Quantum Monte Carlo Calculations at the Petascale

David Ceperley

Department of Physics, NCSA

University of Illinois at Urbana-Champaign

- QMC
- The Endstation Project
- Challenges at the petascale
"The general theory of quantum mechanics is now almost complete. The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble."

Dirac, 1929

\[ \hat{H} = -\sum_{i=1}^{N} \frac{\hbar^2}{2m_i} \nabla_i^2 + \sum_{i<j} \frac{e_ie_j}{r_{ij}} \]

\[ \hat{H}\Psi(r_1, r_2, \ldots) = E\Psi(r_1, r_2, \ldots) \]

Maxwell, Boltzmann and Schrödinger gave us the model (at least for condensed matter physics.) Hopefully, all we must do is numerically solve the mathematical problem and determine the properties. (first principles or ab initio methods) Without numerical calculations, the predictive power of quantum mechanics is limited.
Nature sets a very high standard for accuracy!

1eV = 11,600K  
Room temperature = 0.025eV

Modeling of processes relevant to materials, chemistry, biology,... needs to be accurate at the level of 0.01eV ~ 0.4mH

Examples for small molecules: error in binding energy

O$_3$, H$_2$O$_2$, C$_2$, F$_2$, Be$_2$, ...
Si$_2$, P$_2$, S$_2$, Cl$_2$
As$_2$, Br$_2$, Sb$_2$
TiO, MnO

Current methods are inadequate
QMC typically accurate to 0.1eV
Quantum Monte Carlo

- Premise: need to use simulation techniques to "solve" many-body quantum problems just as you need them classically.
- Both the wavefunction and expectation values are determined by the simulations. Correlation built in from the start.
- QMC gives most accurate method for general quantum many-body systems.
- QMC electronic energy is a standard for approximate DFT calculations. (3rd largest citation in PRL.)
- provide a new understanding of quantum phenomena and a practical tool
- A continuum of stochastic methods:
  - Variational Monte Carlo (VMC)
  - Projector Monte Carlo methods for T=0:
    - Diffusion Monte Carlo (DMC)
    - Reptation MC (RQMC)
    - Auxiliary field QMC (AFQMC)
  - Path Integral Monte Carlo for T>0 (PIMC)
  - Coupled electron-ion Monte Carlo T>0 (CEIMC)

Goal is NOT large N, but higher accuracy and new capabilities
Breakthrough Quantum Monte Carlo simulations:

- Hard-core bosons on a CDC 6600 (1974)
- Electron gas on CRAY-1 (1980)
- Superfluid helium (1984)
- Ground state of solid hydrogen at high pressures, CRAY XMP and CYBER 205 (1987)
- Coupled Electron-Ion Monte Carlo simulations of dense hydrogen on Linux Clusters (2000s)
MD and MC Simulations

- Hard sphere MD/MC ~1953 (Metropolis, Alder)
- Empirical potentials (e.g. Lennard-Jones) ~1960 (Verlet, Rahman)
- Local density functional theory ~1985 (Car-Parrinello)
- Quantum Monte Carlo (CEIMC) ~2000

- Initial simulations used semi-empirical potentials.
- Much progress with “ab initio” molecular dynamics simulations where the effects of electrons are solved for each step.
- However, the potential surface as determined by density functional theory is not always accurate enough
- QMC+MD = CEIMC is a candidate for petascale “killer app”
QMCPACK

- Open-source library and application package to perform Quantum Monte Carlo (QMC) Simulations
- Implements various QMC algorithms: VMC, DMC, RQMC
  - Generic representations of the physical entities and models
  - Object-oriented implementation of QMC algorithms (C++)
  - Generic programing of computational Kernels
- Designed for large-scale QMC simulations of molecules, solids and nanostructures: OpenMP/MPI Hybrid parallelization, effective for multi-core systems
- Standard open-source libraries and utilities for development, compilation and executions
- Adopts XML/HDF5 for I/O
- Developed at UIUC [http://ww.mcc.uiuc.edu/qmc/](http://ww.mcc.uiuc.edu/qmc/)

Principal author: Jeongnim Kim  UIUC
Diffusion Monte Carlo algorithm

Evolution of walker in one step (seconds)

Birth-death process for walkers

Population control

Load balancing
One-body orbitals

- Fixed-node approximation in DMC/RMC uses the “nodes” of
  \[ \Psi_{\Delta S}(\{R\}) = D^\dagger D \]

- Elementary operation: random access to wf tables

- What method is used to obtain \( \{\phi_k(r)\} \) matters for accuracy!
  - Hartree-Fock? DFT? Hybrid ? or beyond HF/DFT
  - How to represent \( \{\phi_k(r)\} \) matters for efficiency!
    - Plane-wave basis set
    - Molecular orbitals in a localized atomic basis set: Gaussian- and Slater-type orbitals and numerical orbitals
    - Real-space grid

- we have to calculate \( \phi_k(r), \nabla \phi_k(r), \text{ and } \nabla^2 \phi_k(r) \text{ at } r \)
  - Can use any combination of mixed basis sets that are optimized for the performance and computing resources.

- This is the time consuming step today
Cost of Evaluations of Wave Function

- **Molecular Orbitals**
  \[ \psi_i(r) = \sum_l \sum_\alpha c_{i,\alpha}^l \phi_{i,\alpha}(r - R_l) \sim N^3 \]

- **Plane-wave**
  \[ \psi_i(r) = \sum_G c_{i,G} \exp^{iG \cdot r} \sim N^3 \]

- **B-spline**
  \[ \psi_i(r) = \sum_l c_{i,l}^l f_l(r) \sim N^2 \]
DMC scales well: hybrid approach

Current work: extend production codes to > 30,000 nodes
What happens when we scale up? Say by a factor of 100

- Can one distribute more walkers across nodes?
  - Will reduce statistical error by factor of 10
  - But wall clock time will be the same
  - Systematic errors will be the same
  - Load balancing/fault tolerance problems
  - For most problems, we will not achieve the science goals

→ Need to find more parallelism to keep errors the same but reduce wall clock time and reduce systematic errors

- Vulnerable to software and hardware errors?
- Are existing pseudo-random number generators adequate at the petascale? Tested SPRNG for 28K random number streams with $10^{10}$ numbers/stream.
DMC algorithm and OpenMP/MPI parallelization

- Memory limitations of the current MPI parallelization
- Will total memory per SMP node will grow with more cores: 8-64 GB will be sufficient for many problems.
- One or more walker \( \{R\} \) per core (OpenMP thread)
- Read-only large-scale data (e.g., wavefunctions) are shared but others objects are allocated per per thread.
DMC Manager Library

- Multi-level scheme of managers, sub-managers, workers.
- Walker point of view:
  - After a walker completes an elementary step, it sends averages, and branching information up the tree
  - Looks for new configurations to send or receive.
- Manager
  - Decides on load balancing and sends transfer information down the tree
  - Provides overall stability of population, checkpointing and I/O of averages
  - When processors die, those branches are pruned
  - Continually monitor statistical “health” of nodes
- Eliminate need for synchronization, blocking calls,… We hope to achieve zero communication costs.
- Can run in an unstable environment
## Multiple QMC Paths to Parallelism

<table>
<thead>
<tr>
<th></th>
<th>Degree of parallelism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walkers/time slices (ensemble)</td>
<td>100-10,000</td>
</tr>
<tr>
<td>Boundary conditions: “twist angles”</td>
<td>100</td>
</tr>
<tr>
<td>Wavefunction pieces</td>
<td>100</td>
</tr>
<tr>
<td>Ionic positions for path integrals</td>
<td>10</td>
</tr>
<tr>
<td>Parameter studies</td>
<td>10</td>
</tr>
</tbody>
</table>
The ENDSTATION TEAM

- Jeongnim Kim, David Ceperley, University of Illinois at Urbana-Champaign
- Henry Krakauer, Shiwei Zhang, College of William and Mary
- Paul Kent, Oak Ridge National Laboratory
- Richard Hennig, Cyrus Umrigar, Cornell University
- Lubos Mitas, North Carolina State University
- Ashok Srinivasan, Florida State University

+ many students, postdocs and other collaborators

The Endstation project is supported by: DOE-DE-FG05-08OR23336

Computer time through the INCITE program