Algorithms for simulating quantum mechanics

Bryan Clark
CSE Seminar
Why care about algorithms for quantum mechanics?

Accurate and efficient quantum mechanical simulations would have revolutionary changes to industrial applications, what we know about our universe and the way physics is done.

We know the rules.
Just need to figure out how to solve them.
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.  – Paul Dirac (1929)
It’s lucky for us that they solved this problem a long time ago...

Unfortunately almost everything is a fermion.

Markov Chain Monte Carlo techniques still amongst the best and most used techniques.
Even water is hard...
Since that time, we’ve seen that there appears to be an exponential wall to simulating quantum systems.

Why should that be?
$2^n$ possible electron configurations

Quantum mechanics tells us we are simultaneously in a superposition of electronic configurations.

$$\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}$$

Superconductor: electrons like to be on top of each other.

Our goal: Find out where the electrons want to be.

Minimal eigenstate of $H = -t \sum_{ij} c_i^\dagger c_j + U \sum_i c_i^\dagger c_i^\dagger c_i^\dagger c_i^\dagger$

$2^n \times 2^n$ matrix
Linear Algebra

\[ H \Psi = E \Psi \] via Lanczos

2^n x 2^n matrix

State of the art: 48 spins

24 electrons

Each step is exponential

Spin liquid?

LiZn_2Mo_3O_8

Phase transitions

\[ \frac{\partial^2 E}{\partial^2 J'} \]
Aside: What’s a Spin Liquid

Insulator

Topological: degeneracy that depends on manifold

Anyonic Excitations
Guess a Wave-function

The true $\Psi$ has the property that $E = \Psi^T H \Psi$ is minimal over all $\Psi$.

We need a compact representation of a $2^n$ state vector. (Important open question)

In some cases, this needs to be an antisymmetric function on the electron positions

Venerable history: BCS Superconductivity
Quantum Hall Effect
Model Wave-functions

People such as Bardeen and Laughlin guessed these wave-functions without a computer by sheer genius (and were rewarded nobel prizes for it). We want to replace nobel prize winner with computers
Protootypical Wave-functions:

\[ \Psi(\uparrow, \downarrow, \uparrow, \uparrow) = 0.3 \]

\[ V_{\uparrow, \downarrow, \uparrow, \uparrow} \]

\[ v_{12} v_{13} v_{23} v_{14} v_{24} \]

\[ v_{\uparrow \downarrow} v_{\uparrow \uparrow} v_{\downarrow \uparrow} v_{\uparrow \uparrow} v_{\downarrow \uparrow} \]

\[ M_{\uparrow} M_{\downarrow} M_{\uparrow} M_{\downarrow} M_{\uparrow} \]

\[ | \Psi \rangle = \exp[-J(R)] \sum \alpha_k \det M_{\uparrow, k} \det M_{\downarrow, k} \]

\[ J(R) = \sum_{i,j} j(r_i - r_j) \]

\[ M_{ij} = \phi_i(r_j) \]

3d function \[ R \equiv \{r_1, r_2, \ldots, r_n\} \]

Unfortunately, these wave-functions are not compact enough. For the most interesting physical systems you need an exponential number of parameters (turn up the matrix size, the number of determinants, etc.) to get converged energies.
Beyond compactness, even for a small number of parameters (1000) we have a hard optimization problem:

Given a parameterization for $\phi_i(r_j)$ or $v_{ij}^A$ how do you find (even local) minima.

You only have stochastic access to the energy and each energy evaluation is slow!

Essentially an online learning problem.

Currently: Stochastic Gradient Descent

$\vec{\alpha} \rightarrow \vec{\alpha} + \delta \nabla_{\vec{\alpha}} E$

Stochastic Reconfiguration

$\alpha_i \rightarrow \alpha_i + \delta \text{sign}(\partial E / \partial \alpha)$

‘Time Evolution’

$\Psi = P(1 - \tau H) \Psi$

(Important open question)
Striped Spin Liquid Crystal

Topological + Nematic!

Herbertsmithite
Volborthite
Zn-Paratacamite
An aside on DMRG

The most important (physics) algorithms still running on one node.

Every other wavefunction gets optimized by stochastic variational means.

DMRG is optimized by alternating least squares.

Outer product 10,000 x 10,000

Diagonalize

SVD
Diffusion Monte Carlo

(Stochastic Power Series)

$$\lim_{M \to \infty} (1 - \tau H)^M \Psi = \Psi_0$$

$$\lim_{M \to \infty} G^M \Psi = \Psi_0$$

$$\sum_{ijklmn} G_{ij} G_{jk} G_{kl} \ldots G_{mn} \Psi_n = (\Psi_0)_i$$

$$P(i \to j) = \frac{|G_{ij}|}{\sum_j |G_{ij}|} \quad w(i \to j) = \sum_j |G_{ij}| \text{sign}(G_{ij})$$

Optimization: Walk w/ $(1 - \tau H)$ on manifold of parameterized states.

DMC: Walk w/ $(1 - \tau H)$
Quantum
The Exciton Bose Liquid

High Tc superconductors have a bad metal.
Can we find a bad metal in a simple Hamiltonian.

Signs of bad metal - X in the structure factor.
Parallelization

Unlike DMRG, diffusion Monte Carlo parallelizes extremely well.
A Sign Problem

The “only” problem in physics

Your errors on this are good

but your errors on this are horrible.

Exponential Relative Variance
Annihilation helps because paths of different signs cancel.

Annihilation fails because you can’t keep enough walkers to get cancellation of all paths.

Without

With but too few walkers

With
Variational Wave Function

\[ A_1 A_2 A_3 A_4 A_5 \]

\[ = 0.01 + 0.03 + \ldots \]

Sample

\[ \frac{1}{\langle MPS_1 | C \rangle} \]

weight

\[ \langle MPS_1 | C \rangle \]

Energy vs. \( \beta \)

- DMRG M=100
- Sample&Evolve M=50
- Sample&Evolve M=70
- Sample&Evolve M=100
Quantum Computing

Will quantum computers solve our problems?

Maybe...

**Modified Church-Turing Thesis**

All computers (physical systems) are essentially equivalent to your laptop.

Quantum Mechanics broke this

Quantum simulations are quantum computers kill application, not factoring!

Quantum computers can (probably) compute g.s. in poly-time.

Two (minor) problems:

- No quantum computers (16 qubits)
- 100 qubit simulation -> $10^{16}$ gates
“A method is more important than a discovery, since the right method will lead to new and even more important discoveries.”

– Lev Landau
Conclusions

- Quantum simulations are important:
  - and you’ve seen some exciting physics we’ve already discovered
- There are many beautiful algorithms
  - some of which we’ve developed.
- But we have a long way to go.
- Deep and interesting questions...
  - algorithms: FOCS/STOCS/ICML style-questions
  - More traditional CSE focus on parallelization
- (Also interested in quantum computing!)
- Interested in collaboration; come find me!
Striped Spin Liquid

\[ \nu_k c_k^\dagger \uparrow c_{-k, \downarrow}^\dagger |0\rangle \]

\[ M_{ij} = \phi(\vec{r}^\uparrow_i - \vec{r}^\downarrow_{ij}) \equiv \phi(\vec{r}_{ij}) \]
(Modified)

Church-Turing Thesis

Quantum computers can compute g.s. in polytime

\[ |0\rangle + |1\rangle \]

\[ |\Psi\rangle \rightarrow e^{iHt} \]

\[ |0\rangle\langle \Psi | + |1\rangle(e^{iE_0t}|\Psi_0\rangle + e^{iE_1t}|\Psi_1\rangle + e^{iE_2t}|\Psi_2\rangle) \]

Minor Problems:

- No quantum computers (16 qubits)
- 100 qubit simulation -> 1016 gates
Everything is quantum mechanics

‘Condensed Matter Physics’
‘Chemistry’
‘Materials’
‘Nuclear Physics’
‘Water’
‘Biology’

We know the rules.
Just need to figure out how to solve them.

Accurate and efficient quantum mechanical simulations would have revolutionary changes to industrial applications, what we know about our universe and the way physics is done.
Variational Monte Carlo

Variational Principle: $E_0 = \langle \Psi_0 | H | \Psi_0 \rangle \leq \langle \Psi_T | H | \Psi_T \rangle$

Highly *nonlinear* optimization with an objective function $\langle E[\Psi[\vec{\alpha}]] \rangle$ and derivatives $\frac{\partial \langle E \rangle}{\partial \alpha_i}$ which can only be evaluated *noisily* and *slowly*.

Aside: Analogous to online learning.
Church-Turing Thesis

All computers (physical systems) are essentially equivalent to your laptop.

Quantum Mechanics broke this

Simulating quantum mechanics is hard

Quantum Computing

Quantum information has started to play an important role in condensed matter.
Guessing a WaveFunction

- Striped Spin Liquid

\[ \Psi_{PBCS} = P \prod_{k} (u_{k} + v_{k}c_{k,\uparrow}^{\dagger}c_{-k,\downarrow}^{\dagger}) |0\rangle \]

\[ \langle R | \Psi_{PBCS} \rangle = \det M \]

\[ M_{ij} = \phi(\vec{r}_{\uparrow,i} - \vec{r}_{\downarrow,j}) \equiv \phi(\vec{r}_{ij}) \]
Some wave-functions

\[ M(i_1, i_2, \ldots, i_n) \]

\[ A(i_1, i_2, i_3, i_4)B(i_3, i_4, i_5, i_6)C(i_5, i_6, \ldots) \]

\[ M(i_1)M(i_2)M(i_3)M(i_4) \]

Optimization by alternating least squares
Very hard to parallelize.
Thinking about stochastic SVD
Our group works on simulating strongly correlated systems. We are currently attacking the Hubbard model as a stepping stone toward more sophisticated models.
An Exponential Problem

There’s an exponential wall to simulating quantum systems.

A (only slightly) biased view on the state of the art to attenuate it.

With a few new algorithms

- Partial Node FCIQMC
- Release FCIQMC
- Release + FN MPS
- Efficient Multi-MPS
- SEMPS
Approach I: Just write down the wave-function

Exponential number of terms

- **Multislater-Jastrow++**
  \[ |\Psi\rangle = \exp[-J(R)] \sum_k \alpha_k \det M_{\uparrow,k} \det M_{\downarrow,k} \]

  Chemical potential

  Exponential number of determinants

- **PEPS or Huse-Elser or MERA**

- **Multi non-orthogonal SD + symmetry projection**

- **MPS**
  Optimize without quantum numbers and project afterwards gains non-trivial energy. On triangular lattice, \(~10\%\)

- **Multi-MPS**
  No sign problem but “bond-dimension” problem.
\[ \alpha|\Psi_{MPS1}\rangle + \beta|\Psi_{MPS2}\rangle + \gamma|\Psi_{MPS3}\rangle \]

How do we choose the MPS

Optimize?

Faster approach to get reasonable states...

**Exact:** \{\(|\Psi_{MPS}\rangle, H|\Psi_{MPS}\rangle, H^2|\Psi_{MPS}\rangle, \ldots \}\}

**Approx:** \{\(|\Psi_{MPS}\rangle, PH|\Psi_{MPS}\rangle, PHPH|\Psi_{MPS}\rangle, \ldots \}\}

Better: Let \( H = h_1 + h_2 + h_3 + h_4 + h_5 \)

\{\(|\Psi_{MPS}\rangle, h_i|\Psi_{MPS}\rangle, h_i h_j|\Psi_{MPS}\rangle, \ldots \}\}

4x8 Hubbard Model:

5 MPO’s of size 6

1 MPO of size 18

For \( n=3 \), factor of 2000x faster!
**Fixed Node:** A (stochastic) sample of the w.f.

**Two recent `improvements`:**

- Fixed node for less-local Hamiltonians
- Fixed node on tensor networks

**Constrained Path:**

Shiwei: Determinants
Garnet: MPS
Approach II: Sample

Sign Problem - Efficiency as $\exp[-\beta \Delta E]$

- PQMC + Annihilation
  - Brings up Delta E
    - Kalos
  - + initiator: Ali Alavi
- AFQMC Free projection
Approach II: Sample

- **PQMC + Annihilation**
  - Brings up Delta E
  - Kalos
  - + initiator: Ali Alavi

- **Partial Node FCIQMC**
  - Importance Sample +
  - Partial Fixed-Node +
  - Annihilation

- **Sample from Tensor Networks + Annihilation**

- **Sign Problem - Efficiency as** \( \exp[-\beta \Delta E] \)

- **AFQMC Free projection**

- **SEMPS**

- **AFQMC release**

- **Sample +**
Approach II: Sample

- **Annihilation + QMC**
  Brings up Delta E
  - Kalos
  - + initiator: Ali Alavi

- **Partial Node FCIQMC**
  Importance Sample + Partial Fixed-Node + Annihilation

- **Sign Problem - Efficiency as** $\exp[-\beta \Delta E]$
  - **AFQMC Free projection**
  - **SEMPs**
  - **AFQMC release**

- **Sample from Tensor Networks + Annihilation**

- **RFCIQMC**
QMC: A Sign Problem

DMRG: A bond dimension problem.

Our goal is to write down an algorithm that has both a sign problem and a bond dimension problem.

The worst of both worlds!
MPS

\[ \sum \]

Product states

Sample

\[ \frac{1}{\langle MPS_1 |C \rangle} \]

weight

Energy:

\[ \sum_i \frac{\langle \Psi_i | H | \Psi_T \rangle w_i}{\sum_i w_i} \]
We’d really like perfect annihilation through all these paths. How can we do this?

We’d really like effectively higher bond dimension. How can we do this?
Venerable history: BCS Superconductivity
Quantum Hall Effect
Model Wave-functions

Particularly valuable if the wave-function is conceptually simple and connects to analytical theory

Replace nobel prize winner with computers
\[ |\Psi\rangle = \exp[-J(R)] \sum \alpha_k \det M_{\uparrow,k} \det M_{\downarrow,k} \]

\[ J(R) = \sum_{ij} j(r_i - r_j) \]

\[ M_{ij} = \phi_i(r_j) \]

3d function \( R \equiv \{ r_1, r_2, \ldots, r_n \} \)

Unfortunately, these wave-functions are not compact enough. For the most interesting physical systems you need an exponential number of parameters (turn up the matrix size, the number of determinants, etc.) to get converged energies.
We’d really like perfect annihilation through all these paths.

\[
|MPS_1\rangle \approx |D_1\rangle + |D_3\rangle + |D_{20}\rangle + \ldots
\]
\[
\exp[-\tau H]|D_1\rangle + \exp[-\tau H]|D_3\rangle + \exp[-\tau H]|D_{20}\rangle + \ldots
\]

Represented ‘exactly’ by MPS of small bond-dimension.

You run out of bond-dimension much slower.

You’re already starting at the best MPS you can get for your bond dimension. You’re guaranteed to be better.

Massively Parallel

You do have a bond-dimension problem.

If \(\text{Sign}(\langle MPS_1|C\rangle) \neq \text{Sign}(\langle \Psi_0|C\rangle)\), you have a weak sign problem.
\( \beta = 0.08 \)

\[ \begin{array}{ccc}
0 & 50 & 100 \\
0.05 & 0.1 & 0.15 \\
0.2 & 0.25 & 0.3 \\
0.35 & 0.4 & 0.45 \\
0.5 & 0.6 & 0.7
\end{array} \]

\( \beta = 4.0 \)

\[ \begin{array}{ccc}
0 & 50 & 100 \\
0.05 & 0.1 & 0.15 \\
0.2 & 0.25 & 0.3 \\
0.35 & 0.4 & 0.45 \\
0.5 & 0.6 & 0.7
\end{array} \]
4 x 32 hubbard model
What to do when you run out of bond dimension?

- Resample here
  - Cone annihilation
  - Exact annihilation

A much smaller sign problem.
The best (or worst) of both worlds: SEMPS

SEMPS: Sample the state, Stochastically evolve

QMC to evaluate DMRG

iTEBD: imaginary time evolve

SEMPS: imaginary time evolve

Release on MPS

SEMPS MC: Stochastically evolve
DRMG+QMC gives us powerful new algorithms including:

- Multi-MPS
- SEMPS
- Fixed-Node w/ MPS
- Release w/ MPS

**Pareto-Optimal:**

- Multi-MPS SEMPS
- Partial Node FCIQMC on Multi-MPS or Multi-Slater Jastrow
- Release of CP AFQMC + SEMPS