Hybrid Quantum-Classical Algorithms for Large Sparse Eigenvalue Problems

Nihar Shah, Phani Motamarri

September 26, 2024

Abstract

With the advent of the Noisy Intermediate-Scale Quantum (NISQ) computing era, quantum computers with 50-100 qubits are expected to surpass the capabilities of classical digital computers. This dissertation proposes a quantum version of Cheybhsehev filtered subspace iteration procedure, a popular iterative orthogonal projection method for solving large sparse eigenvalue poblems typically arising from the discretization of partial differential equations. In particular, the proposed algorithmic implementation leverages quantum computer for subspace construction and Rayleigh-Ritz projection step while the subspace diagonalization is performed on a classical computer.

1 Introduction

We are currently in the Noisy Intermediate-Scale Quantum (NISQ) era of quantum computing, as defined by Preskill in 2018 [Pre18]. In this era, quantum devices, such as IBM's 127-qubit quantum computer, are beginning to surpass the computational power of classical systems. Quantum computing is founded on the principles of quantum mechanics, specifically in a discrete setting, and involves several key concepts fundamental to its operation. Below, we outline these principles:

1. **Superposition:** A classical bit, or simply a bit, is the fundamental unit of information in classical computing. Analogously, the quantum bit, or qubit, serves as the fundamental unit in quantum computing. Classical bits can be realized in various physical forms, such as a switch (on/off) or a voltage level (0 or 5V). Likewise, qubits can be physically realized using different quantum systems, such as trapped ions or superconducting circuits. Fundamentally, a qubit represents a quantum system that could be an electron, an atom, a molecule, or even a group of molecules. Unlike a classical bit, which exists in a state of either 0 or 1, a qubit can exist in a superposition of both states simultaneously. In other words, a qubit can be in state 0 with probability p and in state 1 with probability $1 - p$. Mathematically, a qubit is represented as a vector with complex entries:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

where $\alpha, \beta \in \mathbb{C}$ are complex numbers. While qubits represent two-level quantum systems, higher-dimensional quantum systems, such as qutrits (three-level systems) or qudits (d-level systems), are also possible. For a general d-level quantum system (qudit), the state can be expressed as:

$$|\psi\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle$$

where $\alpha_i \in \mathbb{C}$ and $N = 2^d$ is the total number of classical basis states. In this paper, we will primarily focus on qubits. Thus, a qubit can be considered as a vector in a two-dimensional complex vector space (Hilbert space) with a defined inner product. The notation for multi-qubit states is often written as $|\psi_1\psi_2\ldots\psi_n\rangle$, or equivalently $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$, where the tensor product $\otimes$ denotes the combination of qubit states.

2. **Composite State Space Postulate:** The state space of a multi-qubit system is the tensor product of the state spaces of individual qubits. For example, the state space of $|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_n\rangle$ is given by:

$$|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$$

   This results in a $\mathbb{C}^{\otimes n} = \mathbb{C}^{2^n}$-dimensional complex vector space.

3. **Entanglement:** Unlike classical computing, where bits are independent of each other, quantum computing introduces the concept of entanglement. Entanglement allows qubits to be correlated in such a way that the state of one qubit can instantaneously affect the state of another, regardless of the distance between them. This phenomenon is crucial to the power of quantum computing. Entangled states, known as Bell states, can be represented as:

$$\beta_{xy} = \frac{|0y\rangle + (-1)^x |1\overline{y}\rangle}{\sqrt{2}}$$

   where $x, y \in \{0, 1\}$. Entanglement is a key feature of quantum algorithms and protocols, enabling faster and more efficient solutions to certain problems.

4. **Evolution:** The evolution of a quantum state is governed by the Schrödinger equation:

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = \hat{H} |\psi\rangle$$

   where $\hbar$ is the reduced Planck constant, and $\hat{H}$ is the Hamiltonian operator. For a time-independent Hamiltonian, the solution is given by:

$$|\psi(t)\rangle = e^{-i\frac{\hat{H}}{\hbar}t} |\psi(0)\rangle$$

   where $e^{-i\frac{\hat{H}}{\hbar}t}$ is a unitary operator. Thus, the evolution of a closed quantum system is described by unitary transformations. In a general case, the evolution from time t_1 to t_2 is:

$$|\psi(t_2)\rangle = U(t_1, t_2) |\psi(t_1)\rangle$$

   where the unitary operator $U(t_1, t_2) = e^{-i\frac{\hat{H}}{\hbar}(t_2 - t_1)}$.

5. **Measurement:** Measurement is a fundamental process in quantum mechanics. When we measure a quantum state, it collapses into one of the basis states of the observable we are measuring. In the computational basis, the probability of obtaining a particular classical outcome is given by:

$$p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle$$

   where M_m is a measurement operator associated with outcome m. After measurement, the state collapses to the post-measurement state:

$$\frac{M_m |\psi\rangle}{\sqrt{p(m)}}$$

   For instance, for a qubit in state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, the probability of measuring the classical state 0 is:

$$p(0) = |\alpha|^2$$

   with post-measurement state $|0\rangle$, and the probability of measuring state 1 is:

$$p(1) = |\beta|^2$$

   with post-measurement state $|1\rangle$.

2 Famous Applications

Several key algorithms demonstrate the potential of quantum computing to solve problems more efficiently than classical algorithms. The following subsections highlight some of the most famous quantum algorithms and their applications:

2.1 Grover's Algorithm

Grover's algorithm[Gro96] addresses the problem of unstructured search, where the goal is to find a specific element in an unsorted array. In classical computing, the best deterministic algorithm for this problem is linear search, which has a time complexity of $\mathcal{O}(N)$. Even a probabilistic classical algorithm does not offer substantial improvement, achieving a lower bound of $\Omega(N)$ time complexity.

However, Grover's quantum algorithm offers a quadratic speedup, with a time complexity of $\mathcal{O}(\sqrt{N})$. While this might appear to be a modest improvement, it is, in fact, the best possible performance for this problem. It has been mathematically proven that the lower bound for solving unstructured search problems is $\mathcal{O}(\sqrt{N})$, and Grover's algorithm achieves this bound optimally. Search algorithms are ubiquitous in various fields of computer science, playing a crucial role in tasks ranging from finding nodes in graphs to querying databases in management systems. Grover's algorithm offers a powerful tool to address such problems, providing a quadratic speedup over classical search methods. This speedup can be leveraged in any context where unstructured search problems arise.

For instance, in database management systems, where one might need to search for a specific record among a vast number of entries, Grover's algorithm can significantly reduce the time required to locate the desired item. Similarly, in graph theory, searching for a particular node or path in a large, unstructured network can benefit from Grover's quantum speedup, allowing for more efficient exploration of large datasets.

2.2 Shor's Algorithm

RSA encryption, which is widely used in modern cryptography, relies on the difficulty of factoring large composite numbers into their prime factors. Classically, this problem is considered computationally hard. The best-known classical algorithm for this task is the general number field sieve, which has a time complexity of

$$\mathcal{O}\left(e^{1.9(\log N)^{1/3}(\log\log N)^{2/3}}\right).$$

Shor's quantum algorithm[Sho97] revolutionizes this field by solving the factoring problem with exponentially faster time complexity,

$$\mathcal{O}\left((\log N)^2(\log\log N)\right).$$

This dramatic speedup poses a significant threat to classical cryptographic systems. If sufficiently large quantum computers become practical, Shor's algorithm could efficiently factor large numbers, potentially breaking the security of RSA encryption and other cryptographic systems based on the difficulty of factoring.

2.3 HHL Algorithm

The HHL algorithm[HHL08] (named after its inventors Harrow, Hassidim, and Lloyd) is a quantum algorithm for solving linear systems of equations, a fundamental problem in many fields of science and engineering. Classical direct methods, such as Gaussian Elimination, LU factorization, and QR factorization, have a time complexity of $\mathcal{O}(N^3)$, where N is the number of variables. Iterative methods, such as the steepest descent and conjugate gradient methods, achieve improved time complexities of $\mathcal{O}(sN\kappa\log\epsilon^{-1})$ and $\mathcal{O}(sN\sqrt{\kappa}\log\epsilon^{-1})$, respectively, where s is the matrix sparsity, κ is the condition number, and ϵ is the error tolerance.

The quantum HHL algorithm solves the same problem with time complexity

$$\mathcal{O}(\mathrm{poly}(n)\kappa^2/\epsilon),$$

where $N = 2^n$. Further improvements using techniques such as variable-time amplitude amplification reduce the complexity to

$$\mathcal{O}(\kappa s n\log(\kappa/\epsilon)).$$

This represents a significant quantum advantage, particularly for large sparse systems with a well-conditioned matrix.

However, a notable limitation of the HHL algorithm is that, if we aim to recover the full solution vector x, the algorithm requires an exponential number of quantum runs, limiting its practical applications for direct solution retrieval. Consequently, the HHL algorithm is often used as a subroutine in larger quantum algorithms, particularly in cases where we are interested in specific properties of the solution or matrix operators rather than the full solution itself.

3 Classical vs Quantum Computing

The aforementioned algorithms clearly demonstrate quantum supremacy over several important classical algorithms. However, it remains a challenge to identify classes of problems that cannot be efficiently solved on classical computers but exhibit a quantum advantage.

For instance, simulating quantum circuits on classical computers requires $\mathcal{O}(sn^2 2^n)$ time complexity, where s represents the number of gates and n denotes the number of qubits. This complexity is exponential in both space and time. While some methods may reduce the space complexity to polynomial time, the time complexity remains exponential. Conversely, simulating classical circuits on a quantum computer is asymptotically efficient, capable of being executed in polynomial time.

In a manner analogous to the NAND (NOT + AND) and NOR gates forming a universal gate set in classical computing, there exists a corresponding universal gate set in quantum computing. This set can be composed of gates such as CNOT + Hadamard + T or CCNOT (Toffoli) + Hadamard. The Solovay-Kitaev theorem establishes that all choices of universal gates are equivalent; thus, no particular choice of universal gate set is inherently preferable. In other words, any other gate can be simulated to arbitrary precision in $\mathcal{O}(m \cdot \text{polylog}(1/\epsilon))$, where ϵ denotes the induced norm error.

One might wonder about the probabilistic nature of computations on quantum computers and the potential need for numerous runs to achieve a reliable outcome. However, it can be shown that for the outcome of a computation, even if the probability of obtaining a correct result is $p = \frac{1}{2} + \epsilon$, where $\epsilon > 0$, we can apply the Chernoff bound to demonstrate that

$$p\left(\sum_i X_i \leq \frac{n}{2}\right) \leq e^{-2n\epsilon^2}$$

In this expression, X_i represents a binary outcome variable from the experiment, defined as $X_i = 1$ for obtaining a correct outcome and $X_i = 0$ for obtaining an incorrect one. This relationship underscores the potential for achieving reliable results in quantum computations with fewer runs than might initially be anticipated.

3.1 The Challenges: All is not well!!

Suppose we wish to implement a series of unitary operations $U_i \in \mathbb{C}^{N \times N}$. We can approximate these unitaries as $\tilde{U_i} \in \mathbb{C}^{N \times N}$ such that

$$\|U_i - \tilde{U_i}\| \leq \epsilon \quad \text{for all } i = 1, \ldots, k.$$

Then, the combination of these operations will satisfy

$$\left\|\prod_{i=1}^{k} U_i - \prod_{i=1}^{k} \tilde{U_i}\right\| \leq k\epsilon.$$

This principle is known as Duhamel's principle, which indicates that the error grows linearly with the number of operations. Consequently, we aim to limit the number of gates in our quantum circuits.

Current quantum computers are not noise-free; the noise arises from the interaction of the qubits with their environment. This era of quantum computing is referred to as NISQ (Noisy Intermediate-Scale Quantum), characterized by quantum computers with 50-100 qubits that, despite being noisy, can surpass the capabilities of classical computers. Algorithms designed for NISQ devices typically make the following fault-tolerant assumptions:

- Errors may be introduced due to approximation errors at the mathematical level.

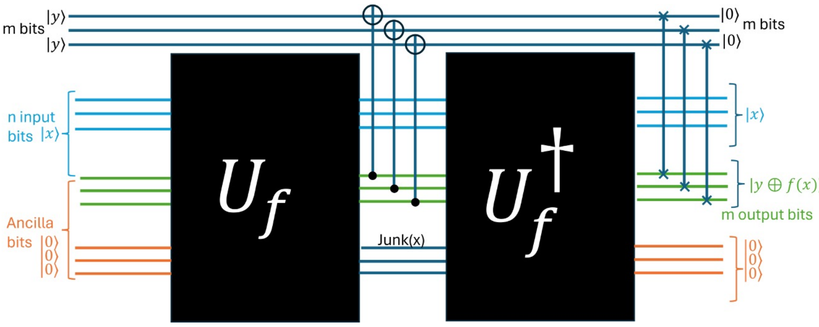


Figure 1: Reversible Computation

- Errors may arise from Monte Carlo errors, which stem from the probabilistic measurements inherent in quantum computing.

A quantum computer can only implement unitary operations, which necessitates that the input dimension must equal the output dimension. This condition is not generally true for classical computations, where the output dimension may differ from the input dimension. For example, consider the AND operation, where the input consists of two bits, and the output is just one bit. Such operations cannot be directly implemented on a quantum computer without modification. In other words, one can conceptualize operations on a quantum computer as being reversible in nature, expressed as:

$$U |x\rangle = |y\rangle \implies |x\rangle = U^{-1} |y\rangle = U^\dagger |y\rangle$$

since $U^{-1} = U^\dagger$ for unitary operations. Therefore, given an input, we can obtain the output, and similarly, given an output, we can derive the unique input on a quantum computer. This requirement imposes additional constraints on quantum computing; for operations to be implemented, they must be reversible, necessitating the use of additional qubits, known as ancilla bits, to facilitate the reversible implementation of non-reversible classical operations as shown in the figure 1.

Without delving into the specifics of the implementation, it is evident that to execute any classical function on a quantum computer requires additional qubits to ensure the computation is reversible, i.e., unitary.

One significant outcome of quantum computing is the **No-Cloning Theorem**, which states that no unitary operation exists that can copy an unknown quantum state. Consequently, we cannot create multiple copies of an unknown quantum state.

Due to the measurement axiom of quantum computing, it can be observed that non-orthogonal quantum states cannot be perfectly distinguished. Thus, no measurement operations can be executed on a quantum computer that perfectly distinguish between two non-orthogonal quantum states.

In a quantum algorithm that implements a classical function, we always assume that the blackbox/oracle conceals the implementation details of the corresponding unitary function U_f. This complexity is referred to as query complexity, which denotes the number of gates utilized in implementing the query. The gates employed to solve the algorithm are termed algorithmic complexity. It is desirable for the classical function f to be efficiently implemented on a quantum computer as U_f (not all classical functions can be efficiently implemented). For the quantum algorithm to confer an advantage, we require that the algorithmic complexity is less than the query complexity. The gate complexity is defined as the sum of algorithmic complexity and query complexity. We aim for the gate complexity to be dominated by the query complexity for the quantum algorithm to be advantageous.

Another requirement for quantum algorithms is depth efficiency. The circuit depth is defined as the maximum number of gates on any path from input to output, which is equivalent to the classical clock time. Quantum states can only be preserved for a limited duration due to noise, referred to as coherence time. It is desired that the circuit depth is less than the coherence time, even if this necessitates running the circuit multiple times.

To utilize fewer qubits, it is beneficial to perform uncomputation to free up ancilla registers (auxiliary registers for implementing U_f) whenever possible. If ancilla registers can be freed through computation, they are termed working registers, as they can subsequently be used for computations in later stages of the algorithms. For further, please refer to the Lin Lin notes [Lin22].

4 Eigenvalue Problems

Let $A \in \mathbb{C}^{m \times m}$ be a square matrix. A nonzero vector $x \in \mathbb{C}^m$ is an eigenvector of A, and $\lambda \in \mathbb{C}$ is the corresponding eigenvalue[TB22] if

$$Ax = \lambda x.$$

Theorem: (Any eigen solver must be iterative) For any $m \geq 5$, there exists a polynomial $p(z)$ of degree m with rational coefficients that has a real root $p(r) = 0$ with the property that r cannot be expressed using any combination of rational numbers, addition, subtraction, multiplication, division, and k-th roots.

Since any eigenvalue problem can be expressed as solving for the roots of the characteristic polynomial, and conversely, any polynomial can be framed as an eigenvalue problem, having a finite step method for solving the eigenvalue problem implies that the root of the characteristic polynomial can be found in a finite number of steps using basic fundamental operations, which would violate the theorem. Hence, any eigenvalue solver must be iterative.

4.1 The Classical Methods

Generally, to solve the eigenvalue problem, we approach it in two phases. Phase 1 involves using a method such as Householder reflections to convert the given matrix into upper Hessenberg form via orthogonal transformations. Phase 2 consists of the iterative step, where most algorithms differ in their implementation.

Using the theorem, if $\|x - q_J\| = \epsilon$, where x is the guessed eigenvector and q_J is the corresponding eigenvector, then

$$\|r(x) - \lambda_J\| = \mathcal{O}(\epsilon^2)$$

where $r(x) = \frac{x^T A x}{x^T x}$ is the Rayleigh quotient and λ_J is the corresponding eigenvalue. In words, if the error between the guessed eigenvector and the actual eigenvector is ϵ, then the error between the Rayleigh quotient (the guessed eigenvalue) and the corresponding eigenvalue is $\mathcal{O}(\epsilon^2)$. The analysis of complexities and convergences of different methods is summarized in the table shown in slides.

4.2 Quantum Methods

The various methods for solving eigenvalue problems on a quantum computer [MKL+24] are listed as follows:

- Quantum Phase Estimation
- Variational Quantum Eigensolver
- Imaginary and Real Time Evolution Methods
- Quantum Subspace Methods (hybrid quantum-classical methods) - Hamiltonian and Quantum Krylov subspace methods
- Time Evolution Methods - Quantum Filter Diagonalization, Quantum Lanczos based on imaginary-time evolution
- Subspace-based Variational Quantum Simulation.

We will discuss Quantum Phase Estimation, Variational Eigensolvers, and provide a general overview of Quantum Subspace methods. Then, I shall propose the Quantum Chebyshev Filtered Subspace iteration method.

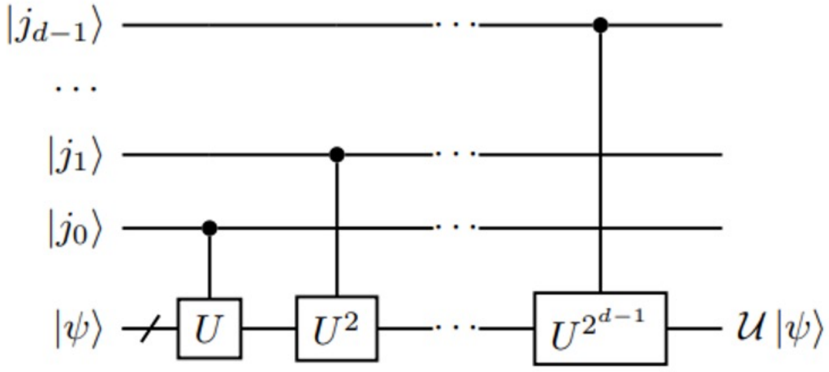


Figure 2: Controlled - U operation

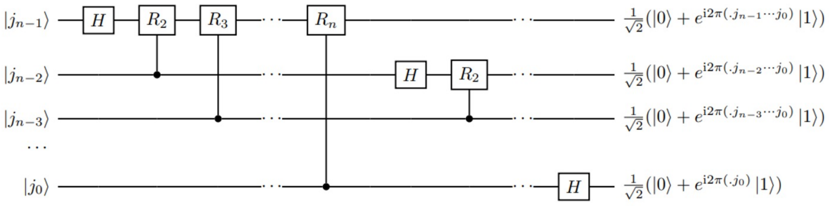

$|j_{n-1}\rangle$
$|j_{n-2}\rangle$
$|j_{n-3}\rangle$
$\cdots$
$|j_0\rangle$
H
R_2
R_3
R_n
H
R_2
H
$\frac{1}{\sqrt{2}}(|0\rangle + e^{i2\pi(.j_{n-1}\cdots j_0)}\,|1\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle + e^{i2\pi(.j_{n-2}\cdots j_0)}\,|1\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle + e^{i2\pi(.j_{n-3}\cdots j_0)}\,|1\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle + e^{i2\pi(.j_0)}\,|1\rangle)$

Figure 3: Quantum Fourier Transform without Swap

4.2.1 Quantum Phase Estimation

One of the first popular method for solving an eigenvalue problem on a quantum computer is the Quantum Phase Estimation. The problem definition is as follows:

Given a quantum circuit that performs the Unitary operation U, we are required to find it's eigenvalue (θ) and its corresponding eigenvector $|\psi\rangle$. Then, the eigenvalue problem is defined as

$$U\,|\psi\rangle = e^{2\pi\iota\theta}\,|\psi\rangle \qquad \theta \in [0,1]$$

This is one of the methods which can be used to solve small scale eigenvalue problems. It involves two steps

1. **Step 1: Implementation of Controlled U operation**, which can be done using the circuit as shown in figure. It has d-ancilla bits which control the power of U to act upon the state $|\psi\rangle$. Thus, has a gate complexity of $mathcalO(d)$ provided that we have an efficient implementation of all the powers of U as shown in figure 2. If not then the gate complexity will be of $\mathcal{O}(d^2)$.

2. **Step 2: Quantum Fourier Transform** Since Quantum Fourier transform is a Unitary circuit it can be easily implemented on a quantum computer using the circuit as shown in the figure 3 where R_Z is the controlled Z rotation $R_j = R_Z(\pi/2^{j-1})$. It has $\mathcal{O}(n^2)$ gate complexity where n is the number of qubits.

3. **Step3: Entire Circuit** The final circuit is as shown in the figure 4 where we first apply a Hadamard transform thus creating an equal superposition of states and then performing a controlled U operation and then an inverse Quantum Fourier transform. Implementation of inverse Quantum Fourier transform is similar to that of the Quantum Fourier transform but with the gates conjugated and reverse in order. We start with a random $|\psi\rangle$. But recall that we can think of $|\psi\rangle$ as a superposition of the eigenstates as $|\psi\rangle = \sum_{i=0}^{2^n-1} \lambda_i\,|\lambda_i\rangle$ since

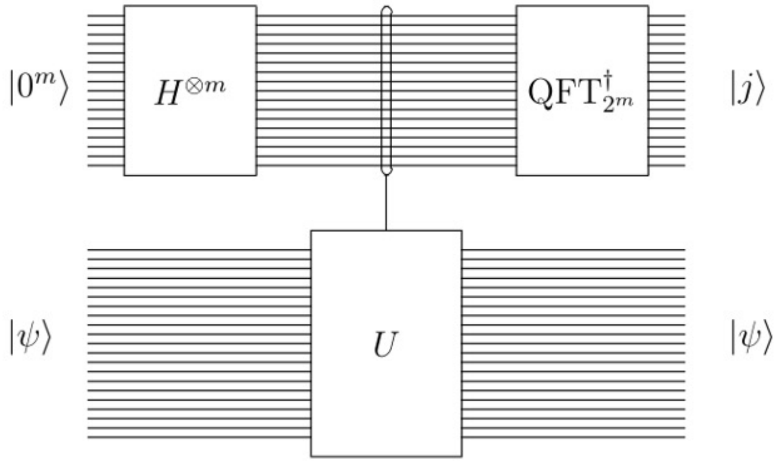


Figure 4: Quantum Phase Estimation Circuit

the matrix given is Unitary. The evolution of states is shown as follows:

$$|0^m\rangle\,|\psi\rangle \xrightarrow{H^{\otimes m}\otimes I} \frac{1}{\sqrt{2^m}}\sum_{i=0}^{2^n-1}\sum_{i=0}^{2^m-1}\alpha_i\,|k\rangle\,|\lambda_i\rangle$$

$$\frac{1}{\sqrt{2^m}}\sum_{i=0}^{2^n-1}\sum_{k=0}^{2^m-1}\alpha_i\,|k\rangle\,|\lambda_i\rangle \xrightarrow{C-U} \frac{1}{\sqrt{2^m}}\sum_{i=0}^{2^n-1}\sum_{k=0}^{2^m-1}\alpha_i e^{2\pi\iota k\lambda_i}\,|k\rangle\,|\lambda_i\rangle$$

$$\frac{1}{\sqrt{2^m}}\sum_{i=0}^{2^n-1}\sum_{k=0}^{2^m-1}\alpha_i e^{2\pi\iota k\lambda_i}\,|k\rangle\,|\lambda_i\rangle \xrightarrow{QFT^{\dagger}\otimes I} \frac{1}{2^m}\sum_{j=0}^{2^m-1}\left(\sum_{i=0}^{2^n-1}\alpha_i\sum_{k=0}^{2^m-1}e^{2\pi\iota\left(k\lambda_i-\frac{jk}{2^m}\right)}\,|j\rangle\right)|\lambda_i\rangle$$

Thus, upon measurement in the classical basis. The probability of getting the jth basis state

$$p_j = \left|\sum_{i=0}^{2^n-1}\alpha_i\sum_{k=0}^{2^m-1}e^{2\pi\iota\left(k\lambda_i-\frac{jk}{2^m}\right)}\,|j\rangle\,|\lambda_i\rangle\right|^2$$

depends on the initial overlap of $|\psi\rangle$ with the eigenstate i.e. $\alpha_i = \langle\lambda_i|\psi\rangle$, approximation of the eigenvalue θ we desire (m bits) and the distribution of the eigenvalues over the unit circle in the complex plane.

4.2.2 Challenges with QPE

- To obtain the phase θ to accuracy $\epsilon = 2^{-d}$ with a success probability at least $1-\delta$, we need $d + \log_2\lceil\delta^{-1}\rceil$, ancilla qubits to store the value of the phase.
- Simulation time required is $T = (\epsilon\delta)^{-1}$.
- This method requires deep circuits with ancilla bits.
- It is hard to execute reliably without quantum error correction.

In order to mitigate some of the problems like deep circuits and requirement of quantum error correction, a new method called Variational Qunatum Eigensolvers were devised which will be now explained.

4.2.3 Variation Quantum Eigensolvers

A common goal of variational algorithms is to find the quantum state with the lowest or highest eigenvalue of a certain observable. A key insight we'll use is the variation theorem of quantum mechanics. before going into its full statement, let us explore some of the mathematical intuition behind it.

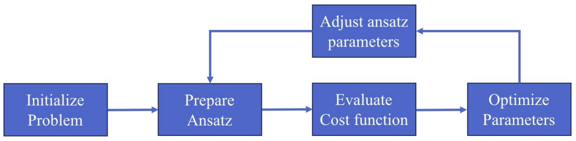


Figure 5: Variation Quantum Eigensolver

In quantum mechanics, energy comes in the form of a quantum observable usually referred to as the *Hamiltonian*, which we'll denote by $\hat{\mathcal{H}}$. let us consider its spectral decomposition:

$$\hat{\mathcal{H}} = \sum_{k=0}^{N-1} \lambda_k |\phi_k\rangle \langle\phi_k|$$

where N is the dimensionality of the space of state, λ_k is the k-th eigenvalue or, physically, the k-th energy level, and $|\phi_k\rangle$ is the corresponding eigen state: $\hat{\mathcal{H}}|\phi_k\rangle = \lambda_k |\phi_k\rangle$, the expected energy of a system in the (normalized) state $ket\psi$ will be:

$$\begin{aligned}\left\langle \psi|\hat{\mathcal{H}}|\psi \middle| \psi|\hat{\mathcal{H}}|\psi \right\rangle &= \left\langle \psi| \left(\sum_{k=0}^{N-1} \lambda_k |\phi_k\rangle \langle k|\right) |\psi \middle| \psi| \left(\sum_{k=0}^{N-1} \lambda_k |\phi_k\rangle \langle k|\right) |\psi \right\rangle \\ &= \sum_{k=0}^{N-1} \lambda_k \langle \psi|\phi_k|\psi|\phi_k\rangle \langle \phi_k|\psi|\phi_k|\psi\rangle \\ &= \sum_{k=0}^{N-1} \lambda_k | \langle \psi|\phi_k|\psi|\phi_k\rangle |^2\end{aligned}$$

If we take into account that $\lambda_0 \leq \lambda_k, \forall k$, we have:

$$\begin{aligned}\left\langle \psi|\hat{\mathcal{H}}|\psi \middle| \psi|\hat{\mathcal{H}}|\psi \right\rangle &= \sum_{k=0}^{N-1} | \langle \psi|\phi_k|\psi|\phi_k\rangle |^2 \\ &\geq \sum_{k=0}^{N-1} \lambda_0 | \langle \psi|\phi_k|\psi|\phi_k\rangle |^2 \\ &= \lambda_0 \sum_{k=0}^{N-1} | \langle \psi|\phi_k|\psi|\phi_k\rangle |^2 \\ &= \lambda_0\end{aligned}$$

Since $\{|\phi_k\rangle\}_{k=0}^{N-1}$ is an orthonormal basis, the probability of measuring $|\phi_k\rangle$ is $p_k = | \langle \psi|\phi_k|\psi|\phi_k\rangle |^2$, and the sum of all the probabilities is such that $\sum_{k=0}^{N-1} | \langle \psi|\phi_k|\psi|\phi_k\rangle |^2 = \sum_{k=0}^{N-1} p_k = 1$. In short, the expected energy of any system is higher than the lowest energy or ground state energy:

$$\left\langle \psi|\hat{\mathcal{H}}\psi \middle| \psi|\hat{\mathcal{H}}\psi \right\rangle \geq \lambda_0$$

the above argument applies to any valid (normalized) quantum state $|\psi\rangle$, so it is perfectly possible to consider parameterized state $\left|\psi(\vec{\theta})\right\rangle$ that depends on a parameter vector $\vec{\theta}$. This is where the "variational" part comes into play. If we consider a cost function given by $C(\vec{\theta}) := \left\langle \psi(\vec{\theta})|\hat{\mathcal{H}}|\psi(\vec{\theta}) \middle| \psi(\vec{\theta})|\hat{\mathcal{H}}|\psi(\vec{\theta}) \right\rangle$ and we want to minimize it, the minimum will always satisfy:

$$\min_{\vec{\theta}} C(\vec{\theta}) = \min_{\vec{\theta}} \left\langle \psi(\vec{\theta}|\hat{\mathcal{H}}|\psi(\vec{\theta}) \middle| \psi(\vec{\theta}|\hat{\mathcal{H}}|\psi(\vec{\theta}) \right\rangle \geq \lambda_0$$

The minimum value of $C(\vec{\theta})$ will be the closest that one can get to λ_0 using the parameterized state $\left|\psi(\vec{\theta})\right\rangle$, and equality will only be reached it there exists a parameter vector $\vec{\theta^*}$ such that $\left|\vec{\theta^*}\right\rangle = |\phi_0\rangle$

Pictorially this can be represented as shown in the figure.

4.2.4 Challenges with VQE and advantages of Quantum Subspace methods

VQE when optimizing parameters on a classical computer requires non-linear parameter optimization which is a NP-hard problem whereas in case of Quantum subspace method there is no non-linear parameter optimization.

In each update of quantum circuit parameter requires a new class of the quantum computer in just one call which incurs overhead communication costs. On the other hand, the Quantum Subspace methods can be naturally parallelized.

Quantum subspace methods on the other hand have two main limitation which is the eigenstates are not stored on a quantum device, so computing properties after a QSE calculation requires additional measurements. Also, accuracy and computational cost depends on the choice of basis states spanning subspace.

5 Proposed Problem

We wish to solve for the following differential equation which encodes the complexities of Poisson as well as Helmholtz and very commonly arrives in many applications, which is an eigenvalue problem and we wish to find the n smallest eigenvalues of the given equation. The differential equation is:

$$\left(-\frac{1}{2}\nabla^2 + V(x)\right)\psi(x) = \lambda\psi(x)$$

Upon, making the following assumptions for discretizing:

1. We use Finite Difference Method (FDM) (using 2nd order accurate Central Difference Scheme for the Laplacian operator)
2. Assuming a Uniform Grid ($\Delta x = \Delta y = \Delta z$).
3. Assuming the 1-dimensional case for simplicity.

The equation reduces to the following matrix equation.

$$\begin{bmatrix} \ddots & \ddots & 0 & \dots & 0 \\ \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & -\frac{1}{2} & 1+V(x) & -\frac{1}{2} & \vdots \\ \vdots & \dots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \\ \psi(x_i - \Delta x) \\ \psi(x_i) \\ \psi(x_i + \Delta x) \\ \vdots \\ \vdots \end{bmatrix} = \lambda \begin{bmatrix} \vdots \\ \vdots \\ \psi(x_i - \Delta x) \\ \psi(x_i) \\ \psi(x_i + \Delta x) \\ \vdots \\ \vdots \end{bmatrix}$$

Note that the discretized matrix is a Real, Symmetric and tridiagonal.

6 Quantum Chebyshev Filtered Subspace Iteration Algorithm

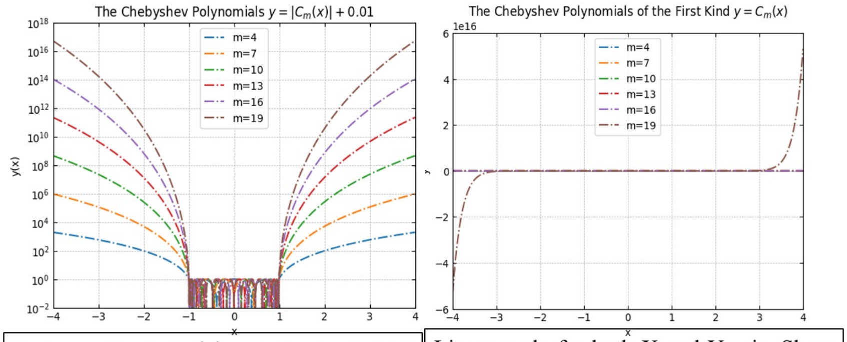
6.1 Cheybshev Filtering

Chebyshev Polynomials [ZCS14] (of the first kind) can be used as spectrum filters which allow us to introduce favourable gaps among the wanted eigenvalues). They are given as follows:

$$C_{m+1}(x) = \left\{ \begin{array}{ll} \cos m \cos^{-1}(x), & |x| < 1 \\ \cosh m \cosh^{-1}(x), & x > 1 \\ (-1)^m \cosh m \cosh^{-1}(-x), & x < -1 \end{array} \right\}$$

The Recursive relation for achieiving higher order chebyshev polynomials of the first kind is given as follows:

$$C_{m+1}(x) = 2xC_m(x) - C_{m-1}(x), \quad m = 1, 2, \dots,$$



Fast growth of $C_m(x)$ outside the $[-1,1]$ interval. Uses a semilog for the y-axis, for which we take absolute value of $C_m(x)$ and add a mall shift 0.01

$$y = |C_m(x)| + 0.01$$

Linear scale for both X and Y-axis. Show the complete dominance of the highest degree polynomial over lower degree counterparts outside the $[-1,1]$ interval.

Figure 6: Chebyshev Polynomials

with the condition that $C_0(x) = 1, C_1(x) = x$. These polynomial shave exponential growth of C_m outside the $[-1, 1]$ interval. It is in fact the fastest exponential grown among all polynomials with degree $\leq m$. This is evident from the graphs as shown in the figure

Say we denote the eigenbasis of a Hermitian H as $\psi's$ with corresponding eigenvalues $\lambda_i's$ i.e. $H\psi_i = \lambda_i\psi_i$. Any initial vector x_0 can be expanded in the eigenbasis as

$$x_0 = \alpha_1\psi_1 + \alpha_2\psi_2 + \ldots + \alpha_n\psi_n$$

Now upon apply polynomial filter: (generally for a random x_0, we have $\alpha \neq 0$), we get,

$$p(H)x_0 = \alpha_1 p(\lambda_1)\psi_1 + \alpha_2 p(\lambda_2)\psi_2 + \ldots + \alpha_n p(\lambda_n)\psi_n$$

After normalization we have, $p(H)x_0 \approx \psi_1$ i.e. the vector to be almost parallel to the largest eigenvector (eigenvector corresponding to the largest eigenvalue). The convergence rate of an eigen-algorithm depends on the gap ratio $= \max_j \frac{\lambda_j}{\lambda_1}, \quad j \neq 1$.

We then apply an affine mapping to map unwanted spectrum inside the $[-1, 1]$ interval, then the exponential growth property automatically applied to map the wanted spectrum (by better gap-ratio).

6.2 Chebyshev Filtered Subspace Iteration

Given the eigenvalue problem of finding n smallest eigenvalues

$$HX = X\Lambda$$

where $H \in \mathbb{C}^{m\times m}$, $X \in \mathbb{C}^{m\times n}$, $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$, with $n << m$. Let the initial guess be

$$\Psi_0 = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ |\psi\rangle & C_1(H)\,|\psi\rangle & \ldots & C_{n-1}(H)\,|\psi\rangle \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Then, for performing the filtering step we are required to perform:

$$\Psi_F = C_m(H)\psi_0 = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ C_m(H)\,|\psi\rangle & C_m(H)C_1(H)\,|\psi\rangle & \ldots & C_m(H)C_{n-1}(H)\,|\psi\rangle \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

where we can denote $|\psi_i\rangle = C_i(H)\,|\psi\rangle$. Then the Projected eigenvalue problem after the Rayleigh-Ritz Projection will be come:

$$\Psi_F^\dagger H \Psi_F X = \Psi_F^\dagger \Psi_F X \Lambda$$

Thus, the projected Eigenvalue problem becomes:

$$\tilde{H}\tilde{X} = \tilde{X}\Lambda$$

where $\tilde{H} = \Psi_F^\dagger H \Psi_F$ and $\tilde{X} = \Psi_F^\dagger \Psi_F X$. This is where we bring in Quantum Method which we call Quantum Chebyshev Filtered Subspace Iteration Method. The computational cost is usually high for filtering step and projection step. To this end, we design an algorithm for these two steps which can leverage a quantum computer. Note that we are required to find the following values which are the entries of the projected matrix $\tilde{H}$ which is

$$\tilde{H}_{ij} = \langle\psi_i| H |\psi_j\rangle = \langle\psi| C_i(H)C_m(H)HC_m(H)C_j(H) |\psi\rangle$$

$$S_{ij} = \langle\psi_i|\psi_j\rangle = \langle\psi| C_i(H)C_m(H)C_m(H)C_j(H) |\psi\rangle$$

for all $i, j = 0, 1, \ldots, n$. Upon using, the property of Chebyshev polynomials and simplifying the above expression we see that we are required to find a total of $2n + 2m + 1$ expectation values. This can be easily done on a quantum computer by doing block encoding of the required matrices as $(\langle G|_a \otimes I_s)(RU)^k |G\rangle_a \otimes I_s) = C_k(H)$ where we assume that the Hermitian Operator can be written in the basis of Pauli operators as

$$H = \sum_{i=0}^{T-1} \alpha_i P_i$$

and R is a reflection operator about the place of $|G\rangle_a$ defined as $R = (2|G\rangle_a \langle G|_a - I_a) \otimes I_s$. Here, we are required to do a state preparation of $|G\rangle_a = \sum_{i=0}^{T-1} \sqrt{\alpha_i} |i\rangle_a$ and the unitary U is defined as $U = \sum_{i=0}^{T-1} |i\rangle_a \langle i|_a \otimes P_i$.

References

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. *Proceedings of the Annual ACM Symposium on Theory of Computing*, Part F129452:212–219, 5 1996.

[HHL08] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for solving linear systems of equations. *Physical Review Letters*, 103(15), nov 2008.

[Lin22] Lin Lin. Lecture notes on quantum algorithms for scientific computation. *arXiv preprint arXiv:2201.08309*, 2022.

[MKL+24] Mario Motta, William Kirby, Ieva Liepuoniute, Kevin J Sung, Jeffrey Cohn, Antonio Mezzacapo, Katherine Klymko, Nam Nguyen, Nobuyuki Yoshioka, and Julia E Rice. Subspace methods for electronic structure simulations on quantum computers. *Electronic Structure*, 6(1):013001, 2024.

[Pre18] John Preskill. Quantum computing in the nisq era and beyond. 1 2018.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997.

[TB22] Lloyd N Trefethen and David Bau. *Numerical linear algebra*. SIAM, 2022.

[ZCS14] Yunkai Zhou, James R Chelikowsky, and Yousef Saad. Chebyshev-filtered subspace iteration method free of sparse diagonalization for dft calculations. 2014.