A Relativistic Fock-Space Coupled Cluster Method: Towards Efficient Execution on GPUs

A. V. Oleynichenko[1,2], S. V. Kozlov[2], A. V. Zaitsevskii[1,2], E. Eliav[3]

[1] NRC "Kurchatov Institute", Petersburg Nuclear Physics Institute, Orlova Roshcha, 188300 Gatchina, Russia
[2] Department of Chemistry, Lomonosov Moscow State University, Leninskie gory 1/3, 119991 Moscow, Russia
[3] School of Chemistry, Tel Aviv University, 69978 Tel Aviv, Israel

alexvoleynichenko@gmail.com

Motivation

Relativistic Fock-space coupled cluster method (FS-RCC) – one of the most promising tools for high-precision electronic structure modelling:

- perfectly suitable for molecular spectroscopy problems
- rather clear physical background
- predictable accuracy
  → up to ~ 10 cm^{-1} for electronic excitation energies
  → depends on the approximation level: CCSD, CCSDT-1...

But: very high computational complexity!

CCSD model – $\mathbf{O(N^6)}$, CCSDT model – $\mathbf{O(N^8)}$
N = number of one-particle basis functions (spinors)
typical size of the problem: $>$ 500 spinors (diatomic molecule)

Parallelization will help us!

A few GPU-parallelized CC codes were implemented to the date:

- single-reference nonrelativistic CCSD [1] and EOM-CCSD [2]
- up to 10x speedup on GPU! [2]

Coupled Cluster Method

Molecular electronic Hamiltonian:

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{4} \sum_{pqrs} V_{pqrs} a_p^\dagger a_q^\dagger a_s a_r$$

h_{pq} one-electron molecular integrals
V_{pqrs} two-electron molecular integrals (electron repulsion)

We solve the Schrödinger-type equation $H\Psi = E\Psi$ assuming that the wavefunction Ψ is of type:

$$\Psi = \{e^T\}\tilde{\psi} \qquad T = \sum_{ia} \boldsymbol{t_i^a} a_a^\dagger a_i + \sum_{ijab} \boldsymbol{t_{ij}^{ab}} a_a^\dagger a_b^\dagger a_j a_i + \ldots$$

We solve equations for $\boldsymbol{t_i^a}$, $\boldsymbol{t_{ij}^{ab}}$ – cluster (excitation) amplitudes

Algorithms

Let us analyze the structure of the FS-RCC amplitude equations:

target amplitudes — two-electron integrals

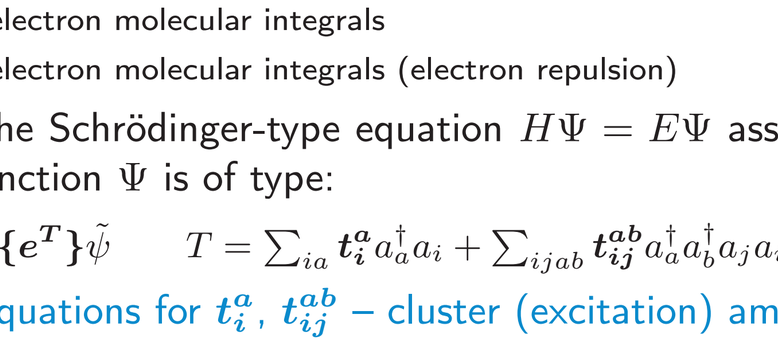
$$(\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b) t_{ij}^{ab} = V_{abij} + \sum_{xy} t_{xy}^{ab} V_{xyij} + \frac{1}{2} \hat{P}(ij|ab) \sum_{ld} t_{lj}^{db} \left(\sum_{kc} t_{ik}^{ac} V_{klcd} \right) + \ldots$$

$$\Delta T[a,b,i,j] = \sum_{x,y} T[\underbrace{a,b}_{A}, \underbrace{x,y}_{K}] * V[\underbrace{x,y}_{K}, \underbrace{i,j}_{I}] = \sum_{K} T_{AK} V_{KI}$$

contribution to amplitudes — zgemm !

FS-RCC = matrix-matrix multiplications + "tensor" transpositions

Multidimensional arrays are splitted into non-zero dense blocks:



Algorithm: tensor contraction on GPU

```
for block_c in symblocks( C ):          // resulting diagram
  for block_a in symblocks( A ):        // operand 1
    for block_b in symblocks( B ):      // operand 2
      cudaMemcpy(block_a^gpu ← block_a)
      cudaMemcpy(block_b^gpu ← block_b)
      cudaMemcpy(block_c^gpu ← block_c)
      block_c^gpu += cublasZgemm(block_a^gpu, block_b^gpu)
      cudaMemcpy(block_c ← block_c^gpu)
```

Time complexity: $\mathbf{O(N^6)}$ for the CCSD model (only t_i^a and t_{ij}^{ab})

Implementation

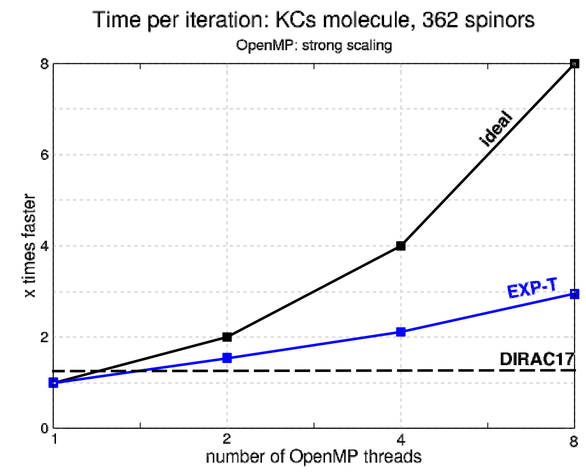
The new coupled cluster program (called EXP-T) was developed:

- FS-MRCC method: up to 3 open shells
- CC cluster operator: up to CCSDT-3 model
- molecular integrals are imported from DIRAC [5]
  → any relativistic Hamiltonians + properties
- CUDA [3] and OpenMP [4] parallelization
- for more information: http://qchem.pnpi.spb.ru

Benchmark: OpenMP

Problem: excitation energies of the KCs molecule (CCSD model)
Problem size: 362 spinors (medium)
CPU: Intel(R) Core(TM) i9-7900X

Compiler: Intel® C++ Compiler 18.0 (-O3 -xHost)
MKL: Intel® Math Kernel Library 18.0 Update 1 for Linux



* DIRAC – the most advanced relativistic quantum chemistry program [5]

To be improved:

- number of disk operations must be reduced

Benchmark: CUDA

Problem: excitation energies of the Rb atom (CCSD model)
Problem size: 182 spinors (small)
GPU: NVIDIA® GeForce® GTX TITAN Black

CPU: AMD FX(TM)-8320(8)
Compiler: Intel® C++ Compiler 19.0 (-O3 -xHost)
MKL: Intel® Math Kernel Library 19.0 Update 3 for Linux

(time in sec) point group	DIRAC17* 1 CPU core	EXP-T 1 CPU core	EXP-T 1 GPU
$C_{\infty v}$	387	334 (1.2x)	1189 (0.3x)
C_s	9931	3087 (3.2x)	2300 (4.3x)
C_1	37302	9726 (3.8x)	4825 **(7.7x)**

Remarks:

- higher symmetry ($C_1 \rightarrow C_{\infty v}$) → more blocks of integrals → smaller blocks & much more metainfo → large overheads
- practically interesting molecules often have C_1 symmetry!

To be improved:

- disk access – asynchronous I/O is required
- asynchronous access to GPU (CUDA streams)
- multi-GPU technology (2, 4, 8 GPUs per node)

Bibliography

[1] A.E. DePrince III et al. J. Chem. Theory Comput. 7, 1287 (2011)
[2] I.A. Kaliman, A.I. Krylov. J. Comput. Chem. 38, 842 (2017)
[3] J. Nickolls et al. ACM Queue. Vol. 6(2), 40 (2008)
[4] L. Dagum, R. Menon. IEEE Comput. Sci. Eng. 5(1), 46-55 (1998)
[5] L. Visscher et al., DIRAC, a relativistic ab initio electronic structure program, http://www.diracprogram.org