Report

Abstract

We simulated the Hamiltonian of the Heisenberg Spin Chain model using topologically optimized Trotterisation on a 7-qubit IBM quantum computer. We then compare the results of this simulation with purely-classical Hamiltonian simulation and classically Trotterised Hamiltonian simulation.

Introduction

Hamiltonian simulation serves as the basis for many problems in science. The universe is of quantum nature, which means that to simulate particles or any other natural entities you need to simulate its Hamiltonian. However, computing the evolution given the Hamiltonian is also not very straightforward.

Due to the discrete nature of classical computers, it is not possible to efficiently simulate any of the above on the same. This gives quantum computers an edge since they can work with qubits to simulate the dynamics of a system given its Hamiltonian. This makes Hamiltonian Simulation the most impacting and significant contribution of quantum computers.

Importance of using quantum computers for simulation

Computational Simulation of the physical systems provides important incites and acts as a bridge between theory and experiments. But can a quantum system be simulated by a classical computer? The answer is certainly, ‘No!’ as said by Bell [2]. Even simulation using pseudo-random variables has exponential computational overhead. So, here comes the Feynman.

> “Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly, it's a wonderful the problem, because it doesn't look so easy.”

- Richard Feynman

Applications of Hamiltonian Simulation

Ground State

Finding the ground state of a system is a computationally hard task. However, if we find a way to simulate the Hamiltonian, then we can do this efficiently.

Biology

In areas like quantum biology and chemistry, it is often important to figure out how a quantum system will evolve from some given initial state. One such important problem is protein folding.

Protein folding involves the physical process by which a protein chain is translated to its native three-dimensional structure, typically a "folded" conformation by which the protein becomes biologically functional. However, finding this folded conformation is an NP-Hard problem.

However, it turns out that finding the native structure of the protein is equivalent to the problem of finding the ground state of the system.

Hence, with Hamiltonian simulation on a quantum computer, we can find the ground state of a protein much faster.

This can vastly accelerate drug discovery and can help us against deadly diseases.

Hamiltonian Simulation

Hamiltonian H describes the physical characteristics of a system and determines its evolution. Given the initial state $|\psi(0)\rangle$, the evolution is governed by the Schrodinger equation.

$$i\hbar \frac{d\,|\psi(t)\rangle}{dt} = H\,|\psi(t)\rangle$$

H itself may change with t, but for simplicity, we will only consider the time-independent case. Then, if we start in some state $|\psi(0)\rangle$, The solution to the Schrodinger equation will give unitary evolution of the state.

$$|\psi(t)\rangle = e^{-iHt}\,|\psi(0)\rangle$$

So, to simulate the evolution classically we need to exponentiate the Hamiltonian. For an n-qubit system, the Hamiltonian will be of dimensions $2^n \times 2^n$ and matrix exponentiation has the complexity of $O(N^3)$ where N is the number of eigenvalues. Hence, for n-qubits, the complexity for hamiltonian simulation will be $O(2^{3n})$ which is exponential and the computation time will be huge even for moderate n.

Quantum Hamiltonian Simulation

There are a few methods that help us take advantage of quantum computers to simulate the Hamiltonian faster. These are described below.

Taylor Series

We can write the unitary as it’s Taylor expansion as

$$e^{-iHt} = \sum_{n=0}^{\infty} \frac{(-iHt)^n}{n!} = I - iHt - \frac{H^2t^2}{2} + \frac{iH^3t^3}{6}...$$

For short periods we can truncate the latter terms.

An efficient quantum algorithm to simulate the Hamiltonian by truncating the latter terms is described in [3].

Quantum Walks

We can implement a unitary operation whose spectrum is related to the Hamiltonian and use phase estimation to adjust the eigenvalues.

Quantum Signal Processing

Trotterisation

This method is the main focus of our project.

In Trotterisation, we decompose our Hamiltonian H into a sum of local Hamiltonians such that

$$H = \sum_{\gamma=1}^{m} H_\gamma$$

$$H = \sum_{\gamma} H_\gamma$$

It is shown in [4] that

$$e^{-itH} = e^{-itH_1} \cdot e^{-itH_2} \cdot .. \cdot e^{-itH_m} + O(t^2)$$

Hence, an approximation can be made for e^{-itH} with an error term of the order of t^2.

Lie-Suzuki-Trotter method

This method describes a quantum algorithm to simulate a Hamiltonian.

Suppose our Hamiltonian is of the form

$$H = \sum_{j=1}^{m} H_j$$

where m is not too big (say, polynomial in n) and H_j are local Hamiltonians, ie, each H_j acts only on a few of the n qubits. For concreteness assume each H_j acts non-trivially on only two of the qubits. Such a Hamiltonian is called 2-local. Note that, for fixed t, the unitary e^{iH_jt} is really just a 2-qubit gate, acting like identity on the other $n{-}2$ qubits; this 2-qubit gate could, in turn, be constructed from CNOTs and single-qubit gates.

Our goal is to implement $U = e^{iHt} = e^{i\Sigma_j H_j t}$. In general, matrix e^{A+B} need not equal $e^A e^B$ if A and B do not commute. The Lie Suzuki-Trotter decomposition gives us a way to handle this. It uses the fact that if A and B have small operator norm, then e^{A+B} and $e^A e^B$ are approximately equal: $e^{A+B} = e^A e^B + E$, where the error-term E is a matrix whose operator norm is $O(||A||\cdot ||B||)$ as described above.

Using the Lie-Trotter formula we can approximate our unitary U as

$$\tilde{U} = (e^{iH_1 t/r} \dots e^{iH_m t/r})^r$$

which is equivalent to applying the trotter approximation for small time-steps of t/r, r times.

We will require $mr = O(m^3 t^2/\epsilon)$ 2-qubit gates, and the error will be $||U - \tilde{U}|| \leq r||E|| \leq \epsilon$.

Why Trotterisation?

Local Hamiltonians are inexpensive

Because we decompose an n-qubit Hamiltonian into a sum of m k-qubit Hamiltonians where $k << m$, we are not exponentiating very huge Hamiltonians anymore. We are exponentiating many small Hamiltonians. This requires much less time compared to exponentiating the huge $2^n \times 2^n$ Hamiltonian.

No Ancillary qubits

Simulating a hamiltonian by trotterisation requires no ancillary qubits. This is very useful, especially in today's time as there is a limit on the number of logical qubits.

Exploiting commutativity

For small t, the trotter error can be represented as

$$e^{-itA}\ e^{-itB} = e^{-it(A+B) - \frac{t^2}{2}[B,A] + i\frac{t^3}{12}[B,[B,A]] - i\frac{t^3}{12}[A,[B,A]] + \dots}$$

This is the Baker-Campbell-Hausdorff.

If $[A, B] = 0$, then the error term in the power will cancel out leaving us with

$$e^{-itA}\ e^{-itB} = e^{-it(A+B)}$$

Heisenberg Chain

Simulation of the Heisenberg chain is a standard benchmark often used in the context of quantum simulation, and we use this example system in our experimentation. It describes a one-dimensional chain of particles and the pairwise interactions of adjacent particles. The Hamiltonian that describes our Heisenberg chain is

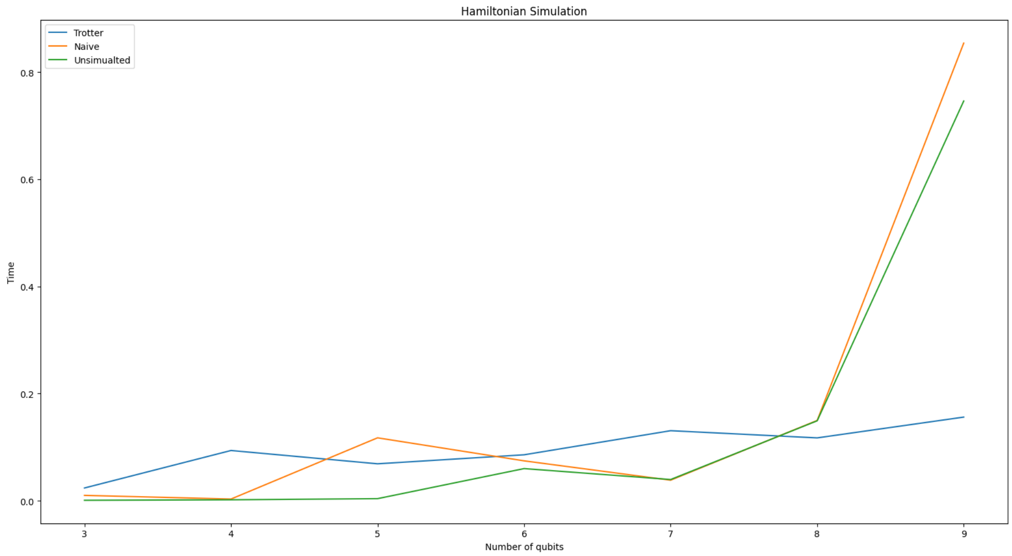
$$H = \sum_{j=1}^{n} [X_j X_{j+1} + Y_j Y_{j+1} + Z_j Z_{j+1} + v_j Z_j]$$

This is the Hamiltonian governing a system containing n qubits. The j^{th} and / qubits interact, where indices are modulo n so that the last qubit interacts with the first.

An instance of this Heisenberg chain is defined by the Hamiltonian, the number of qubits n, and a vector v of the values v_j for each qubit in the system. We allow for disorder in the physical system that makes v a random variable. For our purposes, v_j is a scalar drawn uniformly random within the interval $[-1, 1]$. A simulation is then fully defined by (a) the description of the Heisenberg chain and (b) the period of time t that we wish to simulate the evolution of that chain over. We observe that this Hamiltonian is decomposed into the sum of $4n$ local matrices H_j , enabling us to apply Suzuki decomposition to efficiently simulate the system.

Simulation

Benchmarking



Complexity comparison for exact and approximate solution for growing number of qubits.

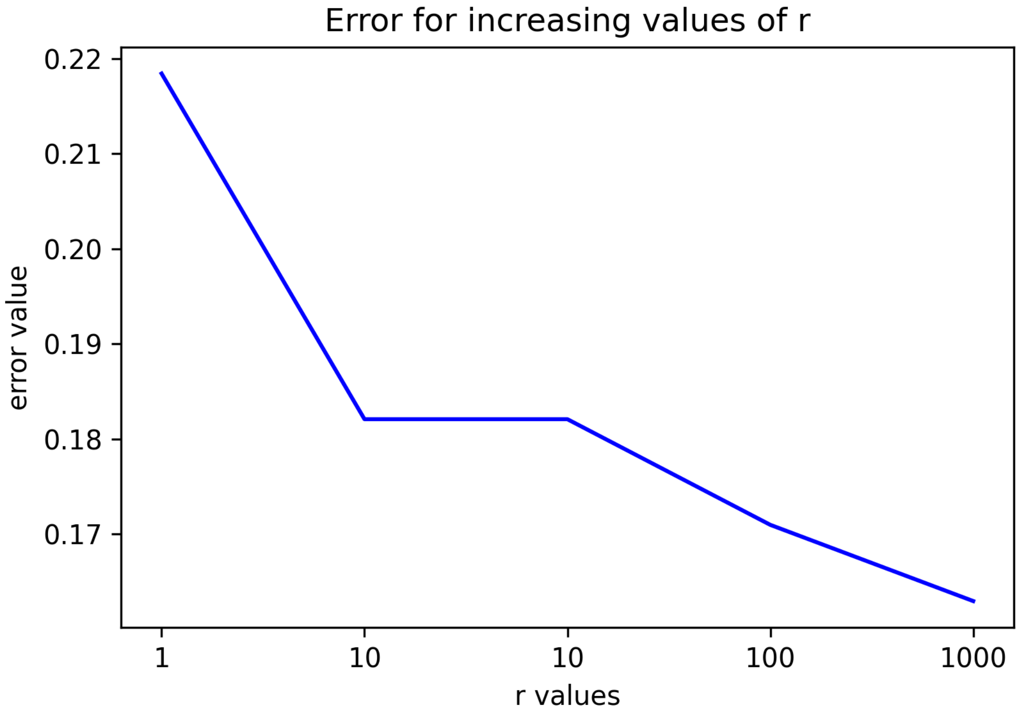
From the above following methods, the focus was primarily on the analysis of the Trotters method and how it improves for larger values of r. The comparison is between the classically implemented Hamiltonian simulation which is exponential in a number of qubits vs the Trotters method of approximation which is dependent on the accuracy that you demand and scales based on r.

To measure how well the algorithm performed, we can check the performance of the same on a classical simulation against its naive counterparts.

IBM's Quantum Simulation

It was required to check how well Trotter's method performs in terms of error on an actual computer with the computed naive method. The error rate clearly goes down even for a noisy quantum computer.

As r increases, the number of gates increases which means that the error rate grows. But due to better approximation of larger r values, the overall error rate becomes lesser even for a noisy quantum computer.



References

[1] Richard P Feynman (1982). "Simulating physics with computers" International Journal of Theoretical Physics

[2] Bell, J. S. (1964). "On the Einstein Podolsky Rosen Paradox". *Physics Physique Физика*. **1** (3): 195–200. doi:10.1103/PhysicsPhysiqueFizika.1.195

[3] Dominic W. Berry et al. (2015) Simulating Hamiltonian dynamics with a truncated Taylor series Phys. Rev. Lett. 114, 090502

[4] Andrew M. Childs et al. (2021)Theory of Trotter Error with Commutator Scaling, Phys. Rev. X 11, 011020