Literature Review

Gabriel Thompson

February 3, 2025

1 Background

In classical computing, when a computer is given a problem and asked to test for many parameters it will run the algorithm a multitude of times until each and every case has ran. Quantum computers can take this same problem with initial conditions, and instead of individually running the algorithm per respective condition, it solves the problem by trialing every condition simultaneously. Using quantum computers therefore majorly decrease time costs in comparison to classical computer run times, apply these quantum algorithm techniques to classical computers to see an increase in efficiency. These quantum-inspired algorithms are very special, as we are not limited to working on quantum computers exclusively – which are a scarce resource – and can port back our algorithms directly, or with minimal fine tuning to quantum computers [1].

1.1 Quantum Bit

In classical computing, computers use bits to store information on a device. Information being stored by these bits are represented in binary as either a zero or one. When working with quantum computers, we use quantum bits – commonly referred to as qubits. Unlike their classical counterparts, qubits are not limited to being stored as a single digit, but rather as linear combination of the two states allowing it to exist in a superposition of both states. This opens up the qubits to a wide range of different stores to be permitted.

1.2 Quantum Register

As with classical bits being able to be stored in a binary series of zeroes and ones, qubits can be stored in a similar manner by encoding probability distributions into a quantum register. A function ψ represented by a quantum register can be written in bra-ket notation,

$$|\psi\rangle = \sum_{k=0}^{2^N-1} a_k|k\rangle = a_0|0\rangle + a_1|1\rangle + \cdots + a_{2^N-1}|2^N-1\rangle \tag{1}$$

with a_k being the probability distribution and N is the number of qubits that are being used in a system.

1.3 Matrix Product State

Another representation of a wave function is called a matrix product state (MPS). MPS is the decomposition of the wave function so that it is represented by a combination of N matrices. The ordering of qubits in a quantum register can be done in two ways, order (A) which is by coordinate followed by significance, or order (B) which is qubit significance, then by coordinate. Two random variables, x_0 and x_1, with three qubits per variable would be represented in the form,

$$\begin{aligned}\text{Order (A)}: |a_0a_1\rangle &\rightarrow |a_0^1\rangle\,|a_0^2\rangle\,|a_0^3\rangle\,|a_1^1\rangle\,|a_1^2\rangle\,|a_1^3\rangle\,, \\ \text{Order (B)}: |a_0a_1\rangle &\rightarrow |a_0^1\rangle\,|a_1^1\rangle\,|a_0^2\rangle\,|a_1^2\rangle\,|a_0^3\rangle\,|a_1^3\rangle\end{aligned} \tag{2}$$

Each $|a_i^j\rangle$ refers to a qubit, with i being coordinate and j the significance. Significance of a qubit refers to the impact that a qubit value will have on its quantum system. Order (A) arranges the qubits first by coordinate, then by significance, while order (B) arranges the qubits by significance before coordinates. Using $N = 28$ qubits, the size of the MPS representation of order (A) was 1Mb which comes at a significantly higher cost in comparison to order (B), which came to 71Kb of data [1]. This reduction in MPS size consequently allows for an increase in the amount of entanglement allowed in the discretization. When using MPS states on classical computers, they are very easy to exploit and can be done very efficiently. This ease of use on classical devices allows for the inclusion of new quantum-inspired algorithms that are of interest, introduced in the following section.

2 Algorithms

The purpose of this project is to explore different approaches of setting up quantum-inspired algorithms to solve partial differential equations (PDEs). We will compare the spectral split-step[1] and linear combination of Hamiltonian simulation (LCHS) methods[4] to the classical algorithms like finite differences method which is a standard technique to solving PDEs. The two quantum-inspired algorithms both use the encoding of an MPS of a quantum register.

2.1 Finite Differences

Finite differences is one of the most commonly used classical techniques for solving PDEs. It is performed by taking approximations of derivatives and and infinitesimal change. For a general function $f(x)$ and a infinitesimally small displacement δx, the finite differences technique can be represented as,

$$f'(x) \simeq \frac{f(x+\delta x) - f(x)}{\delta x} \tag{3}$$

This technique can further be applied to a quantum register to represent the finite differences operator using qubit operators. Let us consider we have n qubits, with the scalar field $\boldsymbol{f}$ in some domain $[0, L]$. We can discretize the field using its respective value at each point as $\boldsymbol{f} = [f_0, f_1, ..., f_{2^n-1}]$[3]. We can now map the scalar field onto the system state which gives the form

$$|f\rangle = \frac{1}{\|\mathbf{f}\|} \sum_{j=0}^{2^n-1} f(x^{[j]})|j\rangle, \tag{4}$$

where $|j\rangle = |j_{n-1}, j_{n-2}, ..., j_0\rangle$, with each element of $j_{n-1}, j_{n-2}, ..., j_0 \in \{0, 1\}$, is the computational basis [4]. We can now apply this to the shift operators $S^- = \sum_{j=1}^{2^n-1} |j-1\rangle\langle j|$ and

$S^{+} = \sum_{j=1}^{2^n-1} |j\rangle\langle j-1|$ and then apply them to the state [4]

$$S^{-} := \sum_{j=1}^{n} I^{\otimes(n-j)} \otimes \sigma_{01} \otimes \sigma_{10}^{\otimes(j-1)}, \tag{5}$$

$$S^{+} := \sum_{j=1}^{n} I^{\otimes(n-j)} \otimes \sigma_{10} \otimes \sigma_{01}^{\otimes(j-1)}, \tag{6}$$

with $\sigma_{01} = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ and $\sigma_{10} = \left(\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right)$ being the basis matrices and I is the identity matrix[4]. With using the shift operators, we can derive multiple representations of the finite differences operator represented by qubit operators,

$$\frac{1}{\delta x}\left(S^{-} - I^{\otimes n}\right)\sum_{j=0}^{2^n-1} f_j\,|j\rangle = \sum_{j=0}^{2^n-1} \frac{f_{j+1} - f_j}{\delta x}\,|j\rangle, \tag{7}$$

this is the exact representation of the forward difference operator [3].

2.2 Spectral Split-Step

Spectral split-step method is used to solve nonlinear PDEs. A general application is the Fokker-Planck equation,

$$\frac{\partial}{\partial t}p(x,t) = -\frac{\partial}{\partial x}[\mu(x)p(x,t)] + \frac{\partial^2}{\partial x^2}[D(x)p(x,t)] \tag{8}$$

where $\mu(x)$ and $D(x)$ are the drift and diffusion constants respectively. This equation has already been solved in a previous paper and will be used as a marker for our progress [4]. Spectral split-step works by first splitting the problem up into small, real steps and then applying a Quantum Fourier Transform (QFT) to move back and fourth through the Fourier space. Compared to the finite differences method discussed in 2.1, this algorithm allows for a greater number of qubits to be used due to the increased efficiency and the numerical stability.

2.2.1 Suzuki-Trotter

Suzuki-Trotter is a method used to expand the bounds of our Hamiltonian-like PDE when it is non-exact. This is done by breaking the time evolving Hamiltonian ($\mathcal{H}$) into kinetic and potential parts to conserve the phase-space.

$$e^{t\mathcal{H}} \simeq \left(e^{\frac{t}{n}\mathcal{K}}e^{\frac{t}{n}\mathcal{V}}\right)^n \tag{9}$$

The kinetic-like operator $e^{\frac{t}{n}\mathcal{K}}$ describes the time evolution over time slice t/n of the Hamiltonian-like with some kinetic-like energy K, and the operator $e^{\frac{t}{n}\mathcal{V}}$ being the portion which conserves the phase-space-like volume[2]. When we apply Suzuki-Trotter decomposition computationally, we are breaking down the Hamiltonian-like PDF and using matrix approximation to calculate each part individually.

2.3 Linear Combination of Hamiltonian Simulation

A special case of linear combination of unitaries, called linear combination of Hamiltonian simulation (LCHS), was used to solve linear ODEs with guaranteeing an optimal preparation state cost [4]. The LCHS method strongly utilizes a powerful quantum operator called oracles. Although oracles, when explored deeply can get quite complex by definition, for the sake of simplicity, we will be treating oracles as a "black box" and using them as a tool [6]. The oracles required for the LCHS method are the state preparation oracle O_{prep}, coefficient oracle for the linear combination of unitaries O_{coef}, the select oracle $SEL_L(s)$ and the Hamiltonian simulation oracle $O_H(s)$ [4].

3 Drawback of Quantum-Inspired Algorithms

While decreasing the runtime of computations in greatly desired, we also need the answer to be accurate. With some quantum-inspired algorithms (QIA), although there is a major decrease in the runtime of the algorithm – there can also be a large increase in the error of the result. The error that appears when communicating two qubits can be reduced by adding ancilla qubits. If the qubits $a|0\rangle + b|1\rangle$ were trying to be sent, additional qubits in state $|0\rangle$ can be placed onto the original information, giving $a|000\rangle + b|100\rangle$. This is a simple 'Bob and Alice' example of only using three qubits, but Bob doesn't know if they have the exact single qubit state that Alice has. However, if error correction was not applied, the probability (p) of failure goes from $O(p^2)$ to $O(p)$ [5]. We could expand the single qubit state to have magnitudes more than just three qubits but not without consequence. As more qubits are added, error in communication of information greatly decreases but the addition of these additional qubits causes an increase in the time cost.

4 Application To Study

Using relevant information from these documents, we can analyze potential differences that appear when comparing the techniques of 2.2 and 2.3 to the classical adaptation of 2.1. Using what has already been extrapolated other's research, we want to find what other possible changes arise aside from efficiency, efficacy and time cost, and what could be the cause of those changes. We can also look at PDEs that have already been solved; the Fokker-Planck equation, the acoustic equation or the non-conservative heat equation, examining the potential variance in time cost.

References

[1] Juan José García-Ripoll. "Quantum-inspired algorithms for multivariate analysis: from interpolation to partial differential equations". In: *Quantum* 5 (Apr. 2021), p. 431. DOI: 10.22331/q-2021-04-15-431. arXiv: 1909.06619 [quant-ph].

[2] Naomichi Hatano and Masuo Suzuki. "Finding Exponential Product Formulas of Higher Orders". In: *Quantum Annealing and Other Optimization Methods*. Springer Berlin Heidelberg, Nov. 2005, pp. 37–68. ISBN: 9783540315155. DOI: 10.1007/11526216_2. URL: http://dx.doi.org/10.1007/11526216_2.

[3] Yuki Sato et al. "Hamiltonian simulation for hyperbolic partial differential equations by scalable quantum circuits". In: *Physical Review Research* 6.3 (Sept. 2024). ISSN: 2643-1564. DOI: 10.

1103/physrevresearch.6.033246. URL: http://dx.doi.org/10.1103/PhysRevResearch.6.033246.

[4] Yuki Sato et al. “Quantum algorithm for partial differential equations of non-conservative systems with spatially varying parameters”. In: *arXiv e-prints*, arXiv:2407.05019 (July 2024), arXiv:2407.05019. DOI: 10.48550/arXiv.2407.05019. arXiv: 2407.05019 [quant-ph].

[5] Andrew Steane. “A Tutorial on Quantum Error Correction A Tutorial on Quantum Error Correction 2”. In: *Nuovo Cimento B* 162 (Jan. 2006). DOI: 10.3254/1-58603-660-2-1.

[6] Jiahao Zhao. “Possible Implementations of Oracles in Quantum Algorithms”. In: *Journal of Physics: Conference Series* 2386.1 (Dec. 2022), p. 012010. DOI: 10.1088/1742-6596/2386/1/012010. URL: https://dx.doi.org/10.1088/1742-6596/2386/1/012010.