Overview of Molcas

Valera Veryazov

Valera.Veryazov@teokem.lu.se

Department of Theoretical Chemistry
Lund University

Today we will learn..

- Installation of Molcas
- Maintenance
- HowTo run Molcas
- HowTo get help
- HowTo create input
- General structure of the input
- HowTo use GUI

What is MOLCAS?

Software for Quantum Chemistry

- for ground and excited states,
- for molecular structure, chemical and photochemical reactions,
- for solution chemistry, chemical bonding, and much more,
- for all elements of the periodic table.

Most important point: multiconfigurational theory CASSCF/RASSCF and CASPT2/RASPT2

Features and methods

- Electronic structure methods
  - Hartree-Fock and DFT for closed and open shell systems.
  - Møller-Plesset second order perturbation theory.
  - Multiconfigurational SCF: CASSCF and RASSCF.
  - Multiconfigurational second order perturbation theory CASPT2 and RASPT2.
  - Multi-reference CI MRCI.‘
  - Coupled-cluster methods CCSD, CCSD(T).
- Geometry optimization
  - Analytical gradients for SCF/DFT and RASSCF.
  - Numerical gradients for CASPT2, Coupled-Cluster methods
  - Automatic geometry optimization of equilibrium geometries, transition states, intersystem crossing, conical intersections, etc.
  - Vibrational frequencies and thermodynamic quantities.

A code called Molcas

- Source code distribution:
- source code (Fortran, C) + scripts (Perl, bash)
- can be configured for
  - ♦ different platforms: Unix, Linux, Windows, MacOS
  - ♦ serial, parallel (SMP or cluster), grid
  - ♦ 32- and 64- bit
  - ♦ all major compilers: gfortran, Intel, PGI, NAG, SunStudio
  - ♦ all major MPI libraries: OpenMPI, intel MPI.
  - ♦ all major BLAS libraries: MKL, ACML, NVBLAS
- Binary ONLY distribution:
  - ♦ Precompiled binaries for main Linux distributions with OpenMPI and MKL
- *Molcas@UU* distribution

Configuration and installation

- Hardware requirements
  - large amount of RAM
  - fast HDD or SSD
  - Cores vs. memory
- MATH libraries: BLAS and LAPACK
- compilers
- MPI libraries
- GPUs support via NVBLAS

Understanding of molcas command

- molcas driver (which version to run)
- molcas.exe (license and parser)
- molcas command
  - molcas input [flags]
  - molcas tool [arguments]
- Multiple installations of Molcas
  - case 1: Current (or parent) directory is MOLCAS
  - case 2: MOLCAS environment is set
  - case 3: latest molcas installation (.Molcas/molcas)

Help!!

- on-line documentation *www.molcas.org*
- molcas help command
  - *molcas help*
  - *molcas help scf*
  - *molcas help scf charge*
  - *molcas help -t b3lyp*
  - *molcas help environment MOLCAS_OUTPUT*
  - *molcas help basis Cu*
- Solving problems
  - Molcas user's billboard
  - Bug report system
  - direct mail: *molcas@molcas.org*

Molcas on farm

- Molcas official farm http://www.molcas.org/farm/
- Course farm:
- Farm run vs. local run: reuse of WorkDir

A first molcas run

molcas Water.DFT.input -f
after run:

- Output
  - Water.DFT.log
  - Water.DFT.err
- Extra files
  - Water.DFT.ScfOrb
  - Water.DFT.scf.molden
- Intermediate files
  - /tmp/Water.DFT.5024/Water.DFT.RunFile
  - /tmp/Water.DFT.5024/Water.DFT.OrdInt

MOLCAS environment variables

- *molcas help environment*
- molcas setuprc: interactive script to make global settings:
  where to store intermediate files, how to name files, etc.

Examples:

- *MOLCAS_WORKDIR* parent directory for all scratch directories
- *MOLCAS_PRINT* - set up print level
- *MOLCAS_OUTPUT* - to set up output directory for additional output files
- *Project* - set up Project name
- *WorkDir* - set up scratch area ($MOLCAS_WORKDIR/$Project)

Usage:

- molcasrc file (created by setuprc)
- export KEY=VALUE
- molcas KEY=VALUE input

Running molcas

- output goes to screen:
  - *molcas input_file*
- output goes to log and error files:
  - *molcas input_file -o logfile -e errorfile*
  - *molcas input_file >logfile 2>errorfile*
- output goes to log file:
  - *molcas input_file -o logfile -e logfile*
  - *molcas input_file >logfile 2>&1*
- output goes to log and error files (.log and .err) :
  - *molcas* **-f** *file.input*
- include environment into command:
  - *molcas MOLCAS_PRINT = 3 input_file -f*
  - *CPUS = 2; export CPUS; molcas input_file*

input example

```
/* this is an example */
&GATEWAY
   COORD = water.xyz
   Basis = ANO-S-MB
&SEWARD
&SCF
   KSDFT = B3LYP
```

Processing of molcas input:

- preprocessing
  - clean and normalize input file
- execute molcas modules or commands
- control of the execution (via return code)
  - interrupt a calculation
  - make loops
  - auto-magically call modules

Understanding molcas input

- Enhanced Molcas Input Language (EMIL)
  - Comment lines: $*$ at the beginning of line, or C++ style //
  - /* */ multiline comments
  - &*MODULE* is a call of module *MODULE*
  - strings $*VAR* replaced by the value of *VAR*
  - Short notation: ; and = replaced by a new line
  - EMIL commands started from > sign
    (>>>>>>>>>>> *COMMAND* <<<<<<<<<<<<)
  - @GEOM or @DFT(B3LYP) - Alias (macros)

EMIL commands: Loops

- geometry optimization loop (terminated by external condition)

```
&GATEWAY
....
>>>> Do While <<<<
&SEWARD
....
&SLAPAF
>>>> EndDo    <<<<
```

EMIL: input example

```
/* this is an example of geometry
   optimization of
   water molecule using DFT */
&GATEWAY
COORD=water.xyz
Basis=ANO-S-MB
>>>>Do While
&SEWARD
&SCF
KSDFT=B3LYP
&ALASKA; &SLAPAF
>>>> EndDo
```

EMIL: Simple calculations

- Syntax:

```
>>>>Export VAR=VALUE
>>>> Eval C=$A+$B
```

- Example

```
>>>>Export DIST=1.0
>>>>Eval R=$Dist+0.1
```

EMIL parallel commands

- ```
  >>>> UNIX echo 'Hello world!'
  >>>> UNIX -SERIAL echo 'Hello world!'
  >>>> COPY File1 File2
  >>>> LINK File1 File2
  >>>> LINK FORCE File1 File2
  >>>> RM File
  ```

- Note all commands are executed in WorkDir
- Submit directory has an alias $CurrDir
  ```

loops with EMIL commands

```
>>>> ForEach VAR IN ( 1, 2, 3 ) <<<<
>>>> ForEach VAR IN ( 1..3 ) <<<<
....
>>>> EndDo    <<<<
```

```
* incremental change of coordinates
>>export DIST=1.0
>>foreach L in ( 1..30 )
>>eval R=$DIST+0.1*$L
&GATEWAY
Coord
2
hydrogen molecule
H 0 0 0
H $R 0 0
BASIS= ANO-S-MB
&SEWARD; &SCF
>>enddo
```

Gateway module

- Coordinates
  - ’Native’ molcas input
  - XYZ input (as a separate file or inline)

```
2  <- Number of atoms
comment line <- ( a.u., transformation)
O  0.0 0.0 0.0 <-  element(†)
C  1.2 0.0 0.0     cartesian coordinates
```

(†) Element name can contain a basis set label.

- Basis set
  - RI/CD basis set
- Symmetry
- External field (PCM,Xfield)

'Native' molcas input

- default symmetry: C_1
- default units: atomic
- only symmetry unique atoms
- atoms with unique labels

```
&GATEWAY
Title =  water, ano-s(dzp) basis set
Symmetry =  x y
Basis set
H.ano-s...2s1p.
H1 -0.783975899  0.000000000 -0.184686472 Angstrom
End of basis
Basis set
O.ano-s...3s2p1d.
O  0.0   0.0 .369372944 Angstrom
End of basis
```

XYZ input

- default symmetry is highest available (D_{2h} subgroup)
- default units: Ångstrom
- 'plain' xyz formatted file (inline or an external file)
- Global or local basis set labels

```
&GATEWAY
COORD
3
water molecule (in Angstrom)
H -0.783975899  0.000000000 -0.184686472
H  0.783975899  0.000000000 -0.184686472
O  0.000000000  0.000000000 .36937294400
Basis
ANO-S-VDZP
```

XYZ input

```
   Basis
   ANO-S-VDZP, H.ANO-S-VDZP, O.ANO-S...3s2p1d.
   Group = C1
*  Group = x y
```

- for C_1 group: Group=C1, or NoSym
- To 'freeze' the molecule: NoMove

Basis sets

- $MOLCAS/basis_library directory
- inline basis sets *vs.* 'standard' basis sets
- Basis set types (basistype.tbl)
  segmented/ANO, pseudopotential, relativistic

```
ANO-RCC            ANO    AE_    RH_
```

- aliases (basis.tbl)

```
H.ANO-S-MB     H.ANO-S...1s.
H.ANO-S-VDZ    H.ANO-S...2s.
H.ANO-S-VDZP   H.ANO-S...2s1p.
H.ANO-S-VTZP   H.ANO-S...3s2p1d.
```

- file aliases (trans.tbl)

```
6-31G*  6-31Gp
```

Communication between codes

```
&GATEWAY
 ...
&SEWARD;  &SCF
```

- GATEWAY
  - create new RUNFILE
  - pass returncode
- SEWARD
  - compute integrals
  - create GssOrb file
  - update RUNFILE
  - pass returncode
- SCF
  - check RUNFILE for starting orbitals
  - check ScfOrb (from another SCF run) or GssOrb file
  - compute WF, and create ScfOrb file
  - update RUNFILE

Reuse of WorkDir

How to set WorkDir?

- Special case: not set
  WorkDir – /tmp/water.$RANDOM
- WorkDir=/scratch/molcas/water/
- MOLCAS_WORKDIR=/scratch/molcas/
  the actual WorkDir name constructed from MOLCAS_WorkDir + the name of Project (input filename).

Should one reuse WorkDir?

- Yes, if want to reuse data, e.g. starting orbitals
- No, if a new calculation is too different

To run calculation with new Workdir

- rm -fr $Workdir
- MOLCAS_NEW_WORKDIR=YES
- *molcas -new input*

GRID_IT module

- Compute cartesian grid from an Orbital file
- for HF/DFT, RASSCF, CASPT2 the resulting file: $Project.grid
- for UHF GRID_IT produces $Project.a.grid and $Project.b.grid
- The code is very expensive for 'nice pictures' (DENSE; ALL)
  - use SPARSE keyword
  - select Orbitals to compute: Select, ORange, ERange
- to run GRID_IT one needs only Gateway and INPORB:
  Link your orbital file to INPORB or use FILEORB keyword

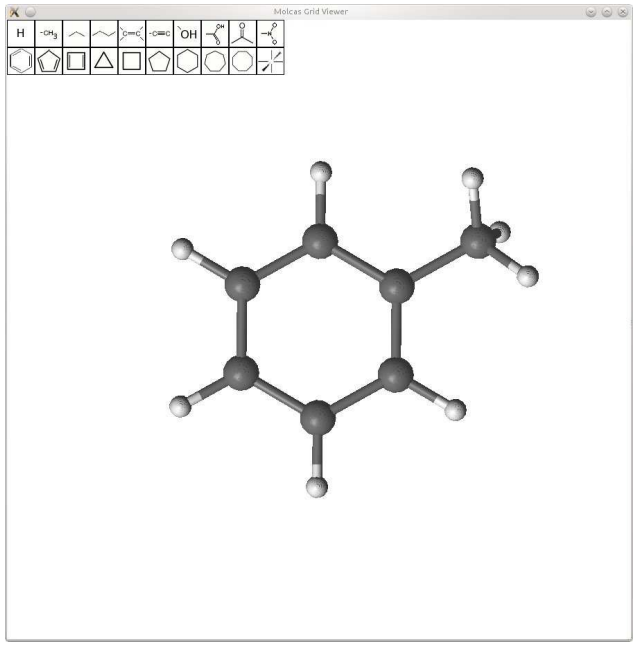
Computed .grid files can be visualized by *gv.exe*. GV code is available at www.molcas.org/GV/

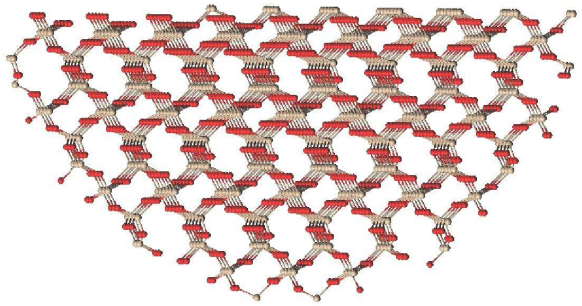
- gv.exe water.grid
- gv.exe -a -1.0 CH3.a.grid CH3.b.grid −−out diff.grid
- gv.exe -a -1.0 CH3.HF.grid CH3.DFT.grid −−out delta.grid

Grid/Geometry Viewer/Editor (GV)

- visualization of coordinates:
  gv.exe file.xyz
- simple editing of coordinates
- visualization of densities and orbitals
  gv.exe file.grid gv.exe -a -1.0 file1.grid file2.grid ––out res.grid
- selection of active space
- visualization of molden files
  gv.exe file.geo.molden
  gv.exe file.freq.molden
  gv.exe file.scf.molden

GV in XYZ mode



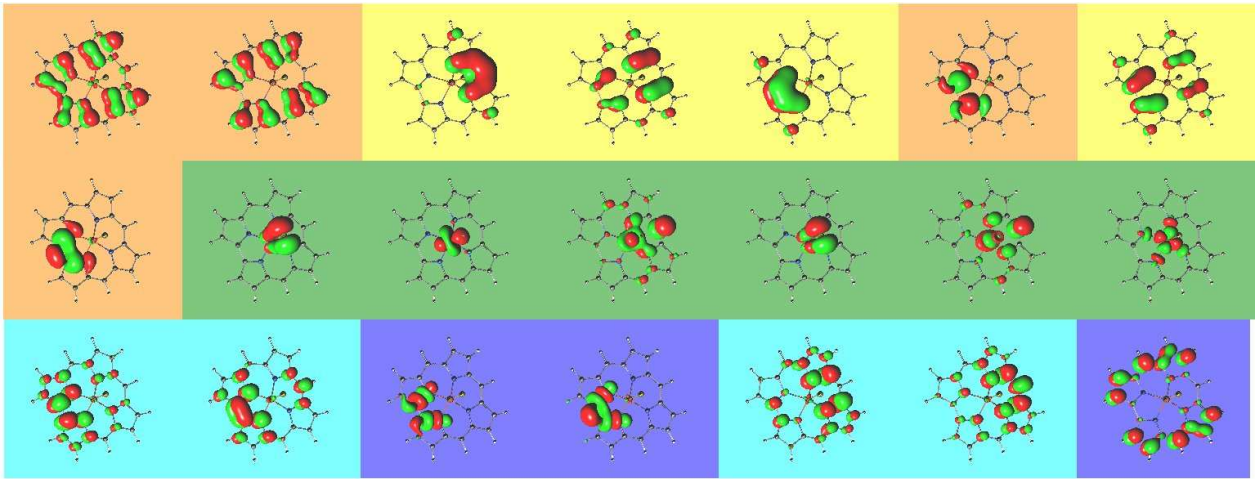


Hints for GV in XYZ mode

- Read the manual and tutorial!
- gv.exe -help, or F1
- if gv edit XYZ file:
  - left mouse click - select an atom (up to 4)
  - left mouse + Shift - mark an atom (for group operations)
  - middle mouse click (or Space) - unselect atoms
  - +/− change a value of the bond/angle
  - PageUp/PageDown change a property of atom/bond
  - Insert/Delete insert/delete selected atom
  - F2 - save XYZ file
  - F8 - check/apply symmetry
  - F3 - fragments menu
  - F4 - edit mode
  - Backspace - Undo

GV in grid mode

Visual selection of the active space



Hints for GV in other modes

- grid file (obtained by call to *GRID_IT* module)
  - $+/-$ change an isovalue
  - PageUp/PageDown change an orbital
  - fi1a3sd - set orbital type
  - F3 - display all computed orbitals on one screen
  - F2 - save Orbital file
- geo.molden files
  - PageUp/PageDown show next geometry
  - F3 - show convergence
- freq.molden files
  - PageUp/PageDown show next vibration
  - F3 - show spectrum

New generation of GV: LUSCUS

- source code at sourceforge.net/projects/luscus/
- nice and intuitive GUI interface
- plug-ins to in- and out- formats
- symmetry library MSYM (any point group)