Computational Challenges of Coupled Cluster Theory

Jeff Hammond

Leadership Computing Facility
Argonne National Laboratory

11 January 2012

Atomistic simulation in chemistry

1. *classical* molecular dynamics (MD) with empirical potentials
2. *quantum* molecular dynamics based upon **density**-function theory (DFT)
3. *quantum* chemistry with **wavefunctions** e.g. perturbation theory (PT), coupled-cluster (CC) or quantum monte carlo (QMC).

Classical molecular dynamics

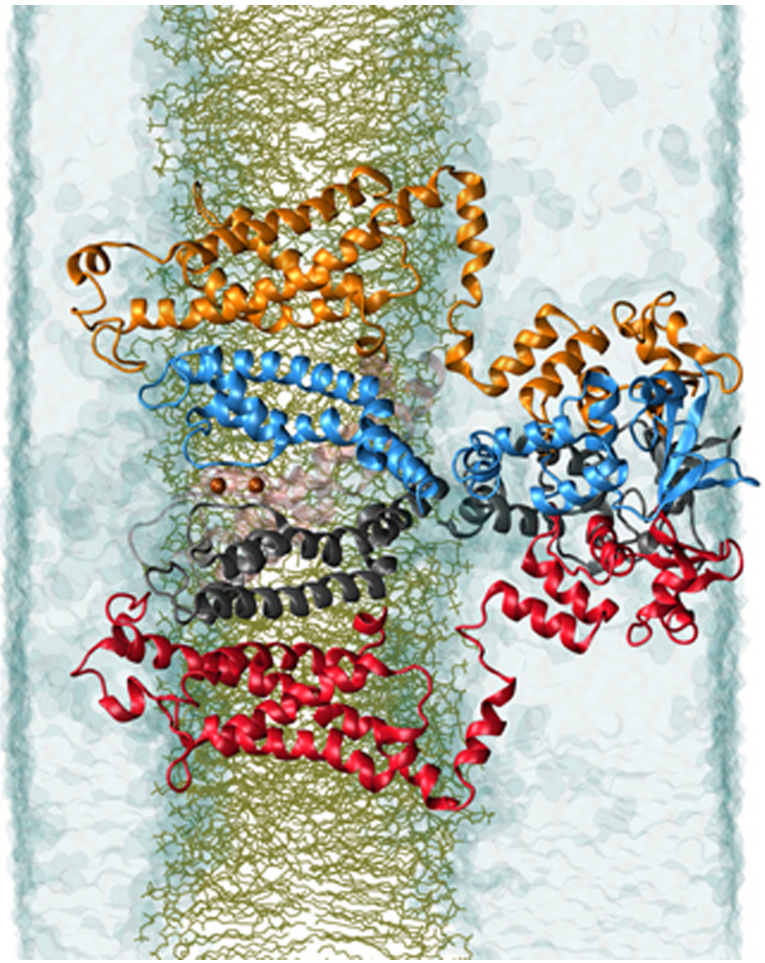


Image courtesy of Benoît Roux via ALCF.

- Solves Newton's equations of motion with empirical terms and classical electrostatics.
- Size: 100K-10M atoms
- Time: 1-10 ns/day
- Scaling: $\sim N_{atoms}$
- Math: N-body

Data from K. Schulten, et al. "Biomolecular modeling in the era of petascale computing." In D. Bader, ed., *Petascale Computing: Algorithms and Applications*.

Car-Parrinello molecular dynamics

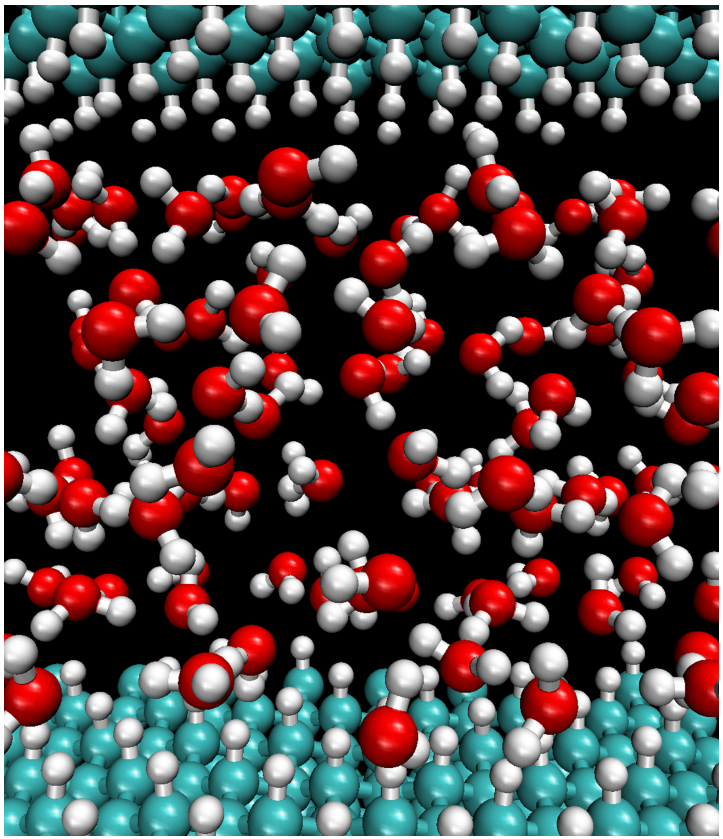


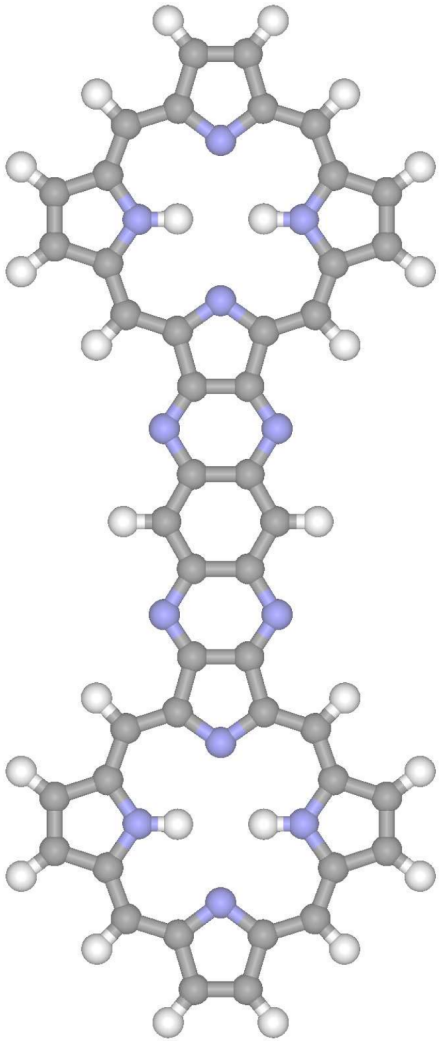
Image courtesy of Giulia Galli via ALCF.

- Forces obtained from solving an approximate single-particle Schrödinger equation.
- Size: 100-1000 atoms
- Time: 0.01-1 ps/day
- Scaling: $\sim N_{el}^{x}$ (x=1-3)
- Math: FFT, eigensolve.

F. Gygi, *IBM J. Res. Dev.* **52**, 137 (2008); E. J. Bylaska et al. *J. Phys.: Conf. Ser.* **180**, 012028 (2009).

Wavefunction theory

,



- MP2 is second-order PT and is accurate via magical cancellation of error.
- CC is infinite-order solution to many-body Schrödinger equation truncated via clusters.
- QMC is Monte Carlo integration applied to the Schrödinger equation.
- Size: 10-100 atoms, maybe 100-1000 atoms with MP2.
- Time: N/A (LOL)
- Scaling: $\sim N_{bf}^{x}$ (x=4-7)
- Math: DLA (tensors)

Image courtesy of Karol Kowalski and Niri Govind.

The Standard Model (of Quantum Chemistry)

Quantum chemistry

1. Separate molecule(s) from environment (closed to both matter and energy)
2. Boundary conditions:
   $\psi(x \to \infty) = 0$ (finite system)
   $\psi(x) = \phi(x + g)$ (infinite, periodic system)
3. Ignore relativity, QED, spin-orbit coupling
4. Separate electronic and nuclear degrees of freedom

$\longrightarrow$ non-relativistic electronic Schrödinger equation in a vacuum at zero temperature.

Quantum chemistry

$$\begin{aligned}\hat{H} &= \hat{T}_{el} + \hat{V}_{el-nuc} + \hat{V}_{el-el} \\ \hat{H} &= -\frac{1}{2}\sum_{i=1}^{M}\nabla_i^2 + \sum_{n=1}^{N}\sum_{i=1}^{M}\frac{Z_n}{R_{ni}} + \sum_{i<j}^{M}\frac{1}{r_{ij}}\end{aligned}$$

$$\Psi\left(\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{x}_{n+1},\ldots,\mathbf{x}_N\right) = -\Psi\left(\mathbf{x}_1,\ldots,\mathbf{x}_{n+1},\mathbf{x}_n,\ldots,\mathbf{x}_N\right)$$

The electron coordinates ($\mathbf{x}_i$) include both space (r) and spin (σ). We will integrate-out spin wherever possible.

Quantum chemistry

Wavefunction antisymmetry is enforced by expanding in determinants, which we now capture in second quantization.

1. project physical operators (e.g. Coulomb) into one-electron basis — usually atom-center Gaussians
2. generate mean-field reference and expand many-body wavefunction in terms of excitations out of that reference

$\longrightarrow$ Full configuration-interation (FCI) ansatz.

1. truncate exponentially-growing FCI ansatz (CI=linear generator, CC=exponential generator)
2. solve CC (or CI) iteratively
3. add more correlation via perturbation theory

$\longrightarrow$ CCSD(T), as one example.

Quantum chemistry

Correct for missing physics using perturbation theory (a posteriori error correction) or mixed (e.g. QM/MM) formalism:

1. relativistic corrections
2. non-adiabatic corrections
3. solvent corrections
4. open BC corrections (less common)

Coupled-cluster theory

Coupled-cluster theory

$$
\begin{aligned}
|\Psi_{CC}\rangle &= \exp(T)|\Psi_{HF}\rangle \\
T &= T_1 + T_2 + \cdots + T_n \quad (n \ll N) \\
T_1 &= \sum_{ia} t_i^a \hat{a}_a^\dagger \hat{a}_i \\
T_2 &= \sum_{ijab} t_{ij}^{ab} \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_j \hat{a}_i \\
|\Psi_{CCD}\rangle &= \exp(T_2)|\Psi_{HF}\rangle \\
&= (1 + T_2 + T_2^2)|\Psi_{HF}\rangle \\
|\Psi_{CCSD}\rangle &= \exp(T_1 + T_2)|\Psi_{HF}\rangle \\
&= (1 + T_1 + \cdots + T_1^4 + T_2 + T_2^2 + T_1 T_2 + T_1^2 T_2)|\Psi_{HF}\rangle
\end{aligned}
$$

Coupled cluster (CCD) implementation

$\exp(T_2)|\Psi_{HF}\rangle$ turns into:

$$R_{ij}^{ab} = V_{ij}^{ab} + P(ia,jb)\left[T_{ij}^{ae} I_e^b - T_{im}^{ab} I_j^m + \frac{1}{2} V_{ef}^{ab} T_{ij}^{ef} + \frac{1}{2} T_{mn}^{ab} I_{ij}^{mn} - T_{mj}^{ae} I_{ie}^{mb} - I_{ie}^{ma} T_{mj}^{eb} + (2T_{mi}^{ea} - T_{im}^{ea}) I_{ej}^{mb}\right]$$

$$\begin{aligned}
I_b^a &= (-2V_{eb}^{mn} + V_{be}^{mn}) T_{mn}^{ea} \\
I_j^i &= (2V_{ef}^{mi} - V_{ef}^{im}) T_{mj}^{ef} \\
I_{kl}^{ij} &= V_{kl}^{ij} + V_{ef}^{ij} T_{kl}^{ef} \\
I_{jb}^{ia} &= V_{jb}^{ia} - \frac{1}{2} V_{eb}^{im} T_{jm}^{ea} \\
I_{bj}^{ia} &= V_{bj}^{ia} + V_{be}^{im} (T_{mj}^{ea} - \frac{1}{2} T_{mj}^{ae}) - \frac{1}{2} V_{be}^{mi} T_{mj}^{ae}
\end{aligned}$$

Tensor Contraction Engine

Tensor Contraction Engine

What does it do?

1. GUI input quantum many-body theory e.g. CCSD.
2. Operator specification of theory (as in a theory paper).
3. Apply Wick's theory to transform operator expressions into array expressions (as in a computational paper).
4. Transform input array expression to operation tree using many types of optimization (i.e. compile).
5. Generate F77/GA/NXTVAL implementation for NWChem or C++/MemoryGrp for MPQC or F90/.. for UTChem.

Developer can intercept at various stages to modify theory, algorithm or implementation (may be painful).

TCE Input

We get 73 lines of serial F90 or 604 lines of parallel F77 from this:

```
1/1 Sum(g1 g2 p3 h4) f(g1 g2) t(p3 h4) {g1+ g2}{p3+ h4}
1/4 Sum(g1 g2 g3 g4 p5 h6) v(g1 g2 g3 g4) t(p5 h6) {g1+
g2+ g4 g3}{p5+ h6}
1/16 Sum(g1 g2 g3 g4 p5 p6 h7 h8) v(g1 g2 g3 g4) t(p5 p6
h7 h8) {g1+ g2+ g4 g3}{p5+ p6+ h8 h7}
1/8 Sum(g1 g2 g3 g4 p5 h6 p7 h8) v(g1 g2 g3 g4) t(p5 h6)
t(p7 h8) {g1+ g2+ g4 g3}{p5+ h6} {p7+ h8}
```

LaTeX equivalent of the first term:

$$\sum_{g_1,g_2,p_3,h_4} f_{g_1,g_2} t_{p_3,h_4} \{g_1^\dagger g_2\}\{p_3^\dagger h_4\}$$

Summary of TCE module

```
http://cloc.sourceforge.net v 1.53  T=30.0 s
-------------------------------------------------
Language     files  blank  comment     code
-------------------------------------------------
Fortran 77   11451   1004   115129  2824724
-------------------------------------------------
SUM:         11451   1004   115129  2824724
-------------------------------------------------
```

Perhaps $<$25 KLOC are hand-written; $\sim$100 KLOC is utility code following TCE data-parallel template.

Expansion from TCE input to massively-parallel F77 is $\sim$ 200 (drops with language abstractions).

TCE template

Pseudocode for $R_{i,j}^{a,b} = T_{i,j}^{c,d} * V_{a,b}^{c,d}$:

```
for i,j in occupied blocks:
   for a,b in virtual blocks:
      for c,d in virtual blocks:
         if symmetry_criteria(i,j,a,b,c,d):
            if dynamic_load_balancer(me):
               Get block t(i,j,c,d) from T
               Permute t(i,j,c,d)
               Get block v(a,b,c,d) from V
               Permute v(a,b,c,d)
               r(i,j,c,d) += t(i,j,c,d) * v(a,b,c,d)
      Permute r(i,j,a,b)
      Accumulate r(i,j,a,b) block to R
```

TCE profile

`ccsd_t2_8` (DGEMM-like):

timer	min	max	avg
dgemm	68.605	91.296	81.282
ga_acc	0.042	0.070	0.050
ga_get	5.845	7.779	6.679
nxtask	0.012	28.710	13.638
tce_sort4	6.184	8.174	7.347
tce_sortacc4	7.892	11.042	9.290

Observations about the TCE template

1. Blocking get means no overlap
2. Dynamic load balancing is **global** (shared counter)
3. Get+Permute of t(i,j,c,d) happens for all (a,b)
4. Get+Permute of v(a,b,c,d) happens for all (i,j)
5. Permute is a nasty operation (desire fused contraction).

We could apply well-known techniques to fix everything...

(There are an uncountable number of good programming techniques not being used in any scientific code.)

TCE Template for MMM

Pseudocode for $C_j^i = A_k^i * B_j^k$:

```
for i in I blocks:
   for j in J blocks:
      for k in K blocks:
          if dynamic_load_balancer(me):
             Get block a(i,k) from A
             Get block b(k,j) from B
             c(i,j) += a(i,k) * b(k,j)
      Accumulate c(i,j) block to C
```

Algorithms trump tuned runtimes and libraries every time.

A better way

- TCE has it right, but only serially:
  tensor contractions are permute + matmul.
- Parallel permute = parallel sorting = well-understood.
- Parallel matmul = well-understood.

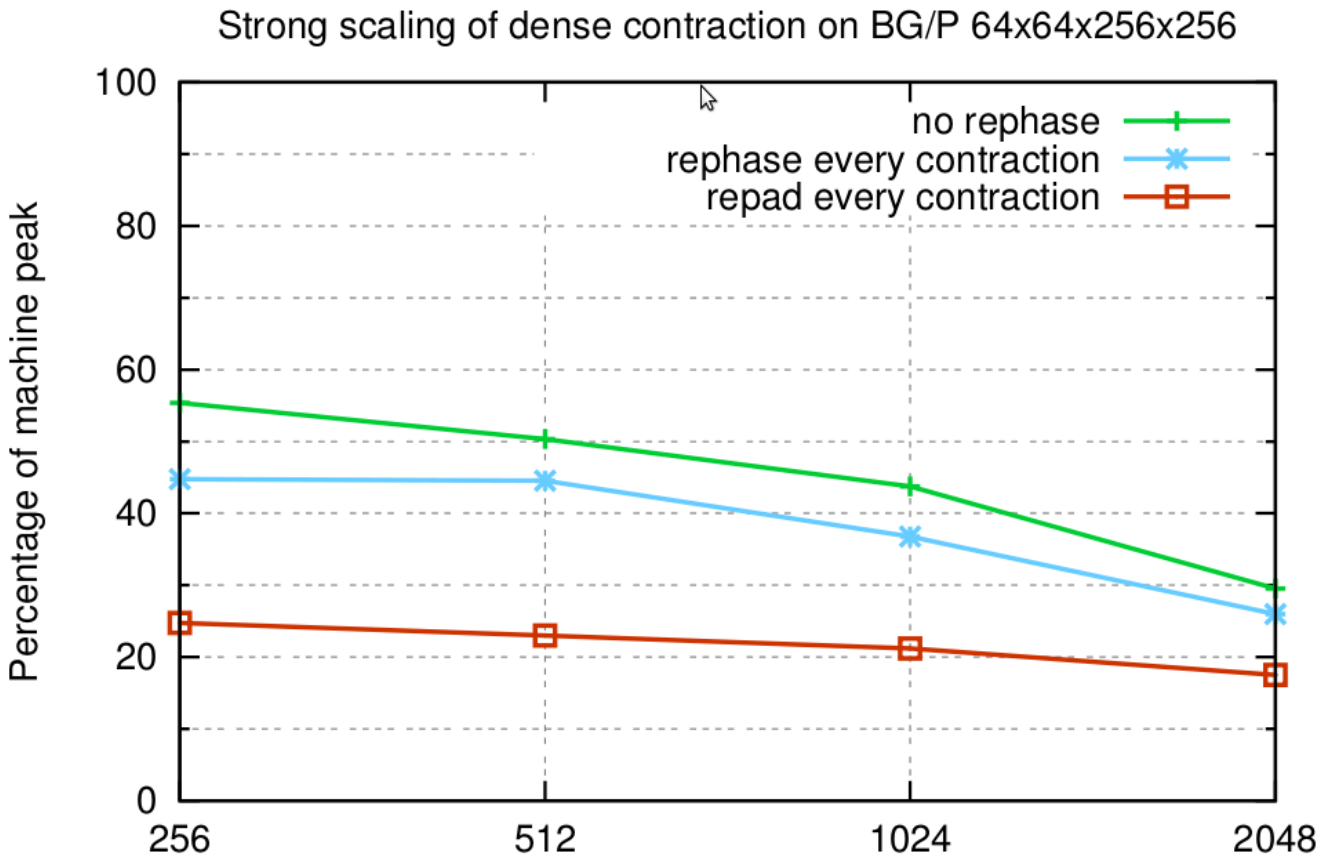
Therefore, parallel tensor contractions are solved, up to the implementation details and future algorithm developments in sorting and matmul.

All existing TCE technology for operation trees are still valid.

Cyclops Tensor Framework

Written by Edgar Solomonik (I am just a cheerleader).

Very preliminary (Summer 2011) strong-scaling results:



Communication

But where's the one-sided communication?!?

Like parallel matmul and sorting, CTF does fine with MPI-1.

There are good uses of one-sided but TCE isn't one*.

* Unless matmul or sorting benefits from it.

Summary

- Dense tensor contractions are dense linear algebra plus some lower-order bookkeeping.
- Permutation symmetry folded into cyclic/elemental distribution in a load-balanced way.
- Parallel dense linear algebra is a well-understood problem that is continuously studied by smart people; parallel libraries exist.
- Parallel dense tensor contractions are **best** implemented in terms of parallel dense linear algebra and **not** as serial dense linear algebra directed by a locality-oblivious dynamic runtime, especially if flops are "free."

The future

Sparse formalisms are much more involved, but there is every reason to believe that representing them in terms of sparse sorting and sparse matmul will be reasonable, whereas the one-sided approach would only move farther and farther away from its sweet-spot (bandwidth-limited bulk operations and infrequent active-messages).

TCE treated matrix-chain multiplication problem in a straightforward way, ignoring possibilities for exotic multi-level fusion.

PNNL and Argonne have efforts using task-parallelism and DAG-scheduling for MPPs and heterogenous nodes, respectively, but much parallelism remains. We can learn from DAG efforts in DLA...

The end of CC

A theory for hard science problems...

"It is our belief that CCSDTQ represents the (practical) end of advancement in coupled-cluster theory. We are unaware of any problems in quantum chemistry that are not treated satisfactorily by this level of theory." – John Stanton

$$
\begin{aligned}
|\Psi_{CC}\rangle &= \exp(T)|\Psi_{HF}\rangle \\
T &= T_1 + T_2 + T_3 + T_4 \\
|\Psi_{CCSDTQ}\rangle &= \exp(T_1 + T_2 + T_3 + T_4)|\Psi_{HF}\rangle
\end{aligned}
$$

Writing this theory down in full details takes longer than some simulations...

CCSDTQ singles

$$
\begin{aligned}
r1_{h_1}^{p_2}, \quad = \quad & f_{h_1}^{p_2}, +f_{h_1}^{h_3}, t_{h_3}^{p_2}, +f_{p_3}^{p_2}, t_{h_1}^{p_3}, -t_{h_4}^{p_3}, v_{h_1 p_3}^{h_4 p_2}, +f_{p_4}^{h_3}, t_{h_3 h_1}^{p_4 p_2}, +\frac{1}{2} t_{h_4 h_5}^{p_3 p_2}, v_{h_1 p_3}^{h_4 h_5}, +\frac{1}{2} t_{h_5 h_1}^{p_3 p_4}, v_{p_3 p_4}^{h_5 p_2}, +\frac{1}{4} t_{h_5 h_6 h_1}^{p_3 p_4 p_2}, v_{p_3 p_4}^{h_5 h_6} \\
& -t_{h_1}^{p_3}, t_{h_4}^{p_2}, f_{p_3}^{h_4}, -t_{h_3}^{p_2}, t_{h_5}^{p_4}, v_{h_1 p_4}^{h_3 h_5}, -t_{h_1}^{p_3}, t_{h_5}^{p_4}, v_{p_3 p_4}^{h_5 p_2}, -\frac{1}{2} t_{h_4 h_5}^{p_3 p_2}, t_{h_1}^{p_6}, v_{p_3 p_6}^{h_4 h_5}, -\frac{1}{2} t_{h_5 h_1}^{p_3 p_4}, t_{h_6}^{p_2}, v_{p_3 p_4}^{h_5 h_6}, +t_{h_4 h_1}^{p_3 p_2}, t \\
& -t_{h_1}^{p_3}, t_{h_4}^{p_2}, t_{h_6}^{p_5}, v_{p_3 p_5}^{h_4 h_6},
\end{aligned}
$$

CCSDTQ doubles

$$
\begin{aligned}
r2^{p_3 p_4}_{h_1 h_2}, \quad = \quad & +v^{p_3 p_4}_{h_1 h_2}, -\left(1 - P^{p_4 p_3 h_1 h_2}_{p_3 p_4 h_1 h_2}\right) t^{p_3}_{h_5}, v^{h_5 p_4}_{h_1 h_2}, +\left(1 - P^{p_3 p_4 h_2 h_1}_{p_3 p_4 h_1 h_2}\right) t^{p_5}_{h_2}, v^{p_3 p_4}_{h_1 p_5}, -\left(1 - P^{p_3 p_4 h_1 h_2}_{p_3 p_4 h_2 h_1}\right) f^{h_5}_{h_1}, t^{p_3 p_4}_{h_5 h_2} \\
& -\left(1 - P^{p_3 p_4 h_1 h_2}_{p_4 p_3 h_1 h_2}\right) f^{p_4}_{p_5}, t^{p_5 p_3}_{h_1 h_2}, +\frac{1}{2} t^{p_3 p_4}_{h_5 h_6}, v^{h_5 h_6}_{h_1 h_2}, +\left(1 - P^{p_4 p_3 h_2 h_1}_{p_3 p_4 h_2 h_1} - P^{p_3 p_4 h_2 h_1}_{p_3 p_4 h_1 h_2} + P^{p_4 p_3 h_2 h_1}_{p_3 p_4 h_1 h_2}\right) t^{p_5 p_3}_{h_6 h_2}, v \\
& +\frac{1}{2} t^{p_5 p_6}_{h_1 h_2}, v^{p_3 p_4}_{p_5 p_6}, +f^{h_5}_{p_6}, t^{p_6 p_3 p_4}_{h_5 h_1 h_2}, +\frac{1}{2}\left(1 - P^{p_3 p_4 h_2 h_1}_{p_3 p_4 h_1 h_2}\right) t^{p_5 p_3 p_4}_{h_6 h_7 h_2}, v^{h_6 h_7}_{h_1 p_5}, -\frac{1}{2}\left(1 - P^{p_4 p_3 h_1 h_2}_{p_3 p_4 h_1 h_2}\right) t^{p_5 p_6 p_3}_{h_7 h_1 h_2}, v^{h} \\
& +\frac{1}{4} t^{p_5 p_6 p_3 p_4}_{h_7 h_8 h_1 h_2}, v^{h_7 h_8}_{p_5 p_6}, +t^{p_3}_{h_5}, t^{p_4}_{h_6}, v^{h_5 h_6}_{h_1 h_2}, -\left(1 - P^{p_4 p_3 h_2 h_1}_{p_3 p_4 h_2 h_1} - P^{p_3 p_4 h_2 h_1}_{p_3 p_4 h_1 h_2} + P^{p_4 p_3 h_2 h_1}_{p_3 p_4 h_1 h_2}\right) t^{p_5}_{h_2}, t^{p_3}_{h_6}, v^{h_6 p_4}_{h_1 p_5}, + \\
& -\left(1 - P^{p_3 p_4 h_2 h_1}_{p_3 p_4 h_1 h_2}\right) f^{h_5}_{p_6}, t^{p_3 p_4}_{h_5 h_2}, t^{p_6}_{h_1}, +\left(1 - P^{p_4 p_3 h_1 h_2}_{p_3 p_4 h_1 h_2}\right) f^{h_5}_{p_6}, t^{p_6 p_3}_{h_1 h_2}, t^{p_4}_{h_5}, +\frac{1}{2}\left(1 - P^{p_3 p_4 h_2 h_1}_{p_3 p_4 h_1 h_2}\right) t^{p_3 p_4}_{h_5 h_6}, t^{p} \\
& -\left(1 - P^{p_4 p_3 h_2 h_1}_{p_3 p_4 h_2 h_1} - P^{p_3 p_4 h_2 h_1}_{p_3 p_4 h_1 h_2} + P^{p_4 p_3 h_2 h_1}_{p_3 p_4 h_1 h_2}\right) t^{p_5 p_3}_{h_6 h_2}, t^{p_4}_{h_7}, v^{h_6 h_7}_{h_1 p_5}, -\left(1 - P^{p_3 p_4 h_2 h_1}_{p_3 p_4 h_1 h_2}\right) t^{p_3 p_4}_{h_5 h_2}, t^{p_6}_{h_7}, v^{h_5 h_7}_{h_1 p_6}, \\
& -\frac{1}{2}\left(1 - P^{p_4 p_3 h_1 h_2}_{p_3 p_4 h_1 h_2}\right) t^{p_5 p_6}_{h_1 h_2}, t^{p_3}_{h_7}, v^{h_7 p_4}_{p_5 p_6}, +\left(1 - P^{p_4 p_3 h_1 h_2}_{p_3 p_4 h_1 h_2}\right) t^{p_5 p_3}_{h_1 h_2}, t^{p_6}_{h_7}, v^{h_7 p_4}_{p_5 p_6}, -\frac{1}{2}\left(1 - P^{p_3 p_4 h_2 h_1}_{p_3 p_4 h_1 h_2}\right) t^{p}_{h} \\
& +\frac{1}{2}\left(1 - P^{p_4 p_3 h_1 h_2}_{p_3 p_4 h_1 h_2}\right) t^{p_5 p_6 p_3}_{h_7 h_1 h_2}, t^{p_4}_{h_8}, v^{h_7 h_8}_{p_5 p_6}, +t^{p_5 p_3 p_4}_{h_6 h_1 h_2}, t^{p_7}_{h_8}, v^{h_6 h_8}_{p_5 p_7}, +\frac{1}{2}\left(1 - P^{p_3 p_4 h_1 h_2}_{p_4 p_3 h_1 h_2}\right) t^{p_5 p_4}_{h_1 h_2}, t^{p_6 p_3}_{h_7 h_8}, v^{h_7}_{p_5} \\
& -\frac{1}{2}\left(1 - P^{p_3 p_4 h_1 h_2}_{p_3 p_4 h_2 h_1}\right) t^{p_3 p_4}_{h_5 h_1}, t^{p_6 p_7}_{h_8 h_2}, v^{h_5 h_8}_{p_6 p_7}, -\left(1 - P^{p_3 p_4 h_1 h_2}_{p_4 p_3 h_1 h_2}\right) t^{p_5 p_4}_{h_6 h_1}, t^{p_7 p_3}_{h_8 h_2}, v^{h_6 h_8}_{p_5 p_7}, +\left(1 - P^{p_3 p_4 h_2 h_1}_{p_3 p_4 h_1 h_2}\right) \\
& +\left(1 - P^{p_3 p_4 h_1 h_2}_{p_4 p_3 h_1 h_2} - P^{p_4 p_3 h_1 h_2}_{p_4 p_3 h_2 h_1} + P^{p_3 p_4 h_1 h_2}_{p_4 p_3 h_2 h_1}\right) t^{p_5}_{h_1}, t^{p_4}_{h_6}, t^{p_7 p_3}_{h_8 h_2}, v^{h_6 h_8}_{p_5 p_7}, \\
& +\left(1 - P^{p_3 p_4 h_1 h_2}_{p_3 p_4 h_2 h_1}\right) t^{p_5}_{h_1}, t^{p_6}_{h_7}, t^{p_3 p_4}_{h_8 h_2}, v^{h_7 h_8}_{p_5 p_6}, +\frac{1}{2} t^{p_3}_{h_5}, t^{p_4}_{h_6}, t^{p_7 p_8}_{h_1 h_2}, v^{h_5 h_6}_{p_7 p_8}, -\left(1 - P^{p_3 p_4 h_1 h_2}_{p_4 p_3 h_1 h_2}\right) t^{p_4}_{h_5}, t^{p_6}_{h_7}, t^{p_8 p_3}_{h_1 h_2}
\end{aligned}
$$

CCSDTQ triples

$$
\begin{aligned}
r3^{p_3 p_4 p_5}_{h_1 h_2 h_3}, \quad = \quad & \Big(1 - P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_3 h_1 h_2) - P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_3 h_1 h_2) - P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_3 h_1 h_2) + P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_2 h_1 h_3) + P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_2 h_1 h_3) - P^{p_4 p_5 p_6}_{p_4 p_5 p_6} \\
& + \Big(1 - P^{p_5 p_4 p_6}_{p_4 p_5 p_6}(h_2 h_3 h_1) - P^{p_6 p_4 p_5}_{p_4 p_6 p_5}(h_2 h_3 h_1) + P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_3 h_2 h_1) - P^{p_5 p_4 p_6}_{p_4 p_5 p_6}(h_3 h_2 h_1) - P^{p_6 p_4 p_5}_{p_4 p_6 p_5}(h_3 h_2 h_1) + P^{p_4 p_5}_{p_4 p_5} \\
& - \Big(1 - P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_2 h_1 h_3) - P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_3 h_1 h_2)\Big) f^{h_7}_{h_1}, t^{p_4 p_5 p_6}_{h_7 h_2 h_3}, + \Big(1 - P^{p_5 p_4 p_6}_{p_6 p_4 p_5}(h_1 h_2 h_3) - P^{p_4 p_5 p_6}_{p_6 p_5 p_4}(h_1 h_2 h_3)\Big) f^{p_6}_{p_7} \\
& + \frac{1}{2}\Big(1 - P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_3 h_1 h_2) - P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_3 h_2 h_1)\Big) t^{p_4 p_5 p_6}_{h_7 h_8 h_3}, v^{h_7 h_8}_{h_1 h_2}, - \Big(1 - P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_2 h_3 h_1) - P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_2 h_3 h_1) \\
& + \frac{1}{2}\Big(1 - P^{p_5 p_4 p_6}_{p_4 p_5 p_6}(h_1 h_2 h_3) - P^{p_6 p_4 p_5}_{p_4 p_6 p_5}(h_1 h_2 h_3)\Big) t^{p_7 p_8 p_4}_{h_1 h_2 h_3}, v^{p_5 p_6}_{p_7 p_8}, + f^{h_7}_{p_8}, t^{p_8 p_4 p_5 p_6}_{h_7 h_1 h_2 h_3}, \\
& + \frac{1}{2}\Big(1 + P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_3 h_2 h_1) + P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_2 h_3 h_1)\Big) t^{p_7 p_4 p_5 p_6}_{h_8 h_9 h_2 h_3}, v^{h_8 h_9}_{h_1 p_7}, + \frac{1}{2}\Big(1 - P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_1 h_2 h_3) - P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_1 h_2 \\
& + \Big(1 - P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_3 h_2 h_1) - P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_3 h_2 h_1) - P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_2 h_3 h_1) + P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_3 h_2 h_1) + P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_2 h_3 h_1) - P^{p_4 p_5}_{p_4 p_5} \\
& - \Big(1 - P^{p_5 p_4 p_6}_{p_4 p_5 p_6}(h_2 h_3 h_1) - P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_2 h_3 h_1) + P^{p_5 p_6 p_4}_{p_4 p_5 p_6}(h_2 h_3 h_1) + P^{p_6 p_4 p_5}_{p_4 p_5 p_6}(h_2 h_3 h_1) - P^{p_6 p_5 p_4}_{p_4 p_5 p_6}(h_2 h_3 h_1) + P^{p_4 p_5}_{p_4 p_5} \\
& - \Big(1 + P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_3 h_2 h_1) + P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_2 h_3 h_1)\Big) f^{h_7}_{p_8}, t^{p_4 p_5 p_6}_{h_7 h_2 h_3}, t^{p_8}_{h_1}, - \Big(1 - P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_1 h_2 h_3) - P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_1 h_2 h_3)\Big) \\
& + \Big(1 - P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_2 h_3 h_1) - P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_2 h_3 h_1) + P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_3 h_2 h_1) - P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_3 h_2 h_1) - P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_3 h_2 h_1) + P^{p_4 p_5}_{p_4 p_5} \\
& - \Big(1 + P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_3 h_2 h_1) + P^{p_4 p_5 p_6}_{p_4 p_5 p_6}(h_2 h_3 h_1)\Big) t^{p_4 p_5 p_6}_{h_7 h_2 h_3}, t^{p_8}_{h_9}, v^{h_7 h_9}_{h_1 p_8}, + \Big(1 - P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_2 h_3 h_1) - P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_2 h_3 h_1 \\
& - \frac{1}{2}\Big(1 - P^{p_4 p_6 p_5}_{p_4 p_5 p_6}(h_1 h_2 h_3) - P^{p_5 p_6 p_4}_{p_5 p_4 p_6}(h_1 h_2 h_3)\Big) t^{p_7 p_8 p_4 p_5}_{h_9 h_1 h_2 h_3}, t^{p_6}_{h_{10}}, v^{h_9 h_{10}}_{p_7 p_8}, + t^{p_7 p_4 p_5 p_6}_{h_8 h_1 h_2 h_3}, t^{p_9}_{h_{10}}, v^{h_8 h_{10}}_{p_7 p_9}, + \Big(1 - \\
& + \frac{1}{2}\Big(1 - P^{p_5 p_4 p_6}_{p_6 p_4 p_5}(h_2 h_3 h_1) - P^{p_4 p_5 p_6}_{p_6 p_5 p_4}(h_2 h_3 h_1) + P^{p_6 p_4 p_5}_{p_6 p_4 p_5}(h_3 h_2 h_1) - P^{p_5 p_4 p_6}_{p_6 p_4 p_5}(h_3 h_2 h_1) - P^{p_4 p_5 p_6}_{p_6 p_5 p_4}(h_3 h_2 h_1) + P^{p_6}_{p_6}
\end{aligned}
$$

CCSDTQ-LR quadruples

Summary:

```
.../ccsdtq_lr> wc -l ccsdtq_t[1234].out
15 ccsdtq_t1.out
38 ccsdtq_t2.out
53 ccsdtq_t3.out
74 ccsdtq_t4.out
180 total
```

These 180 equations – each of which may include dozens of permutations of one contraction – are implemented as 50 KLOC of Fortran 77.

Side note: The operator algebra and code generation for CCSDTQ takes close to an hour. CCSDTQP exhausted the memory on all known machines capabile of running Python.

Summary

Know:

- Don't send a runtime to do an algorithm's job.
- Elemental/cyclic distribution solves critical data distribution problem of symmetric tensors.
- Communication-reducing algorithms for sorting and matmul can be reused for tensors.
- Binary contractions are just one piece of the puzzle.

Want:

- Generalized N-body requires continuous FMM for $r^{-\alpha}$
- Getting to science is hopeless without automation.

Acknowledgments

Berkeley: Edgar Solomonik