Quantum Computing for Solving Cubic Schrödinger Equation

by

Zhalgasbek Ayaz

April, 2025

Submitted to the Department of Mathematics in partial fulfilment of the requirements for the degree of Master of Science in Applied Mathematics

at the

NAZARBAYEV UNIVERSITY

Author ..... Zhalgasbek Ayaz

Thesis supervisor..... Dr. Alejandro Javier Castro Castilla

Contents

Abstract

This thesis explores the numerical approximation of special case of Nonlinear Schrödinger Equation(NLSE) using quantum computing techniques. A Variational Quantum Linear Solver(VQLS) is utilized to address the challenges of solving complex-valued linear systems derived from the discretised form of the CSE. The original complex-valued problem is converted into a real-valued system to perform the approximations. The performance of the quantum simulation of VQLS algorithm is compared with the classical methods of approximation through numerical experiments on $4 \times 4, 8 \times 8, 16 \times 16$ systems via quantum simulations. While classical methods prove to be faster and more accurate for the given systems, the future of the hybrid quantum-classical algorithms are promising. Accuracy and runtime analyses showed the exponential scaling behaviour of the quantum approach as the system became larger. The results suggest that although quantum methods are currently constrained by hardware and algorithmic limitations, they offer a promising foundation for future research on computational sciences.

Acknowledgement

I would like to express my sincere gratitude to everyone who supported me throughout my Master's studies and the completion of this thesis.

First and foremost, I am deeply thankful to my supervisor, Dr. Alejandro Javier Castro Castilla, for providing me with an opportunity to work on this topic, for invaluable guidance, patience, and insightful feedback at every stage of this work. His support has been instrumental in shaping my academic growth.

I would also like to thank Dr. Andrea Mari for his tutorial on the variational quantum linear solver. Without his contribution to the topic and open source materials, this work would have been extremely challenging to finish.

My appreciation also goes to Adilet Akimshe, whose support on the experimental part of the thesis is invaluable.

On a personal note, I am grateful to my family and friends for their unwavering support and encouragement, especially during the challenging moments. Their belief in me kept me going.

1 Introduction

Numerical approximation of differential equations has been pivotal in mathematics and applied industries but increasing model complexity strains classical computers. The numerical approximation of differential equations holds significant importance in mathematics and various applied sectors such as engineering and structural mechanics. While classical computers have historically excelled in tackling diverse problems, the increasing complexity of emerging models demands substantial computational resources. This challenge has prompted the development of quantum computers, offering a promising solution that could revolutionize problem-solving methodologies.

Quantum computers store information using quantum bits (qubits) and exploit specific properties of quantum mechanics for computation [8]. Unlike classical computers, qubits can represent data as both 1 and 0 simultaneously, expanding problem-solving capabilities. This novel approach to storing digital information presents an opportunity to solve linear systems of equations exponentially faster compared to classical computing methods.

Unlike classical mechanics, where the systems are described by definite quantities such as momentum and position, quantum systems are characterized by a wave function $u(x,t)$, which encodes the probability amplitude of a particle's state [8]. Quantum observables, such as energy or momentum, are represented by Hermitian operators acting on complex vector spaces, and the outcomes of measurements correspond to eigenvalues of these operators. A quantum state is a complete mathematical description of a system and is typically represented as a unit vector in a complex Hilbert space. These states can exist in a superposition, meaning a particle can occupy multiple states simultaneously until measured. In practical terms, a quantum state $|x\rangle$ is often expressed in Dirac's bra-ket notation and evolves under the influence of a Hamiltonian operator[10].

1.1 Background and motivation

Quantum computing has emerged as a promising paradigm for solving partial differential equations (PDEs). In certain cases, it offers exponential speedups compared to classical methods. In 2009, Harrow, Hassidim, and Lloyd devised a quantum algorithm specifically for addressing linear systems of equations[9]. Techniques like quantum variational algorithms and Hamiltonian simulation further enable the approximation of complex PDE solutions with potential applications in physics, finance,

and engineering[2]. However, practical implementation faces challenges, including noise, error correction, and scalability of quantum processors[14]. In 2023, a research team led by Daribayev successfully applied the HHL algorithm to solve the Poisson equation[6]. By implementing the algorithm on a quantum simulator, they demonstrated its exponential speedup compared to classical methods.

This trend of solving PDEs using quantum computers motivated me to find modern trends and techniques that can be used to approximate the solution of the special case for Nonlinear Schrödinger Equation(NSE). This thesis project endeavours to offer comprehensive insights into the numerical approximation of NSE using classical and quantum computing techniques. Additionally, this work aims to contribute to the ongoing discourse surrounding the integration of quantum systems into computational sciences and engineering.

1.2 Nonlinear Schrodinger Equation(NLSE)

The Nonlinear Schrödinger Equation (NLSE) is a fundamental equation in mathematical physics that extends the classical linear Schrödinger equation by incorporating nonlinear effects. It appears in various physical contexts, including optics, fluid dynamics, and Bose-Einstein condensates (BECs). The general form of the NLSE from the classical field theory is given by:

$$i\frac{\partial u}{\partial t} = -\frac{\partial^2 u}{\partial x^2} + f(|u|^2)u \tag{1}$$

where u is the complex wave function ($u = u(x,t) = v(x,t) + i\omega(x,t)$) and $f(|u|^2)$ is the nonlinear term that represents the intensity of the wave.

In this thesis project, we consider a special case of NLSE, which is the Cubic

Schrödinger Equation(CSE) of the following general form

$$i\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} + q|u|^2 u = 0;\ L_0 < x < L_1,\ t > t_0 \tag{2}$$

where $i = \sqrt{-1}$ and $q \geq 0$ is a real parameter of cubic nonlinearity [3]. Equation (2) describes the propagation of light pulses in optical fibers, leading to soliton formation[1].

2 Methods

2.1 Classical Approach

There are various numerical methods for solving PDEs, that offer different insights into solutions. Among these methods, the finite difference methods (FDM) and the finite element methods (FEM) are widely used due to their efficiency and accuracy in approximating solutions to PDEs. The choice between these methods depends on the problem's nature, the desired accuracy, and computational efficiency. These methods provide discrete approximations to PDEs, enabling simulations of various complex physical phenomena used in fluid dynamics, wave propagation and quantum mechanics[13].

The FDM is based on approximating derivatives using difference equations on a structured grid. It is particularly effective for problems with regular geometries and is widely used in time-dependent PDEs due to its simplicity and computational efficiency[16]. On the other hand, FEM employs piecewise polynomial basis functions over an unstructured mesh, making it suitable for irregular domains and problems involving complex boundary conditions[5]. While FEM provides higher flexibility and

accuracy for complex geometries, it typically requires more computational resources compared to FDM.

For this thesis work, the FDM was chosen to approximate the solution of CSE as its efficiency and ease of implementation is suitable for these types of equations. CSE requires accurate time-stepping schemes such as the Crank-Nicolson method or the split-step Fourier method. Additionally, FDM accommodates periodic boundary conditions, which are common in CSE applications. It also provides an efficient framework for handling the CSE's dispersive and nonlinear terms while maintaining computational feasibility for high-dimensional problems[15].

It was decided to take the existing numerical approximation of CSE and replicate its results. From [3] the initial condition is described in the following form

$$u(x, t_0) = f(x) = f_R(x) + if_I(x); L_0 \leq x \leq L_1 \tag{3}$$

in which f_R and f_I are real-valued, continuous functions of x and boundary conditions

$$\frac{\partial u(L_0, t)}{\partial x} = 0 \text{ and } \frac{\partial u(L_1, t)}{\partial x} = 0; \ t \geq t_0. \tag{4}$$

The initial/boundary-value problem(IBVP), described in equations (2) and (4), gives rise to soliton solutions in which the solution and its derivatives with respect to x vanish as $|x| \to \infty$ [3]. For the single-soliton case when $q \neq 0$, the analytical function is of the form

$$u(x, t) = \left(\frac{2a}{q}\right)^{\frac{1}{2}} \exp\left\{i\left[\frac{1}{2}cx - \theta t\right]\right\} \operatorname{sech}[a^{\frac{1}{2}}(x - ct)], \tag{5}$$

where $\theta = \frac{c^2}{4} - a$ satisfies the IVBP.

Through the Crank-Nicolson scheme equation (2) is written as

$$\frac{i}{l}\left(u_m^{n+1} - u_m^n\right) + \frac{1}{2}\left(u_m^{n+1} + u_m^n\right)_{xx} + \frac{q}{2}\left(|u_m^{n+1}|^2 u_m^{n+1} + |u_m^n|^2 u_m^n\right) = 0, \tag{6}$$

where for the space derivative the following central-difference formula is used

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{u(x-h,t) - 2u(x,t) + u(x+h,t)}{h^2}, \tag{7}$$

where $h = \Delta x$ and $l = \Delta t$. Let U be a numerical approximation of u. Now, the linearized form of the scheme is obtained by using Taylor's expansion of $|U_m^{n+1}|^2 U_m^{n+1}$ about the n-th time-level(for the detailed numerical scheme and analysis see [3]).

Finally, we obtain our three-level finite-difference scheme

$$\begin{aligned} p(U_{m-1}^{n+1} + U_{m+1}^{n+1}) + 2\,(i-p)\,U_m^{n+1} = -p\left(U_{m-1}^n + U_{m-1}^n\right) + 2(i+p)U_m^n \\ -lq[(2+3\lambda)|U_m^n|^2 - 2\lambda|U_m^n|U_m^{n-1}]U_m^n + \lambda lq|U_m^n|^2 U_m^{n-1}. \end{aligned} \tag{8}$$

with $p = \frac{l}{h^2}$, where λ is an appropriate positive real number. When (8) is applied to all points of the grid, it leads to a linear algebraic system of equations of the form

$$A\mathbf{U}^{n+1} = F(\mathbf{U}^n, \mathbf{U}^{n-1}) \tag{9}$$

where A is a tridiagonal matrix of order $N+2$ given by

$$A = \begin{bmatrix} 2(i-p) & 2p & & & & \\ p & 2(i-p) & p & & & \\ & . & . & . & & \\ & & p & 2(i-p) & p & \\ & & & 2p & 2(i-p) & \end{bmatrix}. \tag{10}$$

The von Neumann method with linearization of the non-linear term $|U^{n+2}|^2$ was used for the stability analysis of the provided method. For the full stability analysis of the method see [3].

2.2 Complex valued problems

The classical scheme described in (8) contains complex values. Since it is not a trivial task for a quantum algorithm to work with complex-valued matrices, it is more efficient to convert complex-valued matrices into real-valued matrices, preserving their structure and properties. Now, we can convert the complex values scheme to a real-valued scheme by applying the following substitution

$$U_m^n = \omega_m^n + v_m^n i \tag{11}$$

where ω represents the real part and v represents the imaginary part of original scheme. Our original scheme becomes

$$p((w_{m-1}^{n+1} + iv_{m-1}^{n+1}) + (w_{m+1}^{n+1} + iv_{m+1}^{n+1})) + 2(i-p)(w_m^{n+1} + iv_m^{n+1}) =$$

$$- p((w_{m-1}^{n} + iv_{m-1}^{n}) + (w_{m+1}^{n} + iv_{m+1}^{n})) + 2(i+p)(w_m^{n} + iv_m^{n})$$

$$- lq\left[(2+3\lambda)|(w_m^n + iv_m^n)|^2 - 2\lambda|w_m^n + iv_m^n|(w_m^{n-1} + iv_m^{n-1})\right](w_m^n + iv_m^n)$$

$$+ \lambda lq|w_m^n + iv_m^n|^2(w_m^{n-1} + iv_m^{n-1}). \quad (12)$$

Now, we can separate the real parts of the scheme from the imaginary parts. So the real part of the scheme becomes

$$p(v_{m-1}^{n+1} + v_{m+1}^{n+1}) + 2(w_m^{n+1} - p(v_m^{n+1})) =$$

$$- pv_{m-1}^{n} - pv_{m+1}^{n} + 2w_m^n + 2pv_m^n - lq((2+3\lambda) * ((w_m^n)^2 + (v_m^n)^2)$$

$$- 2\lambda\sqrt{(w_m^n)^2 + (v_m^n)^2}\sqrt{(w_m^{n-1})^2 + (v_m^{n-1})^2)}v_m^n + \lambda l((w_m^n)^2 + (v_m^n)^2)v_m^{n-1} \quad (13)$$

and imaginary part

$$p(v_{m-1}^{n+1} + v_{m+1}^{n+1}) + 2(w_m^{n+1} - p(v_m^{n+1})) =$$

$$- pv_{m-1}^n - pv_{m+1}^n + 2w_m^n + 2pv_m^n - lq((2+3\lambda) * ((w_m^n)^2 + (v_m^n)^2)$$

$$- 2\lambda\sqrt{(w_m^n)^2 + (v_m^n)^2}\sqrt{(w_m^{n-1})^2 + (v_m^{n-1})^2})v_m^n + \lambda l((w_m^n)^2 + (v_m^n)^2)v_m^{n-1}. \quad (14)$$

Now, when real part and imaginary parts are separated, the new real valued matrix is obtained

$$Ab = \left[\begin{array}{cccc|cccc} -2p & 2p & 0 & ... & 2 & 0 & 0 & ... \\ p & -2p & p & ... & 0 & 2 & 0 & ... \\ ... & ... & ... & ... & ... & ... & ... & ... \\ ... & p & -2p & p & ... & 0 & 2 & 0 \\ ... & 0 & 2p & -2p & ... & 0 & 0 & 2 \\ \hline 2 & 0 & 0 & ... & -2p & 2p & 0 & ... \\ 0 & 2 & 0 & ... & p & -2p & p & ... \\ ... & ... & ... & ... & ... & ... & ... & .. \\ ... & 0 & 2 & 0 & ... & p & -2p & p \\ ... & 0 & 0 & 2 & ... & 0 & 2p & -2p \end{array}\right] \left[\begin{array}{c} w_1 \\ w_2 \\ ... \\ w_n \\ w_{n+1} \\ \hline v_1 \\ v_2 \\ .. \\ v_n \\ v_{n+1} \end{array}\right]. \quad (15)$$

Similarly, the b vector is divided into real and imaginary parts. Now, every complex valued matrix $n \times n$ becomes $2n \times 2n$ real valued matrix.

2.3 Quantum Approach(Quantum Linear Solver)

Numerous quantum algorithms have been developed for solving linear systems of equations. The cornerstone of modern ways to solve linear systems on quantum devices is the HHL algorithm[9]. Unlike the classical way of solving linear systems of equations, where given a matrix A and a vector b, we find a vector x such that $Ax = b$, the proposed algorithm does not need to know the solution x, but rather an approximation the expectation value of some operator associated with vector x. There are conditions that must be followed in order for the algorithm to succeed. The matrix A must be Hermitian matrix, sparse and efficiently row computable. If the matrix is not Hermitian, then a new matrix can be defined as follows:

$$\tilde{A} = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}.$$

Once these conditions are satisfied, we can proceed with the algorithm itself. First, the given Hermitian matrix $\tilde{A}$ is transformed into a unitary operator e^{iAt}. Since $\tilde{A}$ is Hermitian, the problem can be solved

$$\tilde{A}y = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

to obtain

$$y = \begin{bmatrix} 0 \\ x \end{bmatrix}.$$

Next important step is to to prepare quantum state $|b\rangle$ as $|b\rangle = \sum_{i=1}^{N} b_i|i\rangle$. Then, the techniques of Hamiltonian simulation[7] are used to apply e^{iAt} to $|b\rangle$ for a superposition of different times t. This ability to exponentiate A translates, via the well-known technique of phase estimation[12], into the ability to decompose $|b\rangle$ in the eigenbasis

of A and to find the corresponding eigenvalues λ_j. Informally, the state of the system after this stage is close to

$$\sum_{j=1}^{N} \beta_j |u_j\rangle |\lambda_j\rangle,$$

where u_j is the eigenvector basis of A, and

$$|b\rangle = \sum_{j=1}^{N} \beta_j |u_j\rangle.$$

Then the linear map is performed taking $|\lambda_j\rangle$ to $C\lambda_j^{-1}|\lambda_j\rangle$, where C is a normalizing constant. After it succeeds, it is needed uncompute the $|\lambda_j\rangle$ register and are left with a state proportional to

$$\sum_{j=1}^{N} \beta_j \lambda_j^{-1} |u_j\rangle = A^{-1}|b\rangle = |x\rangle.$$

However, in the modern state of Noisy Intermediate-Scale Quantum (NISQ) devices that operate on relatively fewer qubits, implementing a pure HHL algorithm for large linear systems is challenging due to the algorithm's resource intensity (large number of qubits). Therefore, many alternative approaches exist for solving linear systems on existing quantum devices such as adiabatic-inspired quantum algorithms that was implemented for an 8×8 system, which is still quite small for our proposed problem. Another alternative, that this thesis will be using, is variational hybrid quantum-classical algorithms(VHQCAs)[4]. In contrast with previously mentioned gate based quantum algorithms, VHQCAs use additional classical optimization techniques to reduce the depth of quantum circuits to improve their time complexity.

For this thesis, the Variational Quantum Linear Solver[3] will be used as a main tool for solving our problem. The proposed VQLS hybrid methods seeks to variationally prepare $|x\rangle$ such that $A|x\rangle \propto |b\rangle$[4]. Figure 1 demonstrates the schematic diagram

for the VQLS algorithm. The input to VQLS is a matrix A written as a linear combination of unitaries A_l and a short-depth quantum circuit U which prepares the state $|b\rangle$. The output of VQLS is a quantum state $|x\rangle$ that is approximately proportional to the solution of the linear system $Ax = b$. Parameters α in the ansatz $V(\alpha)$ are adjusted in a hybrid quantum-classical optimization loop until the cost $C(\alpha)$ (local or global) is below a user-specified threshold. When this loop terminates, the resulting gate sequence $V(\alpha_{opt})$ prepares the state $|x\rangle = x/||x||_2$, from which observable quantities can be computed. Furthermore, the final value of the cost $C(\alpha_{opt})$ provides an upper bound on the deviation of observables measured on $|x\rangle$ from observables measured on the exact solution. The schematics of the algorithm is shown in Figure 1.

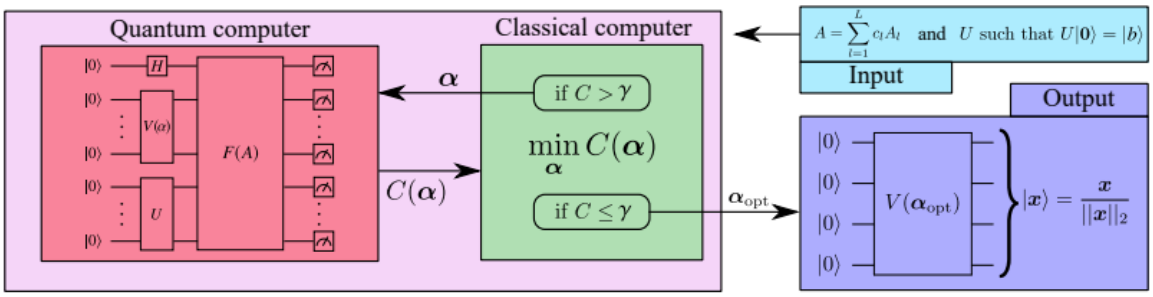

Figure 1: Schematic diagram of algorithms proposed in [4]

It is also, important to ensure that the linear system is properly constructed. Thus, in the VQLS framework, the matrix A is expressed as a linear combination of unitary operators, often using Pauli strings. To be specific, $A = \sum_l c_l A_l$, where each A_l is a unitary operator - a tensor product of Pauli matrices, and c_l are complex coefficients. Such decomposition allows shallow circuits to operate efficiently during the execution of the algorithm on NISQ devices. These Pauli operators consist of

three 2 by 2 Hermitian and unitary matrices:

$$\text{Pauli-X}(\sigma_x) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\text{Pauli-Y}(\sigma_y) = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$\text{Pauli-Z}(\sigma_z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The combination of these matrices and the 2×2 identity matrix forms a basis for the vector space of 2 by 2 Hermitian matrices. This property allows any single-qubit Hermitian operator to be expressed as a linear combination of Pauli matrices, which is instrumental for the algorithm which we are implementing.

The proposed algorithm heavily relies on an optimization of cost functions while preparing a quantum state $|x(\alpha)\rangle$ that approximates the solution to the quantum linear system $A|x\rangle = |b\rangle$. There are several cost functions that can benefit this algorithm. The global cost function $\hat{C}_G$ is defined to measure the overlap of the trial state $|\psi\rangle = A|x\rangle$, which explicitly looks as

$$C_G = 1 - \frac{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger U|0\rangle\langle 0|U^\dagger A_l V|0\rangle}{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger A_l V|0\rangle}. \tag{16}$$

Here, all expectation values could be estimated with a Hadamard test. However, the global cost function is experimentally challenging since it requires applying all the unitaries ($U^\dagger, A_l$ and V) in a controlled way, which can be further implemented as a development of this thesis work.

Another method is to minimize the local version of the cost function, which is easier to measure and also leads to the same optimal solution. This local cost function can be obtained by replacing $|0\rangle\langle 0|$ from 16 by the following operator

$$P = \frac{1}{2} + \frac{1}{2n}\sum_{j=0}^{n-1} Z_j, \tag{17}$$

where Z_j is the Pauli operator Z applied to the jth qubit. Now, our new cost function becomes

$$C_L = 1 - \frac{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger U P U^\dagger A_l V|0\rangle}{\sum_{l,l'} c_l c_{l'}^* \langle 0|V^\dagger A_{l'}^\dagger A_l V|0\rangle}. \tag{18}$$

Then we can solve our problem using C_L. By substituting P into the expression (18) we get

$$C_L = \frac{1}{2} - \frac{1}{2n}\frac{\sum_{j=0}^{n-1}\sum_{l,l'} c_l c_{l'}^* \mu_{l,l',j}}{\sum_{l,l'} c_l c_{l'}^* \mu_{l,l',-1}}, \tag{19}$$

which we can compute whenever we are able to measure the following coefficients

$$\mu_{l,l',j} = \langle 0|V^\dagger A_{l'}^\dagger U Z_j U^\dagger A_l V|0\rangle,$$

where we used the convention that if $j = -1$, Z_{-1} is replaced with the identity. These, coefficients can be measured using a Hadamard test. The advantage of this local cost function is that only the unitary operations $A_l, A_{l'}^\dagger, Z_j$ need control on an ancillary qubit. V, U and their adjoints can be directly applied to the system.

2.4 Technical setup

The implementation of classical numerical schemes and quantum simulations was performed on the mid-range processor based on AMD's Zen 4 architecture, Ryzen 5

7500f. Below are the specifications of the device.

Processor	AMD Ryzen5 7500f
Architecture	Zen 4(5nm process)
Cores/Threads	6 cores /12 threads
Base clock	3.7 GHz
Boost clock	Up to 5.0 GHz
Cache	32MB L3, 6MB L2
RAM	32Gb
RAM frequency	6000Hz

Table 1: Technical specification of first device

The high clock speeds and efficient number of cores, combined with 12 threads, provide good performance for large enough systems. With 32GB of 6000Hz RAM, the computation speed allows fast approximation of large systems. For this project, GPU acceleration was not used in any part of the process.

3 Results

All of the code used for this thesis can be found in the following GitHub repository Quantum-computing-for-solving-Cubic-Schrodinger-Equation.

The IBVP (3) - (5) was solved numerically with the following values. Baundaries lie between $-80(L_0)$ and $100(L1)$, and the initial value is $U(x,t_0) = u(x,t_0); x \in (-80, 100)$, which means the numerical solution at time step $t = 0$ is equal to the theoretical solution. Additionally, since our scheme is implicit in 2 time levels, the numerical solution at time step $t = 1$ is also equal to the theoretical solution. This

gives us the following

$$\begin{cases} \frac{\partial u(-80,t)}{\partial x} = 0 \text{ and } \frac{\partial u(100,t)}{\partial x} = 0 \\ u(x,t_0) = u(x,t_1) = f(x) = f_R(x) + if_I(x); -80 \leq x \leq 100 \\ u(x,t) = \left(\frac{2a}{q}\right)^{\frac{1}{2}} \exp\left\{i\left[\frac{1}{2}cx - \theta t\right]\right\} \operatorname{sech}[a^{\frac{1}{2}}(x-ct)] \end{cases} \tag{20}$$

For the accuracy, it was decided to use the following percentage relative error $L_\epsilon = L_\infty \times 100/|u_m^n|$, where $L_\infty = |u_m^n - U_m^n|$.

The original approximation provided in [3], with theoretical parameters $q = 1, a = 0.01, c = 0.1$ to time $t = 36$ with time step $l = 0.01$ and space step $h = 1$ is provided in figures 2 and 3 below. The relative percentage error for the conducted experiment was $L_\epsilon = 0.145$.

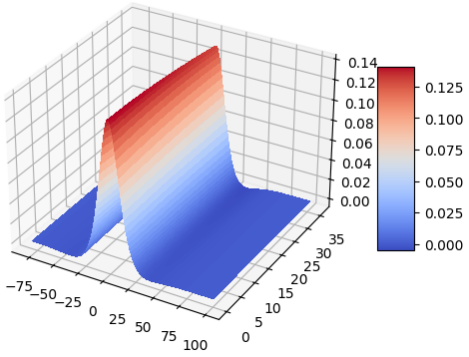

Figure 2: Real part of solution

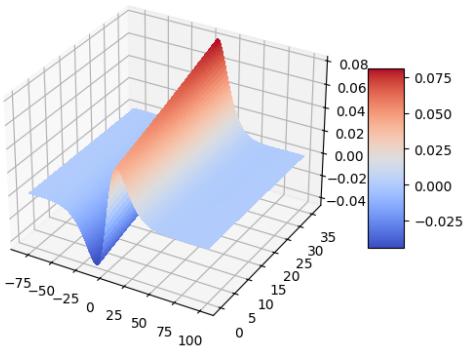

Figure 3: Imaginary part of solution

However, due to the technical limitations of the VQLS algorithm, the size of the problem was decreased. For the original problem boundaries were $L_0 = -80$ and $L_1 = 100$ with space step $h = 1$; the size of the matrix is 181 by 181, which cannot be described as 2^n for $n \in \mathbb{Z}$, which is crucial for implementation of VQLS. Also, the time is changed to $t = 10$ with time step $l = 1$. Thus, for the following experiments, different values of space step h were used to satisfy the requirements for the VQLS

algorithm. The VQLS algorithm was used to approximate solutions for the 4 by 4, 8 by 8 and 16 by 16 complex-valued matrices. For the quantum simulation, the number of system qubits was chosen according to the size of the matrix.

3.1 4 by 4 matrix

The 4 by 4 complex-valued matrix becomes an 8 by 8 real matrix. Applying the substitution from (11) we obtain the following real-valued matrix

$$A = \left[\begin{array}{cccc|cccc} -2p & 2p & 0 & 0 & 2 & 0 & 0 & 0 \\ p & -2p & p & 0 & 0 & 2 & 0 & 0 \\ 0 & p & -2p & p & 0 & 0 & 2 & 0 \\ 0 & 0 & 2p & -2p & 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 0 & -2p & 2p & 0 & 0 \\ 0 & 2 & 0 & 0 & p & -2p & p & 0 \\ 0 & 0 & 2 & 0 & 0 & p & -2p & p \\ 0 & 0 & 0 & 2 & 0 & 0 & 2p & -2p \end{array}\right]$$

where $p = \frac{l}{h^2}$. Then we decomposed the matrix into unitary operators. After the decomposition we can enter our vector b computed using scheme (8). We then implement the Hadamard test, where unitary operators A_j are controlled by the state of an ancillary qubit. Then we generated the solution state $|x\rangle = V(w)|0\rangle$. To generate solution we optimized the local cost function from (18). To minimize the cost function, the gradient-descent optimizer was used. The optimization results with respect to the optimization steps can be seen in Figure 4. The threshold value for the termination of local cost function optimization was set to 300. Additionally, loop has an option to terminate if $C_L < 10^{-5}$. From Figure 4 we can see that the cost function optimization terminated before reaching 300 steps as it already reached the threshold

value of 10^{-5}.

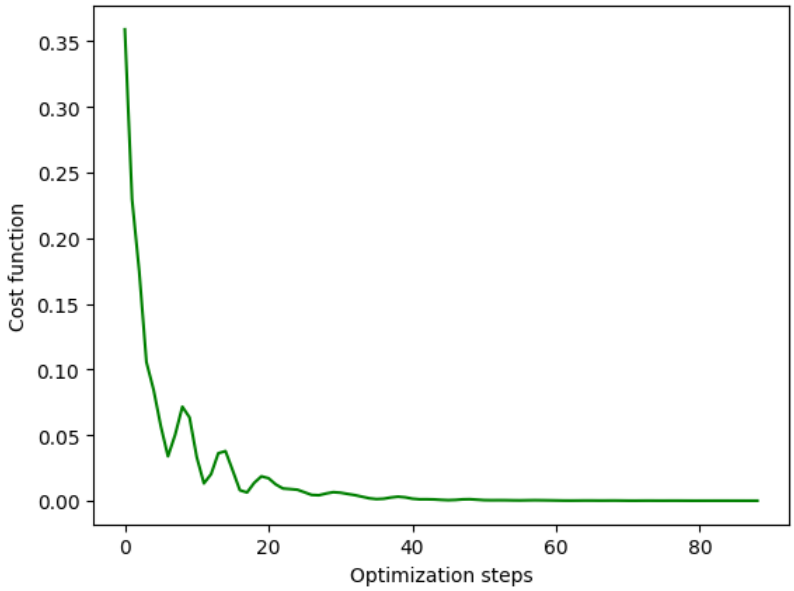

Figure 4: Cost function optimization

After the variational weights were optimized, we generated the quantum state $|x\rangle$. Measurement of $|x\rangle$ is not a solution itself, but the probability of each basis state. The probability distribution over the basis state is estimated by computing the frequency of each outcome. After the probabilities are obtained, we can compare the results with the classical approach. The side by side charts of classical and quantum probabilities are represented in Figure 5. Finally, by unnormalizing the quantum probabilities we obtained the solution for our 4 by 4 complex-valued matrix. The plots of the solution is represented in Figures 6 and 7. The same procedure was repeated for 8 by 8 and 16 by 16 matrices. For each problem the adjustment of hyperparamters should be made. One of the main hyperparameters is learning rate(eta), a hyperparameter that controls how much the model's parameters are adjusted during each step of the optimization process to minimize the error. Also, during the experiment a predefined structure of circuit(ansatz), which provides variational freedom to accurately approximate the

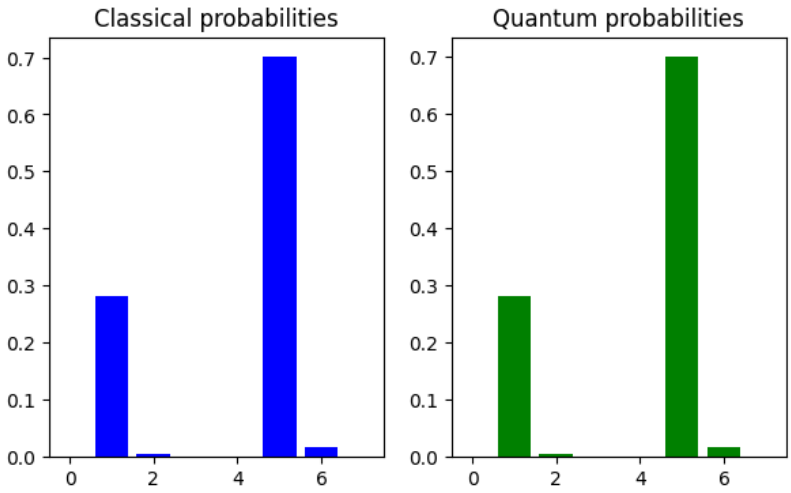

Figure 5: Classical and quantum vector probabilities

ground state without introducing excessive circuit depth. For this particular system circuit operated on 8 system qubits, and an additional ancillary qubit, with a learning eta of 0.3 and 2 layers of ansatz.

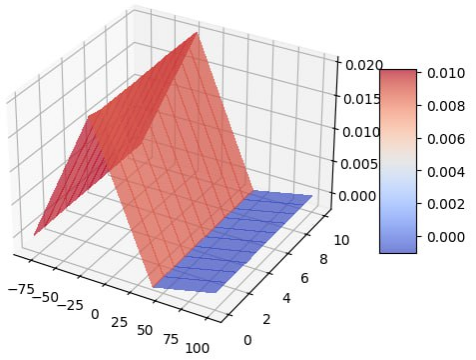

Figure 6: Real part of solution

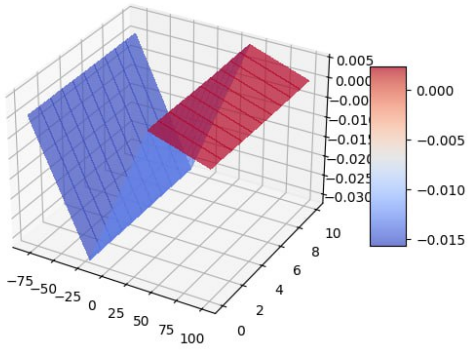

Figure 7: Imaginary part of solution

From Figures 6 and 7 we can see that the general wave type shape is preserved despite being too sharp. It is caused due to the smaller size of the system compared to the original classical method.

3.2 8 by 8 matrix

This 8 by 8 complex system is converted into a 16 by 16 real system, with the same structure as for the 4 by 4 case. For solving this problem circuit operated on the following hyperparameters: number of qubits = 16, ancillary qubit = 1, number of optimization steps = 300, eta = 0.2, layers of ansatz = 3. The number of optimization steps and the threshold value for C_L remained unchanged. The optimization results with respect to the optimization steps can be seen in Figure 8. As we can see from 8, the cost function terminated before reaching the threshold value, and outputted relatively accurate vector probabilities from Figure 9.

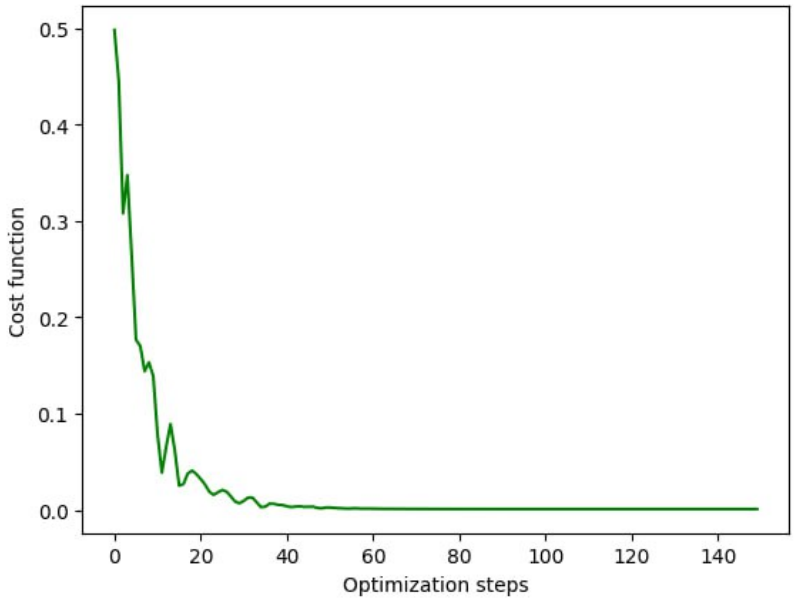

Figure 8: Cost function optimization

From Figure 9 we can see that the approximation of the probabilities was visually accurate. From Figures 10 and 11, we can notice that as the size of the grid increased, the shape of the waves also changed. It is smoother compared to the previous result, and the change in the imaginary part of the solution is more noticeable.

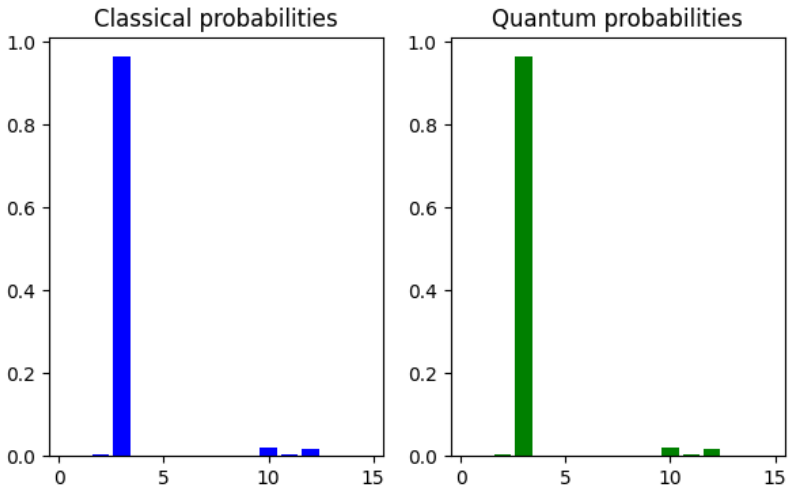

Figure 9: Classical and quantum vector probabilities

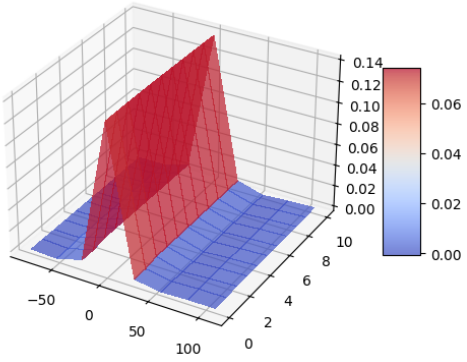

Figure 10: Real part of solution

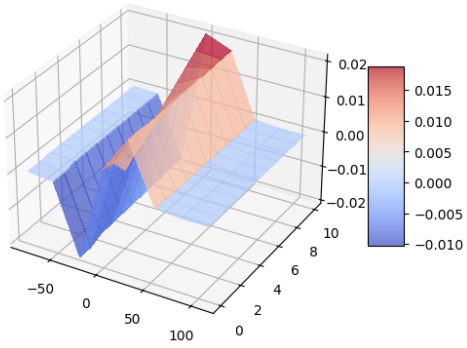

Figure 11: Imaginary part of solution

3.3 16 by 16 matrix

16 by 16 complex-valued matrix was converted into a 32 by 32 real-valued matrix, which means the number of system qubits used was 32, the number of optimization steps remained unchanged, eta = 0.01, layers of ansatz = 4. For this problem, the termination condition of the cost function loop was changed to $C_L < 10^{-3}$, due

to the time constraints, as the original condition did not allow the experiment to finish within the given time. From Figure 12, we can see that despite increasing the threshold value, the cost function loop terminated at 300th step.

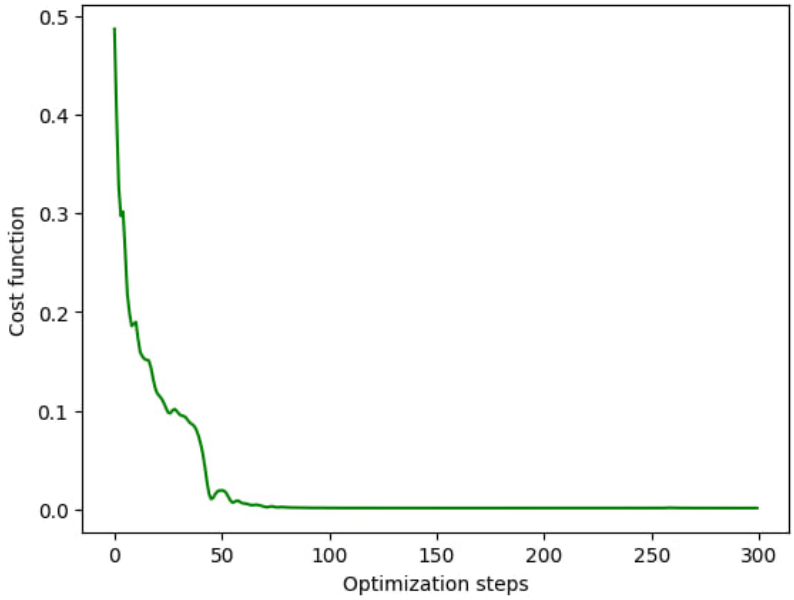

Figure 12: Cost function optimization

Due to the change of termination criterion from 10^{-5} to 10^{-3}, the accuracy of the approximation of probabilities has suffered. Despite the general shape and directions being similar, from Figure 13 we can notice that some components of the vectors for classical and quantum probabilities differ. However, these differences did not cause any global changes when it comes to the approximation of the whole solution over the chosen grid. The solution preserved its shape and amplitude. The visualization of the solution for a 16 by 16 complex matrix can be seen in Figures 14 and 15.

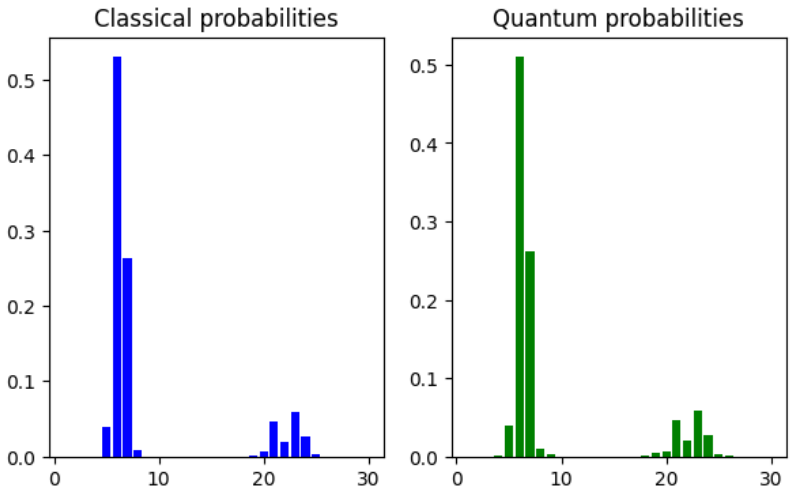

Figure 13: Classical and quantum vector probabilities

We can also see that the

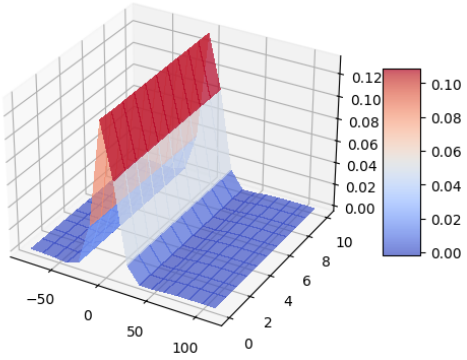

Figure 14: Real part of solution

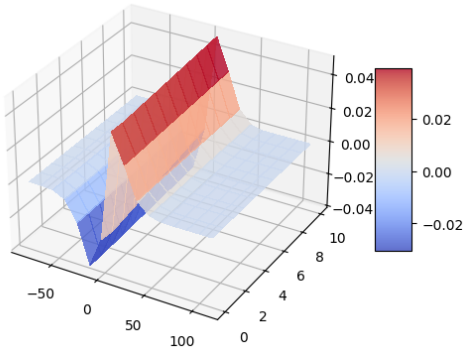

Figure 15: Imaginary part of solution

3.4 Accuracy

The time taken to conduct each experiments and the accuracy of solutions can bee seen from Table 2.

Size	Classical	Quantum	L_ϵ(C)	L_ϵ(Q)
4 by 4	3.61s	4h 36m 18s	0.307	0.324
8 by 8	4.8s	19h 24m 18s	0.274	0.288
16 by 16	3.45s	117h 7m 39s	0.245	0.313
32 by 32	2.87s	–	0.219	–
64 by 64	5.07s	–	0.198	–

Table 2: Accuracy and time complexity

We can see that as the size of the matrix increases, the time it takes to solve the problem increases. The change in time is exponential rather than linear. As the size of the problem increases, the adjustment of hyperparameters should be more precise to obtain an optimal solution. Also, the termination threshold affected the accuracy of the approximation. For the first two experiments, where the termination parameters were the same, the trend is decreasing. However, due to the change of termination threshold from 10^{-5} to 10^{-3} for the cost function for the 16 by 16 system, L_ϵ has increased. From the table we can also note that the accuracy of the approximation is relatively close.

3.5 Limitations

The main limitation of this project is the lack of access to the real NISQ devices, where the computational speed for the given circuit could have been exponentially faster. Previous successful experiments in that field were implemented on various NISQ devices such as Rigetti's quantum computer and IBM's Q System One. However, throughout the course of this work, obtaining access to the real NISQ devices was not possible. Also, the time taken to simulate the systems that operate on 32 or more qubits does not allow for conducting more experiments. For the experiment

on a 16 by 16 complex-valued matrix, the threshold value for the optimisation was sacrificed to obtain the results, to efficiently use the given time. Additionally, the quantum circuits that were used during the experiments initially were designed to solve small problems, which might be the reason behind challenging hyperparameter tuning.

4 Conclusion

This thesis work investigated the numerical approximation of the Cubic Schrödinger equation using both a classical approach and a simulation of the quantum approach. The classical implicit finite difference method has proven to be time-efficient and accurate for solving these types of equations on a classical computer. On the other hand, the quantum approach utilises more complex tools to achieve the same result, which in our case is not as efficient classical approach.

Through extensive experiments, the application of the VQLS algorithm on 4 by 4, 8 by 8 and 16 by 16 systems demonstrated its potential for solving linear systems. Although through the quantum approach we were able to preserve the general shape and behaviour of the classical solutions, it suffered from limitations in terms of runtime, precision and hardware dependence. Notably, the execution time increased exponentially as the size of the systems increased.

This study highlighted several key limitations that could be improved for the future development of the hybrid classical-quantum algorithm. The high computational cost of the quantum simulations on classical computers can be easily solved when the accessible quantum devices become available. Access to real quantum hardware and further refinement of cost functions and optimization strategies will be essential for

practical scalability.

In conclusion, while classical methods currently outperform the quantum methods of solving linear systems, this work provides promising evidence that quantum computing - especially hybrid approaches like VQLS - can play a significant role in the future of scientific computing.

References

[1] G. P. Agrawal. *Nonlinear fiber optics*. Academic press, 2019.

[2] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Phys. Rev. Lett.*, 114:090502, Mar 2015.

[3] A. G. Bratsos. A linearized finite-difference scheme for the numerical solution of the nonlinear cubic schrödinger equation. *Korean Journal of Computational and Applied Mathematics*, 2001.

[4] Carlos Bravo-Prieto, Ryan LaRose, M. Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J. Coles. Variational Quantum Linear Solver. *Quantum*, 7:1188, November 2023.

[5] S. C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer New York, NY, 2008.

[6] Beimbet Daribayev, Aksultan Mukhanbet, and Timur Imankulov. Implementation of the hhl algorithm for solving the poisson equation on quantum simulators. *Applied Sciences*, 13(20), 2023.

[7] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21, 1982.

[8] E. Grumbling and M Horowitz. *Quantum Computing: Progress and Prospects*. The National Academies Press, Washington, DC, 2019.

[9] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[10] Jack D. Hidary. *Quantum Computing: An Applied Approach.* Springer, Palo Alto, CA, USA, 2021.

[11] Jack D. Hidary. *Quantum Computing: An Applied Approach.* Springer, Palo Alto, CA, USA, 2021.

[12] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem, 1995.

[13] Randall J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations.* Society for Industrial and Applied Mathematics, 2007.

[14] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.

[15] Thiab R Taha and Mark I Ablowitz. Analytical and numerical aspects of certain nonlinear evolution equations. ii. numerical, nonlinear schrödinger equation. *Journal of Computational Physics*, 55(2):203–230, 1984.

[16] Thomas J. W. *Numerical Partial Differential Equations: Finite Difference Methods.* Springer New York, NY, 1995.