Simulating quantum and classical field theories on a quantum computer

Stephen Jordan

With:

John Preskill, Keith Lee, Ali Moosavian
Pedro Costa, Aaron Ostrander
Field Theory

**Classical**
- Value at each point in space
- Classical simulation in polynomial time and polynomial memory
- Represent by amplitudes: quantum simulation in polynomial time and logarithmic memory

**Quantum**
- Qubit(s) at each point in space
- Classical simulation in exponential time (memory can be polynomial)
- Represent by qubits: quantum simulation in polynomial time and polynomial memory
A QFT Computational Problem

**Input:** a list of momenta of incoming particles.

**Output:** a list of momenta of outgoing particles.
Our Results

• Efficient simulation algorithms for example QFTs:
  • Bosonic: Massive $\phi^4$
  • Fermionic: Massive Gross-Neveu

• Recent Developments
  • BQP-hardness: classical computers cannot perform certain QFT simulations efficiently
    [Jordan, Krovi, Lee, Preskill, *Quantum* 2, 44, 2018]
  • Better Speed and broken symmetries
Representing Quantum Fields

A field is a list of values, one for each location in space.

A quantum field is a superposition over classical fields.

A superposition over bit strings is a state of a quantum computer.
Our Algorithms

1) **Choose a lattice discretization.**
   Bound discretization error (renormalization group)

2) **Prepare physically realistic initial state.**
   Is the most time-consuming step.
   This depends strongly on which QFT is simulated.

3) **Implement time-evolution by a quantum circuit.**
   Can use Suzuki-Trotter formulae.

4) **Perform measurements on final state.**
   One must be careful about variance.
Adiabatic State Preparation

\[ H(s) = H_{\text{free}} + sH_{\text{interaction}} \]

Prepare wavepackets in free theory, then adiabatically turn on interaction. **Problem:**

![Diagram showing the evolution of wavepackets from s = 0 to s = 1]
Adiabatic State Preparation

Solution: intersperse backward time evolutions with time-independent Hamiltonians.

This winds back dynamical phase on each eigenstate without undoing adiabatic change of basis.
Runtimes

Weak Coupling:

<table>
<thead>
<tr>
<th>$d$</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(1/\epsilon)^{1.5}$</td>
</tr>
<tr>
<td>2</td>
<td>$(1/\epsilon)^{2.376}$</td>
</tr>
<tr>
<td>3</td>
<td>$(1/\epsilon)^{5.5}$</td>
</tr>
</tbody>
</table>

Strong Coupling:

<table>
<thead>
<tr>
<th>$d$</th>
<th>$\lambda_c - \lambda_0$</th>
<th>$p$</th>
<th>$n_{out}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\left(\frac{1}{\lambda_c - \lambda_0}\right)^9$</td>
<td>$p^4$</td>
<td>$n_{out}^5$</td>
</tr>
<tr>
<td>2</td>
<td>$\left(\frac{1}{\lambda_c - \lambda_0}\right)^6.3$</td>
<td>$p^6$</td>
<td>$n_{out}^{7.128}$</td>
</tr>
</tbody>
</table>
Improved State Preparation

• Two problems with adiabatic state preparation:
  • Cannot reach symmetry-broken phase
  • Runtime bound not practical $O(\epsilon^{-8})$

• A solution for both problems, complexity $O(\epsilon^{-3.23})$:
  • Classically compute a Matrix Product State description of the (interacting) vacuum
  • Compile this MPS directly into a quantum circuit that prepares the state
  • Excite single-particle wavepackets by simulating an oscillatory source term

[Moosavian, Jordan, arXiv:1711.04006]
From MPS to Quantum Circuit

\[ \langle \psi_I | M_1 M_2 M_3 | \psi_F \rangle \]

\[ n \chi^3 \quad \text{(DMRG)} \]
\[ n \chi^3 \quad \text{(SVD)} \]
\[ n \chi^2 \]

[Schon, Hamerer, Wolf, Cirac, Solano, 2006]
Bond Dimension

• It suffices to take $\chi = ke^{S_{1/2}}$ where errors shrink superpolynomially with $k$, resulting in $k \sim \epsilon^{-\sqrt{N/3}}$ [Swingle, arXiv:1304.6402]

• For correlation lengths large compared to lattice spacing, estimates of $S_{1/2}$ are available from conformal field theory:

$$S_{1/2} = \frac{N}{6} \log \left( \frac{1}{ma} \right) \quad ma \ll 1$$

• $\epsilon \sim a$, hence for complexity of preparing interacting vacuum is:

$$\epsilon^{-N/2-1-\sqrt{3N}} \quad \text{as} \quad N \to \infty \quad \epsilon^{-3.23...}$$
Next steps

The program is ongoing!

- Greater generality
- Greater asymptotic efficiency
- Greater practicality
  (see also: analog simulators, classical algorithms)

Let’s simulate the whole Standard Model!
Solving PDEs classically

Vast swaths of engineering are done by solving PDEs using finite element and finite difference methods.

Is this a promising application for quantum computers?
Quantum linear algebra


- Given: oracle access to $s$-sparse $N \times N$ matrix $A$, and ability to make quantum state (proportional to) $\vec{b}$
- Produces quantum state $\epsilon$-close to $\vec{x}$, where $A\vec{x} = \vec{b}$
- Complexity:
  
  $$O(\log N) \quad \text{qubits}$$
  $$\tilde{O}(sk^2/\epsilon) \quad \text{gates}$$

- Generated a lot of buzz. (Google scholar shows 428 citations as of February 2018.)
Apply HHL to finite element


• Consider FEM for electromagnetic scattering problem with separation of variables:

\[ \Phi(\vec{x}, t) = \sin(\omega t) f(\vec{x}) \]

• Propose using Sparse Approximate Inverse Preconditioner to reduce \( \kappa \)

• Good idea! Analysis incomplete.
Preconditioners

• Jacobs, Clader, and Sprouse suggested SPAI may reduce $\kappa$ to $\text{polylog}(N)$, where $N$ is lattice size.

• Pedro Costa and I tried it.

$\kappa \sim \frac{N^2}{s}$

$N =$ number of lattice sites
$s =$ sparsity of preconditioner

Complexity $\sim \kappa s$ breaks even
Invert and truncate vs SPAI
Diffusion is Hard

• Simulating diffusion-like processes could solve lots of other problems.

• **Graph isomorphism:**

• If, for a given graph $G$, you could make:

$$|S_n G\rangle = \frac{1}{\sqrt{n!}} \sum_{\pi \in S_n} |\pi G\rangle$$

then you could solve GI by a Hadamrd test or swap test, because:

$$\langle S_n G | S_n G' \rangle = \begin{cases} 1 & \text{if } G, G' \text{ isomorphic} \\ 0 & \text{otherwise} \end{cases}$$
Also...

It was shown by Aharonov and Regev that producing Gaussian superpositions around lattice points would yield solutions to hard (and cryptographically important) versions of the Shortest Vector and Closest Vector problems (but not the NP-hard versions). These could probably be made by diffusion.
Non-HHL quantum PDE algorithm

Costa, Jordan, Ostrander [arXiv:1711.05394]

• We consider the wave equation:

\[ \frac{\partial^2}{\partial t^2} \Phi = \nabla^2 \Phi \]

• Rather than using HHL we recast it directly into a Hamiltonian simulation problem.

• By doing so, we get quadratically better performance with lattice spacing than is obtained using the algorithms of Berry et al.
Wave scattering

• Conservation laws mapped to unitarity.

• Coarsegrained output: scattering crosssection.

• More general problem than considered by Jacobs, Clader, and Sprouse: full time dependence rather than sinusoidal.
Core Idea

• Wave equation:

\[ \frac{\partial^2 \Phi}{\partial t^2} = \frac{\partial^2 \Phi}{\partial x^2} \]

• Schrödinger’s equation:

\[ \frac{\partial}{\partial t} \begin{bmatrix} \Phi \\ \Psi \end{bmatrix} = -i \begin{bmatrix} 0 & -\frac{\partial}{\partial x} \\ \frac{\partial}{\partial x} & 0 \end{bmatrix} \begin{bmatrix} \Phi \\ \Psi \end{bmatrix} \]

\[ \frac{\partial^2}{\partial t^2} \begin{bmatrix} \Phi \\ \Psi \end{bmatrix} = \begin{bmatrix} \frac{\partial^2}{\partial x^2} & 0 \\ 0 & \frac{\partial^2}{\partial x^2} \end{bmatrix} \begin{bmatrix} \Phi \\ \Psi \end{bmatrix} \]
General case, discretized

Graph Laplacian:

$$\frac{\partial^2}{\partial t^2} \Phi = \nabla^2 \Phi$$

$$\frac{\partial^2}{\partial t^2} \Phi = -\frac{1}{h^2} L \Phi$$

**e.g.**

\[
\begin{bmatrix}
1 & -1 & & & \\
-1 & 2 & -1 & & \\
& -1 & 2 & -1 & \\
& & & \ddots & \\
& & & & -1 & 1
\end{bmatrix}
\]
Incidence matrix

- Rows indexed by edges
- Columns indexed by vertices
- +1 source, -1 sink, 0 otherwise

\[
B = 
\begin{bmatrix}
    a \\ b \\ c \\
\end{bmatrix}
\begin{bmatrix}
    1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\
\end{bmatrix}
\]

\[B^T B = L\]
Quantum algorithm

• Prepare initial state
• Simulate Hamiltonian time evolution by standard techniques

\[
\frac{d}{dt} \begin{bmatrix} \Phi \\ \Psi \end{bmatrix} = -i \frac{1}{\hbar} \begin{bmatrix} 0 & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \Phi \\ \Psi \end{bmatrix}
\]

• Do projective measurement on detector region.
# Performance

<table>
<thead>
<tr>
<th></th>
<th>quantum</th>
<th>classical</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>$T \frac{D^2}{h}$</td>
<td>$T \left(\frac{\ell}{h}\right)^D$</td>
</tr>
<tr>
<td>space</td>
<td>$D \log(\ell/h)$</td>
<td>$\left(\frac{\ell}{h}\right)^D$</td>
</tr>
</tbody>
</table>

$h = \text{lattice spacing} \quad \ell = \text{diameter of region} \\
D = \# \text{dimensions} \quad T = \text{duration of process}$
Higher order Laplacians

- Standard discretized laplacian:

\[
\frac{f(x + h) - 2f(x) + f(x + h)}{h^2} = \frac{df}{dx}(x) + O(h^2)
\]

- Higher order:

\[
-\frac{5}{2} f(x) + \frac{4}{3} f(x + h) + \frac{4}{3} f(x - h) - \frac{1}{12} f(x + 2h) - \frac{1}{12} f(x - 2h)
\]

\[
= \frac{df}{dx}(x) + O(h^4)
\]
Next Steps

• Cast Galerkin method variationally and apply low depth quantum circuits?

• Hadamard test is a key quantum advantage: L1 distance between efficiently samplable distributions is SZK-complete to estimate. So, use HHL together with Hadamard test for stability analysis?

• Quantify resource count (gates, qubits) using Q# and tracer

Thanks!
Exciting Particles

• Simulate dynamics with an oscillatory source term:

\[ H(t) = H_0 + \lambda \cos(\omega t)W \]  

(Rabi Oscillation)

• Ensure resonance with desired state:

\[ \omega = \sqrt{p^2 + m^2} \]

• Ensure W selects desired momentum:

\[ W = \int dx \left( f(x)\psi(x) + f^*\psi^\dagger(x) \right) \]

\[ f(x) \propto e^{ipx-x^2/\sigma^2} \]
Exciting Particles

• How hard to drive the system (choosing $\lambda$)?
• How long to drive the system?

• Strategy:
  • Make 2-level approximation. Derive error bound:
    $$|\langle \psi(t) | \psi_2(t) \rangle| \geq 1 - (2\lambda + 3\lambda^2 t) \frac{1}{\delta}$$
  • Analyze 2-level system with Floquet theory:
    $$|\langle 1 | \psi(\pi/\lambda) \rangle| \geq 1 - O\left(\frac{\lambda^2}{\omega^2}\right)$$
Is $A$ Hermitian?  

- no: 
  \[
  \begin{bmatrix}
  0 & A \\
  A^\dagger & 0
  \end{bmatrix}
  \begin{bmatrix}
  \vec{y}
  \end{bmatrix} = 
  \begin{bmatrix}
  \vec{b} \\
  0
  \end{bmatrix}
  \Rightarrow \begin{bmatrix}
  \vec{y}
  \end{bmatrix} = 
  \begin{bmatrix}
  0 \\
  \vec{x}
  \end{bmatrix}
  \]

- yes: 
  Can $\ket{b}$ be made efficiently?

- no: Sorry, you’re out of luck.

- yes: 
  Make $\ket{x}$.
  Use it wisely.
Apply HHL to finite element and Finite Difference


