CPSFM Center for Predictive Simulation of Functional Materials

EXASCALE COMPUTING PROJECT

QMCPACK Users Workshop 2019

Oak Ridge National Laboratory

May 14-15, 2019

Auxiliary-Field quantum Monte Carlo

Miguel A. Morales, Fionn D. Malone
Lawrence Livermore National Laboratory

QMCPACK

Lawrence Livermore National Laboratory

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

CPSFM Center for Predictive Simulation of Functional Materials

Overview

- AFQMC Basics
  - Single particle basis sets, second quantization, etc.
  - Slater determinants, Hubbard-Stratonovich transformation, etc.
- Real vs Orbital Space
  - Advantages/Disadvantages of each approach.
- AFQMC in QMCPACK
  - Workflow, features, examples.
- Tools (Fionn D. Malone)

QMCPACK

Lawrence Livermore National Laboratory

Useful references

- Zhang, S.; Krakauer, H. Quantum Monte Carlo Method using Phase-Free Random Walks with Slater Determinants. Phys. Rev. Lett. 2003, 90, 136401.
- Purwanto, W.; Zhang, S. Quantum Monte Carlo method for the ground state of many- boson systems. Phys. Rev. E 2004, 70, 056702.
- Suewattana, M.; Purwanto, W.; Zhang, S.; Krakauer, H.; Walter, E. J. Phaseless auxiliary-field quantum Monte Carlo calculations with plane waves and pseudopotentials: Applications to atoms and molecules. Phys. Rev. B 2007, 75, 245123.
- Motta, M.; Zhang, S. Ab initio computations of molecular systems by the auxiliary-field quantum Monte Carlo method. WIREs Comput. Mol. Sci. 2018, 8, e1364.
- Motta, M.; Zhang, S. Computation of Ground-State Properties in Molecular Systems: Back-Propagation with Auxiliary-Field Quantum Monte Carlo. J. Chem. Theory Com- put. 2017, 13, 5367.
- Vitali, E.; Shi, H.; Qin, M.; Zhang, S. Computation of dynamical correlation functions for many-fermion systems with auxiliary-field quantum Monte Carlo. Phys. Rev. B 2016, 94, 085140.
- Motta, M.; Zhang, S. Communication: Calculation of interatomic forces and optimization of molecular geometry with auxiliary-field quantum Monte Carlo. J. Chem. Phys. 2018, 148, 181101.
- C.-C. Chang, B. M. Rubenstein, and M. A. Morales, “Auxiliary-field-based trial wave functions in quantum Monte Carlo calculations,” Phys. Rev. B **94**, 235144 (2016).
- Shuai Zhang, Fionn D Malone, Miguel A Morales, "Auxiliary-field quantum Monte Carlo calculations of the structural properties of nickel oxide", *J. Chem. Phys.* **149**, 164102 (2018).
- F.D. Malone, Shuai Zhang, Miguel A Morales, et al. “Overcoming the Memory Bottleneck in Auxiliary Field Quantum Monte Carlo Simulations with Interpolative Separable Density Fitting”, J. *Chem. Theory Comput.*, 2019, 15 (1), pp 256–264.
- Edgar Josue Landinez-Borda, John Gomes, Miguel A. Morales, Non-orthogonal multi-Slater determinant expansions in auxiliary field quantum Monte Carlo**,** J. Chem. Phys. **150**, 074105 (2019)

AFQMC: Projection QMC in Orbital Space

Shares many features with DMC

- Discrete walker representation of imaginary-time evolving distribution

$$\boxed{|\Psi_0\rangle = \int f(\phi)|\phi\rangle d\phi} \longrightarrow \boxed{|\Psi_0\rangle = \sum_{\phi} w_\phi |\phi\rangle,}$$

- Short time approximation to imaginary-time propagator with Trotter-like factorization

$$\boxed{|\Psi_0\rangle \propto \lim_{\tau\to\infty} e^{-\tau\hat{H}}|\Psi_T\rangle;} \qquad \boxed{|\Psi^{(n+1)}\rangle = e^{-\Delta\tau\hat{H}}|\Psi^{(n)}\rangle,}$$

- Mixed estimator for the energy. Backward walking or mixed estimator for general observables.

$$\boxed{\langle\hat{O}\rangle = \lim_{n\to\infty}\frac{\langle\Psi^{(n)}|\hat{O}|\Psi^{(n)}\rangle}{\langle\Psi^{(n)}|\Psi^{(n)}\rangle}} \qquad \boxed{E_0 = \lim_{n\to\infty}\frac{\langle\Psi_T|\hat{H}|\Psi^{(n)}\rangle}{\langle\Psi_T|\Psi^{(n)}\rangle}}$$

Walker Representation

- (N, N_α, N_β): # of electrons (total, spin up, spin down)

- $|\xi_i\rangle$ $(i = 1, \dots, M)$: Single-particle basis set.
  - Lattice site on a model Hamiltonian, plane waves, LCAO, etc.
  - Molecular orbitals, localized orbitals, etc.

- $c_i, c_i^\dagger$: annihilation and creation operators for the basis set .

- $|\psi_n\rangle$: Single-particle orbital set.
  - $|\psi_n\rangle = \sum_{i=1}^{M} D_{i,n} |\xi_i\rangle$

- In AFQMC, walkers represent Slater determinants. A walker is defined by a weight and the (M, N) Slater matrix $(D_{i,n})$ defining its occupied orbitals.
  - $|D\rangle = \prod_{n=1}^{N} \psi_n^\dagger \, |0\rangle$ Slater Matrix= $\begin{bmatrix} D_{1,1} & \cdots & D_{1,N} \\ \vdots & \ddots & \vdots \\ D_{M,1} & \cdots & D_{M,N} \end{bmatrix}$

Second Quantization

- Real space electronic Hamiltonian:
  - $\widehat{H} = \widehat{H}_1 + \widehat{H}_2 = -\frac{\hbar^2}{2m}\sum_i \nabla_i^2 + \sum_i V_{ext}(\overrightarrow{r_i}) + \sum_{i<j} V_{e-e}(\,|\overrightarrow{r_i} - \overrightarrow{r_j}|\,)$

- Given a single particle basis set, the corresponding second quantized Hamiltonian becomes:
  - $\widehat{H} = \sum_{ij} T_{ij}\, c_i^\dagger c_j + \frac{1}{2}\sum_{ijkl}(ik|jl)\, c_i^\dagger c_j^\dagger c_l c_k$
  - $\widehat{H}_2 = \frac{1}{2}\hat{v}_0 + \frac{1}{2}\sum_n \hat{v}_n^2$

- The Hamiltonian is defined through its matrix elements in the basis:
  - $(ik|jl) = <\xi_i\ \xi_j|\widehat{H}_2|\xi_k\ \xi_l>$
    $= \int d\vec{r}_1\vec{r}_2\ \xi_i^*(\vec{r}_1)\,\xi_k(\vec{r}_1)\,V_{e-e}(\,|\overrightarrow{r_1} - \overrightarrow{r_2}|\,)\ \xi_j^*(\vec{r}_2)\xi_l(\vec{r}_2)$
  - $T_{ij} = <\xi_i\,|\widehat{H}_1|\,\xi_j>$
    $= \int d\vec{r}_1\ \xi_i^*(\vec{r}_1)\,[-\frac{\hbar^2}{2m}\nabla_i^2 + V_{ext}(\overrightarrow{r_1})]\xi_j(\vec{r}_1)$

Walker Propagation - I

- One body propagators are easy!
  - $|D'\rangle = e^{(\sum_{ij} c_i^\dagger B_{ij} c_j)}|D\rangle \;\rightarrow\; D' = e^B D$
  - Main idea of AFQMC: Represent many-body propagator only with 1-body operators!

- Hubbard-Stratonovich (HS) transformation:
  - $e^{-\left(\frac{\tau}{2}\right)\hat{v}^2} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} d\sigma\, e^{-\sigma^2/2} e^{\sigma\sqrt{-\tau}\,\hat{v}}$

- Standard Approximations
  - $e^{-\beta\hat{H}} = \prod_n e^{-\tau\hat{H}}, \quad \beta = n\tau$ Break-up in timesteps
  - $e^{-\tau\hat{H}} = e^{-\frac{\tau}{2}\hat{H}_1} e^{-\tau\hat{H}_2} e^{-\frac{\tau}{2}\hat{H}_1} + O[\tau^2]$ Trotter approximation

- AFQMC: Many-body propagator as an integral over 1-body propagators
  - $e^{-\beta\hat{H}} = \prod_n \int d\vec{\sigma} P(\vec{\sigma})\hat{B}(\vec{\sigma}) + O[\tau^2]$
  - $\hat{B}(\vec{\sigma}) = e^{-\frac{\tau}{2}\hat{H}_1} e^{i\sqrt{\tau}\,(\vec{\sigma}\cdot\hat{v})} e^{-\frac{\tau}{2}\hat{H}_1}$

Walker Propagation - II

- Efficiency improvements: Similar to DMC
  - Mean-field subtraction: $v_n^{MF} = \langle\Psi_T|\hat{v}_n|\Psi_T\rangle / \langle\Psi_T|\Psi_T\rangle$

$$\hat{H}_2 = \left(\frac{1}{2}\hat{v}_0 + \sum_n v_n^{MF}\,\hat{v}_n - \frac{1}{2}\sum_n (v_n^{MF})^2\right) + \frac{1}{2}\sum_n (\hat{v}_n - v_n^{MF})^2$$

  - Importance sampling and force bias : $v_b = -\langle\Psi_T|\hat{v}|\phi_n\rangle / \langle\Psi_T|\phi_n\rangle$

$$\int d\sigma\, e^{-\sigma^2/2} e^{\sigma\hat{v}} \rightarrow \int d\sigma\, \frac{\langle\Psi_T|\hat{B}(\sigma)|\phi\rangle}{\langle\Psi_T|\phi\rangle} e^{-\sigma^2/2} e^{v_b\sigma - v_b^2/2} e^{\,(\sigma - v_b)\hat{v}}$$

$$|\Phi_{AF}\rangle = \frac{1}{N_w}\sum_n w_n|\phi_n\rangle \rightarrow \frac{1}{N_w}\sum_n w_n \frac{|\phi_n\rangle}{\langle\Psi_T|\phi_n\rangle}$$

  - Propagation step:
    1. $|\phi_n\rangle \rightarrow \hat{B}(\vec{\sigma})|\phi_n\rangle$
    2. $w_n \rightarrow w_n * \frac{\langle\Psi_T|\hat{B}(\sigma)|\phi\rangle}{\langle\Psi_T|\phi\rangle} e^{v_b\sigma - v_b^2/2} \approx e^{-[E_L(\phi)+E_L(\phi')]/2}$

"Local Energy Approximation"

"Hybrid Propagation"

Walker Propagation - III

- Sign problem: Phaseless approximation
  - $\hat{B}(\vec{\sigma}) = e^{-\frac{\tau}{2}\hat{H}_1} e^{i\sqrt{\tau}\,(\vec{\sigma}\cdot\hat{v})} e^{-\frac{\tau}{2}\hat{H}_1}$
  - Complex HS transformation leads to sign problem. Same origin as DMC sign problem. Same solution, but in Slater determinant space.

$$E = \frac{\langle \Psi_T | \hat{H} | \Phi_{AF} \rangle}{\langle \Psi_T | \Phi_{AF} \rangle} = \frac{\sum_n |w_n \langle \Psi_T | \phi \rangle|\ e^{i\theta(\phi)}\ E_L(\phi_n)}{\sum_n |w_n \langle \Psi_T | \phi \rangle|\ e^{i\theta(\phi)}}$$

  - Project random walk to real axis. Leads to real weights.

$$w_n \rightarrow w_n * \left| \frac{\langle \Psi_T | \hat{B}(\sigma) | \phi \rangle}{\langle \Psi_T | \phi \rangle}\ e^{v_b \sigma - \frac{v_b^2}{2}} \right| * \max(0, \cos(\Delta\theta))$$

$$\Delta\theta = \text{phase of } \frac{\langle \Psi_T | \hat{B}(\sigma) | \phi \rangle}{\langle \Psi_T | \phi \rangle}$$

  - Importance sampling leads to a seamless application of the approximation.

Observables

- Mixed distribution
  - $\left\langle \hat{O} \right\rangle_M = \frac{\left\langle \Psi_T \middle| \hat{O} \middle| \Phi_{AF} \right\rangle}{\langle \Psi_T | \Phi_{AF} \rangle} = \frac{\sum_n w_n O_M(\phi_n)}{\sum_n w_n}$, $O_M(\phi_n) = \frac{\left\langle \Psi_T \middle| \hat{O} \middle| \phi_n \right\rangle}{\langle \Psi_T | \phi_n \rangle}$
  - $\left\langle \hat{O} \right\rangle_M$: quadratic in $|\Psi_T - \Psi_{Exact}|$ for operators that commute with Hamiltonian. Linear for all others.
  - Small overhead, but can depend significantly on quality of Ψ_T.

- Back propagation:
  - $\left\langle \hat{O} \right\rangle_{BP} = \frac{\left\langle \Phi'_{AF} \middle| \hat{O} \middle| \Phi_{AF} \right\rangle}{\left\langle \Phi'_{AF} \middle| \Phi_{AF} \right\rangle} = \frac{\sum_n w_n O(\phi_m, \phi_n)}{\sum_n w_n}$, $O(\phi_m, \phi_n) = \frac{\left\langle \phi_m \middle| \hat{O} \middle| \phi_n \right\rangle}{\langle \phi_m | \phi_n \rangle}$
  - $\left\langle \hat{O} \right\rangle_{BP}$: quadratic in $|\Psi_T - \Psi_{Exact}|$.
  - Typically requires an overhead of 50-75%.

Real vs Orbital Space

AFQMC

- Direct connection between ab-initio and model Hamiltonians
- Flexible treatment of core electrons:
  - All-e, frozen-core, ECP, NCPP, LAPW, etc.
- Spin-orbit coupling easy to incorporate.
- Typically, smaller bias from phaseless approximation.
- Efficient/simple code.
  - GEMM, QR, Inverse

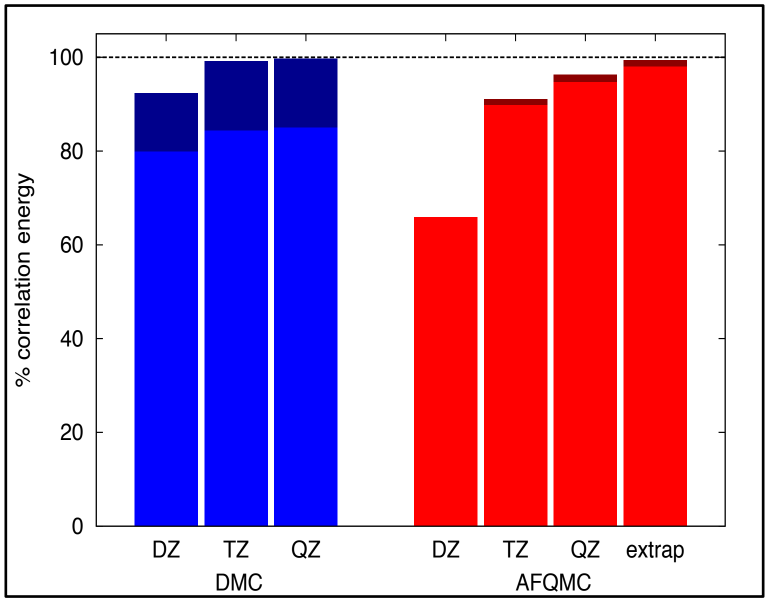
- Small ab-initio community.
- Basis set error.
  - Error cancellation is “transferred” to the basis.
- Requires 2-electron integrals.
  - ~~M^3 memory cost for propagation~~
  - ~~M^4 memory cost for energy~~
  - M^{2-3} memory cost
  - No “direct” algorithm.
  - Forces require xN_{atoms} more memory.

DMC

- Trivial explicit correlation.
  - Jastrow is almost free!
- No basis set extrapolation. Works at CBS limit.
- Memory friendly.
- Intuitive.

- Hard to simplify.
- Uncontrolled approximation needed for non-local ECP channels.
- Fixed-node error is large.
  - Error cancellation is needed for accurate results.
- Spin-orbit is not trivial (to me!!!).
- Divergent potentials.
  - Forces are noisy!

Real vs Orbital Space



Illustrative DMC – AFQMC comparison

- Cohesive energy of 2-atom unit cell of Carbon in diamond structure
- Bright (dark) region corresponds to a calculation with a single determinant (converged MSD).
    - SD DMC consistently recovers ~80% of the total correlation energy.
    - Fixed node error of ~20%.
    - Converged DMC-MSD almost reaches CBS at TZ.
    - Large basis set error for AFQMC.
        - Requires basis set extrapolation.
    - SD-AFQMC shows small phaseless error.
        - Short determinant expansion recovers remaining error.

Disclaimer: For illustration purposes only!

AFQMC in QMCPACK: Design Principles

- Code still under active development.
  - Always check developers version for latest functionality.
  - Significant extensions every 3-6 months.
- Core AFQMC algorithms within QMCPACK.
  - Mean-field calculations performed by external codes.
    - PySCF, Molpro, GAMESS, VASP, etc.
  - Basic input: 1-2 electron integrals and trial wave functions.
  - AFQMC code doesn’t ”know” about atoms, positions, basis sets, pseudopotentials, etc.
- Focus on efficient, large scale calculations.
  - Multiple levels of parallelization and data distribution.
    - Distribution of integrals over nodes.
  - xml input, hdf5 data, python analysis.

Sample Workflow

- Simple workflow. Relies on converter tools and post-processing scripts.
  - Many still under development.
  - Contributions are welcome.
  - Properties require custom (simple) post-processing scripts. Examples are offered for charge and spin densities.
- Sample workflow:
  1. python scf.py
  2. python pyscf_to_afqmc.py   [options]
  3. qmcpack afqmc.xml
  4. qmca [options] sample.scalar.dat
  5. python [options] –i sample.scalar.h5 analysis.py

Input Files - I

- Request AFQMC calculation with:
  - <simulation method="afqmc">
- Non-execute blocks must be “named” and are parsed first.
- “Execute” blocks are executed sequentially afterwards.
- Most decisions are made when pyscf_to_afqmc.py is executed.

```xml
<AFQMCInfo name="info0">
  <parameter name="NMO">2048</parameter>
  <parameter name="NAEA">256</parameter>
  <parameter name="NAEB">256</parameter>
</AFQMCInfo>

<Hamiltonian name="ham0" info="info0">
  <parameter name="filename">integrals.h5</parameter>
</Hamiltonian>

<Wavefunction name="wfn0" type="MSD" info="info0">
  <parameter name="filetype">ascii</parameter>
  <parameter name="filename">wfn.dat</parameter>
</Wavefunction>

<WalkerSet name="wset0">
  <parameter name="walker_type">closed</parameter>
</WalkerSet>

<Propagator name="prop0" info="info0">
</Propagator>
```

Input Files - II

- Execute blocks typically require some minimal customization.
- Observables are defined locally in these blocks.
- Wavefunctions used in observables can be independent from the one used for propagation (phaseless constraint and importance sampling).
- Triple loop structure:
  - blocks: Observables are calculated and data is written to disk. Checkpoint occurs if requested.
  - steps: Branching and load balancing at every step. Orthogonalization occurs with given frequency.
  - substeps: Only propagation.

```xml
<execute wset="wset0" ham="ham0" wfn="wfn0" prop="prop0" info="info0">
  <parameter name="timestep">0.005</parameter>
  <parameter name="blocks">1000</parameter>
  <parameter name="steps">10</parameter>
  <parameter name="substeps">4</parameter>
  <parameter name="nWalkers">100</parameter>
  <Estimator name="energy" wfn="wfn_estimator">
    <parameter name="print_components">yes</parameter>
  </Estimator>
</execute>
```

Hamiltonians

Cholesky Factorization

$$(ik|jl) = M_{\mathrm{ik},jl} = \sum \mathrm{L}_{ik}^{n} \mathrm{L}_{lj}^{n*} .$$

Sparse Representation: (Only option for molecular problems)

- Storage of L_{ik}^{n} and $M_{ik,jl}$ as a sparse matrices. Elements below a certain cutoff are set to zero.
- Automatically incorporates any symmetries present in the system.
- Relies on less efficient sparse linear algebra (compared to dense storage).
- Very large setup cost! Must generate M from L, $O[N^2 M^2 n_{chol}]$.

K-Point Representation: (Only for periodic systems with k-points)

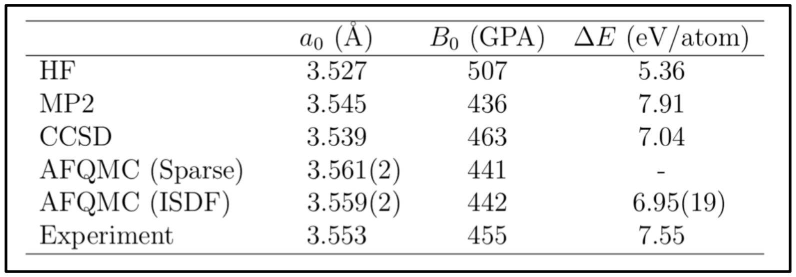
- Explicit use momentum conservation. Ideal for small unit cells with lots of k-points.
- Reduction of $1/N_k$ in memory and computation cost compared to direct algorithm.
- Based on highly efficient batched linear algebra operations.
- Scales poorly with size of the primitive cell. Not practical for large supercells.

Tensor Hyper-Contraction

$$(ik|jl)= \sum \phi_{i,\mu}^{*} \phi_{k,\mu} M_{\mu,\nu} \phi_{j,\nu}^{*} \phi_{l,\nu}$$

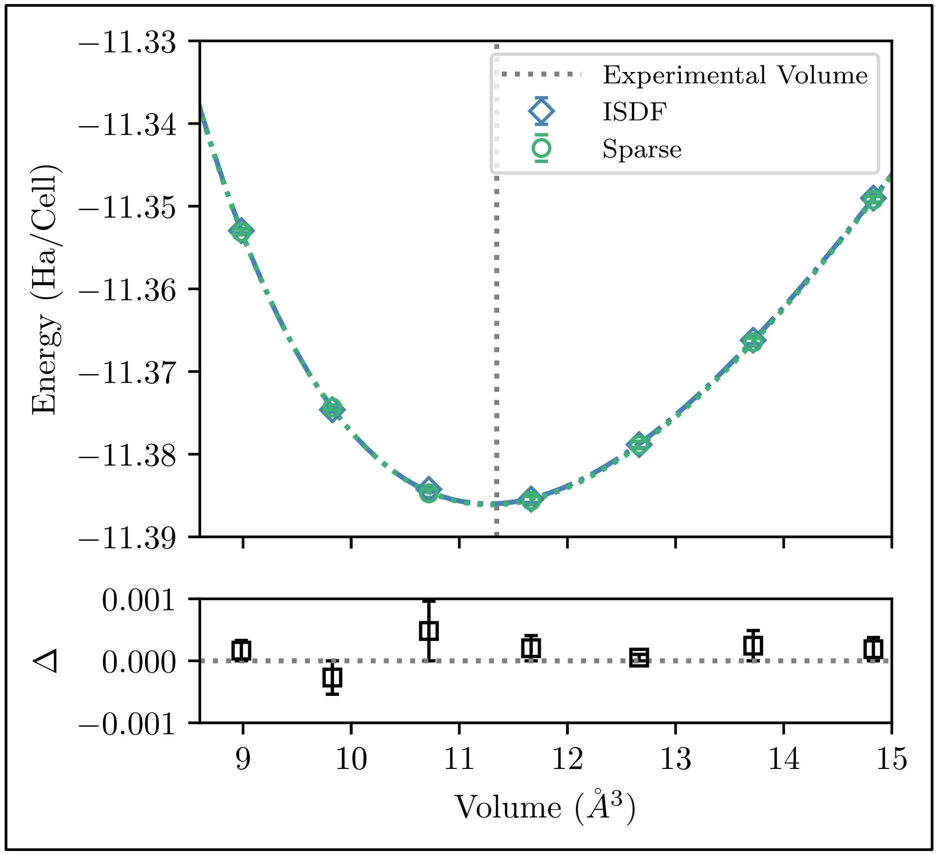
- Factorize integral tensor as a contraction of 2nd-order tensors. Quadratic memory requirements. Cubic computational cost.
- Popularized in QC by Todd Martinez. Made practical in PBC by Lin Lin.
- Ideal for large supercells at the Gamma point or large molecular problems with no symmetry.
- Does not currently benefit from symmetries in the system.
- Large prefactor, specially for small system sizes. Currently problematic for heavier atoms.

Cholesky vs THC



	a_0 (Å)	B_0 (GPA)	ΔE (eV/atom)
HF	3.527	507	5.36
MP2	3.545	436	7.91
CCSD	3.539	463	7.04
AFQMC (Sparse)	3.561(2)	441	-
AFQMC (ISDF)	3.559(2)	442	6.95(19)
Experiment	3.553	455	7.55

- THC (ISDF) is only moderately slower even though it does not make use of any symmetries.
- Only practical choice for large unit cells.
  - Actual calculations are done in a supercell representation.
- Same accurate answer regardless of approach.



F.D. Malone, Shuai Zhang, Miguel A Morales, et al. “Overcoming the Memory Bottleneck in Auxiliary Field Quantum Monte Carlo Simulations with Interpolative Separable Density Fitting”, J. *Chem. Theory Comput.*, 2019, 15 (1), pp 256–264. 18

Hamiltonians

Systems without K-Point Symmetry

Representation	Memory		Computation	
	HS Potential	Energy	HS Potential	Energy
Sparse Cholesky	$s\ x_c\ M^3$	$s'\ N^2M^2$	$s\ x_c\ M^3$	$s' N^2M^2$
Dense Cholesky	$x_c\ M^3$	$x_c\ NM^2$	$x_c\ M^3$	$x_c\ N^2M^2$
THC	$x_\mu^2\ M^2$	$x_\mu^2\ M^2$	$x_\mu\ M^2$	$x_\mu^2\ M^2$

Systems with K-Point Symmetry (N_k: # kpoints)

Representation	Memory		Computation	
	HS Potential	Energy	HS Potential	Energy
Sparse Cholesky	$s\ x_c\ M^3$	$s'\ N^2M^2$	$s\ x_c\ M^3$	$s' N^2M^2$
Dense Cholesky	$x_c\ N_k^2\ m^3$	$x_c\ N_k^2\ nm^2$	$x_c\ N_k^2\ m^3$	$x_c\ N_k^3\ n^2\ m^2$

- Always useful to use K-Point representation.
- Dense representation is generally more efficient.

s, s': Sparsity of corresponding structures
x_μ: THC prefactor, typically 5-20.
x_c: Cholesky prefactor, typically 5-20.

Orthogonal vs Non-Orthogonal Determinants

$$|\Phi_T\rangle = \sum_i c_i |D_i\rangle$$

Orthogonal (PHMSD)

$$\langle D_i | D_j \rangle = \delta_{ij}$$

- Robust and well-controlled.
- Some form of CI typically involved in construction.
  - Truncated-CI, CASCI, selected CI, etc.
- Size consistency problems.
  - Possibly fatal for bulk.
- Fast evaluation algorithms:
  - Cost proportional to rank of the update
- Lower memory overhead for large expansions.

Non-Orthogonal (NOMSD)

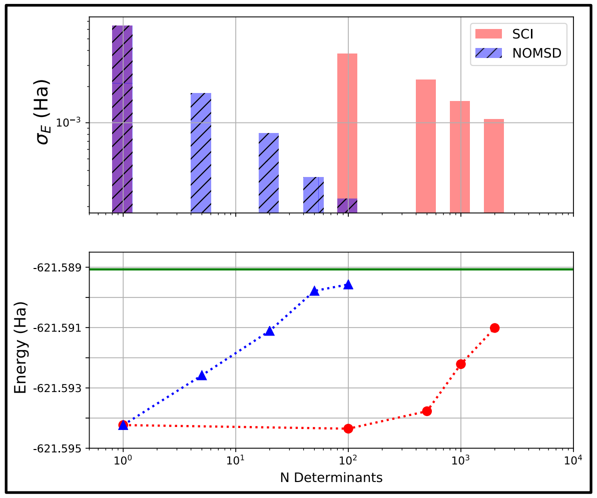
$$\langle D_i | D_j \rangle \neq \delta_{ij}$$

- Not common/routine in QC community.
- Several algorithms:
  - Iterative HF: 1-det at a time
  - Resonating HF: N-dets simult.
  - Complicated optimization problem.
- Size consistency problem less severe.
- Fastest convergence in MSD ansatz.
- Linear cost with #dets.
  - Both computation and memory.
  - Large memory overhead ultimately makes the approach impractical for large systems.

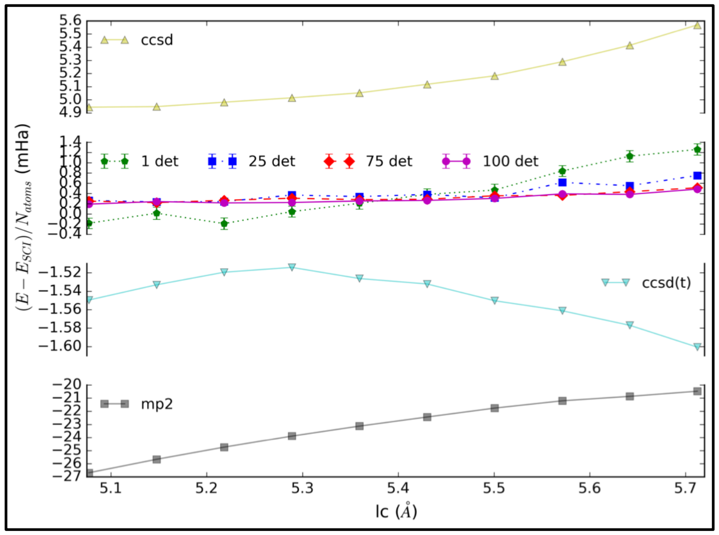
- Different wavefunctions can be combined in a single execution.
  - NOMSD for propagation, PHMSD for energy evaluation.

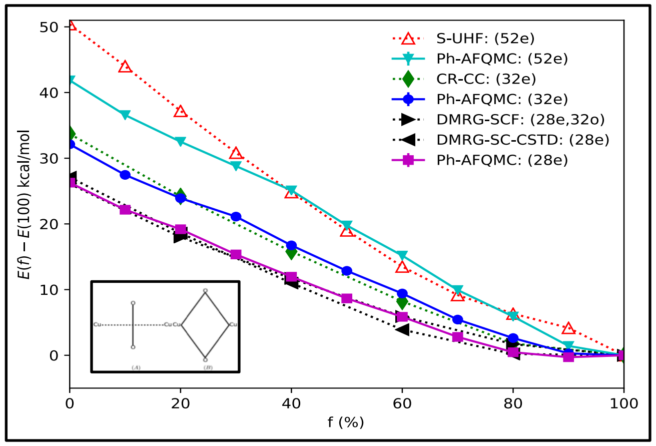
Systematic Improvable Wavefunctions

NOMSD vs PHMSD



Si (diamond) with NOMSD





Cu_2O_2 with NOMSD

- NOMSD offer great promise for compact and accurate trial wavefunctions.
- Mostly unexplored territory.
    - Great opportunity!

Observables

- Slowly expanding list of observables.
  - Currently: energy, static 1-RDM
    - 1-RDM can be used to calculate many properties: spin and charge densities, occupancies, momentum distribution, magnetic moments, etc.
  - In development: dynamic 1-RDM, 2-RDM, contractions of the 1-/2-RDM, forces.
- Back propagation (in principle) available for any observable.
  - Currently limited to 1-RDMs.

```
<Estimator name="back_propagation">
  <parameter name="nsteps">500</parameter>
  <parameter name="naverages">5</parameter>
  <parameter name="ortho">25</parameter>
  <parameter name="block_size">10</parameter>
</Estimator>
```

Data Distribution

- Large data structures can be distributed over multiple memory spaces (nodes, GPUs, etc).
  - *“**nnodes**”* parameter in wavefunction and propagator.
  - Distributed algorithms are automatically chosen if nnodes > 1.
  - Enables calculations on large systems.
    - Some communication overhead. Typically modest.
  - Does not change the number of walkers, only static data layout.
- Walker sets can be shared over multiple cores in a node.
  - Use “***ncores***” parameter in the execution block.
  - Operations on the walker set are distributed over all cores in the group.
    - nWalkers defines the number of walkers per “working” group.
  - Improved time to solution.
    - Allows 1 walker per node, useful for large systems.

Warning: Use with caution!

GPUs

- AFQMC code is actively being ported to GPU.
  - Currently: Production level GPU port for K-Point dense Hamiltonian.
    - 15-40x speed-up for DP, 10-25x speed-up for SP compared to good CPU.
    - Larger performance gains for large systems.
  - Mixed precision build recommended for GPUs.
    - Memory is the more important resource!
  - No real build on GPUs yet (no molecules or gamma point).
  - Summit/Sierra enable AFQMC calculations on systems with 250 atoms.
    - Bcc Fe with 6x6x6 k-point grid.
    - C (diamond) with 5x5x5 k-point grid.
    - AFM-NiO with 3x3x3 k-point grid including back propagation and magnetic moment.

Upcoming Features

- Full GPU coverage (3-6 months)
- Finite-T algorithm (3 months)
- Spin-Orbit coupling and non-collinear magnetism (6 months)
- Forces (6-12 months)
- VASP interface (3 months)
- K-Point THC (6-12 months)