On the numerical accuracy of implemented QC algorithms with respect to precision of input/processed data

Victor P. Vysotskiy, Valera Veryazov, Per Åke Malmqvist

Theoretical Chemistry, Chemical Center, P.O.B. 124, Lund 22100, Sweden, E-mail: Victor.Vysotskiy@teokem.lu.se

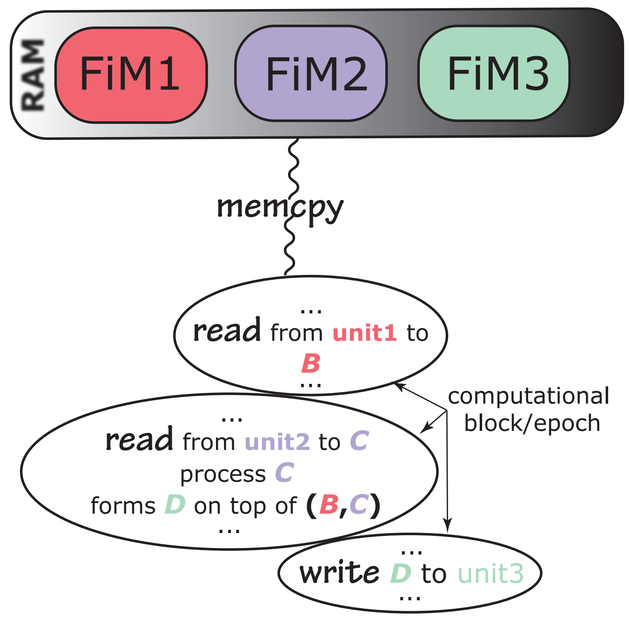
• Files In Memory

Recently we developed a RAM-based I/O layer in MOLCAS called Files In Memory (**FiM**). In framework of FiM scratch data generated and processed by MOLCAS module reside entirely in RAM throughout the computation.

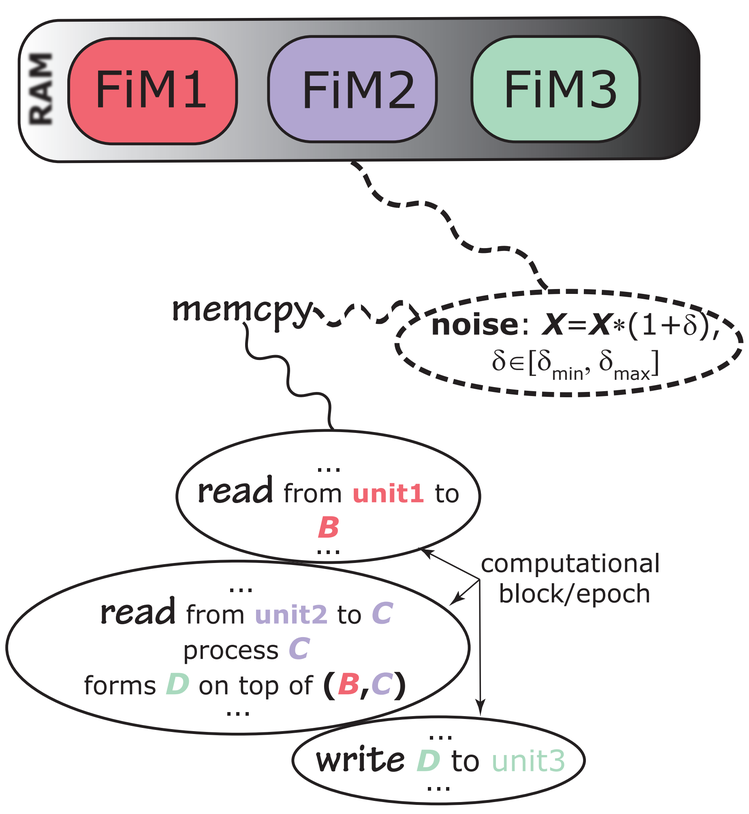
Unlike to the memory-resident I/O layer of CRAY FFIO [1], **FiM is a general framework** and **can be used on any POSIX compliant operation system** such as Linux, AIX, Windows, Solaris.

The beauty of **FiM** that it is **easy to use** for both MOLCAS end user and developer: there is no need to change source code, one just needs to edit an external resource file! In addition, **FiM** provides **environment variables** that control the execution of MOLCAS and automatic (dynamical) switching between I/O layers at **runtime**.

By design, FiM provides precise control over data needed for solving a problem and provides functionality for: efficient debugging, analysis of intermediate data and verifying the numerical stability of quantum chemistry algorithms implemented .

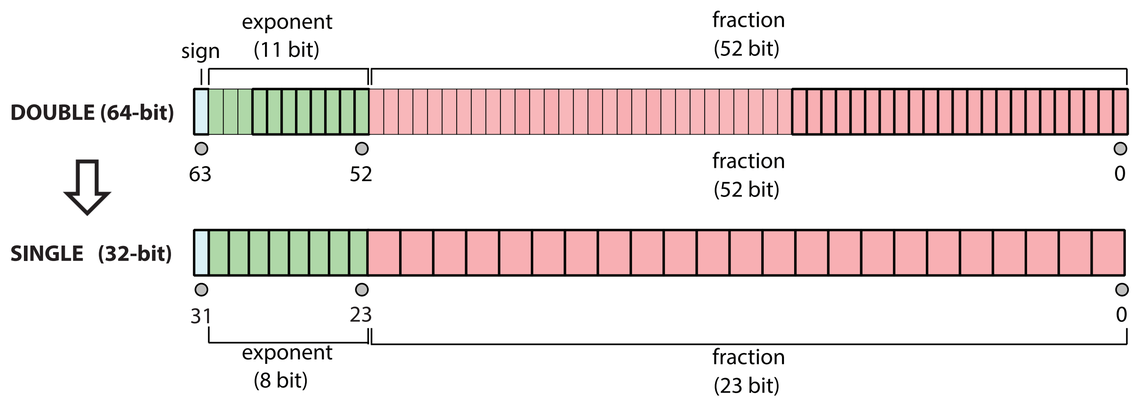


• Approach



In particular, test for numerical stability is accomplished by adding artificial numerical noise to requested/used data at every '***read/write***' I/O operation. The numerical noise is constructed as random floating point number from a user-specified interval $[\delta_{min},\delta_{max}]$.

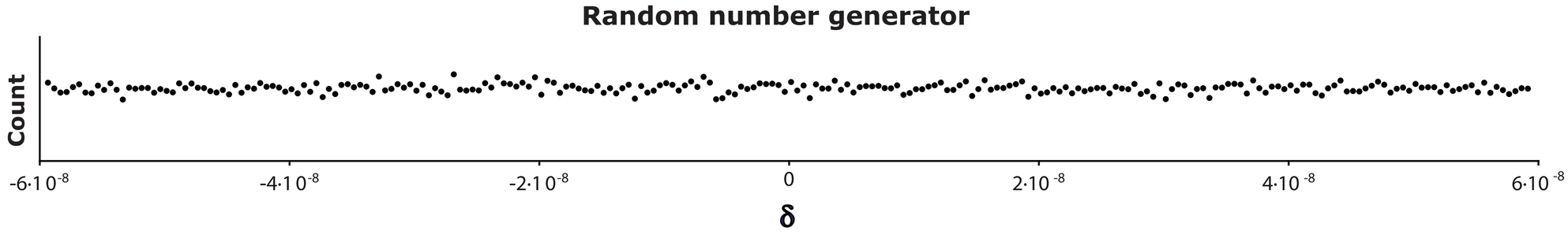
IEEE Standard 754 Floating-point



$$\delta_{64\leftrightarrow 32}=2^{-24}\approx 5.960\times 10^{-8}$$

For instance, for single precision the corresponding numerical noise will be in the range $[-2^{-24},2^{-24}]$.

Random number generator



Our approach does not introduce substantial computational overhead and thus can be applied for large systems!

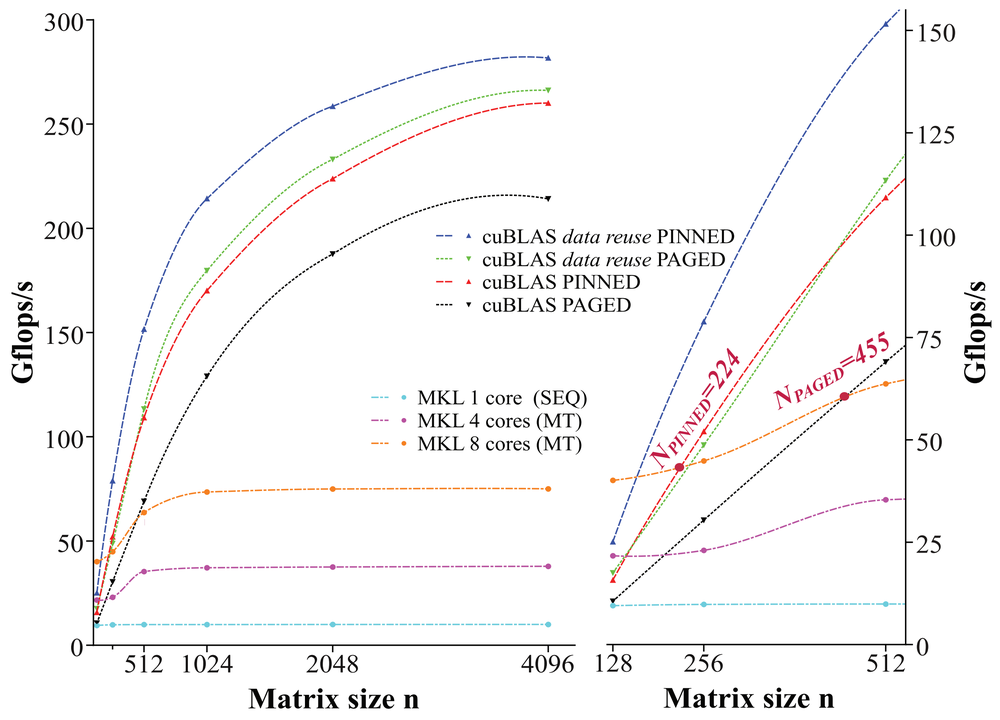
• Why 32-bit? GPU

✓ Halves memory demands;

✓ On modern processing units single precision arithmetic at leasst 2x times faster than double precision arithmetic;

✓ High Throughput 64⇔32 conversion: 3.5 Gb/s

✓ CASPT2 code spends up to 80% of the computational time in DGEMM



✓ Intel MKL v10.2 on Intel Xeon E5520 (2.27 GHz)

✓ CUDA BLAS v4.1 on NVIDIA Tesla M2050.

✓ *Pinned memory* means that allocated memory pages remain in real RAM all the time.

✓ In *Data reuse* scenario the C matrix was resided on the GPU device

✓ *The previous GPU* hardware generation and corresponding *CUDA* libraries was ***4x** times slower* than the current one. In particular, the N_{pinned} and N_{paged} crossing points for Tesla S1070& &cuBLAS v2 are ***796*** and ***1845***, respectively.

• Results

By using **FiM** with perturbing the CD-CASPT2 data used and the benchmarks suite for the electronically excited states (196 valence excitations in 26 organic molecules) [2], we found that **single precision can be sufficient for the CD-CASPT2 method**. Specifically, the **error** introduced in CD-CASPT2 total and excitations energies are typically **on the order of 10^{-6} hartree or even less**.

• References

[1] Cray T3E™ Fortran Optimization Guide - 004-2518-002, Chapter 5. Input/Output.
[2] M. Schreiber, M. R. Silva-Junior, S. P. A. Sauer, W. Thiel *J. Chem. Phys*. 128, 134110(1-25) (2008).

http://www.molcas.org