TITLE
 p-DMABN
SYMMETRY
    1
SPIN
    1
NACTEL
   10    0    0
FROZEN
    8    0    3    0
INACTIVE
   12    1    9    1
RAS2
    0    2    0    7
THRS
1.0E-06,1.0E-03,1.0E-03
ITER
50,25
LUMORB
END OF INPUT
```

In the SEWARD input the WELL INTEGRALS must include first the number of Gaussians used (four), followed by the coefficient and exponent of the Gaussian and the radius of the cavity in the sequence explained above: first the most compact Gaussian with the radius plus 2.0 au, and so on to the least compact Gaussian. Here, we have defined a cavity size of 10 au (cavity centered at coordinate origin). The RASSCF program will read the RCTFLD input, prepared this time for acetonitrile ($\epsilon = 38.8$), a cavity size of 10.0 au (the same as in the SEWARD input) and a multipole expansion up to the fourth order which is considered sufficient [260]. The active space includes the π space over the molecular plane, excluding the π orbital of the CN group which lies in the molecular plane.

We repeat the calculation for different cavity sizes in order to find the radius which gives the lowest absolute energy at the CASSCF level. The presence of the repulsive terms allows the cavity radius to be computed by energy minimization. For the calculations using different cavity sizes it is not necessary to repeat the calculation of all the integrals, just those related to the well potential. Therefore, the keyword ONEONLY can be included in the SEWARD input. The *ONEINT* file will be modified and the *ORDINT* file is kept the same for each molecular geometry. The energies obtained are in Table 10.17.

Taking the gas-phase value (no cav.) as the reference, the CASSCF energy obtained with a 10.0 au cavity radius is higher. This is an effect of the repulsive potential, meaning that the molecule is too close to the boundaries. Therefore we discard this value and use the values from 11.0 to 16.0 to make a simple second order fit and obtain a minimum for the cavity radius at 13.8 au.

Once we have this value we also need to optimize the position of the molecule in the cavity. Some parts of the molecule, especially those with more negative charge, tend to move close to the boundary. Remember than the sphere representing the cavity has its origin in the

Radius (au)	CASSCF energies (au)
no cav.	-455.653242
10.0	-455.645550
11.0	-455.653486
12.0	-455.654483
14.0	-455.654369
16.0	-455.654063

Table 10.17: Ground state CASSCF energies for DMABN with different cavity sizes.

cartesian coordinates origin. We use the radius of 13.8 au and compute the CASSCF energy at different displacements along the coordinate axis. Fortunately enough, this molecule has C_{2v} symmetry. That means that displacements along two of the axis (x and y) are restricted by symmetry. Therefore it is necessary to analyze only the displacements along the z coordinate. In a less symmetric molecule all the displacements should be studied even including combination of the displacements. The result may even be a three dimensional net, although no great accuracy is really required. The results for DMABN n C_{2v} symmetry are compiled in Table 10.18.

Disp. in z (au)	CASSCF energies (au)
+0.5	-455.654325
0.0	-455.654400
-0.5	-455.654456
-1.0	-455.654486
-1.5	-455.654465

Table 10.18: Ground state CASSCF energies for different translations with respect to the initial position of of the DMABN molecule in a 13.8 au cavity.

Fitting these values to a curve we obtain an optimal displacement of -1.0 au. We move the molecule and reoptimize the cavity radius at the new position of the molecule. The results are listed in Table 10.19.

Radius (au)	CASSCF energies (au)
11.8	-455.653367
12.8	-455.654478
13.8	-455.654486
14.8	-455.654318

Table 10.19: Ground state CASSCF energies for DMABN with different cavity sizes. The molecule position in the cavity has been optimized.

There is no significant change. The cavity radius is then selected as 13.8 au and the position of the molecule with respect to the cavity is kept as in the last calculation. The calculation is carried out with the new values. The SCF or RASSCF outputs will contain the information about the contributions to the solvation energy. The CASSCF energy obtained will include the reaction field effects and an analysis of the contribution to the solvation energy for each value of the multipole expansion:

```
Reaction field specifications:
------------------------------

 Dielectric Constant :                  .388E+02
 Radius of Cavity(au):                  .138E+02
 Truncation after    :                 4

Multipole analysis of the contributions to the dielectric solvation energy

--------------------------------------
   l             dE
--------------------------------------
   0            .0000000
   1           -.0013597
   2           -.0001255
   3           -.0000265
   4           -.0000013
--------------------------------------
```

10.6.4 Solvation effects in ground states. PCM model in formaldehyde.

The reaction field parameters are added to the SEWARD program input through the keyword

```
RF-Input
```

To invoke the PCM model the keyword

```
PCM-model
```

is required. A possible input is

```
RF-input
PCM-model
solvent
acetone
AAre
0.2
End of rf-input
```

which requests a PCM calculation with acetone as solvent, with tesserae of average area 0.2Å^2. Note that the default parameters are solvent=water, average area 0.4Å^2; see the SEWARD manual section for further PCM keywords. By default the PCM adds non-electrostatic terms (i. e. cavity formation energy, and dispersion and repulsion solute-solvent interactions) to the computed free-energy in solution.

A complete input for a ground state CASPT2 calculation on formaldehyde (H_2CO) in water is

```
&GATEWAY
Title
formaldehyde
Coord
4
H      0.000000    0.924258   -1.100293    Angstrom
H      0.000000   -0.924258   -1.100293    Angstrom
```

```
C      0.000000    0.000000   -0.519589    Angstrom
O      0.000000    0.000000    0.664765    Angstrom
Basis set
 6-31G*
Group
 X Y
RF-input
PCM-model
solvent
water
end of rf-input
End of input
&SEWARD
End of input
&SCF
Title
formaldehyde
Occupied
5 1 2 0
End of input
&RASSCF
Title
formaldehyde
nActEl
4 0 0
Inactive
4 0 2 0
Ras2
1 2 0 0
LumOrb
End of input
&CASPT2
Frozen
4  0  0  0
RFPErt
End of input
```

10.6.5 Solvation effects in excited states. PCM model and acrolein.

In the PCM picture, the solvent reaction field is expressed in terms of a polarization charge density $\sigma(\mathbf{s})$ spread on the cavity surface, which, in the most recent version of the method, depends on the electrostatic potential $V(\mathbf{s})$ generated by the solute on the cavity according to

$$\left[\frac{\epsilon+1}{\epsilon-1}\hat{S}-\frac{1}{2\pi}\hat{S}\hat{D}^{*}\right]\sigma(\mathbf{s}) \quad = \quad \left[-1+\frac{1}{2\pi}\hat{D}\right]V(\mathbf{s}) \tag{10.5}$$

where ϵ is the solvent dielectric constant and $V(\mathbf{s})$ is the (electronic+nuclear) solute potential at point $\mathbf{s}$ on the cavity surface. The $\hat{S}$ and $\hat{D}^{*}$ operators are related respectively to the electrostatic potential $V^{\sigma}(\mathbf{s})$ and to the normal component of the electric field $E^{\sigma}_{\perp}(\mathbf{s})$ generated by the surface charge density $\sigma(\mathbf{s})$. It is noteworthy that in this PCM formulation the polarization charge density $\sigma(\mathbf{s})$ is designed to take into account implicitly the effects of the fraction of solute electronic density lying outside the cavity.

In the computational practice, the surface charge distribution $\sigma(\mathbf{s})$ is expressed in terms of a set of point charges $\mathbf{q}$ placed at the center of each surface tessera, so that operators are replaced by the corresponding square matrices. Once the solvation charges ($\mathbf{q}$) have been determined, they can be used to compute energies and properties in solution.

The interaction energy between the solute and the solvation charges can be written

$$E_{int} = \mathbf{V}^{\dagger}\mathbf{q} = \sum_{\mathbf{i}}^{\mathbf{N_{TS}}} \mathbf{V_i q_i} \tag{10.6}$$

where V_i is the solute potential calculated at the representative point of tessera i. The charges act as perturbations on the solute electron density ρ: since the charges depend in turn on ρ through the electrostatic potential, the solute density and the charges must be adjusted until self consistency. It can be shown[261] that for any SCF procedure including a perturbation linearly depending on the electron density, the quantity that is variationally minimized corresponds to a free energy (i.e. E_{int} minus the work spent to polarize the dielectric and to create the charges). If $E^0 = E[\rho^0] + V_{NN}$ is the solute energy in vacuo, the free energy minimized in solution is

$$\mathcal{G} = E[\rho] + V_{NN} + \frac{1}{2}E_{int} \tag{10.7}$$

where V_{NN} is the solute nuclear repulsion energy, ρ^0 is the solute electronic density for the isolated molecule, and ρ is the density perturbed by the solvent.
The inclusion of non-equilibrium solvation effects, like those occurring during electronic excitations, is introduced in the model by splitting the solvation charge on each surface element into two components: $q_{i,f}$ is the charge due to electronic (fast) component of solvent polarization, in equilibrium with the solute electronic density upon excitations, and $q_{i,s}$, the charge arising from the orientational (slow) part, which is delayed when the solute undergoes a sudden transformation.

The photophysics and photochemistry of acrolein are mainly controlled by the relative position of the $^1(n-\pi^*)$, $^3(n-\pi^*)$ and $^3(\pi-\pi^*)$ states, which is, in turn, very sensitive to the presence and the nature of the solvent. We choose this molecule in order to show an example of how to use the PCM model in a CASPT2 calculation of vertical excitation energies.

The three states we want to compute are low-lying singlet and triplet excited states of the *s-trans* isomer. The π space (4 π MOs /4 π-electrons) with the inclusion of the lone-pair MO (n_y) is a suitable choice for the active space in this calculation. For the calculation in aqueous solution, we need first to compute the CASPT2 energy of the ground state in presence of the solvent water. This is done by including in the SEWARD input for the corresponding gas-phase calculation the section

```
RF-input
PCM-model
solvent
 water
DIELectric constant
 78.39
CONDuctor version
AARE
 0.4
End of rf-input
```

If not specified, the default solvent is chosen to be water. Some options are available. The value of the dielectric constant can be changed for calculations at temperatures other than 298 K. For calculations in polar solvents like water, the use of the conductor model (C-PCM) is recommended. This is an approximation that employs conductor rather than dielectric boundary conditions. It works very well for polar solvents (i. e. dielectric constant greater than about 5), and is based on a simpler and more robust implementation. It can be useful also in cases when the dielectric model shows some convergence problems. Another parameter that can be varied in presence of convergency problem is the average area of the tesserae of which the surface of the cavity is composed. However, a lower value for this parameter may give poorer results.

Specific keywords are in general needed for the other modules to work with PCM, except for the SCF. The keyword NONEquilibrium is necessary when computing excited states energies in RASSCF. For a state specific calculation of the ground state CASSCF energy, the solvent effects must be computed with an equilibrium solvation approach, so this keyword must be omitted. None the less, the keyword RFpert must be included in the CASPT2 input in order to add the reaction field effects to the one-electron hamiltonian as a constant perturbation.

```
 &RASSCF &END
Title
Acrolein GS + PCM
Spin
 1
Symmetry
 1
nActEl
 6 0 0
Frozen
 4 0
Inactive
 8 0
Ras2
 1 4
LUMORB
THRS
1.0e-06 1.0e-04 1.0e-04
ITERation
 100 100
End of input
 &CASPT2 &END
Title
 ground state + PCM
RFpert
End of Input
```

Information about the reaction field calculation employing a PCM-model appear first in the SCF output

```
Polarizable Continuum Model (PCM) activated
Solvent:water
Version: Conductor
Average area for surface element on the cavity boundary: 0.4000 Angstrom2
Minimum radius for added spheres: 0.2000 Angstrom
Polarized Continuum Model Cavity
================================
 Nord Group  Hybr  Charge Alpha Radius     Bonded to
   1   O     sp2   0.00   1.20  1.590   C   [d]
   2   CH    sp2   0.00   1.20  1.815   O   [d]  C   [s]
   3   CH    sp2   0.00   1.20  1.815   C   [s]  C   [d]
   4   CH2   sp2   0.00   1.20  2.040   C   [d]
 ------------------------------------------------------------------------------
```

The following input is used for the CASPT2 calculation of the $^3A''(n \rightarrow \pi^*)$ state. Provided that the same $WorkDir has been using, which contains all the files of of the calculation done for the ground state, the excited state calculation is done by using inputs for the RASSCF and the CASPT2 calculations:

```
 &RASSCF &END
Title
Acrolein n->pi* triplet state + PCM
Spin
 3
Symmetry
 2
nActEl
 6 0 0
Frozen
 4 0
Inactive
 8 0
Ras2
 1 4
NONEquilibrium
LUMORB
ITERation
 100 100
End of input
 &CASPT2 &END
Title
 triplet state
RFpert
End of Input
```

Note the RASSF keyword NONEQ, requesting that the slow part of the reaction field be frozen as in the ground state, while the fast part is equilibrated to the new electronic distribution. In this case the fast dielectric constant is the square of the refraction index, whose value is tabulated for all the allowed solvents (anyway, it can be modified by the user through the keyword "DIELectric" in SEWARD).

The RASSCF output include the line:

```
Calculation type: non-equilibrium (slow component from JobOld)
 Reaction field from state:           1
```

This piece of information means that the program computes the solvent effects on the energy of the $^3A''(n \rightarrow \pi^*)$ by using a non-equilibrium approach. The slow component of the solvent response is kept frozen in terms of the charges that have been computed for the previous equilibrium calculation of the ground state. The remaining part of the solvent response, due to the fast charges, is instead computed self-consistently for the state of interest (which is state 1 of the specified spatial and spin symmetry in this case).

The vertical excitations to the lowest valence states in aqueous solution for *s-trans* acrolein are listed in the Table 10.20 and compared with experimental data. As expected by qualitative reasoning, the vertical excitation energy to the $^1A''(n \rightarrow \pi^*)$ state exhibits a blue shift in water. The value of the vertical transition energy computed with the inclusion of the PCM reaction field is computed to be 3.96 eV at the CASPT2 level of theory. The solvatochromic shift is thus of +0.33 eV. Experimental data are available for the excitation energy to the $^1A''(n \rightarrow \pi^*)$ state. The band shift in going from isooctane to water is reported to be +0.24 eV which is in fair agreement with the PCM result.

No experimental data are available for the excitation energies to the triplet states of acrolein in aqueous solution. However it is of interest to see how the ordering of these two states depends on solvent effects. The opposing solvatochromic shifts produced by the solvent on these two electronic transitions place the two triplet states closer in energy. This result might suggest that a dynamical interconversion between the $n\pi^*$ and $\pi\pi^*$ may occur more favorable in solution.

Table 10.20: Vertical excitation energies/eV (solvatochromic shifts) of s-trans acrolein in gas-phase and in aqueous solution

State	Gas-phase	Water	Expt.[a]
$^1A''(n_y \rightarrow \pi^*)$	3.63	3.96 (+0.33)	3.94 (+0.24)[b]
$T_1\ ^3A''(n_y \rightarrow \pi^*)$	3.39	3.45 (+0.06)	
$T_2\ ^3A'(\pi \rightarrow \pi^*)$	3.81	3.71 (-0.10)	

[a] Ref.[264]

[b] Solvatochromic shifts derived by comparison of the absorption wave lengths in water and isooctane

10.7 Computing relativistic effects in molecules.

Molcas-8.2 is intended for calculations on systems including all atoms of the periodic table. This is only possible if relativistic effects can be added in a way that is accurate and at the same time applies to all the methods used in Molcas-8.2, in particular the CASSCF and CASPT2 approaches. Molcas-8.2 includes relativistic effects within the same wave function framework as used in non-relativistic calculations. This has been possible by partitioning the relativistic effects into two parts: the scalar relativistic effects and spin-orbit coupling. This partitioning is based on the Douglas-Kroll (DK) transformation of the relativistic Hamiltonian [265, 266].

10.7.1 Scalar relativistic effects

The scalar relativistic effects are included by adding the corresponding terms of the DK Hamiltonian to the one-electron integrals in Seward (use the keyword Douglas-Kroll). This has no effect on the form of the wave function and can be used with all Molcas-8.2 modules. Note however that it is necessary to use a basis set with a corresponding relativistic contraction. Molcas-8.2 provides the ANO-RCC basis set, which has been constructed using the DK Hamiltonian. Use this basis set in your relativistic calculations. It has the same accuracy as the non-relativistic ANO-L basis set. Scalar relativistic effects become important already for atoms of the second row. With ANO type basis sets it is actually preferred to use the DK Hamiltonian and ANO-RCC in all your calculations.

10.7.2 Spin-Orbit coupling (SOC)

In order to keep the structure of Molcas as intact as possible, it was decided to incorporate SOC as an *a posteriori* procedure which can be added after a series of CASSCF calculations.

The program RASSI has been modified to include the spin-orbit part of the DK Hamiltonian [267]. The method is thus based on the concept of electronic states interacting via SOC. In practice this means that one first performs a series of CASSCF calculations in the electronic states one expects to interact via SOC. They are then used as the basis states in the RASSI calculations. Dynamic electron correlation effects can be added by a shift of the diagonal of the SOC Hamiltonian to energies obtained in a CASPT2 or MRCI calculation. If MS-CASPT2 is used, a special output file (*JOBMIX*) is provided that is to be used as the input file for RASSI. The procedure will below be illustrated in a calculation on the lower excited states of the PbO molecule.

The SO Hamiltonian has been approximated by a one-electron effective Hamiltonian [268], which also avoids the calculation of multi-center integrals (the Atomic Mean Field Approximation – AMFI) [268, 269].

10.7.3 The PbO molecule

Results from a calculation of the potentials for the ground and lower excited states of PbO, following the procedure outlined above, has recently been published [270]. The ground state of PbO dissociates to O(^3P) and Pb(^3P). However in the Pb atom there is strong SOC between the ^3P, ^1D, and ^1S term of the $(6s)^2(6p)^2$ electronic configuration. All levels with the Ω value O^+ arising from these terms will therefore contribute to the ground state potential. The first task is therefore to construct the electronic states that are obtained by coupling O(^3P) to any of the ^3P, ^1D, and ^1S terms of Pb. In the table below we give the states. They have been labeled both in linear symmetry and in C_2 symmetry, which is the symmetry used in the calculation because it makes it possible to average over degenerate components.

Spin	C_2 sym	Labels in linear symmetry	No. of states
2	1	$^5\Delta$, $2\times{}^5\Sigma^+$, $^5\Sigma^-$	5
2	2	$2\times{}^5\Pi$	4
1	1	$3\times{}^3\Delta$, $3\times{}^3\Sigma^+$, $4\times{}^3\Sigma^-$	13
1	2	$6\times{}^3\Pi$, $^3\Phi$	14
0	1	$^1\Delta$, $2\times{}^1\Sigma^+$, $^1\Sigma^-$	5
0	2	$2\times{}^1\Pi$	4

The total number of states is 45. One thus has to perform 6 CASSCF (and MS-CASPT2) calculations according to the spin and symmetries given in the table. The RASSI-SO calculation will yield 134 levels with Ω ranging from 0 to 4. Only the lower of these levels will be accurate because of the limitations in the selection of electronic states.

The active space used in these calculations is 6s,6p for Pb and 2p for O. This is the natural choice and works well for all main group elements in most molecules. The s-orbital should be active in groups IIa-Va, but may be left inactive for the heavier atoms (groups VIa-VIIa). The ANO-RCC basis sets have been constructed to include correlation of the semi-core electrons. For Pb they are the 5d, which should then not be frozen in the CASPT2 calculations. All other core electrons should be frozen, because there are no basis functions to describe their correlation. Including them in the correlation treatment may lead to large BSSE errors.

The input file for these calculations is quite lengthy, so we show here only one set of CASSCF/CASPT2 calculations but the whole RASSI input for all six cases.

```
&GATEWAY
```

```
  Title= PbO
  Coord= $CurrDir/PbO.xyz
  Basis set
  ANO-RCC-VQZP
  Group= XY
  AngMom
 0.00  0.00  0.00
End of Input
&SEWARD
End of Input
&SCF
  Title
  PbO
  Occupied
    24 21
  Iterations
    20
  Prorbitals
    2 1.d+10
End of Input
&RASSCF
  Title
  PbO
  Symmetry
    1
  Spin
    5
  CIROOT
    5 5 1
  nActEl
    8 0 0
  Inactive
    23 18
  Ras2
    3 4
  Lumorb
  THRS
    1.0e-8 1.0e-04 1.0e-04
  Levshft
    1.50
  ITERation
    200 50
  CIMX
    200
  SDAV
    500
End of Input
&CASPT2
  Title
  PbO
  MAXITER
    25
  FROZEN
    19 16
  Focktype
  G1
  Multistate
    5 1 2 3 4 5
  Imaginary Shift
    0.1
End of Input
>> COPY $Project.JobMix $CurrDir/JobMix.12
&RASSCF
  Title
  PbO
  Symmetry
```

```
    2
  Spin
    5
  CIROOT
    4 4 1
  nActEl
    8 0 0
  Inactive
    23 18
  Ras2
    3 4
  Lumorb
  THRS
    1.0e-8 1.0e-04 1.0e-04
  Levshft
    1.50
  ITERation
    200 50
  CIMX
    200
  SDAV
    500
End of Input
&CASPT2
  Title
  PbO
  MAXITER
    25
  FROZEN
    19 16
  Focktype
  G1
  Multistate
    4 1 2 3 4
  Imaginary Shift
    0.1
>> COPY $Project.JobMix $CurrDir/JobMix.22
&RASSCF
  Title
  PbO
  Symmetry
    1
  Spin
    3
  CIROOT
    13 13 1
  nActEl
    8 0 0
  Inactive
    23 18
  Ras2
    3 4
  Lumorb
  THRS
    1.0e-8 1.0e-04 1.0e-04
  Levshft
    1.50
  ITERation
    200 50
  CIMX
    200
  SDAV
    500
End of Input
&CASPT2
  Title
  PbO
  MAXITER
    25
  FROZEN
    19 16
  Focktype
  G1
  Multistate
    13 1 2 3 4 5 6 7 8 9 10 11 12 13
  Imaginary Shift
    0.1
End of Input
>> COPY $Project.JobMix $CurrDir/JobMix.11
&RASSCF
  Title
  PbO
  Symmetry
    2
  Spin
    3
  CIROOT
    14 14 1
  nActEl
    8 0 0
  Inactive
    23 18
  Ras2
    3 4
  Lumorb
  THRS
    1.0e-8 1.0e-04 1.0e-04
  Levshft
    1.50
  ITERation
    200 50
  CIMX
    200
  SDAV
    500
End of Input
&CASPT2
  Title
  PbO
  MAXITER
    25
  FROZEN
    19 16
  Focktype
  G1
  Multistate
    14 1 2 3 4 5 6 7 8 9 10 11 12 13 14
  Imaginary Shift
    0.1
>> COPY $Project.JobMix $CurrDir/JobMix.21
&RASSCF
  Title
  PbO
  Symmetry
    1
  Spin
    1
  CIROOT
    5 5 1
  nActEl
    8 0 0
  Inactive
```

```
    23 18
  Ras2
    3 4
  Lumorb
  THRS
    1.0e-8 1.0e-04 1.0e-04
  Levshft
    1.50
  ITERation
    200 50
  CIMX
    200
  SDAV
    500
End of Input
&CASPT2
  Title
  PbO
  MAXITER
    25
  FROZEN
    19 16
  Focktype
  G1
  Multistate
    5 1 2 3 4 5
  Imaginary Shift
    0.1
End of Input
>> COPY $Project.JobMix $CurrDir/JobMix.10
&RASSCF
  Title
  PbO
  Symmetry
    2
  Spin
    1
  CIROOT
    4 4 1
  nActEl
    8 0 0
  Inactive
    23 18
  Ras2
    3 4
  Lumorb
  THRS
    1.0e-8 1.0e-04 1.0e-04
  Levshft
    1.50
  ITERation
    200 50
  CIMX
    200
  SDAV
    500
End of Input
&CASPT2
  Title
  PbO
  MAXITER
    25
  FROZEN
    19 16
  Focktype
  G1
  Multistate
    4 1 2 3 4
  Imaginary Shift
    0.1
End of Input
>> COPY $Project.JobMix $CurrDir/JobMix.20
>> COPY $CurrDir/JobMix.12 JOB001
>> COPY $CurrDir/JobMix.11 JOB002
>> COPY $CurrDir/JobMix.21 JOB003
>> COPY $CurrDir/JobMix.10 JOB004
>> COPY $CurrDir/JobMix.22 JOB005
>> COPY $CurrDir/JobMix.20 JOB006
&RASSI
  Nrof JobIphs
    6 5 13 14 5 4 4
    1 2 3 4 5
    1 2 3 4 5 6 7 8 9 10 11 12 13
    1 2 3 4 5 6 7 8 9 10 11 12 13 14
    1 2 3 4 5
    1 2 3 4
    1 2 3 4
  Spin Orbit
  Ejob
End of Input
```

In the above definitions of the JobMix files the labels correspond to symmetry and spin. Thus *JobMix.12* is for quintets (S=2) in symmetry 1, etc. The keyword EJOB ensures that the MS-CASPT2 energies from the *JobMix* files are used as the diagonal elements in the SO Hamiltonian matrix. The output file of one such calculation is quite lengthy (6 CASSCF/MS-CASPT2 calculations and one RASSI). Important sections of the RASSI output are the spin-free energies (look for the word "SPIN-FREE" in the listing) and the SOC energies (found by looking for "COMPLEX"). The complex SO wave functions are also given and can be used to analyze the wave function. For linear molecules one wants to know the Ω values of the different solutions. Here the computed transition moments can be quite helpful (using the selection rules). It is important in a calculation of many excited states, as the one above, to check for intruder state problems in the CASPT2 results.

This example includes a large number of states, because the aim was to compute full potential curves. If one is only interested in the properties near equilibrium, one can safely reduce the number of states. For lighter atoms it is often enough to include the spin-free states that are close in energy in the calculation of the SOC. An intersystem crossing can usually be treated by including only the two crossing states. The choice of basis states for the RASSI calculation depends on the strength of the SO interaction and the energy separation between the states.

The above input is for one distance. The shell script loops over distances according to:

```
Dist='50.0 10.0 8.00 7.00 6.00 5.50 5.00 4.40 4.20 4.00 3.90 3.80 3.75 3.70
3.65 3.60 3.55 3.50 3.40 3.30 3.10'
 for R in $Dist
 do
 cat $CurrDir/template | sed -e "s/Dist/$R/" >$CurrDir/input
rm -rf $WorkDir
mkdir  $WorkDir
cd     $WorkDir
echo "R=$R" >>$CurrDir/energies
molcas $CurrDir/input >$CurrDir/out_$R
grep "Reference energy" $CurrDir/out_$R >>$CurrDir/energies
grep "Total energy" $CurrDir/out_$R >>$CurrDir/energies
grep "Reference weight" $CurrDir/out_$R >>$CurrDir/energies
```

```
done
```

Thus, the whole potential curves can be run as one job (provided that there are no problems with intruder states, convergence, etc). Notice that the *JOBIPH* files for one distance are used as input (*JOBOLD*) for the next distance. The shell script collects all CASSCF and CASPT2 energies and reference weights in the file *energies*.

We shall not give any detailed account of the results obtained in the calculation of the properties of the PbO molecule. The reader is referred to the original article for details [270]. However it might be of interest to know that the computed dissociation energy (D_0) was 5.0 eV without SOC and 4.0 eV with (experiment is 3.83 eV). The properties at equilibrium are much less affected by SOC: the bond distance is increased with 0.003 Å, the frequency is decreased with 11 cm^{-1}. The results have also been used to assign the 10 lowest excited levels.

10.8 Extra information about basis sets and integrals

Basis set format The **Inline** option for a basis set will read the basis set as defined by the following pseudo code.

```
Read Charge, lAng
Do iAng = 0, lAng
   Read nPrim, nContr
   Read (Exp(iPrim),iPrim=1,nPrim)
   Do iPrim=1,nPrim
      Read (Coeff(iPrim,iContr),iContr=1,nContr)
   End Do
End Do
```

where *Charge* is the nuclear charge, *lAng* is the highest angular momentum quantum number, *nPrim* is the number of primitive functions (exponents) for a given shell, and *nContr* is the number of contracted functions for a given shell.

The following is an example of a DPZ basis set for carbon. Normally, however, the basis set will be read from a library file following the specified label (like, e.g., C.DZP...4s2p1d.), and not be inserted inline at the input file.

```
Basis set                                -- Start defining a basis set
C.DZP.Someone.9s5p1d.4s2p1d. / inline    -- Definition in input stream
       6.0             2                 -- charge, max l-quantum no.
    9    4                               -- no. of prim. and contr. s-functions
4232.61                                  -- s-exponents
634.882
146.097
42.4974
14.1892
1.9666
5.1477
0.4962
0.1533
  .002029   .0       .0       .0         -- s-contraction matrix
  .015535   .0       .0       .0
  .075411   .0       .0       .0
  .257121   .0       .0       .0
  .596555   .0       .0       .0
  .242517   .0       .0       .0
  .0       1.0       .0       .0
  .0        .0      1.0       .0
  .0        .0       .0      1.0
    5    2                               -- no. of prim. and contr. p-functions
18.1557                                  -- p-exponents
3.98640
1.14290
0.3594
0.1146
 .018534   .0                            -- p-contraction matrix
 .115442   .0
 .386206   .0
 .640089   .0
 .0       1.0
    1    1                               -- no. of prim. and contr. d-functions
   .75                                   -- d-exponents
  1.0                                    -- d-contraction matrix
C1 0.00000 0.00000 0.00000               -- atom-label, Cartesian coordinates
C2 1.00000 0.00000 0.00000               -- atom-label, Cartesian coordinates
End Of Basis                             -- end of basis set definition
```

The basis set label and the ECP libraries The label within the *ECP* library is given as input in the line following the keyword BASIS SET. The label defines either the valence basis set and core potential which is assigned to a frozen-core atom or the embedding potential which is assigned to an environmental froze-ion. Here, all the comments made about this label in the section **The basis set label and the basis set library** for all-electron basis sets stand, except for the following changes:

1. The identifier `type` must be `ECP` or `PP`.

2. The identifier `aux` specifies the kind of the potential. It is used, for instance, to choose between non-relativistic, Cowan-Griffin, or no-pair Douglas-Kroll relativistic core potentials
   (i.e. `Pt.ECP.Casarrubios.13s10p9d5f.1s2p2d1f.16e-NR-AIMP.`
   or `Pt.ECP.Casarrubios.13s10p9d5f.1s2p2d1f.16e-CG-AIMP.`
   or `Pt.ECP.Rakowitz.13s10p9d6f.5s4p4d2f.18e-NP-AIMP.`)
   and to pick up one among all the embedding potentials available for a given ion
   (i.e. `F.ECP.Lopez-Moraza.0s.0s.0e-AIMP-KMgF3.`
   or `F.ECP.Lopez-Moraza.0s.0s.0e-AIMP-CsCaF3.`).

3. The identifier `contracted` is used here in order to produce the actual basis set out of the basis set included in the *ECP* library, which is a minimal basis set (in general contraction form) augmented with some polarization, diffuse, ... function. It indicates the number of `s, p, ...,` contracted functions in the actual basis set, the result being always a many-primitive contracted function followed by a number of primitives. As an example, `At.ECP.Barandiaran.13s12p8d5f.`**`3s4p3d2f`**`.17e-CG-AIMP.` will generate a (13,1,1/12,1,1,1/8,1,1/5,1) formal contraction pattern which is in this case a (13,1,1/12,1,1,1/7,1,1/5,1) real pattern. Other contraction patters should be input "Inline".

4. The user is suggested to read carefully section 10.9 of the tutorials and examples manual before using the ECP utilities.

One-Electron Integral Labels

The storage of one-electron integrals on disk is facilitated by the one-electron integral I/O facility. The internal structure of the one-electron file and the management is something which the user normally do not need to worry about. However, for the general input section of the FFPT, the user need to know the name and structure of the internal labels which the one-electron integral I/O facility associates with each type of one-electron integral. The labels are listed and explained here below for reference. The component index is also utilized by the one-electron integral I/O facility to discriminate the various components of the one-electron integrals of a certain type, for example, the dipole moment integrals have three components (1=x-component, 2=y-component, 3=z-component). The component index is enumerated as a canonical index over the powers of the Cartesian components of the operator (e.g. multipole moment, velocity, electric field, etc.). The order is defined by following pseudo code,

```
Do ix = nOrder, 0, -1
   Do iy = nOrder-ix, 0, -1
      iz = nOrder-ix-iy
   End Do
End Do,
```

where nOrder is the total order of the operator, for example, nOrder=2 for the electric field gradient and the quadrupole moment operator.

Label	Explanation
'Mltpl *nn*'	the *nn*'th order Cartesian multipole moments.
'MltplS '	the overlap matrix used in the semi-empirical NDDO method.
'Kinetic '	the kinetic energy integrals.
'Attract '	the electron attraction integrals.
'AttractS'	the electron attraction integrals used in the semi-empirical NDDO method.
'PrjInt '	the projection integrals used in ECP calculations.
'M1Int '	the M1 integrals used in ECP calculations.
'M2Int '	the M2 integrals used in ECP calculations.
'SROInt '	the spectrally resolved operator integrals used in ECP calculations.
'XFdInt '	the external electric field integrals.
'MassVel '	the mass-velocity integrals.
'Darwin '	the Darwin one-electron contact integrals.
'Velocity'	the velocity integrals.
'EF0nnnnn'	the electric potential at center *nnnnn*.
'EF1nnnnn'	the electric field at center *nnnnn*.
'EF2nnnnn'	the electric field gradient at center *nnnnn*.
'AngMom '	the angular momentum integrals.
'DMS '	the diamagnetic shielding integrals.
'Wellnnnn'	the *nnnn*'th set of spherical well integrals.
'OneHam '	the one-electron Hamiltonian.
'AMProd '	the hermitized product of angular momentum integrals.
'AMFI '	the atomic mean field integrals.

10.9 Core and Embedding Potentials within the SEWARD Program

8.2 is able to perform *effective core potential* (ECP) and *embedded cluster* (EC) calculations. In ECP calculations [271, 272] the *core* electrons of a molecule are kept frozen and represented by a set of atomic effective potentials, while only the valence electrons are explicitly handled in the quantum mechanical calculation. In EC calculations only the electrons assigned to a piece of the whole system, the *cluster*, are explicitly treated in a quantum mechanical calculation, while the rest of the whole system, the *environment*, is kept frozen and represented by embedding potentials which act onto the *cluster*. For an explanation of the type of potentials and approaches used in MOLCAS the reader is referred to the section 8.49 of the user's guide.

To use such type of effective potentials implies to compute a set of atomic integrals and therefore involves only the SEWARD program. The remaining MOLCAS programs will simply use the integrals in the standard way and no indication of the use of ECP will appear in the outputs further on; the difference is of course that the absolute energies obtained for the different methods are not comparable to those obtained in an all-electron calculation. Therefore, the only input required to use ECP or EC is the SEWARD input, according to the examples given below. In the input files of the subsequent MOLCAS programs the orbitals corresponding to the excluded core orbitals should of course not be included, and not the excluded electrons.

10.9.1 SEWARD input for Effective Core Potential calculations

Astatine (At) is the atomic element number 85 which has the main configuration in its electronic ground state: [*core*] $6s^2 5d^{10} 6p^5$. In the *core* 68 electrons are included, corresponding to the xenon configuration plus the $4f^{14}$ lantanide shell. To perform an ECP calculation in a molecular system containing At it is necessary to specify which type of effective potential will substitute the *core* electrons and which valence basis set will complement it. Although the core ECP's (strictly AIMP's, see section 8.49 of the user's guide) can be safely mixed together with all-electron basis set, the valence basis sets included in the MOLCAS AIMP library have been explicitly optimized to complement the AIMP potentials.

The file ***ECP*** in the MOLCAS directory $MOLCAS/basis_library contains the list of available core potentials and valence basis sets. Both the relativistic (CG-AIMP's) and the nonrelativistic (NR-AIMP's) potentials are included. As an example, this is the head of the entry corresponding to the relativistic ECP for At:

```
/At.ECP.Barandiaran.13s12p8d5f.1s1p2d1f.17e-CG-AIMP.
Z.Barandiaran, L.Seijo, J.Chem.Phys. 101(1994)4049; L.S. JCP 102(1995)8078.
core[Xe,4f] val[5d,6s,6p]  SO-corr  (11,1,1/9111/611*/4o1)=3s4p3d2f recommended
*
* - spin-orbit basis set correction from
*   L.Seijo, JCP 102(1995)8078.
*
* - (5o) f orthogonality function is the 4f core orbital
*
*ATQR-DSP(A3/A2/71/5)-SO       (A111/9111/611/41)
```

The first line is the label line written in the usual SEWARD format: element symbol, basis label, first author, size of the primitive set, size of the contracted set (in both cases referred to the valence basis set), and type of ECP used. In this case there are 17 valence electrons and the effective potential is a Cowan-Griffin-relativistic core AIMP. The number of primitive functions for the valence basis set ($13s12p8d5f$ here) will split into different subsets (within a segmented contraction scheme) according to the number of contracted functions. In the library, the contracted basis functions have been set to the minimal basis size: $1s1p2d1f$ for the valence electrons in At. This means the following partition: $1s$ contracted function including 13 primitive functions; $1p$ contracted function including 12 primitive functions; $2d$ contracted functions, the first one containing seven primitive functions and the second one primitive function (see the library), and finally $1f$ contracted function containing five primitive functions.

In the `SEWARD` input the user can modify the contraction scheme simply varying the number of contracted functions. There is a recommended size for the valence basis set which is printed in the third line for each atom entry on the library: $3s4p3d2f$ for At. For example, the simplest way to include the atom core potential and valence basis set in the `SEWARD` input would be:

```
At.ECP...3s4p3d2f.17e-CG-AIMP.
```

This means a partition for the valence basis set as showed in figure 10.14.

Figure 10.14: Partition of a valence basis set using the ECP's library

```
Basis set:AT.ECP...3S4P3D2F.17E-CG-AIMP.
                 Type
                  s
          No.      Exponent    Contraction Coefficients
           1   .133037396D+07  -.000154   .000000   .000000
           2   .993126141D+05  -.001030   .000000   .000000
           3   .128814005D+05  -.005278   .000000   .000000
           4   .247485916D+04  -.014124   .000000   .000000
           5   .214733934D+03   .069168   .000000   .000000
           6   .111579706D+03   .020375   .000000   .000000
           7   .370830653D+02  -.259246   .000000   .000000
           8   .113961072D+02   .055751   .000000   .000000
           9   .709430236D+01   .649870   .000000   .000000
          10   .448517638D+01  -.204733   .000000   .000000
          11   .157439587D+01  -.924035   .000000   .000000
          12   .276339384D+00   .000000  1.000000   .000000
          13   .108928284D+00   .000000   .000000  1.000000
                 Type
                  p
          No.      Exponent    Contraction Coefficients
          14   .608157825D+04   .000747   .000000   .000000   .000000
          15   .128559298D+04   .009304   .000000   .000000   .000000
          16   .377428675D+03   .026201   .000000   .000000   .000000
          17   .552551834D+02  -.087130   .000000   .000000   .000000
          18   .233740022D+02  -.044778   .000000   .000000   .000000
          19   .152762905D+02   .108761   .000000   .000000   .000000
          20   .838467359D+01   .167650   .000000   .000000   .000000
          21   .234820847D+01  -.290968   .000000   .000000   .000000
          22   .119926577D+01  -.237719   .000000   .000000   .000000
          23   .389521915D+00   .000000  1.000000   .000000   .000000
          24   .170352883D+00   .000000   .000000  1.000000   .000000
          25   .680660800D-01   .000000   .000000   .000000  1.000000
                 Type
                  d
          No.      Exponent    Contraction Coefficients
          26   .782389711D+03   .007926   .000000   .000000
```

```
27   .225872717D+03   .048785   .000000   .000000
28   .821302011D+02   .109617   .000000   .000000
29   .173902999D+02  -.139021   .000000   .000000
30   .104111329D+02  -.241043   .000000   .000000
31   .195037661D+01   .646388   .000000   .000000
32   .689437556D+00   .000000  1.000000   .000000
33   .225000000D+00   .000000   .000000  1.000000
       Type
        f
No.      Exponent    Contraction Coefficients
34   .115100000D+03   .065463   .000000
35   .383200000D+02   .270118   .000000
36   .151600000D+02   .468472   .000000
37   .622900000D+01   .387073   .000000
38   .242100000D+01   .000000  1.000000
```

Therefore, the primitive set will always be split following the scheme: the first contracted function will contain the total number of primitives minus the number of remaining contracted functions and each of the remaining contracted functions will contain one single uncontracted primitive function. In the present example possible contraction patterns are: contracted $1s1p2d1f$ (13/12/8,1/5 primitives per contracted function, respectively), $2s2p3d2f$ (12,1/11,1/7,1,1/4,1), $3s3p4d2f$ (11,1,1/10,1,1/6,1,1,1/4,1), etc. Any other scheme which cannot be generated in this way must be included in the input using the Inline format for basis sets or an additional user's library. When the Inline option is used both the valence basis set and the AIMP potential must be included in the input, as it will be shown in the next section.

For an explanation of the remaining items in the library the reader is referred to the section 8.49 of the user's guide.

Figure 10.15 contains the sample input required to compute the SCF wave function for the astatine hydride molecule at an internuclear distance of 3.2 au. The Cowan-Griffin-relativistic core-AIMP has been used for the At atom with a size for the valence basis set recommended in the *ECP* library: $3s4p3d2f$.

Figure 10.15: Sample input required by SEWARD and SCF programs to compute the SCF wave function of HAt using a relativistic ECP

```
 &GATEWAY
Title
HAt molecule using 17e-Cowan-Griffin-relativistic core-AIMP
coord
2
coordinates in bohr
At 0 0 0
H  0 0 3.2
group
X Y
Basis set
H.ano-l-vtzp
Basis set
At.ECP...3s4p3d2f.17e-CG-AIMP.
 &SEWARD
 &SCF
Title
 HAt g.s. (At-val=5d,6s,6p)
Occupied
 4 2 2 1
```

10.9.2 SEWARD input for Embedded Cluster calculations

To perform embedded cluster (EC) calculations requires certain degree of experience and therefore the reader is referred to the literature quoted in section 8.49 of the user's guide. On the following a detailed example is however presented. It corresponds to EC calculations useful for local properties associated to a Tl^+ impurity in $KMgF_3$. First, a cluster must be specified. This is the piece of the system which is explicitly treated by the quantum mechanical calculation. In the present example the cluster will be formed by the unit $(TlF_{12})^{11-}$. A flexible basis for the cluster must be determined. Figure 10.17 contains the basis set selection for the thallium and fluorine atoms. In this case ECP-type basis sets have been selected. For Tl a valence basis set of size $3s4p4d2f$ has been used combined with the relativistic core-AIMP potentials as they appear in the *ECP* library. For the F atom the valence basis set has been modified from that appearing in the *ECP* library. In this case the exponent of the p-diffuse function and the p contraction coefficients of the F basis set have been optimized in calculations on the fluorine anion included in the specific lattice in order to obtain a more flexible description of the anion. This basis set must be introduced Inline, and then also the ECP potential must be added to the input. The user can compare the basis set and ECP for F in figure 10.17 with the entry of *ECP* under /F.ECP.Huzinaga.5s6p1d.1s2p1d.7e-NR-AIMP. The entry for the Inline format must finish with the line End of Spectral Representation Operator.

Once the cluster has been defined it is necessary to represent the embedding lattice. Presently, MOLCAS includes embedding potentials for ions of several elpasolites, fluoro-perovskites, rocksalt structure oxides and halides, and fluorites. The embedding potentials for any other structure can be included in the input using the Inline format or included in a private user library. In the selected example a fluoro-perovskite lattice has been selected: $KMgF_3$. Here, the Tl^+ impurity substitutes a K^+ ion in an O_h site with 12 coordination. The first coordination shell of fluorine ions has been included into the cluster structure and the interactions to the Tl atom will be computed by quantum mechanical methods. The rest of the lattice will be represented by the structure $KMgF_3$ with five shells of ions at experimental sites. The shells have been divided in two types. Those shells closer to the cluster are included as embedding potentials from the library *ECP*. For example the potassium centers will use the entry on figure 10.16.

Figure 10.16: Sample input for an embedded core potential for a shell of potassium cations

```
Basis set
K.ECP..0s.0s.0e-AIMP-KMgF3.
PSEUdocharge
K2-1    0.0000000000   0.0000000000   7.5078420000
K2-2    0.0000000000   7.5078420000   0.0000000000
K2-3    0.0000000000   7.5078420000   7.5078420000
K2-4    7.5078420000   0.0000000000   0.0000000000
K2-5    7.5078420000   0.0000000000   7.5078420000
K2-6    7.5078420000   7.5078420000   0.0000000000
K2-7    7.5078420000   7.5078420000   7.5078420000
End Of Basis
```

No basis set is employed to represent the potassium centers on figure 10.16, which just act as potentials embedding the cluster. The keyword PSEUDOCHARGE ensures that the interaction energy between the embedding potentials is not included in the "Nuclear repulsion energy" and that their location is not varied in a geometry optimization (SLAPAF). The first shells of Mg^{+2} and F^{-} will be introduced in the same way.

The remaining ions of the lattice will be treated as point charges. To add a point charge on the SEWARD input it is possible to proceed in two ways. One possibility is to employ the usual label to introduce an atom with its basis functions set to zero and the keyword CHARGE set to the value desired for the charge of the center. This way of introducing point charges must not be used when geometry optimizations with the SLAPAF program is going to be performed because SLAPAF will recognize the point charges as atoms whose positions should be optimized. Instead the keyword XFIELD can be used as it is illustrated in figure 10.17. XFIELD must be followed by a line containing the number of point charges, and by subsequent lines containing the cartesian coordinates and the introduced charge or the three components of the dipole moment at the specified geometry. In any case the seven positions in each line must be fulfilled. To ensure the neutral character of the whole system the point charges placed on the terminal edges, corners or faces of the lattice must have the proper fractional values.

Figure 10.17 contains the complete sample input to perform a SCF energy calculation on the system $(TlF_{12})^{11-} : KMgF_3$.

Figure 10.17: Sample input for a SCF geometry optimization of the $(TlF_{12})^{11-} : KMgF_3$ system

```
 &GATEWAY
Title
|                          Test run TlF12:KMgF3.1                               |
|** Molecule **   (TlF12)11- cluster embedded in a lattice of KMgF3            |
|** Basis set and ECP **                                                       |
|  * Tl * (11,1,1/9,1,1,1/5,1,1,1/4,1)                            from ECP    |
|         13e-Cowan-Griffin-relativistic core-AIMP                 from ECP    |
|  * F *  (4,1/4,1,1) diffuse-p optimized in KMgF3:F(-)                 inline|
|          7e-nonrelativistic core-AIMP                                  inline|
|  KMgF3 embedding-AIMPs                                          from ECP    |
|** cluster geometry **   r(Tl-F)/b= 5.444 = 3.84948932 * sqrt(2)             |
|** lattice **  (perovskite structure) 5 shells of ions at experimental sites  |
Symmetry
X Y Z
Basis set
Tl.ECP.Barandiaran.13s12p8d5f.3s4p4d2f.13e-CG-AIMP.
Tl     0.00000   0.00000   0.00000
End Of Basis
Basis set
F.ECP.... / Inline
*    basis set and core-AIMP as in: F.ECP.Huzinaga.5s6p1d.2s4p1d.7e-NR-AIMP.
*    except that the p-diffuse and the p contraction coeffs. have been
*    optimized in KMgF3-embedded F(-) scf calculations.
  7.000000         1
    5    2
   405.4771610
   61.23686380
   13.47117730
   1.095173720
   .3400847530
  -.013805187800   .000000000000
  -.089245064800   .000000000000
  -.247937861000   .000000000000
   .632895340000   .000000000000
   .000000000000   .465026336000
    6    3
   44.13600920
   9.982597110
   2.947082680
   .9185111850
   .2685213550
   .142
   .015323038700   .000000000000   .000000000000
   .095384703000   .000000000000   .000000000000
   .291214218000   .000000000000   .000000000000
   .441351868000   .000000000000   .000000000000
   .000000000000   .427012588000   .000000000000
   .000000000000   .000000000000  1.000000000000
*
* Core AIMP: F-1S
*
* Local Potential Paramenters : (ECP convention)
*                               A(AIMP)=-Zeff*A(ECP)
M1
    7
   279347.4000
   31889.74900
   5649.977600
   1169.273000
   269.0513200
   71.29884600
   22.12150700
   .004654725000
   .007196816857
   .015371258571
   .032771900000
   .070383742857
   .108683807143
   .046652035714
M2
    0
COREREP
   1.0
PROJOP
    0
   14    1
  52.7654040
   210965.4100
   31872.59200
   7315.837400
   2077.215300
   669.9991000
   232.1363900
   84.99573000
   32.90124100
   13.36331800
   5.588141500
   2.319058700
   .9500928100
   .3825419200
   .1478404000
   .000025861368
   .000198149380
   .001031418900
   .004341016600
   .016073698000
   .053856655000
   .151324390000
   .318558040000
   .404070310000
   .190635320000
   .011728993000
   .002954046500
  -.000536098280
   .000278474090
*
Spectral Representation Operator
Valence primitive basis
Exchange
End of Spectral Representation Operator
F_1        3.849489320       3.849489320        .000000000
F_2         .000000000       3.849489320       3.849489320
F_3        3.849489320        .000000000       3.849489320
* 3*4 = 12
End Of Basis
* end of cluster data: TlF12
* beginning of lattice embedding data: KMgF3
Basis set
K.ECP.Lopez-Moraza.0s.0s.0e-AIMP-KMgF3.
pseudocharge
* K(+) ions as embedding AIMPs
K2-1    0.0000000000   0.0000000000   7.5078420000
K2-2    0.0000000000   7.5078420000   0.0000000000
K2-3    0.0000000000   7.5078420000   7.5078420000
K2-4    7.5078420000   0.0000000000   0.0000000000
K2-5    7.5078420000   0.0000000000   7.5078420000
K2-6    7.5078420000   7.5078420000   0.0000000000
K2-7    7.5078420000   7.5078420000   7.5078420000
* 3*2 + 3*4 + 1*8 = 26
End Of Basis
Basis set
Mg.ECP.Lopez-Moraza.0s.0s.0e-AIMP-KMgF3.
pseudocharge
* Mg(2+) ions as embedding AIMPs
MG1-1   3.7539210000   3.7539210000   3.7539210000
MG3-1   3.7539210000   3.7539210000  11.2617630000
MG3-2   3.7539210000  11.2617630000   3.7539210000
MG3-3   3.7539210000  11.2617630000  11.2617630000
MG3-4  11.2617630000   3.7539210000   3.7539210000
MG3-5  11.2617630000   3.7539210000  11.2617630000
MG3-6  11.2617630000  11.2617630000   3.7539210000
MG3-7  11.2617630000  11.2617630000  11.2617630000
* 8*8 = 64
End Of Basis
Basis set
F.ECP.Lopez-Moraza.0s.0s.0e-AIMP-KMgF3.
pseudocharge
* F(-) ions as embedding AIMPs
F2-1    3.7539210000   3.7539210000   7.5078420000
F2-2    3.7539210000   7.5078420000   3.7539210000
F2-3    7.5078420000   3.7539210000   3.7539210000
F3-1    0.0000000000   3.7539210000  11.2617630000
F3-2    3.7539210000   0.0000000000  11.2617630000
F3-3    3.7539210000  11.2617630000   0.0000000000
F3-4    0.0000000000  11.2617630000   3.7539210000
F3-5    3.7539210000  11.2617630000   7.5078420000
F3-6    0.0000000000  11.2617630000  11.2617630000
F3-7    3.7539210000   7.5078420000  11.2617630000
F3-8   11.2617630000   3.7539210000   0.0000000000
F3-9   11.2617630000   0.0000000000   3.7539210000
F3-10   11.2617630000   3.7539210000   7.5078420000
F3-11    7.5078420000   3.7539210000  11.2617630000
F3-12   11.2617630000   0.0000000000  11.2617630000
F3-13   11.2617630000  11.2617630000   0.0000000000
F3-14    7.5078420000  11.2617630000   3.7539210000
F3-15   11.2617630000   7.5078420000   3.7539210000
F3-16   11.2617630000  11.2617630000   7.5078420000
F3-17    7.5078420000  11.2617630000  11.2617630000
F3-18   11.2617630000   7.5078420000  11.2617630000
```

```
* 9*4 +  12*8 = 132
End Of Basis
* The rest of the embedding lattice will be represented by point charges,
* which enter into the calculation in the form of a XField.
*
XField
 95
*
* K(+) ions as point charges
    0.0000000000   0.0000000000  15.0156840000       +1.0  0.  0.  0.
    0.0000000000   7.5078420000  15.0156840000       +1.0  0.  0.  0.
    0.0000000000  15.0156840000   0.0000000000       +1.0  0.  0.  0.
    0.0000000000  15.0156840000   7.5078420000       +1.0  0.  0.  0.
    0.0000000000  15.0156840000  15.0156840000       +1.0  0.  0.  0.
    7.5078420000   0.0000000000  15.0156840000       +1.0  0.  0.  0.
    7.5078420000   7.5078420000  15.0156840000       +1.0  0.  0.  0.
    7.5078420000  15.0156840000   0.0000000000       +1.0  0.  0.  0.
    7.5078420000  15.0156840000   7.5078420000       +1.0  0.  0.  0.
    7.5078420000  15.0156840000  15.0156840000       +1.0  0.  0.  0.
   15.0156840000   0.0000000000   0.0000000000       +1.0  0.  0.  0.
   15.0156840000   0.0000000000   7.5078420000       +1.0  0.  0.  0.
   15.0156840000   0.0000000000  15.0156840000       +1.0  0.  0.  0.
   15.0156840000   7.5078420000   0.0000000000       +1.0  0.  0.  0.
   15.0156840000   7.5078420000   7.5078420000       +1.0  0.  0.  0.
   15.0156840000   7.5078420000  15.0156840000       +1.0  0.  0.  0.
   15.0156840000  15.0156840000   0.0000000000       +1.0  0.  0.  0.
   15.0156840000  15.0156840000   7.5078420000       +1.0  0.  0.  0.
   15.0156840000  15.0156840000  15.0156840000       +1.0  0.  0.  0.
*
* F(-) ions as point charges
   3.7539210000   3.7539210000  15.0156840000       -1.0  0.  0.  0.
   3.7539210000  11.2617630000  15.0156840000       -1.0  0.  0.  0.
   3.7539210000  15.0156840000   3.7539210000       -1.0  0.  0.  0.
   3.7539210000  15.0156840000  11.2617630000       -1.0  0.  0.  0.
  11.2617630000   3.7539210000  15.0156840000       -1.0  0.  0.  0.
  11.2617630000  11.2617630000  15.0156840000       -1.0  0.  0.  0.
  11.2617630000  15.0156840000   3.7539210000       -1.0  0.  0.  0.
  11.2617630000  15.0156840000  11.2617630000       -1.0  0.  0.  0.
  15.0156840000   3.7539210000   3.7539210000       -1.0  0.  0.  0.
   15.0156840000   3.7539210000  11.2617630000       -1.0  0.  0.  0.
   15.0156840000  11.2617630000   3.7539210000       -1.0  0.  0.  0.
   15.0156840000  11.2617630000  11.2617630000       -1.0  0.  0.  0.
*
* Mg(2+) ions in face, as fractional point charges
   3.7539210000   3.7539210000  18.7696050000        +1.0  0.  0.  0.
   3.7539210000  11.2617630000  18.7696050000        +1.0  0.  0.  0.
   3.7539210000  18.7696050000   3.7539210000        +1.0  0.  0.  0.
   3.7539210000  18.7696050000  11.2617630000        +1.0  0.  0.  0.
  11.2617630000   3.7539210000  18.7696050000        +1.0  0.  0.  0.
  11.2617630000  11.2617630000  18.7696050000        +1.0  0.  0.  0.
  11.2617630000  18.7696050000   3.7539210000        +1.0  0.  0.  0.
  11.2617630000  18.7696050000  11.2617630000        +1.0  0.  0.  0.
  18.7696050000   3.7539210000   3.7539210000        +1.0  0.  0.  0.
  18.7696050000   3.7539210000  11.2617630000        +1.0  0.  0.  0.
  18.7696050000  11.2617630000   3.7539210000        +1.0  0.  0.  0.
  18.7696050000  11.2617630000  11.2617630000        +1.0  0.  0.  0.
*
* Mg(2+) ions in edge, as fractional point charges
   3.7539210000  18.7696050000  18.7696050000     +0.5  0.  0.  0.
  11.2617630000  18.7696050000  18.7696050000     +0.5  0.  0.  0.
  18.7696050000   3.7539210000  18.7696050000     +0.5  0.  0.  0.
  18.7696050000  11.2617630000  18.7696050000     +0.5  0.  0.  0.
  18.7696050000  18.7696050000   3.7539210000     +0.5  0.  0.  0.
  18.7696050000  18.7696050000  11.2617630000     +0.5  0.  0.  0.
*
* Mg(2+) ions in corner, as fractional point charges
```

```
  18.7696050000  18.7696050000  18.7696050000      +0.25  0. 0. 0.
*
* F(-) ions in face, as fractional point charges
   0.0000000000   3.7539210000  18.7696050000       -0.5  0. 0. 0.
   3.7539210000   0.0000000000  18.7696050000       -0.5  0. 0. 0.
   0.0000000000  11.2617630000  18.7696050000       -0.5  0. 0. 0.
   3.7539210000   7.5078420000  18.7696050000       -0.5  0. 0. 0.
   3.7539210000  18.7696050000   0.0000000000       -0.5  0. 0. 0.
   0.0000000000  18.7696050000   3.7539210000       -0.5  0. 0. 0.
   3.7539210000  18.7696050000   7.5078420000       -0.5  0. 0. 0.
   0.0000000000  18.7696050000  11.2617630000       -0.5  0. 0. 0.
   3.7539210000  18.7696050000  15.0156840000       -0.5  0. 0. 0.
   3.7539210000  15.0156840000  18.7696050000       -0.5  0. 0. 0.
   7.5078420000   3.7539210000  18.7696050000       -0.5  0. 0. 0.
  11.2617630000   0.0000000000  18.7696050000       -0.5  0. 0. 0.
   7.5078420000  11.2617630000  18.7696050000       -0.5  0. 0. 0.
  11.2617630000   7.5078420000  18.7696050000       -0.5  0. 0. 0.
  11.2617630000  18.7696050000   0.0000000000       -0.5  0. 0. 0.
   7.5078420000  18.7696050000   3.7539210000       -0.5  0. 0. 0.
  11.2617630000  18.7696050000   7.5078420000       -0.5  0. 0. 0.
   7.5078420000  18.7696050000  11.2617630000       -0.5  0. 0. 0.
  11.2617630000  18.7696050000  15.0156840000       -0.5  0. 0. 0.
  11.2617630000  15.0156840000  18.7696050000       -0.5  0. 0. 0.
  18.7696050000   3.7539210000   0.0000000000       -0.5  0. 0. 0.
  18.7696050000   0.0000000000   3.7539210000       -0.5  0. 0. 0.
  18.7696050000   3.7539210000   7.5078420000       -0.5  0. 0. 0.
  18.7696050000   0.0000000000  11.2617630000       -0.5  0. 0. 0.
  18.7696050000   3.7539210000  15.0156840000       -0.5  0. 0. 0.
  15.0156840000   3.7539210000  18.7696050000       -0.5  0. 0. 0.
  18.7696050000  11.2617630000   0.0000000000       -0.5  0. 0. 0.
  18.7696050000   7.5078420000   3.7539210000       -0.5  0. 0. 0.
  18.7696050000  11.2617630000   7.5078420000       -0.5  0. 0. 0.
  18.7696050000   7.5078420000  11.2617630000       -0.5  0. 0. 0.
  18.7696050000  11.2617630000  15.0156840000       -0.5  0. 0. 0.
  15.0156840000  11.2617630000  18.7696050000       -0.5  0. 0. 0.
  15.0156840000  18.7696050000   3.7539210000       -0.5  0. 0. 0.
  18.7696050000  15.0156840000   3.7539210000       -0.5  0. 0. 0.
  15.0156840000  18.7696050000  11.2617630000       -0.5  0. 0. 0.
  18.7696050000  15.0156840000  11.2617630000       -0.5  0. 0. 0.
*
* F(-) ions in edge, as fractional point charges
   0.0000000000  18.7696050000  18.7696050000       -0.25  0. 0. 0.
   7.5078420000  18.7696050000  18.7696050000       -0.25  0. 0. 0.
  18.7696050000   0.0000000000  18.7696050000       -0.25  0. 0. 0.
  18.7696050000   7.5078420000  18.7696050000       -0.25  0. 0. 0.
  18.7696050000  18.7696050000   0.0000000000       -0.25  0. 0. 0.
  18.7696050000  18.7696050000   7.5078420000       -0.25  0. 0. 0.
  18.7696050000  18.7696050000  15.0156840000       -0.25  0. 0. 0.
  15.0156840000  18.7696050000  18.7696050000       -0.25  0. 0. 0.
  18.7696050000  15.0156840000  18.7696050000       -0.25  0. 0. 0.
*  end of lattice embedding data: KMgF3
* 13 cluster components  and 881 lattice components
 &SEWARD
 &SCF
Title
 (TlF12)11- run as D2h
Occupied
 12    7    7    6    7    6    6    3
```

Bibliography

[1] Björn O. Roos, Valera Veryazov, and Per-Olof Widmark. Relativistic atomic natural orbital type basis sets for the alkaline and alkaline-earth atoms applied to the ground-state potentials for the corresponding dimers. *Theor. Chem. Acc.*, 111:345–351, 2004.

[2] Björn O. Roos, Roland Lindh, Per-Åke Malmqvist, Valera Veryazov, and Per-Olof Widmark. Main group atoms and dimers studied with a new relativistic ANO basis set. *J. Phys. Chem. A*, 108:2851–2858, 2004.

[3] Björn O. Roos, Roland Lindh, Per-Åke Malmqvist, Valera Veryazov, and Per-Olof Widmark. New relativistic ANO basis sets for transition metal atoms. *J. Phys. Chem. A*, 109:6575–6579, 2005.

[4] Björn O. Roos, Roland Lindh, Per-Åke Malmqvist, Valera Veryazov, and Per-Olof Widmark. New relativistic ANO basis sets for actinide atoms. *Chem. Phys. Letters*, 409:295–299, 2005.

[5] Björn O. Roos, Roland Lindh, Per-Åke Malmqvist, Valera Veryazov, Per-Olof Widmark, and Antonio Carlos Borin. New relativistic atomic natural orbital basis sets for lanthanide atoms with applications to the ce diatom and luf$_3$. *J. Phys. Chem. A*, 112:11431–11435, 2008.

[6] Björn O. Roos, Per-Åke Malmqvist, and Laura Gagliardi. Heavy element quantum chemistry – the multiconfigurational approach. In Erkki J. Brändas and Eugene S. Kryachko, editors, *Fundamental World of Quantum Chemistry. Vol. II*, pages 425–442. Kluwer Academic Publishers, Dordrecht, Netherlands, 2003.

[7] Francesco Aquilante, Roland Lindh, and Thomas Bondo Pedersen. Unbiased auxiliary basis sets for accurate two-electron integral approximations. *J. Chem. Phys.*, 127:114107, 2007.

[8] Francesco Aquilante, Per-Åke Malmqvist, Thomas Bondo Pedersen, Abhik Ghosh, and Björn O. Roos. Cholesky decomposition-based multiconfiguration second-order perturbation theory (CD-CASPT2): Application to the spin-state energetics of co$^{\text{iii}}$(diiminato)(nph). *J. Chem. Theory Comput.*, 4:694–702, 2008.

[9] Francesco Aquilante, Thomas Bondo Pedersen, Björn O. Roos, Alfredo Sánchez de Merás, and Henrik Koch. Accurate *ab initio* density fitting for multiconfigurational self-consistent field methods. *J. Chem. Phys.*, 129:024113, 2008.

[10] Dongxia Ma, Giovanni Li Manni, and Laura Gagliardi. The generalized active space concept in multiconfigurational self-consistent field methods. *J. Chem. Phys.*, 135:044128, 2011.

[11] Björn O. Roos. The multiconfigurational (MC) self-consistent field (SCF) theory. In Björn O. Roos, editor, *Lecture Notes in Quantum Chemistry. European Summer School in Quantum Chemistry*, volume 58 of *Lecture Notes in Chemistry*, pages 177–254. Springer-Verlag, Berlin, Germany, 1992.

[12] James Finley, Per-Åke Malmqvist, Björn O. Roos, and Luis Serrano-Andrés. The multi-state CASPT2 method. *Chem. Phys. Letters*, 288:299–306, 1998.

[13] John D. Watts, Jürgen Gauss, and Rodney J. Bartlett. Coupled-cluster methods with noniterative triple excitations for restricted open-shell Hartree–Fock and other general single determinant reference functions. Energies and analytical gradients. *J. Chem. Phys.*, 98:8718–8733, 1993.

[14] Pavel Neogrády and Miroslav Urban. Spin-adapted restricted Hartree–Fock reference coupled-cluster theory for open-shell systems: Noniterative triples for noncanonical orbitals. *Int. J. Quantum Chem.*, 55:187–203, 1995.

[15] Roland Lindh. The reduced multiplication scheme of the Rys–Gauss quadrature for 1st order integral derivatives. *Theor. Chim. Acta*, 85:423–440, 1993.

[16] Shervin Fatehi and Joseph E. Subotnik. Derivative couplings with built-in electron-translation factors: Application to benzene. *J. Phys. Chem. Lett.*, 3(15):2039–2043, 2012.

[17] Michael Stenrup, Roland Lindh, and Ignacio Fdez. Galván. Constrained numerical gradients and composite gradients: Practical tools for geometry optimization. *J. Comput. Chem.*, 36(22):1698–1708, 2015.

[18] Kerstin Andersson, Per-Åke Malmqvist, Björn O. Roos, Andrzej Sadlej, and Krzysztof Wolinski. Second-order perturbation theory with a CASSCF reference function. *J. Phys. Chem.*, 94:5483–5486, 1990.

[19] Kerstin Andersson, Per-Åke Malmqvist, and Björn O. Roos. Second-order perturbation theory with a complete active space self-consistent field reference function. *J. Chem. Phys.*, 96:1218–1226, 1992.

[20] Per Åke Malmqvist, Kristine Pierloot, Abdul Rehaman Moughal Shahi, Christopher J. Cramer, , and Laura Gagliardi. The restricted active space followed by second-order perturbation theory method: Theory and application to the study of cuo_2 and cu_2o_2 systems. *J. Chem. Phys.*, 128:204109, 2008.

[21] Vicenta Sauri, Luis Serrano-Andrés, Abdul Rehaman Moughal Shahi, Laura Gagliardi, Steven Vancoillie, and Kristine Pierloot. Multiconfigurational second-order perturbation theory restricted active space (RASPT2) method for electronic excited states: A benchmark study. *J. Chem. Theory Comput.*, 7:153–168, 2011.

[22] Björn O. Roos, Markus P. Fülscher, Per-Åke Malmqvist, Manuela Merchán, and Luis Serrano-Andrés. Theoretical studies of the electronic spectra of organic molecules. In Stephen R. Langhoff, editor, *Quantum Mechanical Electronic Structure Calculations with Chemical Accuracy*, volume 13 of *Understanding Chemical Reactivity*, pages 357–438. Kluwer Academic Publishers, Dordrecht, Netherlands, 1995.

[23] Björn O. Roos, Kerstin Andersson, Markus P. Fülscher, Per-Åke Malmqvist, Luis Serrano-Andrés, Kristine Pierloot, and Manuela Merchán. Multiconfigurational perturbation theory: Applications in electronic spectroscopy. In I. Prigogine and Stuart A. Rice, editors, *New Methods in Computational Quantum Mechanics*, volume 93 of *Advances in Chemical Physics*, pages 213–331. John Wiley & Sons, Hoboken, NJ, USA, 1996.

[24] Kerstin Andersson and Björn O. Roos. Multiconfigurational second-order perturbation theory: A test of geometries and binding energies. *Int. J. Quantum Chem.*, 45:591–607, 1993.

[25] Giovanni Ghigo, Björn O. Roos, and Per-Åke Malmqvist. A modified definition of the zeroth-order Hamiltonian in multiconfigurational perturbation theory (CASPT2). *Chem. Phys. Letters*, 396:142–149, 2004.

[26] K. Andersson. Different forms of the zeroth-order Hamiltonian in second-order perturbation theory with a complete active space self-consistent field reference function. *Theor. Chim. Acta*, 91:31–46, 1995.

[27] Björn O. Roos and Kerstin Andersson. Multiconfigurational perturbation theory with level shift — the cr_2 potential revisited. *Chem. Phys. Letters*, 245:215–223, 1995.

[28] Björn O. Roos, Kerstin Andersson, Markus P. Fülscher, Luis Serrano-Andrés, Kristine Pierloot, Manuela Merchán, and Vicent Molina. Applications of level shift corrected perturbation theory in electronic spectroscopy. *J. Mol. Struct. Theochem*, 388:257–276, 1996.

[29] Niclas Forsberg and Per-Åke Malmqvist. Multiconfiguration perturbation theory with imaginary level shift. *Chem. Phys. Letters*, 274:196–204, 1997.

[30] Thorstein Thorsteinsson, David L. Cooper, Joseph Gerratt, Peter B. Karadakov, and Mario Raimondi. Modern valence bond representations of CASSCF wavefunctions. *Theor. Chim. Acta*, 93:343–366, 1996.

[31] David L. Cooper, Thorstein Thorsteinsson, and Joseph Gerratt. Fully variational optimization of modern VB wave functions using the CASVB strategy. *Int. J. Quantum Chem.*, 65:439–451, 1997.

[32] David L. Cooper, Thorstein Thorsteinsson, and Joseph Gerratt. Modern VB representations of CASSCF wave functions and the fully-variational optimization of modern VB wave functions using the CASVB strategy. *Adv. Quantum Chem.*, 32:51–67, 1998.

[33] T. Thorsteinsson and D. L. Cooper. An overview of the CASVB approach to modern valence bond calculations. In Alfonso Hernández-Laguna, Jean Maruani, Roy McWeeny, and Stephen Wilson, editors, *Quantum Systems in Chemistry and Physics. Vol. 1: Basic problems and models systems*, pages 303–326. Kluwer Academic Publishers, Dordrecht, Netherlands, 2000.

[34] Pavel Neogrády, Miroslav Urban, and Ivan Hubač. Spin adapted restricted Hartree–Fock reference coupled cluster theory for open shell systems. *J. Chem. Phys.*, 100:3706–3716, 1994.

[35] Pavel Neogrády, Miroslav Urban, and Ivan Hubač. Spin adapted restricted open shell coupled cluster theory. Linear version. *J. Chem. Phys.*, 97:5074–5080, 1992.

[36] Peter J. Knowles, Claudia Hampel, and Hans-Joachim Werner. Coupled cluster theory for high spin, open shell reference wave functions. *J. Chem. Phys.*, 99:5219–5227, 1993.

[37] Miroslav Urban, Jozef Noga, Samuel J. Cole, and Rodney J. Bartlett. Towards a full CCSDT model for electron correlation. *J. Chem. Phys.*, 83:4041–4046, 1985.

[38] Krishnan Raghavachari, Gary W. Trucks, John A. Pople, and Martin Head-Gordon. A fifth-order perturbation comparison of electron correlation theories. *Chem. Phys. Letters*, 157:479–483, 1989.

[39] Reinhart Ahlrichs, Peter Scharf, and Claus Ehrhardt. The coupled pair functional (CPF). A size consistent modification of the CI(SD) based on an energy functional. *J. Chem. Phys.*, 82:890–898, 1985.

[40] Delano P. Chong and Stephen R. Langhoff. A modified coupled pair functional approach. *J. Chem. Phys.*, 84:5606–5610, 1986.

[41] Robert J. Gdanitz and Reinhart Ahlrichs. The averaged coupled-pair functional (ACPF): A size-extensive modification of MR CI(SD). *Chem. Phys. Letters*, 143:413–420, 1988.

[42] B. Roos. A new method for large-scale Cl calculations. *Chem. Phys. Letters*, 15:153–159, 1972.

[43] Isaiah Shavitt. Graph theoretical concepts for the unitary group approach to the many-electron correlation problem. *Int. J. Quantum Chem.*, 12-S11:131–148, 1977.

[44] Per E. M. Siegbahn. Generalizations of the direct CI method based on the graphical unitary group approach. II. Single and double replacements from any set of reference configurations. *J. Chem. Phys.*, 72:1647–1656, 1980.

[45] William C. Swope, Hans C. Andersen, Peter H. Berens, and Kent R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.*, 76:637–649, 1982.

[46] I. V. Abarenkov. Unit cell for a lattice electrostatic potential. *Phys. Rev. B*, 76:165127, 2007.

[47] Peter V. Sushko and Igor V. Abarenkov. General purpose electrostatic embedding potential. *J. Chem. Theory Comput.*, 6:1323–1333, 2010.

[48] Jorge M. del Campo and Andreas M. Köster. A hierarchical transition state search algorithm. *J. Chem. Phys.*, 129:024107, 2008.

[49] Richard C. Raffenetti. General contraction of Gaussian atomic orbitals: Core, valence, polarization, and diffuse basis sets; molecular integral evaluation. *J. Chem. Phys.*, 58:4452–4458, 1973.

[50] Jan Almlöf and Peter R. Taylor. General contraction of Gaussian basis sets. I. Atomic natural orbitals for first- and second-row atoms. *J. Chem. Phys.*, 86:4070–4077, 1987.

[51] Per-Olof Widmark, Per-Åke Malmqvist, and Björn O. Roos. Density matrix averaged atomic natural orbital (ANO) basis sets for correlated molecular wave functions. I. First row atoms. *Theor. Chim. Acta*, 77:291, 1990.

[52] Per-Olof Widmark, B. Joakim Persson, and Björn O. Roos. Density matrix averaged atomic natural orbital (ANO) basis sets for correlated molecular wave functions. II. Second row atoms. *Theor. Chim. Acta*, 79:419–432, 1991.

[53] Rosendo Pou-Amérigo, Manuela Merchán, Ignacio Nebot-Gil, Per-Olof Widmark, and Björn O. Roos. Density matrix averaged atomic natural orbital (ANO) basis sets for correlated molecular wave functions. III. First row transition metal atoms. *Theor. Chim. Acta*, 92:149–181, 1995.

[54] Kristine Pierloot, Birgit Dumez, Per-Olof Widmark, and Björn O. Roos. Density matrix averaged atomic natural orbital (ANO) basis sets for correlated molecular wave functions. IV. Medium size basis sets for the atoms h–kr. *Theor. Chim. Acta*, 90:87–114, 1995.

[55] Victor P. Vysotskiy, Jonas Boström, and Valera Veryazov. A new module for constrained multi-fragment geometry optimization in internal coordinates implemented in the MOLCAS package. *J. Comput. Chem.*, 34:2657–2665, 2013.

[56] Yubin Wang, Gaohong Zhai, Binbin Suo, Zhengting Gan, and Zhenyi Wen. Hole–particle correspondence in CI calculations. *Chem. Phys. Letters*, 375:134–140, 2003.

[57] Bing Suo, Gaohong Zhai, Yubin Wang, Zhenyi Wen, Xiangqian Hu, and Lemin Li. Parallelization of MRCI based on hole–particle symmetry. *J. Comput. Chem.*, 26:88–96, 2005.

[58] János Pipek and Paul G. Mezey. A fast intrinsic localization procedure applicable for *ab initio* and semiempirical linear combination of atomic orbital wave functions. *J. Chem. Phys.*, 90:4916–4926, 1989.

[59] S. F. Boys. Construction of some molecular orbitals to be approximately invariant for changes from one molecule to another. *Rev. Mod. Phys.*, 32:296–299, 1960.

[60] J. M. Foster and S. F. Boys. Canonical configurational interaction procedure. *Rev. Mod. Phys.*, 32:300–302, 1960.

[61] Clyde Edmiston and Klaus Ruedenberg. Localized atomic and molecular orbitals. *Rev. Mod. Phys.*, 35:457–465, 1963.

[62] Francesco Aquilante, Thomas Bondo Pedersen, Alfredo Sánchez de Merás, and Henrik Koch. Fast noniterative orbital localization for large molecules. *J. Chem. Phys.*, 125:174101, 2006.

[63] Joseph E. Subotnik, Yihan Shao, WanZhen Liang, and Martin Head-Gordon. An efficient method for calculating maxima of homogeneous functions of orthogonal matrices: Applications to localized occupied orbitals. *J. Chem. Phys.*, 121:9220–9229, 2004.

[64] Laura Gagliardi, Roland Lindh, and Gunnar Karlström. Local properties of quantum chemical systems: The LoProp approach. *J. Chem. Phys.*, 121:4494–4500, 2004.

[65] Axel D. Becke and Erin R. Johnson. Exchange-hole dipole moment and the dispersion interaction. *J. Chem. Phys.*, 122:154104, 2005.

[66] Anders Bernhardsson, Roland Lindh, Jeppe Olsen, and Markus Fülscher. A direct implementation of the second-order derivatives of multiconfigurational SCF energies and an analysis of the preconditioning in the associated response equation. *Mol. Phys.*, 96:617–628, 1999.

[67] Jonna Stålring, Anders Bernhardsson, and Roland Lindh. Analytical gradients of a state average MCSCF state and a state average diagnostic. *Mol. Phys.*, 99:103–114, 2001.

[68] Jeppe Olsen, Björn O. Roos, Poul Jørgensen, and Hans Jørgen Aa. Jensen. Determinant based configuration interaction algorithms for complete and restricted configuration interaction spaces. *J. Chem. Phys.*, 89:2185–2192, 1988.

[69] Giovanni Li Manni, Rebecca K. Carlson, Sijie Luo, Dongxia Ma, Jeppe Olsen, Donald G. Truhlar, and Laura Gagliardi. Multi-configuration pair-density functional theory. *J. Chem. Theory Comput.*, 10:3669–3680, 2014.

[70] Rebecca K. Carlson, Giovanni Li Manni, Andrew L. Sonnenberger, Donald G. Truhlar, and Laura Gagliardi. Multiconfiguration pair-density functional theory: Barrier heights and main group and transition metal energetics. *J. Chem. Theory Comput.*, 11:82–90, 2015.

[71] Rebecca K. Carlson, Donald G. Truhlar, and Laura Gagliardi. Multiconfiguration pair-density functional theory: A fully translated gradient approximation and its performance for transition metal dimers and the spectroscopy of $\mathrm{re_2cl_8^{2-}}$. *J. Chem. Theory Comput.*, 11(9):4077, 2015.

[72] A. Wallqvist, P. Ahlström, and G. Karlström. New intermolecular energy calculation scheme: Applications to potential surface and liquid properties of water. *J. Phys. Chem.*, 94:1649–1656, 1990.

[73] Nigel W. Moriarty and Gunnar Karlström. Electronic polarization of a water molecule in water. A combined quantum chemical and statistical mechanical treatment. *J. Phys. Chem.*, 100:17791–17796, 1996.

[74] Anders Öhrn and Gunnar Karlström. A theoretical study of the solvent shift to the $n \to \pi^*$ transition in formaldehyde with an effective discrete quantum chemical solvent model including non-electrostatic perturbation. *Mol. Phys.*, 104:3087–3099, 2006.

[75] Anders Öhrn and Francesco Aquilante. p-benzoquinone in aqueous solution: Stark shifts in spectra, asymmetry in solvent structure. *Phys. Chem. Chem. Phys.*, 9:470–480, 2007.

[76] Anders Öhrn. *Development and Application of a First Principle Molecular Model for Solvent Effects.* PhD thesis, Lunds universitet, Theor. Chemistry, Chem. Center, P.O.B. 124,S-221 00 Lund, Sweden, 2008.

[77] Per-Åke Malmqvist, Alistair Rendell, and Björn O. Roos. The restricted active space self-consistent-field method, implemented with a split graph unitary group approach. *J. Phys. Chem.*, 94:5477–5482, 1990.

[78] Björn O. Roos, Peter R. Taylor, and Per E. M. Siegbahn. A complete active space SCF method (CASSCF) using a density matrix formulated super-CI approach. *Chem. Phys.*, 48:157–173, 1980.

[79] Björn O. Roos. The complete active space self-consistent field method and its applications in electronic structure calculations. In K. P. Lawley, editor, *Ab Initio Methods in Quantum Chemistry Part II*, volume 69 of *Advances in Chemical Physics*, pages 399–445. John Wiley & Sons, Hoboken, NJ, USA, 1987.

[80] Björn O. Roos. The complete active space SCF method in a Fock-matrix-based super-CI formulation. *Int. J. Quantum Chem.*, 18-S14:175–189, 1980.

[81] Francesco Aquilante, Thomas Bondo Pedersen, and Roland Lindh. Low-cost evaluation of the exchange Fock matrix from Cholesky and density fitting representations of the electron repulsion integrals. *J. Chem. Phys.*, 126:194106, 2007.

[82] Per-Åke Malmqvist. Calculation of transition density matrices by nonunitary orbital transformations. *Int. J. Quantum Chem.*, 30:479–494, 1986.

[83] Per-Åke Malmqvist and Björn O. Roos. The CASSCF state interaction method. *Chem. Phys. Letters*, 155:189–194, 1989.

[84] Steven Vancoillie, Per-Åke Malmqvist, and Kristine Pierloot. Calculation of EPR g tensors for transition-metal complexes based on multiconfigurational perturbation theory (CASPT2). *ChemPhysChem*, 8:1803–1815, 2007.

[85] Steven Vancoillie, Lubomír Rulíšek, Frank Neese, and Kristine Pierloot. Theoretical description of the structure and magnetic properties of nitroxide–cu(ii)–nitroxide spin triads by means of multiconfigurational ab initio calculations. *J. Phys. Chem. A*, 113:6149–6157, 2009.

[86] Chad E. Hoyer, Xuefei Xu, Dongxia Ma, Laura Gagliardi, and Donald G. Truhlar. Diabatization based on the dipole and quadrupole: The DQ method. *J. Chem. Phys.*, 141(11):114104, 2014.

[87] Joseph E. Subotnik, Sina Yeganeh, Robert J. Cave, and Mark A. Ratner. Constructing diabatic states from adiabatic states: extending generalized Mulliken-Hush to multiple charge centers with Boys localization. *J. Chem. Phys.*, 129(24):244101, 2008.

[88] J. Almlöf, K. Faegri, Jr., and K. Korsell. Principles for a direct SCF approach to LICAO–MO *ab-initio* calculations. *J. Comput. Chem.*, 3:385–399, 1982.

[89] Dieter Cremer and Jürgen Gauss. An unconventional SCF method for calculations on large molecules. *J. Comput. Chem.*, 7:274–282, 1986.

[90] Marco Häser and Reinhart Ahlrichs. Improvements on the direct SCF method. *J. Comput. Chem.*, 10:104–111, 1989.

[91] Gunnar Karlström. Dynamical damping based on energy minimization for use ab initio molecular orbital SCF calculations. *Chem. Phys. Letters*, 67:348–350, 1979.

[92] Harrell Sellers. The C^2-DIIS convergence acceleration algorithm. *Int. J. Quantum Chem.*, 45:31–41, 1993.

[93] Thomas H. Fischer and Jan Almlöf. General methods for geometry and wave function optimization. *J. Phys. Chem.*, 96:9768–9774, 1992.

[94] S. H. Vosko, L. Wilk, and M. Nusair. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: A critical analysis. *Can. J. Phys.*, 58:1200–1211, 1980.

[95] A. D. Becke. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A*, 38:3098–3100, 1988.

[96] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, 1964.

[97] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, 1965.

[98] J. C. Slater. *Quantum Theory of Molecular and Solids. Vol. 4. The Self-Consistent Field for Molecular and Solids.* McGraw–Hill, New York, NY, USA, 1974.

[99] A. D. Becke. Density functional calculations of molecular bond energies. *J. Chem. Phys.*, 84:4524–4529, 1986.

[100] Axel D. Becke and Erin R. Johnson. A unified density-functional treatment of dynamical, nondynamical, and dispersion correlations. *J. Chem. Phys.*, 127:124108, 2007.

[101] Nicholas C. Handy and Aron J. Cohen. Left–right correlation energy. *Mol. Phys.*, 99:403–412, 2001.

[102] Chengteh Lee, Weitao Yang, and Robert G. Parr. Development of the colle–salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B*, 37:785–789, 1988.

[103] Burkhard Miehlich, Andreas Savin, Hermann Stoll, and Heinzwerner Preuss. Results obtained with the correlation energy density functionals of Becke and Lee, Yang and Parr. *Chem. Phys. Letters*, 157:200–206, 1989.

[104] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Letters*, 77:3865–3868, 1996.

[105] Axel D. Becke. Density-functional thermochemistry. III. The role of exact exchange. *J. Chem. Phys.*, 98:5648–5652, 1993.

[106] Stefan Grimme. Semiempirical hybrid density functional with perturbative second-order correlation. *J. Chem. Phys.*, 124:034108, 2006.

[107] Phillip A. Stewart and Peter M. W. Gill. Becke–Wigner: A simple but powerful density functional. *J. Chem. Soc. Faraday Trans.*, 91:4337–4341, 1995.

[108] Peter M. W. Gill. A new gradient-corrected exchange functional. *Mol. Phys.*, 89:433–445, 1996.

[109] Wee-Meng Hoe, Aaron J. Cohen, and Nicholas C. Handy. Assessment of a new local exchange functional OPTX. *Chem. Phys. Letters*, 341:319–328, 2001.

[110] Mark J. Allen, Thomas W. Keal, and David J. Tozer. Improved NMR chemical shifts in density functional theory. *Chem. Phys. Letters*, 380:70–77, 2003.

[111] Thomas W. Keal and David J. Tozer. A semiempirical generalized gradient approximation exchange-correlation functional. *J. Chem. Phys.*, 121:5654–5660, 2004.

[112] John P. Perdew, Matthias Ernzerhof, and Kieron Burke. Rationale for mixing exact exchange with density functional approximations. *J. Chem. Phys.*, 105:9982–9985, 1996.

[113] Adrienn Ruzsinszky, Gábor I. Csonka, and Gustavo E. Scuseria. Regularized gradient expansion for atoms, molecules, and solids. *J. Chem. Theory Comput.*, 5:763–769, 2009.

[114] Vincent Tognetti, Pietro Cortona, and Carlo Adamo. A new parameter-free correlation functional based on an average atomic reduced density gradient analysis. *J. Chem. Phys.*, 128:034101, 2008.

[115] Marcel Swart, Miquel Solà, and F. Matthias Bickelhaupt. A new all-round density functional based on spin states and S_N2 barriers. *J. Chem. Phys.*, 131:049103, 2009.

[116] Yan Zhao and Donald G. Truhlar. A new local density functional for main-group thermochemistry, transition metal bonding, thermochemical kinetics, and noncovalent interactions. *J. Chem. Phys.*, 125:194101, 2006.

[117] Yan Zhao and Donald G. Truhlar. Density functional for spectroscopy: No long-range self-interaction error, good performance for Rydberg and charge-transfer states, and better performance on average than B3LYP for ground states. *J. Phys. Chem. A*, 110:13126–13130, 2006.

[118] Yan Zhao and Donald G. Truhlar. The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: Two new functionals and systematic testing of four M06-class functionals and 12 other functionals. *Theor. Chem. Acc.*, 120:215–241, 2008.

[119] Yan Zhao and Donald G. Truhlar. Density functionals with broad applicability in chemistry. *Acc. Chem. Res.*, 41:157–167, 2008.

[120] R. Lindh, U. Ryu, and B. Liu. The reduced multiplication scheme of the Rys quadrature and new recurrence relations for auxiliary function based two-electron integral evaluation. *J. Chem. Phys.*, 95:5889–5897, 1991.

[121] Ernest R. Davidson. Use of double cosets in constructing integrals over symmetry orbitals. *J. Chem. Phys.*, 62:400–403, 1975.

[122] Benny G. Johnson, Peter M. W. Gill, and John A. Pople. The performance of a family of density functional methods. *J. Chem. Phys.*, 98:5612–5626, 1993.

[123] Nicholas C. Handy, David J. Tozer, Gregory J. Laming, Christopher W. Murray, and Roger D. Amos. Analytic second derivatives of the potential energy surface. *Isr. J. Chem.*, 33:331–344, 1993.

[124] Jon Baker, Jan Andzelm, Andrew Scheiner, and Bernard Delley. The effect of grid quality and weight derivatives in density functional calculations. *J. Chem. Phys.*, 101:8894–8902, 1994.

[125] Michael E. Mura and Peter J. Knowles. Improved radial grids for quadrature in molecular density-functional calculations. *J. Chem. Phys.*, 104:9848–9858, 1996.

[126] A. D. Becke. A multicenter numerical integration scheme for polyatomic molecules. *J. Chem. Phys.*, 88:2547–2553, 1988.

[127] Christopher W. Murray, Nicholas C. Handy, and Gregory J. Laming. Quadrature schemes for integrals of density functional theory. *Mol. Phys.*, 78:997–1014, 1993.

[128] Oliver Treutler and Reinhart Ahlrichs. Efficient molecular numerical integration schemes. *J. Chem. Phys.*, 102:346–354, 1995.

[129] Roland Lindh, Per-Åke Malmqvist, and Laura Gagliardi. Molecular integrals by numerical quadrature. I. Radial integration. *Theor. Chem. Acc.*, 106:178–187, 2001.

[130] Daoling Peng and Markus Reiher. Exact decoupling of the relativistic Fock operator. *Theor. Chem. Acc.*, 131:1081, 2012.

[131] Daoling Peng and Kimihiko Hirao. An arbitrary order Douglas–Kroll method with polynomial cost. *J. Chem. Phys.*, 130:044102, 2009.

[132] Markus Reiher and Alexander Wolf. Exact decoupling of the Dirac Hamiltonian. I. General theory. *J. Chem. Phys.*, 121:2037–2047, 2004.

[133] Markus Reiher and Alexander Wolf. Exact decoupling of the Dirac Hamiltonian. II. The generalized Douglas–Kroll–Hess transformation up to arbitrary order. *J. Chem. Phys.*, 121:10945–10956, 2004.

[134] Markus Reiher. Douglas–Kroll–Hess theory: a relativistic electrons-only theory for chemistry. *Theor. Chem. Acc.*, 116:241–252, 2006.

[135] Alexander Wolf, Markus Reiher, and Bernd Artur Hess. The generalized Douglas–Kroll transformation. *J. Chem. Phys.*, 117:9215–9226, 2002.

[136] Alexander Wolf and Markus Reiher. Exact decoupling of the Dirac Hamiltonian. III. Molecular properties. *J. Chem. Phys.*, 124:064102, 2006.

[137] Alexander Wolf and Markus Reiher. Exact decoupling of the Dirac Hamiltonian. IV. Automated evaluation of molecular properties within the Douglas–Kroll–Hess theory up to arbitrary order. *J. Chem. Phys.*, 124:064103, 2006.

[138] Werner Kutzelnigg and Wenjian Liu. Quasirelativistic theory equivalent to fully relativistic theory. *J. Chem. Phys.*, 123:241102, 2005.

[139] Wenjian Liu and Daoling Peng. Infinite-order quasirelativistic density functional method based on the exact matrix quasirelativistic theory. *J. Chem. Phys.*, 125:044102, 2006.

[140] Daoling Peng, Wenjian Liu, Yunlong Xiao, and Lan Cheng. Making four- and two-component relativistic density functional methods fully equivalent based on the idea of "from atoms to molecule". *J. Chem. Phys.*, 127:104106, 2007.

[141] Maria Barysz, Andrzej J. Sadlej, and Jaap G. Snijders. Nonsingular two/one-component relativistic Hamiltonians accurate through arbitrary high order in α^2. *Int. J. Quantum Chem.*, 65:225–239, 1997.

[142] Dariusz Këdziera and Maria Barysz. Non-iterative approach to the infinite-order two-component (IOTC) relativistic theory and the non-symmetric algebraic Riccati equation. *Chem. Phys. Letters*, 446:176–181, 2007.

[143] Daoling Peng and Markus Reiher. Local relativistic exact decoupling. *J. Chem. Phys.*, 136:244108, 2012.

[144] Liviu F. Chibotaru, Liviu Ungur, and Alessandro Soncini. The origin of nonmagnetic Kramers doublets in the ground state of dysprosium triangles: Evidence for a toroidal magnetic moment. *Angew. Chem. Int. Ed.*, 47:4126–4129, 2008.

[145] Liviu F. Chibotaru, Liviu Ungur, Christophe Aronica, Hani Elmoll, Guillaume Pillet, and Dominique Luneau. Structure, magnetism, and theoretical study of a mixed-valence $\mathrm{co}_3^{\mathrm{ii}}\mathrm{co}_4^{\mathrm{iii}}$ heptanuclear wheel: Lack of SMM behavior despite negative magnetic anisotropy. *J. Am. Chem. Soc.*, 130:12445–12455, 2008.

[146] L. F. Chibotaru and L. Ungur. *Ab initio* calculation of anisotropic magnetic properties of complexes. I. Unique definition of pseudospin Hamiltonians and their derivation. *J. Chem. Phys.*, 137:064112, 2012.

[147] Roland Lindh, Anders Bernhardsson, Gunnar Karlström, and Per-Åke Malmqvist. On the use of a Hessian model function in molecular geometry optimizations. *Chem. Phys. Letters*, 241:423–428, 1995.

[148] Chunyang Peng, Philippe Y. Ayala, H. Bernhard Schlegel, and Michael J. Frisch. Using redundant internal coordinates to optimize equilibrium geometries and transition states. *J. Comput. Chem.*, 17:49–56, 1996.

[149] P. Pulay and G. Fogarasi. Geometry optimization in redundant internal coordinates. *J. Chem. Phys.*, 96:2856–2860, 1992.

[150] Jon Baker, Alain Kessi, and Bernard Delley. The generation and use of delocalized internal coordinates in geometry optimization. *J. Chem. Phys.*, 105:192–212, 1996.

[151] Roland Lindh, Anders Bernhardsson, and Martin Schütz. Force-constant weighted redundant coordinates in molecular geometry optimizations. *Chem. Phys. Letters*, 303:567–575, 1999.

[152] Jon Baker. Techniques for geometry optimization: A comparison of Cartesian and natural internal coordinates. *J. Comput. Chem.*, 14:1085–1100, 1993.

[153] M. J. Frisch, G. W. Trucks, H. B. Schlegel, P. M. W. Gill, B. G. Johnson, M. A. Robb, J. R. Cheeseman, T. A. Keith, G. A. Petersson, J. A. Montgomery, Jr., K. Raghavachari, M. A. Al-Laham, V. G. Zakrzewski, J. V. Ortiz, J. B. Foresman, J. Cioslowski, B. B. Stefanov, A. Nanayakkara, M. Challacombe, C. Y. Peng, P. Y. Ayala, W. Chen, M. W. Wong, J. L. Andres, E. S. Replogle, R. Gomperts, R. L. Martin, D. J. Fox, J. S. Binkley, D. J. Defrees, J. Baker, J. P. Stewart, M. Head-Gordon, C. Gonzalez, and J. A. Pople. *Gaussian 94 (Revision A.1)*. Gaussian, Inc., Pittsburgh, PA, USA, 1995.

[154] Josep Maria Bofill. Updated Hessian matrix and the restricted step method for locating transition structures. *J. Comput. Chem.*, 15:1–11, 1994.

[155] Josep Maria Bofill. Remarks on the updated Hessian matrix methods. *Int. J. Quantum Chem.*, 94:324–332, 2003.

[156] Ajit Banerjee, Noah Adams, Jack Simons, and Ron Shepard. Search for stationary points on surfaces. *J. Phys. Chem.*, 89:52–57, 1985.

[157] Emili Besalú and Josep Maria Bofill. On the automatic restricted-step rational-function-optimization method. *Theor. Chem. Acc.*, 100:265–274, 1998.

[158] Pál Császár and Péter Pulay. Geometry optimization by direct inversion in the iterative subspace. *J. Mol. Struct.*, 114:31–34, 1984.

[159] Péter Pulay. Convergence acceleration of iterative sequences. The case of SCF iteration. *Chem. Phys. Letters*, 73:393–398, 1980.

[160] P. Pulay. Improved SCF convergence acceleration. *J. Comput. Chem.*, 3:556–560, 1982.

[161] Charles J. Cerjan and William H. Miller. On finding transition states. *J. Chem. Phys.*, 75:2800–2806, 1981.

[162] Satoshi Maeda, Koichi Ohno, and Keiji Morokuma. Updated branching plane for finding conical intersections without coupling derivative vectors. *J. Chem. Theory Comput.*, 6(5):1538–1545, 2010.

[163] John C. Tully. Molecular dynamics with electronic transitions. *J. Chem. Phys.*, 93(2):1061–1071, 1990.

[164] Sharon Hammes-Schiffer and John C. Tully. Proton transfer in solution: Molecular dynamics with quantum transitions. *J. Chem. Phys.*, 101(6):4657–4667, 1994.

[165] Giovanni Granucci and Maurizio Persico. Critical appraisal of the fewest switches algorithm for surface hopping. *J. Chem. Phys.*, 126(13):134114, 2007.

[166] Andrzej J. Sadlej. Medium-size polarized basis sets for high-level correlated calculations of molecular electric properties. *Collect. Czech. Chem. Commun.*, 53:1995–2016, 1988.

[167] Andrzej J. Sadlej. Medium-size polarized basis sets for high-level-correlated calculations of molecular electric properties. II. Second-row atoms: Si through cl. *Theor. Chim. Acta*, 79:123–140, 1991.

[168] Andrzej J. Sadlej and Miroslav Urban. Medium-size polarized basis sets for high-level-correlated calculations of molecular electric properties: III. Alkali (li, na, k, rb) and alkaline-earth (be, mg, ca, sr) atoms. *J. Mol. Struct. Theochem*, 234:147–171, 1991.

[169] Andrzej J. Sadlej. Medium-size polarized basis sets for high-level-correlated calculations of molecular electric properties. IV. Third-row atoms: Ge through br. *Theor. Chim. Acta*, 81:45–63, 1991.

[170] Andrzej J. Sadlej. Medium-size polarized basis sets for high-level-correlated calculations of molecular electric properties. V. Fourth-row atoms: Sn through i. *Theor. Chim. Acta*, 81:339–354, 1992.

[171] Vladimir Kellö and Andrzej J. Sadlej. Estimates of relativistic contributions to molecular properties. *J. Chem. Phys.*, 93:8122–8132, 1990.

[172] Andrzej J. Sadlej and Miroslav Urban. Mutual dependence of relativistic and electron correlation contributions to molecular properties: The dipole moment of agh. *Chem. Phys. Letters*, 176:293–302, 1991.

[173] Sigeru Huzinaga, Luis Seijo, Zoila Barandiarán, and Mariusz Klobukowski. The *ab initio* model potential method. Main group elements. *J. Chem. Phys.*, 86:2132–2145, 1987.

[174] Zoila Barandiarán and Luis Seijo. The *ab initio* model potential representation of the crystalline environment. Theoretical study of the local distortion on nacl:cu$^+$. *J. Chem. Phys.*, 89:5739–5746, 1988.

[175] Zoila Barandiarán, Luis Seijo, and Sigeru Huzinaga. The *ab initio* model potential method. Second series transition metal elements. *J. Chem. Phys.*, 93:5843–5850, 1990.

[176] Christina Wittborn and Ulf Wahlgren. New relativistic effective core potentials for heavy elements. *Chem. Phys.*, 201:357–362, 1995.

[177] Frank Rakowitz, Christel M. Marian, Luis Seijo, and Ulf Wahlgren. Spin-free relativistic no-pair *ab initio* core model potentials and valence basis sets for the transition metal elements sc to hg. Part I. *J. Chem. Phys.*, 110:3678–3686, 1999.

[178] Frank Rakowitz, Christel M. Marian, and Luis Seijo. Spin-free relativistic no-pair *ab initio* core model potentials and valence basis sets for the transition metal elements sc to hg. II. *J. Chem. Phys.*, 111:10436–10443, 1999.

[179] Z. Barandiarán and L. Seijo. Local properties of imperfect crystals. In S. Fraga, editor, *Computational Chemistry: Structure, Interactions and Reactivity*, volume 77B of *Studies in Physical and Theoretical Chemistry*, pages 435–461. Elsevier, Amsterdam, Netherlands, 1992.

[180] James C. Phillips and Leonard Kleinman. New method for calculating wave functions in crystals and molecules. *Phys. Rev.*, 116:287–294, 1959.

[181] S. Huzinaga and A. A. Cantu. Theory of separability of many-electron systems. *J. Chem. Phys.*, 55:5543–5549, 1971.

[182] Sigeru Huzinaga, Dennis McWilliams, and Antonio A. Cantu. Projection operators in Hartree–Fock theory. *Adv. Quantum Chem.*, 7:187–220, 1973.

[183] José Luis Pascual, Luis Seijo, and Zoila Barandiarán. Ab initio model potential study of environmental effects on the Jahn–Teller parameters of cu^{2+} and ag^{2+} impurities in mgo, cao, and sro hosts. *J. Chem. Phys.*, 98:9715–9724, 1993.

[184] M. Pelissier, N. Komiha, and J.-P. Daudey. One-center expansion for pseudopotential matrix elements. *J. Comput. Chem.*, 9:298–302, 1988.

[185] P. Jeffrey Hay and Willard R. Wadt. *Ab initio* effective core potentials for molecular calculations. Potentials for the transition metal atoms sc to hg. *J. Chem. Phys.*, 82:270–283, 1985.

[186] P. Jeffrey Hay and Willard R. Wadt. *Ab initio* effective core potentials for molecular calculations. Potentials for main group elements na to bi. *J. Chem. Phys.*, 82:284–298, 1985.

[187] P. Jeffrey Hay and Willard R. Wadt. *Ab initio* effective core potentials for molecular calculations. Potentials for k to au including the outermost core orbitals. *J. Chem. Phys.*, 82:299–310, 1985.

[188] Patricio Fuentealba, Heinzwerner Preuss, Hermann Stoll, and László Von Szentpály. A proper account of core-polarization with pseudopotentials: Single valence-electron alkali compounds. *Chem. Phys. Letters*, 89:418–422, 1982.

[189] P. Fuentealba, L. von Szentpály, H. Preuss, and H. Stoll. Pseudopotential calculations for alkaline-earth atoms. *J. Phys. B: At. Mol. Phys.*, 18:1287–1296, 1985.

[190] G. Igel-Mann, H. Stoll, and H. Preuss. Pseudopotentials for main group elements (IIIa through VIIa). *Mol. Phys.*, 65:1321–1328, 1988.

[191] Andreas Bergner, Michael Dolg, Wolfgang Küchle, Hermann Stoll, and Heinzwerner Preuß. *Ab initio* energy-adjusted pseudopotentials for elements of groups 13–17. *Mol. Phys.*, 80:1431–1441, 1993.

[192] P. Fuentealba, H. Stoll, L. von Szentpály, P. Schwerdtfeger, and H. Preuss. On the reliability of semi-empirical pseudopotentials: Simulation of Hartree–Fock and Dirac–Fock results. *J. Phys. B: At. Mol. Phys.*, 16:L323–L328, 1983.

[193] M. Kaupp, P v. R. Schleyer, H. Stoll, and H. Preuss. Pseudopotential approaches to ca, sr, and ba hydrides. Why are some alkaline earth mx$_2$ compounds bent? *J. Chem. Phys.*, 94:1360–1366, 1991.

[194] M. Dolg, U. Wedig, H. Stoll, and H. Preuss. Energy-adjusted *ab initio* pseudopotentials for the first row transition elements. *J. Chem. Phys.*, 86:866–872, 1987.

[195] Ulrich Wedig, Michael Dolg, Hermann Stoll, and Heinzwerner Preuss. Energy-adjusted pseudopotentials for transition-metal elements. In A. Veillard, editor, *Quantum Chemistry: The Challenge of Transition Metals and Coordination Chemistry*, volume 176 of *NATO ASI Series*, pages 79–89. D. Reidel, Dordrecht, Netherlands, 1986.

[196] László Von Szentpály, Patricio Fuentealba, Heinzwerner Preuss, and Hermann Stoll. Pseudopotential calculations on rb$_2^+$, cs$_2^+$, rbh$^+$, csh$^+$ and the mixed alkali dimer ions. *Chem. Phys. Letters*, 93:555–559, 1982.

[197] D. Andrae, U. Häußermann, M. Dolg, H. Stoll, and H. Preuß. Energy-adjusted *ab initio* pseudopotentials for the second and third row transition elements. *Theor. Chim. Acta*, 77:123–141, 1990.

[198] H. Stoll, P. Fuentealba, P. Schwerdtfeger, J. Flad, L. v. Szentpály, and H. Preuss. Cu and ag as one-valence-electron atoms: CI results and quadrupole corrections for cu$_2$, ag$_2$, cuh, and agh. *J. Chem. Phys.*, 81:2732–2736, 1984.

[199] W. Küchle, M. Dolg, H. Stoll, and H. Preuss. *Ab initio* pseudopotentials for hg through rn. I. Parameter sets and atomic calculations. *Mol. Phys.*, 74:1245–1263, 1991.

[200] Gudrun Igel-Mann. *Semiempirische Pseudopotentiale; Untersuchungen an Hauptgruppenelementen und Nebengruppenelementen mit abgeschlossener d-Schale.* PhD thesis, Universität Stuttgart, Institut für Theoretische Chemie, 1987.

[201] M. Dolg, H. Stoll, and H. Preuss. A combination of quasirelativistic pseudopotential and ligand field calculations for lanthanoid compounds. *Theor. Chim. Acta*, 85:441–450, 1993.

[202] M. Dolg, H. Stoll, and H. Preuss. Energy-adjusted *ab initio* pseudopotentials for the rare earth elements. *J. Chem. Phys.*, 90:1730–1734, 1989.

[203] Michael Dolg, Peter Fulde, Wolfgang Küchle, Carl-Stefan Neumann, and Hermann Stoll. Ground state calculations of di-π-cyclooctatetraene cerium. *J. Chem. Phys.*, 94:3011–3017, 1991.

[204] M. Dolg, H. Stoll, A. Savin, and H. Preuss. Energy-adjusted pseudopotentials for the rare earth elements. *Theor. Chim. Acta*, 75:173–194, 1989.

[205] Michael Dolg, Hermann Stoll, Heinz-Jüregn Flad, and Heinzwerner Preuss. *Ab initio* pseudopotential study of yb and ybo. *J. Chem. Phys.*, 97:1162–1173, 1992.

[206] Michael Dolg, Hermann Stoll, Heinzwerner Preuss, and Russell M. Pitzer. Relativistic and correlation effects for element 105 (hahnium, ha): A comparative study of m and mo (m = nb, ta, ha) using energy-adjusted ab initio pseudopotentials. *J. Phys. Chem.*, 97:5852–5859, 1993.

[207] Peter Schwerdtfeger, Michael Dolg, W. H. Eugen Schwarz, Graham A. Bowmaker, and Peter D. W. Boyd. Relativistic effects in gold chemistry. I. Diatomic gold compounds. *J. Chem. Phys.*, 91:1762–1774, 1989.

[208] U. Häussermann, M. Dolg, H. Stoll, H. Preuss, P. Schwerdtfeger, and R. M. Pitzer. Accuracy of energy-adjusted quasirelativistic *ab initio* pseudopotentials. All-electron and pseudopotential benchmark calculations for hg, hgh and their cations. *Mol. Phys.*, 78:1211–1224, 1993.

[209] W. Küchle, M. Dolg, H. Stoll, and H. Preuss. Energy-adjusted pseudopotentials for the actinides. Parameter sets and test calculations for thorium and thorium monoxide. *J. Chem. Phys.*, 100:7535–7542, 1994.

[210] Andreas Nicklass, Michael Dolg, Hermann Stoll, and Heinzwerner Preuss. *Ab initio* energy-adjusted pseudopotentials for the noble gases ne through xe: Calculation of atomic dipole and quadrupole polarizabilities. *J. Chem. Phys.*, 102:8942–8952, 1995.

[211] Thierry Leininger, Andreas Nicklass, Hermann Stoll, Michael Dolg, and Peter Schwerdtfeger. The accuracy of the pseudopotential approximation. II. A comparison of various core sizes for indium pseudopotentials in calculations for spectroscopic constants of inh, inf, and incl. *J. Chem. Phys.*, 105:1052–1059, 1996.

[212] Xiaoyan Cao, Michael Dolg, and Hermann Stoll. Valence basis sets for relativistic energy-consistent small-core actinide pseudopotentials. *J. Chem. Phys.*, 118:487–496, 2003.

[213] L. R. Kahn and W. A. Goddard, III. A direct test of the validity of the use of pseudopotentials in molecules. *Chem. Phys. Letters*, 2:667–670, 1968.

[214] Phillip A. Christiansen, Yoon S. Lee, and Kenneth S. Pitzer. Improved *ab initio* effective core potentials for molecular calculations. *J. Chem. Phys.*, 71:4445–4450, 1979.

[215] Philippe Durand and Jean-Claude Barthelat. A theoretical method to determine atomic pseudopotentials for electronic structure calculations of molecules and solids. *Theor. Chim. Acta*, 38:283–302, 1975.

[216] Chris-Kriton Skylaris, Laura Gagliardi, Nicholas C. Handy, Andrew G. Ioannou, Steven Spencer, Andrew Willetts, and Adrian M. Simper. An efficient method for calculating effective core potential integrals which involve projection operators. *Chem. Phys. Letters*, 296:445–451, 1998.

[217] Harry Partridge, Stephen R. Langhoff, and Charles W. Bauschlicher, Jr. Electronic spectroscopy of diatomic molecules. In Stephen R. Langhoff, editor, *Quantum Mechanical Electronic Structure Calculations with Chemical Accuracy*, volume 13 of *Understanding Chemical Reactivity*, pages 209–260. Kluwer Academic Publishers, Dordrecht, Netherlands, 1995.

[218] Peter R. Taylor. Molecular symmetry and quantum chemistry. In Björn O. Roos, editor, *Lecture Notes in Quantum Chemistry. European Summer School in Quantum Chemistry*, volume 58 of *Lecture Notes in Chemistry*, pages 89–176. Springer-Verlag, Berlin, Germany, 1992.

[219] Margareta R. A. Blomberg, Per E. M. Siegbahn, and Björn O. Roos. A theoretical study of nih optical spectrum and potential curves. *Mol. Phys.*, 47:127–143, 1982.

[220] Rosendo Pou-Amérigo, Manuela Merchán, Ignacio Nebot-Gil, Per-Åke Malmqvist, and Björn O. Roos. The chemical bonds in cuh, cu_2, nih, and ni_2 studied with multiconfigurational second order perturbation theory. *J. Chem. Phys.*, 101:4893–4902, 1994.

[221] Kerstin Andersson and Björn O. Roos. Excitation energies in the nickel atom studied with the complete active space SCF method and second-order perturbation theory. *Chem. Phys. Letters*, 191:507–514, 1992.

[222] Gerhard Herzberg. *Molecular Spectra and Molecular Structure. Vol I. Spectra of Diatomic Molecules.* D. Van Nostrand, Princeton, NJ, USA, 2nd edition, 1966.

[223] Peter R. Taylor. Accurate calculations and calibration. In Björn O. Roos, editor, *Lecture Notes in Quantum Chemistry. European Summer School in Quantum Chemistry*, volume 58 of *Lecture Notes in Chemistry*, pages 325–412. Springer-Verlag, Berlin, Germany, 1992.

[224] Remedios González-Luque, Manuela Merchán, and Björn O. Roos. A theoretical determination of the dissociation energy of the nitric oxide dimer. *Theor. Chim. Acta*, 88:425–435, 1994.

[225] M. Perić, B. Engels, and S. D. Peyerimhoff. Theoretical spectroscopy on small molecules: *Ab initio* investigations of vibronic structure, spin–orbit splittings and magnetic hyperfine effects in the electronic spectra of triatomic molecules. In Stephen R. Langhoff, editor, *Quantum Mechanical Electronic Structure Calculations with Chemical Accuracy*, volume 13 of *Understanding Chemical Reactivity*, pages 261–356. Kluwer Academic Publishers, Dordrecht, Netherlands, 1995.

[226] Trygve Helgaker. Optimization of minima and saddle points. In Björn O. Roos, editor, *Lecture Notes in Quantum Chemistry. European Summer School in Quantum Chemistry*, volume 58 of *Lecture Notes in Chemistry*, pages 295–324. Springer-Verlag, Berlin, Germany, 1992.

[227] Mercedes Rubio, Manuela Merchán, Enrique Ortí, and Björn O. Roos. A theoretical study of the electronic spectrum of naphthalene. *Chem. Phys.*, 179:395–409, 1994.

[228] Luis Serrano-Andrés, Manuela Merchán, Ignacio Nebot-Gil, Roland Lindh, and Björn O. Roos. Towards an accurate molecular orbital theory for excited states: Ethene, butadiene, and hexatriene. *J. Chem. Phys.*, 98:3151–3162, 1993.

[229] Luis Serrano-Andrés and Björn O. Roos. Theoretical study of the absorption and emission spectra of indole in the gas phase and in a solvent. *J. Am. Chem. Soc.*, 118:185–195, 1996.

[230] Christopher S. Page, Manuela Merchán, Luis Serrano-Andrés, and Massimo Olivucci. A theoretical study of the low-lying excited states of *trans*- and *cis*-urocanic acid. *J. Phys. Chem. A*, 103:9864–9871, 1999.

[231] Rosendo Pou-Amérigo, Manuela Merchán, and Enrique Ortí. Theoretical study of the electronic spectrum of *p*-benzoquinone. *J. Chem. Phys.*, 110:9536–9546, 1999.

[232] C. E. Blom and A. Bauder. Microwave spectrum, rotational constants and dipole moment of s-*cis* acrolein. *Chem. Phys. Letters*, 88:55–58, 1982.

[233] Vicent Molina and Manuela Merchán. Theoretical analysis of the electronic spectra of benzaldehyde. *J. Phys. Chem. A*, 105:3745–3751, 2001.

[234] Francis Ford, Tetsuro Yuzawa, Matthew S. Platz, Stephan Matzinger, and Markus Fülscher. Rearrangement of dimethylcarbene to propene: Study by laser flash photolysis and *ab Initio* molecular orbital theory. *J. Am. Chem. Soc.*, 120:4430–4438, 1998.

[235] Timothy J. Lee and Peter R. Taylor. A diagnostic for determining the quality of single-reference electron correlation methods. *Int. J. Quantum Chem.*, 36-S23:199–207, 1989.

[236] Timothy J. Lee and Gustavo E. Scuseria. Achieving chemical accuracy with coupled-cluster theory. In Stephen R. Langhoff, editor, *Quantum Mechanical Electronic Structure Calculations with Chemical Accuracy*, volume 13 of *Understanding Chemical Reactivity*, pages 47–108. Kluwer Academic Publishers, Dordrecht, Netherlands, 1995.

[237] S. Matzinger and M. P. Fülscher. Methyl substitution in carbenes. A theoretical prediction of the singlet–triplet energy separation of dimethylcarbene. *J. Phys. Chem.*, 99:10747–10751, 1995.

[238] David J. Tozer, Roger D. Amos, Nicholas C. Handy, Björn O. Roos, and Luis Serrano-Andrés. Does density functional theory contribute to the understanding of excited states of unsaturated organic compounds? *Mol. Phys.*, 97:859–868, 1999.

[239] Luis Serrano-Andrés, Markus P. Fülscher, Björn O. Roos, and Manuela Merchán. Theoretical study of the electronic spectrum of imidazole. *J. Phys. Chem.*, 100:6484–6491, 1996.

[240] Luis Serrano-Andrés. *Estudio teórico del espectro electrónico de sistemas orgánicos.* PhD thesis, Universitat de València, 1994.

[241] Luis Serrano-Andrés, Manuela Merchán, Markus Fülscher, and Björn O. Roos. A theoretical study of the electronic spectrum of thiophene. *Chem. Phys. Letters*, 211:125–134, 1993.

[242] Karl Kaufmann, Werner Baumeister, and Martin Jungen. Universal Gaussian basis sets for an optimum representation of Rydberg and continuum wavefunctions. *J. Phys. B: At. Mol. Opt. Phys.*, 22:2223–2240, 1989.

[243] M. P. Fülscher and B. O. Roos. The excited states of pyrazine: A basis set study. *Theor. Chim. Acta*, 87:403–413, 1994.

[244] Kerstin Andersson. *Multiconfigurational perturbation theory.* PhD thesis, Karlstads universitet, Fakulteten för ekonomi, kommunikation och IT, 1992.

[245] Markus P. Fülscher, Luis Serrano-Andrés, and Björn O. Roos. A theoretical study of the electronic spectra of adenine and guanine. *J. Am. Chem. Soc.*, 119:6168–6176, 1997.

[246] Luis Serrano-Andrés and Markus P. Fülscher. Theoretical study of the electronic spectroscopy of peptides. 1. The peptidic bond: Primary, secondary, and tertiary amides. *J. Am. Chem. Soc.*, 118:12190–12199, 1996.

[247] Manuela Merchán, Enrique Ortí, and Björn O. Roos. Theoretical determination of the electronic spectrum of free base porphin. *Chem. Phys. Letters*, 226:27–37, 1994.

[248] Luis Serrano-Andrés and Björn O. Roos. A theoretical study of the indigoid dyes and their chromophore. *Chem. Eur. J.*, 3:717–725, 1997.

[249] K. Pierloot, E. Van Praet, L. G. Vanquickenborne, and B. O. Roos. Systematic ab initio study of the ligand field spectra of hexacyanometalate complexess. *J. Phys. Chem.*, 97:12220–12228, 1993.

[250] Kristine Pierloot, Jan O. A. De Kerpel, Ulf Ryde, and Björn O. Roos. Theoretical study of the electronic spectrum of plastocyanin. *J. Am. Chem. Soc.*, 119:218–226, 1997.

[251] Kristine Pierloot, Eftimios Tsokos, and Björn O. Roos. 3p–3d intershell correlation effects in transition metal ions. *Chem. Phys. Letters*, 214:583–590, 1993.

[252] Manuela Merchán and Remedios González-Luque. *Ab initio* study on the low-lying excited states of retinal. *J. Chem. Phys.*, 106:1112–1122, 1997.

[253] Luis Serrano-Andrés, Manuela Merchán, Björn O. Roos, and Roland Lindh. Theoretical study of the internal charge transfer in aminobenzonitriles. *J. Am. Chem. Soc.*, 117:3189–3204, 1995.

[254] Manuela Merchán, Rosendo Pou-Amérigo, and Björn O. Roos. A theoretical study of the dissociation energy of ni_2^+. A case of broken symmetry. *Chem. Phys. Letters*, 252:405–414, 1996.

[255] M. P. Fülscher, S. Matzinger, and T. Bally. Excited states in polyene radical cations. An ab initio theoretical study. *Chem. Phys. Letters*, 236:167–176, 1995.

[256] Mercedes Rubio, Manuela Merchán, Enrique Ortí, and Björn O. Roos. A theoretical study of the electronic spectra of the biphenyl cation and anion. *J. Phys. Chem.*, 99:14980, 1995.

[257] Vincenzo Barone and Maurizio Cossi. Quantum calculation of molecular energies and energy gradients in solution by a conductor solvent model. *J. Phys. Chem. A*, 102:1995–2001, 1998.

[258] Maurizio Cossi, Nadia Rega, Giovanni Scalmani, and Vincenzo Barone. Polarizable dielectric model of solvation with inclusion of charge penetration effects. *J. Chem. Phys.*, 114:5691–5701, 2001.

[259] Gunnar Karlström. New approach to the modeling of dielectric media effects in ab initio quantum chemical calculations. *J. Phys. Chem.*, 92:1315–1318, 1988.

[260] Luis Serrano-Andrés, Markus P. Fülscher, and Gunnar Karlström. Solvent effects on electronic spectra studied by multiconfigurational perturbation theory. *Int. J. Quantum Chem.*, 65:167–181, 1997.

[261] Jacopo Tomasi and Maurizio Persico. Molecular interactions in solution: An overview of methods based on continuous distributions of the solvent. *Chem. Rev.*, 94:2027–2094, 1994.

[262] Maurizio Cossi and Vincenzo Barone. Solvent effect on vertical electronic transitions by the polarizable continuum model. *J. Chem. Phys.*, 112:2427–2435, 2000.

[263] Anders Bernhardsson, Roland Lindh, Gunnar Karlström, and Björn O. Roos. Direct self-consistent reaction field with Pauli repulsion: Solvation effects on methylene peroxide. *Chem. Phys. Letters*, 251:141–149, 1996.

[264] W. F. Forbes and R. Shilton. Electronic spectra and molecular dimensions. III. Steric effects in methyl-substituted α,β-unsaturated aldehydes. *J. Am. Chem. Soc.*, 81:786–790, 1959.

[265] Marvin Douglas and Norman M. Kroll. Quantum electrodynamical corrections to the fine structure of helium. *Ann. Phys.*, 82:89–155, 1974.

[266] Bernd A. Hess. Relativistic electronic-structure calculations employing a two-component no-pair formalism with external-field projection operators. *Phys. Rev. A*, 33:3742–3748, 1986.

[267] Per-Åke Malmqvist, Björn O. Roos, and Bernd Schimmelpfennig. The restricted active space (RAS) state interaction approach with spin–orbit coupling. *Chem. Phys. Letters*, 357:230–240, 2002.

[268] Bernd A. Heß, Christel M. Marian, Ulf Wahlgren, and Odd Gropen. A mean-field spin-orbit method applicable to correlated wavefunctions. *Chem. Phys. Letters*, 251:365–371, 1996.

[269] B. Schimmelpfennig. Amfi, an atomic mean-field spin-orbit integral program. Computer code, 1996. University of Stockholm.

[270] Björn O. Roos and Per-Åke Malmqvist. On the effects of spin–orbit coupling on molecular properties: Dipole moment and polarizability of pbo and spectroscopic constants for the ground and excited states. *Adv. Quantum Chem.*, 47:37–49, 2004.

[271] Ulf Wahlgren. The effective core potential method. In Björn O. Roos, editor, *Lecture Notes in Quantum Chemistry. European Summer School in Quantum Chemistry*, volume 58 of *Lecture Notes in Chemistry*, pages 413–421. Springer-Verlag, Berlin, Germany, 1992.

[272] Luis Seijo and Zoila Barandiarán. The *Ab Initio* model potential method: A common strategy for effective core potential and embedded cluster calculations. In Jerzy Leszczynski, editor, *Computational Chemistry: Reviews of Current Trends*, volume 4, pages 55–152. World Scientific, Singapore, 1999.

[273] A. D. Buckingham. Permanent and induced molecular moments and long-range intermolecular forces. *Adv. Chem. Phys.*, 12:107–142, 1967.

[274] Guido Raos, Joseph Gerratt, David L. Cooper, and Mario Raimondi. Spin correlation in π-electron systems from spin-coupled wavefunctions. I. Theory and first applications. *Chem. Phys.*, 186:233–250, 1994.

[275] Guido Raos, Joseph Gerratt, David L. Cooper, and Mario Raimondi. Spin correlation in π-electron systems from spin-coupled wavefunctions. II. Further applications. *Chem. Phys.*, 186:251–273, 1994.

[276] R. Fletcher. A new approach to variable metric algorithms. *Comput. J.*, 13:317–322, 1970.

[277] Gunnar Karlström, Roland Lindh, Per-Åke Malmqvist, Björn O. Roos, Ulf Ryde, Valera Veryazov, Per-Olof Widmark, Maurizio Cossi, Bernd Schimmelpfennig, Pavel Neogrády, and Luis Seijo. MOLCAS: a program package for computational chemistry. *Comput. Mater. Sci.*, 28:222–239, 2003.

[278] G. A. Gallup and J. M. Norbeck. Population analyses of valence-bond wavefunctions and beh$_2$. *Chem. Phys. Letters*, 21:495–500, 1973.

[279] Thomas A. Halgren and William N. Lipscomb. The synchronous-transit method for determining reaction pathways and locating molecular transition states. *Chem. Phys. Letters*, 50:225–232, 1977.

[280] Ola Engkvist, Per-Olof Åstrand, and Gunnar Karlström. Accurate intermolecular potentials obtained from molecular wave functions: Bridging the gap between quantum chemistry and molecular simulations. *Chem. Rev.*, 100:4087–4108, 2000.

[281] Isaiah Shavitt. Matrix element evaluation in the unitary group approach to the electron correlation problem. *Int. J. Quantum Chem.*, 14-S12:5–32, 1978.

[282] David L. Cooper, Robert Ponec, Thorstein Thorsteinsson, and Guido Raos. Pair populations and effective valencies from ab initio SCF and spin-coupled wave functions. *Int. J. Quantum Chem.*, 57:501–518, 1996.

[283] Thorstein Thorsteinsson and David L. Cooper. Modern valence bond descriptions of molecular excited states: An application of CASVB. *Int. J. Quantum Chem.*, 70:637–650, 1998.

[284] Ulf Ryde and Mats H. M. Olsson. Structure, strain, and reorganization energy of blue copper models in the protein. *Int. J. Quantum Chem.*, 81:335–347, 2001.

[285] Valera Veryazov, Per-Olof Widmark, Luis Serrano-Andrés, Roland Lindh, and Björn O. Roos. MOLCAS as a development platform for quantum chemistry software. *Int. J. Quantum Chem.*, 100:626–635, 2004.

[286] Ulf Ryde, Mats H. M. Olsson, Björn O. Roos, Jan O. A. De Kerpel, and Kristine Pierloot. On the role of strain in blue copper proteins. *J. Biol. Inorg. Chem.*, 5:565–574, 2000.

[287] Per E. M. Siegbahn, Jan Almlöf, J. Heiberg, and Björn O. Roos. The complete active space SCF (CASSCF) method in a Newton–Raphson formulation with application to the hno molecule. *J. Chem. Phys.*, 74:2384–2396, 1981.

[288] C. G. Broyden. The convergence of a class of double-rank minimization algorithms 2. The new algorithm. *J. Inst. Math. Appl.*, 6:222–231, 1970.

[289] Ulf Ryde, Mats H. M. Olsson, Kristine Pierloot, and Björn O. Roos. The cupric geometry of blue copper proteins is not strained. *J. Mol. Biol.*, 261:586–596, 1996.

[290] Thorstein Thorsteinsson and David L. Cooper. Nonorthogonal weights of modern VB wavefunctions. Implementation and applications within CASVB. *J. Math. Chem.*, 23:105–106, 1998.

[291] Donald Goldfarb. A family of variable-metric methods derived by variational means. *Math. Comput.*, 24:23–26, 1970.

[292] D. F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Math. Comput.*, 24:647–656, 1970.

[293] M. J. D. Powell. Recent advances in unconstrained optimization. *Math. Program.*, 1:26–57, 1971.

[294] Ove Christiansen, Jürgen Gauss, and Bernd Schimmelpfennig. Spin–orbit coupling constants from coupled-cluster response theory. *Phys. Chem. Chem. Phys.*, 2:965–971, 2000.

[295] B. H. Chirgwin and C. A. Coulson. The electronic structure of conjugated systems. VI. *Proc. Roy. Soc. Lond. A*, 201:196–209, 1950.

[296] J. E. Dennis, Jr. and R. B. Schnabel. Least change secant updates for quasi-Newton methods. *SIAM Rev.*, 21:443–459, 1979.

[297] Thorstein Thorsteinsson, David L. Cooper, Joseph Gerratt, and Mario Raimondi. Symmetry adaptation and the utilization of point group symmetry in valence bond calculations, including CASVB. *Theor. Chim. Acta*, 95:131–150, 1997.

[298] Martin Schütz and Roland Lindh. An integral direct, distributed-data, parallel MP2 algorithm. *Theor. Chim. Acta*, 95:13–34, 1997.

[299] H. Bernhard Schlegel. Optimization of equilibrium geometries and transition structures. In K. P. Lawley, editor, *Ab Initio Methods in Quantum Chemistry Part I*, volume 67 of *Advances in Chemical Physics*, pages 249–286. John Wiley & Sons, Hoboken, NJ, USA, 1987.

[300] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, Chichester, West Sussex, England, 2nd edition, 1987.

[301] Warren J. Hehre. *Practical Strategies for Electronic Structure Calculations*. Wavefunction, Irvine, CA, USA, 1995.

[302] Francesco Aquilante, Jochen Autschbach, Rebecca K. Carlson, Liviu F. Chibotaru, Mickaël G. Delcey, Luca De Vico, Ignacio Fdez. Galván, Nicolas Ferré, Luis Manuel Frutos, Laura Gagliardi, Marco Garavelli, Angelo Giussani, Chad E. Hoyer, Giovanni Li Manni, Hans Lischka, Dongxia Ma, Per Åke Malmqvist, Thomas Müller, Artur Nenov, Massimo Olivucci, Thomas Bondo Pedersen, Daoling Peng, Felix Plasser, Ben Pritchard, Markus Reiher, Ivan Rivalta, Igor Schapiro, Javier Segarra-Martí, Michael Stenrup, Donald G. Truhlar, Liviu Ungur, Alessio Valentini, Steven Vancoillie, Valera

Veryazov, Victor P. Vysotskiy, Oliver Weingart, Felipe Zapata, and Roland Lindh. Molcas 8: New capabilities for multiconfigurational quantum chemical calculations across the periodic table. *J. Comput. Chem.*, 37(5):506–541, 2016.

List of Figures

4.1 *The acrolein molecule.* 38

6.1 Flowchart for Module Dependencies in MOLCAS 70

6.2 The geometry of the water molecule 73

6.3 Sample input requesting the SCF module to calculate the ground Hartree-Fock energy for a neutral water molecule in C_{2v} symmetry. 76

6.4 Symmetry adapted Basis Functions from a GATEWAY output. 76

6.5 Molecular orbitals from the first symmetry species of a calculation of water using C_{2v} symmetry and a minimal basis set. 78

6.6 Sample input requested by the MBPT2 module to calculate the MP2 energy for the ground state of the water in C_{2v} symmetry. 80

6.7 Sample input requesting the RASSCF module to calculate the eight-electrons-in-six-orbitals CASSCF energy of the second excited triplet state in the second symmetry group of a water molecule in C_{2v} symmetry. 80

6.8 RASSCF orbital space including keywords and electron occupancy ranges. . . 82

6.9 RASSCF portion of output relating to CI configurations and electron occupation of natural orbitals. 83

6.10 Sample input requesting the CASPT2 module to calculate the CASPT2 energy of a water molecule in C_{2v} symmetry with one frozen orbital. 85

6.11 Sample input requesting the RASSI module to calculate the matrix elements and expectation values for eight interacting RASSCF states 87

6.12 Sample input requesting the RASSI module to calculate and diagonalize the spin-orbit Hamiltonian the ground and triplet excited state in water. 87

6.13 Sample input containing the files required by the SEWARD, SCF, RASSCF, MOTRA, CCSORT, CCSD, and CCT3 programs to compute the ground state of the HF^+ cation. 96

6.14 Sample input requesting the GENANO module to average three sets of natural orbitals on the oxygen atom. 100

8.1 *Pictorial representation of GAS active space.* 326

10.1 Sample input of the SEWARD program for the magnesium porphirin molecule in the D_{2h} symmetry . . . 450

10.2 1,3-cyclopentadiene . . . 452

10.3 Twisted biphenyl molecule . . . 461

10.4 Acrolein geometrical isomers . . . 468

10.5 Reactant . . . 471

10.6 Product . . . 471

10.7 Transition state . . . 477

10.8 Dimethylcarbene to propene reaction path . . . 477

10.9 Dimethylcarbene atom labeling . . . 478

10.10Thiophene . . . 491

10.11Radial extent of the Rydberg orbitals . . . 496

10.12Guanine . . . 519

10.13N,N-dimethylaminobenzonitrile (DMABN) . . . 525

10.14Partition of a valence basis set using the ECP's library . . . 544

10.15Sample input required by SEWARD and SCF programs to compute the SCF wave function of HAt using a relativistic ECP . . . 545

10.16Sample input for an embedded core potential for a shell of potassium cations 546

10.17Sample input for a SCF geometry optimization of the $(TlF_{12})^{11-} : KMgF_3$ system . . . 547

List of Tables

1.1 Modules in MOLCAS which benefit from parallel processing. 9

6.1 Symmetries available in MOLCAS including generators, MOLCAS keywords and symmetry elements. 72

6.2 Examples of types of wave functions obtainable using the RAS1 and RAS3 spaces in the RASSCF module. 81

10.1 Classification of the spherical harmonics in the $C_{\infty v}$ group. 424

10.2 Classification of the spherical harmonics and $C_{\infty v}$ orbitals in the C_{2v} group. . 424

10.3 MOLCAS labeling of the spherical harmonics. 425

10.4 Resolution of the $C_{\infty v}$ species in the C_{2v} species. 430

10.5 Classification of the spherical harmonics in the $D_{\infty h}$ group^a. 438

10.6 Classification of the spherical harmonics and $D_{\infty h}$ orbitals in the D_{2h} group^a. 439

10.7 Resolution of the $D_{\infty h}$ species in the D_{2h} species. 440

10.8 Geometrical parameters for the ground state of acrolein 467

10.9 Bond distances (Å) and bond angles (deg) of dimethylcarbene, propene, and their transition state^a 479

10.10Total (au) and relative (Kcal/mol, in braces) energies obtained at the different theory levels for the reaction path from dimethylcarbene to propene 486

10.11Mulliken's population analysis (partial charges) for the reaction path from dimethylcarbene to propene. MRCI wave functions. 490

10.12Selection of active spaces in thiophene. 493

10.13Labeling for the configurations in caspt2. 508

10.14Excitation energies and reference weights of thiophene for different level shift values. 512

10.15CASSCF and CASPT2 excitation energies (eV), oscillator strengths (f), dipole moments (μ(D)), and transition moment directions (Θ) of singlet valence excited states of guanine^a. The Rydberg orbitals have not been included in the active space. 519

10.16 CASSCF and CASPT2 excitation energies (eV), oscillator strengths (f), dipole moments (μ(D)), and transition moment directions (Θ) of singlet valence excited states of guanine^{a,b}. The Rydberg orbitals have been first included in the active space and then deleted. 520

10.17 Ground state CASSCF energies for DMABN with different cavity sizes. . . . 527

10.18 Ground state CASSCF energies for different translations with respect to the initial position of of the DMABN molecule in a 13.8 au cavity. 527

10.19 Ground state CASSCF energies for DMABN with different cavity sizes. The molecule position in the cavity has been optimized. 527

10.20 Vertical excitation energies/eV (solvatochromic shifts) of s-trans acrolein in gas-phase and in aqueous solution 533