RESOURCE EVALUATION OF QUANTUM LINEAR SYSTEMS ALGORITHM FOR APPLICATION TO ELECTROMAGNETIC SCATTERING PROBLEMS

THESIS

Casey J.R. Riggs, 2nd Lieutenant, USAF

AFIT-ENV-MS-17-M-216

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENV-MS-17-M-216

RESOURCE EVALUATION OF QUANTUM LINEAR SYSTEMS ALGORITHM FOR APPLICATION TO ELECTROMAGNETIC SCATTERING PROBLEMS

THESIS

Presented to the Faculty

Department of Systems Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Casey J.R. Riggs, BS

2nd Lieutenant, USAF

March 2017

AFIT-ENV-MS-17-M-216

RESOURCE EVALUATION OF QUANTUM LINEAR SYSTEMS ALGORITHM FOR APPLICATION TO ELECTROMAGNETIC SCATTERING PROBLEMS

THESIS

Casey J.R. Riggs, BS

2nd Lieutenant, USAF

Committee Membership:

Major Logan O. Mailloux
Co-Chair

Major Charlton D. Lewis
Co-Chair

Dr. Ojas D. Parekh
Member

Abstract

Current limitations in quantum computing technology do not allow for very large applications of quantum algorithms, and it is the nature of quantum algorithms not only to be able to solve problems of interest much more quickly than classical means but also to do so with less resources which makes them so promising. One such problem of interest is the application of the Quantum Linear Systems Algorithm, along with a few other subroutines, to the calculation of an electromagnetic scattering cross-section via finite element methods. This work composes a resource analysis of the algorithm as well as required subroutines. Additionally, this work details the primary contributors to the resources involved as well as methods to decrease these resource requirements. The particular problem of interest to this work is the EM scattering of an aerodynamic cone within a square computational finite element region of 50×50 to 400×400 grid points. The mesh sizes resulted in a range of resource requirement from 453 to 477 logical qubits. However, varying the desired bit precision independently between 8-bit to 128-bit created resource requirements from 133 to 853 logical qubits. The desired precision of the calculations created a much larger effect on the resource requirement of the application of the algorithm.

What is best in life?

"Crush your enemies, see them driven before you, and to hear the lamentation of their women!"

- Conan the Barbarian

Table of Contents

Page

List of Figures

Page

List of Tables

Page

RESOURCE EVALUATION OF QUANTUM LINEAR SYSTEMS ALGORITHM FOR APPLICATION TO ELECTROMAGNETIC SCATTERING PROBLEMS

I. Introduction

Background

Understanding the application of quantum algorithms requires knowledge of both quantum mechanical systems and computationally intensive problems. An example of both the necessary quantum mechanical properties and a computationally intensive problem (factoring) are presented in [1]. Described in [1] are basic quantum phenomena including superposition, entanglement, and measurement as well the application of Shor's algorithm for factoring. More specifically, the computationally intensive process of breaking RSA encryption is a difficult problem, and this problem can be solved easily via Shor's factoring algorithm [2], which utilizes quantum algorithm subroutines.

The field of quantum computation is primarily dominated by a subset of theoretical physicists, mathematicians, and computer scientists and combined with the reality that universal quantum computers of applicable size are not readily available, it is not surprising that quantum computation has not garnered more widespread attention. Recent publicity surrounding the applicability of Shor's algorithm, in a modern environment that is conscious of security, highlights one of the many advantages of quantum computers.

Quantum algorithms take advantage of the superposition of states provided by quantum mechanical systems in order to process information. Measuring a quantum state

results in a singular value, which does not utilize the full capability of quantum computing. Extracting a useful property rather than a set of all the different values contained in the wave function is often more useful—so as to take advantage of the parallel processing ability of quantum computers. For example, extracting the period of a modular function as in Shor's factoring algorithm, rather than specific modular exponents.

Much of the mathematical discussion following the proofs and the explanations of quantum algorithms may be difficult to manage, especially if one does not have an extensive mathematical background in linear systems, eigenvalues/eigenvectors, Hilbert spaces, etc. Although many of the quantum algorithms currently known can be decomposed into a more readable fashion and even heuristic examples, (there are several sources for learning about this [3] [4] [5] [6]) this is still a difficult field to enter. It is the purpose of the conference paper in this work [1] to present a viable introduction for those unaccustomed to the mathematical rigor which is often assumed of those interested in this field.

Large and computationally intensive problems often rely on solving large systems of linear equations. For example, computational fluid dynamics, finite element analysis of structures, protein folding, and electromagnetic scattering cross sections all rely on solving linear systems of equations. Although each individual problem may have its own alternate steps and nuances, the most computationally intensive part is the inversion of a large matrix in order to solve the linear system. The field of space complexity or resource analysis in quantum computing is often ignored. While the space complexity of quantum

algorithms is generally given as big "O" estimates, applications of the algorithms can be deterministically evaluated for logical resources depending on the particular quantum algorithm and associated subroutines. The logical resource evaluation is important to the field of quantum computing because currently only very small (i.e. consisting of a few qubits) "universal" quantum computers exist. The size and scalability of these quantum computers is progressing and the computation of useful problems is closer.

Theoretically, with sufficient resources, there exists a range of mathematical problems which can be solved efficiently on universal quantum computing devices. Built on these mathematical problems are computational problems which can be applicable to real world analysis. The application of such problems, such as factoring, to the security of the RSA encryption scheme is where interest in the field of quantum computing really takes hold.

Problem Statement

The topic of this thesis is the resource analysis of the application of a specific quantum algorithm for solving linear systems of equations known as the Quantum Linear System Algorithm (QLSA) originally presented in [7]. This application along with other quantum subroutines can be used to calculate the radar cross-section of a 2-D body using a Finite Element Method (FEM). A resource analysis of the QLSA for a RCS calculation has only been evaluated for a simple 2-D square [8]. This evaluation was incomplete and further study is necessary into the space complexity of the application.

Research Objectives & Investigative Questions

The three investigative questions of this thesis are:

1. What quantum phenomena are necessary to understand in order to study quantum algorithms?
2. How would one prepare an implementation of the QLSA for EM scattering?
3. What resources are required for using the QLSA for RCS?

This research was motivated by the interest of the United States Air Force (USAF) in the application of quantum computer technology to the design process of complex systems [9]. The objective of this research is to generalize the resources required to implement the QLSA for this particular problem, notably the sizes and utilization of quantum registers. In the process of understanding this problem a brief foray into how one should approach this complex field is also developed.

Methodology Overview

The nature of technological progress at the time of writing dictates that this analysis be a purely theoretical one. Current quantum computing technology does not yet exist to construct large enough quantum circuits to run all the required quantum algorithms in order to properly apply the QLSA to the RCS problem—including the separate subroutines required for non-trivial problems. The process is verified mathematically, and the reliability of the quantum algorithms are based on the assumption of fault-tolerant quantum bits and gates. The reliability of the algorithm as a

whole is taken into account with regard to the quantum nature of the algorithm for the specified problem in order to create a precise approximation.

The pre-existing raw data from the Finite Element Analysis (FEA) of a 2-D aerodynamic cone of a classical computation is used in the resource analysis of a potential quantum computation. More specifically, attributes of that data are used in the calculation of the resource requirements. An effort to minimize the resource requirements is conducted through the evaluation of both sequential and parallel computation of repetitive quantum subroutines. The resource analysis is then generalized to a form more easily applicable to various problem sizes and parameters.

Assumptions & Limitations

The field of quantum computing is quite broad in scope. The theoretical study of the application of quantum mechanics to real-world problems is ongoing and the challenge of engineering reliable and scalable quantum systems is progressing rapidly. The theoretical application of quantum algorithms to real-world problems is a very small and specific field of study, and largely ignored are the constraints on the resources involved. The quantum computation community operates under the assumption that the necessary quantum mechanical resources will be available in time, and is largely focused on speeding up the time complexity of the algorithms.

In order to extract resource requirements an understanding of how the algorithms operate and specifically the notation of quantum circuit diagrams is required. These circuit diagrams are extremely useful yet they themselves often leave out important details. The generalized QLSA algorithm along with the necessary subroutines which

complete the specific application to RCS's have been proposed in [10], and the complex issue of resource analysis has been attempted once [8] to the knowledge of the author.

The issue of resource analysis varies according to the characteristics of the problem and the desired precision. This work attempts to bridge the gap between an earlier attempt at resource analysis [8] and a more complete analysis of a generalized 2-D problem, via an example using the electromagnetic scattering of an aerodynamic cone.

The effort to construct a reliable generalized resource evaluation is two-fold. On the one hand the resource evaluation provides a goal for which to strive in the short-term for solving real RCS problems; on the other hand, such resource estimates may prove to motivate more research into the quantum computation field.

Impact of Research

The implication of a generalized resource requirement for the application of QLSA to RCS is that more realistic estimates can be generated for arbitrary electromagnetic scattering problems. The resource evaluation of complex quantum algorithms can be introduced and lead to optimization of quantum resources for running these algorithms. This work hinges on critical assumptions such as the efficient construction of particular oracles (an oracle is a term which is used to describe a specific black-box function used in a quantum subroutines), which themselves may incur extra resource costs as research progresses in finding efficient means to implement them. The time complexity of the algorithm, while noted, is not the focus of this work—a stark contrast to most works in quantum computing at the moment.

Organization of Thesis

This thesis is composed first of an abstract, summarizing the efforts and results. Next is the introduction, where the relevance and scope of the problem is proposed. Following that is the literature review, where introductory material as well as brief summaries of quantum algorithms are presented. Following this is the first published paper, which is a case study of learning about quantum algorithms, using the infamous Shor's algorithm. The second published paper (pending approval), the heart of the thesis, composes a resource estimate, generalized resource requirements, and an optimization of quantum resources for the particular application of the QLSA to the EM scattering problem. The final section of the paper is the conclusion and answers the research questions as well as proposes areas for future research in this field.

II. Literature Review

Chapter Overview

The purpose of this chapter is to introduce the reader to the field of quantum computing. Quantum computing, without a significant background in mathematics and computer science requires quite a bit of learning. This chapter is written to alleviate the need to sift through many extremely technical papers in order to understand the necessary quantum algorithms and subroutines required for the calculation of RCS's via a quantum computer; it is also written to extract the important details of the quantum algorithms as it relates to the resource calculations.

There are many topics in this chapter, and the intent is to start from the basics of quantum computing and progress all the way through the building of the full algorithm for the problem at hand. First, the basic notions of quantum mechanics are discussed as applied to quantum computations. Then quantum algorithms are introduced and as a prime example the principles of Deutsch's algorithm are presented, and from there one is able to proceed to understanding an algorithm such as Shor's. The first paper included in this thesis is a case study on exactly that, and while a more thorough analysis of the basic concepts involved are included in this chapter, the paper is not a bad place to start in itself.

The field of quantum algorithms specifically requires two areas of knowledge, one of quantum mechanical process, and the other of the computational problem which needs to be solved. This chapter is heavily focused on the quantum algorithms and a brief overview of the radar-cross section calculation is presented. The goal of many of the

applications of quantum algorithms are to reduce the computational complexity in terms of run-time consideration, and for this many applications often boil down to a particular mathematical problem. For example, the calculation of an RCS via FEM boils down to the inversion of a large matrix, which is solved quantum mechanically by the QLSA.

A few extra steps are necessary in the calculation however, and these are approached by various other quantum algorithms. The integration of all these algorithms together has already been done for this application, with a few assumed oracle constructions. From knowledge of how each of the algorithms work and how they are used in conjunction with one another, one can create a more complete resource analysis.

This chapter begins with an explanation of fundamental quantum mechanics principles such as quantum superposition and the phenomena known as "entanglement". Following this are the building blocks of quantum computers—quantum gates and circuits, which then proceeds into a short discussion of quantum error correction and physical requirements. The bulk of the chapter then includes explanations of the quantum subroutines which are necessary for the problem, including the Quantum Phase Estimation Algorithm (QPEA), Quantum Linear Systems Algorithm (QLSA), Quantum Sate Preparation Algorithm (QSPA), a corrected rotation, the Quantum Swap Test, and the Quantum Amplitude Estimation Algorithm (QAEA)--via a more thorough understanding of Quantum Amplitude Amplification (QAA). At the end of the chapter is a brief discussion of the RCS problem set-up.

Superposition and Entanglement

The most basic building block of the quantum computer is the quantum bit, called the "qubit". The qubit is the quantum counterpart to the classical bit. While a classical bit is restricted to existing in one of two states (either a 0 or a 1, off and on respectively), a qubit is a quantum-mechanical system that exists in a linear superposition of states (a continuum between 0 and 1). A visual representation of the quantum bit is the Bloch sphere:

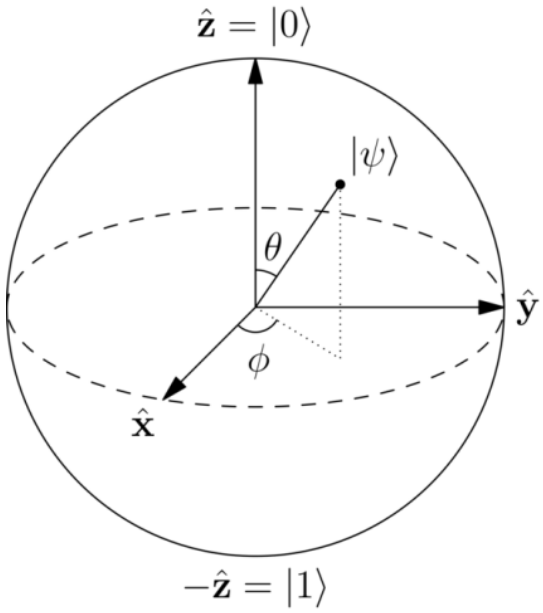


Figure 1. A Bloch sphere, the state of the quantum bit is represented as $|\psi\rangle$.

Notice how the qubit not only varies by the rotation angle θ but also the phase angle ϕ. Denoted at the top and bottom of the Bloch sphere are the two orthonormal basis vectors, or the measurement basis, $|0\rangle$ and $|1\rangle$. After a measurement it is important to note that the qubit may only collapse into one of these two measurement basis and thus loses any superposition it had before the measurement. The superposition essentially

contains the relational information of the system and measurements are often taken at the last possible opportunity to best utilize this unique property of the quantum mechanical system.

The $|\psi\rangle$ vector is called the wave vector of the quantum state. The state of an individual qubit in a perfect superposition of states $|0\rangle$ and $|1\rangle$ (represented by a wave function with $\theta = 90°$ and $\phi = 0°$) is mathematically written as:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad \text{Eq. 2.1}$$

The normalization term $\frac{1}{\sqrt{2}}$ is used to denote the amplitude of the wave vector. Because the probability of finding a quantum system in a particular state is the square of the amplitude, this wave vector acts as an interpretation of the quantum superposition of a qubit which has equal probability ($P = \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}$) of being measured as either a $|0\rangle$ or $|1\rangle$. Mathematically the state is represented as a linear superposition of both the basis states $|0\rangle$ and $|1\rangle$.

At this point it is important to discuss the quantum phenomena known as "entanglement". Entanglement is the "spooky action at a distance" as Einstein put it [11], meaning that when one quantum bit is entangled with another the actions taken on either qubit affect both simultaneously. Although the propagation of this entanglement interaction is not completely understood, for the practical consideration of this thesis, the qubits will interact instantaneously in the quantum systems.

The entanglement of multiple bits to each other can be extremely useful, and in fact forms a foundation for much of quantum computation. Being able to operate within

the space provided by having multiple bits entangled is what gives quantum computation such an advantage over classical means.

If a register contains 3 qubits, all in a perfect superposition, the quantum state can be expanded as:

$$|\psi\rangle = \frac{1}{\sqrt{2^3}} \sum_{x=0}^{2^3-1} |x\rangle \qquad \text{Eq. 2.2}$$

In this wave vector, the normalization constant $\frac{1}{\sqrt{2^3}}$ gives equal amplitude to each of the possible states in the vector. The states of the vector are all the possible combinations of those 3 qubits:

$$|x\rangle = (|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \qquad \text{Eq. 2.3}$$

The summation of these states satisfies the quantum mechanical expansion signifying a superposition with one another. A more general expansion of a register containing n qubits in superposition may be represented as:

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \qquad \text{Eq. 2.4}$$

Where each state in the superposition has an amplitude of $\frac{1}{\sqrt{2^n}}$ and thus a probability of measurement of $\frac{1}{2^n}$. For this basic case, the quantum register exists as all the possible solutions; however, it is the aim of the quantum algorithms to specifically reduce this quantum state to the particular values of interest. In order to do this, the quantum state needs to be operated on.

Gates and Quantum Circuits

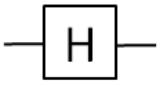
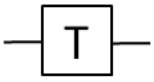
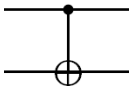
Quantum gates are the circuit notation for unitary operations on quantum states. These gates can follow one another sequentially on a specific qubit or in parallel on different registers to constitute a quantum circuit which can implement interesting algorithms on quantum bits. It is possible to construct complex unitary operations out of a small number of single and double qubit gates according to the Solovay-Kitaev theorem (for a more detailed description of this theorem see [3]). One such family of unitary quantum gates includes the Hadamard gate (H), the $\pi/8$ gate ($R(\pi/4)$), and the controlled-not ($CNOT$) gate. These three gates constitute a family in that they can be used in combination to simulate any unitary quantum operation. It is important to note that larger, more complex operations such as the Toffoli gate and the Fredkin gate can be decomposed into these two-qubit operations. This allows the focus of quantum computing to narrow its attention on creating a few reliable quantum operations rather than constructing an infinite number of large multi-qubit operations.

Although the construction of large unitary operations can be efficiently decomposed into these smaller operations, there may yet be better optimization schemes for creating these large operations with even fewer gates. This plays a large part in the runtime considerations for algorithms such as the QLSA or specifically those involving oracles and Hamiltonian simulation. The current methods for constructing these quantum algorithms require further study and more research into optimization. One attempt at optimizing the construction of larger unitary operations was taken in [12] in the

application of the Group Leader Optimization Theorem to the construction of an efficient quantum circuit.

These single and double qubit operators are represented as matrices. The quantum state being represented as wave function can then be easily manipulated by these matrix operators. For example, the matrix depiction of three operator family mentioned before is presented in Table 1.

Table 1. A family consisting of two 1-qubit gates and a 2-qubit gate

	Hadamard Gate	$\pi/8$ Gate $= R(\pi/4)$	CNOT Gate
Symbol			
Matrix representation	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$	$\begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}$	$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$

Although these are just a few examples, there are many other single, double, and multi-qubit operators [3]. As an example, the construction of one of the most fundamental quantum operations, the Quantum Fourier Transform (QFT), can be decomposed into single and double qubit operations as shown by a simple QFT on a 3-qubit register:

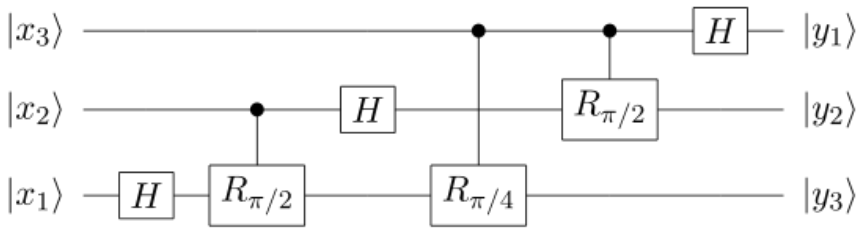


Figure 2. A simple QFT on a 3-qubit register [3].

The QFT is akin to the discrete Fourier transform in classical notation. The mathematical representation of the effect of the QFT on an input is:

$$QFT\ |j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k\rangle \qquad \text{Eq. 2.5}$$

The QFT makes a spectacular appearance in Shor's algorithm [2] and is a key component in the QLSA. Specifically, the QFT plays a major part in the QPEA as well as the QAEA components of the QSLA as applied to the EM scattering problem. Discussion of the particular quantum subroutines applicable to the problem at hand is discussed in more detail in later sections.

Figure 2 is an example of the quantum circuit diagram notation for used commonly in quantum algorithms. The figure is read left to right, beginning with the input register denoted by the Dirac notation on the left (sometimes for conciseness, multi-qubit registers are represented by a single line with a slash at the beginning). The operators are shown as boxes (such as those presented in Table 1) and perform their respective operations on the qubit on which line they sit, sometimes controlled by the values in another quantum register as denoted by the line with the solid dot (e.g. a CNOT gate, or a controlled rotation).

Quantum Bits and Physical Implementation/Quantum Error Correction

There are currently many avenues being pursued for the physical realization of reliable quantum bits. Some of these include polarized photons, trapped ions, electrons, superconducting materials, and atomic nuclei [3]. Largely, singling out individual particles and keeping them in a highly controlled environment creates a hard problem for engineers. These quantum bits must be kept in such controlled conditions because qubits are extremely sensitive to outside interference such as electromagnetic waves, variation in temperature, and light. In theory, these systems will become reliable enough in the future and will be scalable to larger registers of quantum bits. The intersection of these two fields is called Quantum Error Correction (QEC) and is composed of the techniques which seek to control quantum bits in an effort to make them more reliable.

An approach to insuring the reliability of a quantum bits is to construct a larger entity known as a logical qubit. Such logical qubits may be composed of many individual qubits. In a logical qubit, all the qubits align to the same state. This method, combined with QEC algorithms can be used to more reliably create quantum bit registers [13]. However, this does drive the technological requirements up significantly for many quantum algorithms. While simple laboratory tests of small quantum algorithms may be able to utilize a miniscule number of quantum bits [14] [15] [16], larger experiments and certainly applied quantum algorithms will need more sufficient error correction and thus may require logical qubits composed of several qubits themselves.

While error correction in a standard computer may involve a “voting” process, qubits must be treated differently. The quantum mechanical nature of the superposition of

the qubit must not be disturbed however the qubits need to conform to one another within the logical qubit. There are two types of errors which can affect the quantum state of a logical qubit. One is the bit flip and the other is a sign flip. Alternatively, these could be looked at as disturbed rotation or phase respectively of the quantum state. Simple examples of quantum error correction algorithms to fix these issues utilizing a logical qubit of 3 individual qubits are presented in Figures 3 and 4.

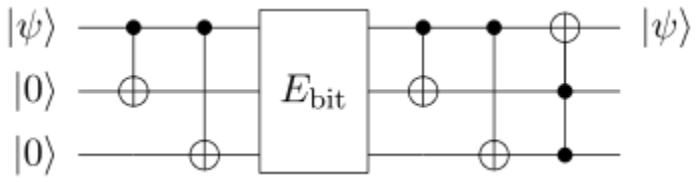


Figure 3. Bit flip QEC circuit [3].

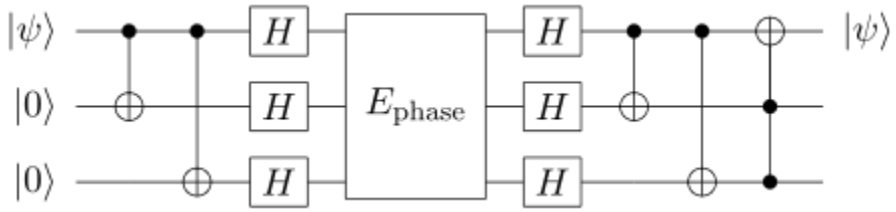


Figure 4. Sign flip QEC circuit [3].

These QEC algorithms maintain the state of the logical qubit by performing operations on the individual qubits in order to ensure conformity. Integration of both of these QEC algorithms into a singular QEC algorithm utilizing nine individual qubits is known as the Shor code [13] and is presented in Figure 5. The Shor code can correct for a bit flip, sign flip, or both within the logical entity.

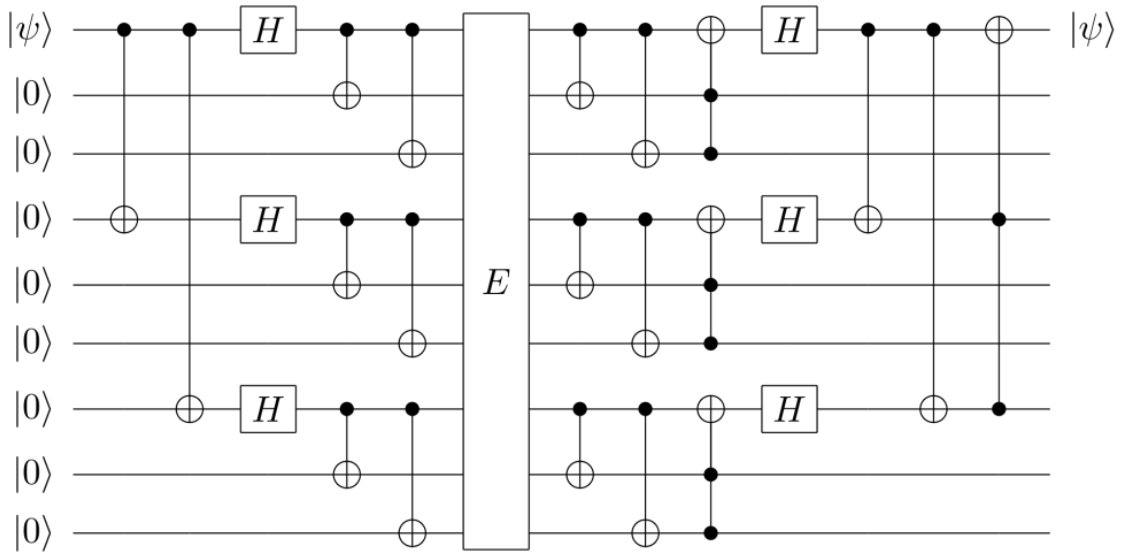


Figure 5. Shor code QEC circuit diagram.

While these simple algorithms represent robust approaches to QEC they expand the resource requirement significantly (in the case of the Shor code the required qubit resources grows by a factor of 9). Further study is needed in this field to generate more reliable, less resource intensive logical qubits.

The field of QEC also includes the operations themselves on the qubits. Such fault tolerant quantum gates are also being worked toward with incredible efficiency [17]. The minimum efficiency of the quantum gate is quite high for reliable operations, on the order of 99.9%; however, when viewing complex operations which may contain millions or billions of operations, such efficiency is required to prevent unnecessary iteration when using the algorithms.

Quantum Subroutines

Involved in the process of solving a linear system of equations quantum mechanically are many quantum subroutines. Larger research efforts have gone into refining these individual subroutines in order to make them more efficient as well as evolve them in parallel with current technological standards. Originally, some components of these quantum subroutines were judged "black-box" operations, which meant that although there was not an intuitive solution for these operations, one would come eventually.

The discussion of the specific quantum subroutines will start with the QPEA (Quantum Phase Estimation Algorithm) subroutine, which is vital to the function of the QLSA. Putting these together will ultimately create the foundation for the QLSA. Expanding on one of the assumed conditions of the original QLSA is a discussion of certain preconditioning steps proposed by Clader [10], specifically the QSPA (Quantum State Preparation Algorithm). The quantum swap test is discussed briefly following the QSPA and the background of the QAEA (Quantum Amplitude Estimation Algorithm) is last, in order to properly prepare for the application of the QLSA to the EM scattering problem.

Quantum Phase Estimation

The QPEA uniquely identifies the eigenphase of an eigenvector of a particular unitary, also known as the Abelian stabilizer problem. For the purposes of application, the QPEA will be used to extract the eigenvalues from the simulated Hermitian matrix A in an effort to invert the matrix. Although the algorithm is essentially introduced in

Shor's algorithm for quantum factoring [2], the generalization of the algorithm proper is presented by Kitaev in [18].

If the RCS matrix A can be conditioned well enough to be a sufficiently sparse Hermitian matrix, then it can be applied as a unitary operator in the form of e^{iAt} in a process known as Hamiltonian simulation. There exists a method to force the matrix into a Hermitian form outlined in [7]. The Hamiltonian simulation process paired with the application of the inverse quantum Fourier transform composes the main components of the phase estimation algorithm. The general quantum circuit diagram which illustrates the process is presented in Figure 6.

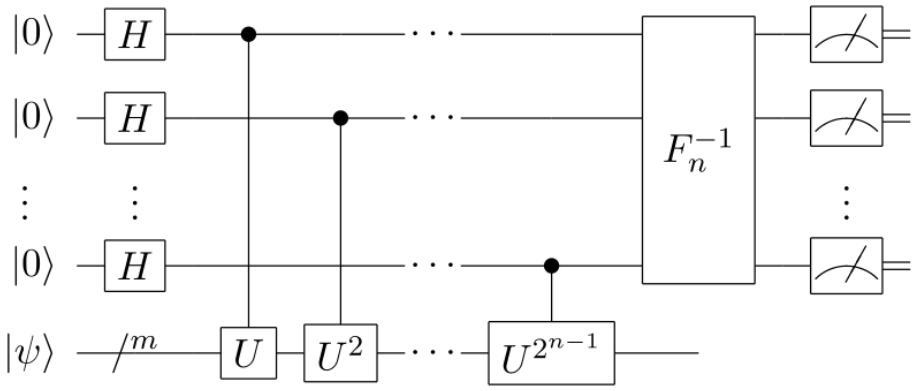


Figure 6. The quantum phase estimation circuit.

The function of unitary simulation and inverse Fourier transform may not be intuitively obvious, however a detailed explanation is given in [3]. The phase estimation process is summarized in Table 2.

Table 2. Summary of quantum phase estimation.

Steps	State of the system	Description	Assumptions
1.	$\|0\rangle\|b\rangle$	Initial state	State $\|b\rangle$ can be efficiently prepared
2.	$\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}\|j\rangle\|b\rangle$	After Hadamard gates ($\|j\rangle$ is superposition state)	
3.	$\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}\|j\rangle U^j\|b\rangle$ $= \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1} e^{2\pi i j\varphi_b}\|j\rangle\|b\rangle$	Black-box simulation of [A] matrix	[A] is Hamiltonian, sparse, and effectively row-computable
4.	$\|\widetilde{\varphi_b}\rangle\|b\rangle$	After the application of the Inverse Fourier Transform	
5.	$\widetilde{\varphi_b}$	Measurement of first register	

The process used to estimate the eigenvalues of the matrix A with the eigenvector $|b\rangle$ is the first step in the inversion process of the matrix. In order to complete the inversion of the matrix, the eigenvalues need to be inverted. Because the eigenvalues are contained in a quantum state, they need to be inverted quantum mechanically, which is not trivial. What follows involves a rotation about the approximate inverse of the eigenvalues and is discussed in greater detail later, for now the simple concept of the rotation will be sufficient. In order to effectively simulate A a quantum oracle is required

to determine the magnitudes and phases of the values contained in A and is discussed next.

Generalized Quantum Simulation Oracle

The oracle used to simulate the unitary operator U in the phase estimation was abstracted to be able to simulate any unitary matrix as an operation. The simulation of the A matrix is used specifically in the QLSA [7] to estimate the eigenvalues of the A matrix. The oracle used to implement the simulation is given by [10] in the supplementary material (based on [19]). In general, the method is used to simulate a given Hermitian matrix.

Given a particular Hermitian matrix A, which can be subdivided into c 1-sparse sub-matrices, as well as two specific unitaries: one to calculate the magnitude and one to calculate the phase of the particular elements of the 1-sparse sub-matrix A_c, one can perform the operations to specifically simulate A according to Equation 6.

$$e^{-iA_c t}|a,0,0,0\rangle = cos(x_c(a)t)|a,0,0,0\rangle - isin(x_c(a)t)e^{i\phi_c(a)}|v_c(a),0,0,0\rangle \qquad \text{Eq. 2.6}$$

This algorithm contains both operators given above, a phase shift operation which utilizes a spare ancilla, and a swap operation between the first two registers. The quantum circuit notation in Figure 7 summarizes the oracle.

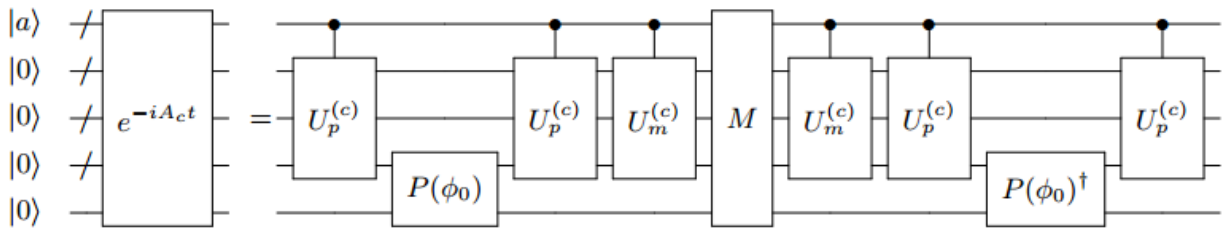


Figure 7. Circuit diagram of Oracle for Hamiltonian Simulation [10].

This particular construction may not be ideal, however further research into the field of more efficiently simulating Hamiltonians is ongoing. Notably, efforts have been made to reduce the computational complexity of the operation in [20] [21] and [22]. The implementation of the oracle in this way adds several more registers to the logical requirement of the algorithm. More details on the construction of the oracle and the specific unitaries used in its construction can be found in the supplementary material of [10].

Quantum Linear Systems Algorithm

The original design for the QLSA proposed in [7] intended to solve a system of linear equations in the form $Ax = b$ quantum mechanically. The new system of linear equations can be summarized by $A|x\rangle = |b\rangle$, where a given matrix A is a Hermitian $N \times N$ matrix, and $|x\rangle$ and $|b\rangle$ are vectors in a Hilbert space represented by a quantum superposition of values. The QLSA effectively inverts the matrix A, thereby creating the solution $|x\rangle = A^{-1}|b\rangle$. In this way the solution $|x\rangle$ is represented quantum mechanically and reading out every individual value would require at a minimum N iterations, which defeats the speedup of the algorithm; therefore, it is more useful to extract some

expectation value from the solution rather than each individual value contained in the quantum state.

The original design of the QLSA involved the use of the QAA (Quantum Amplitude Amplification) to ensure with better probability the measurement of a $|1\rangle$ in an ancilla register indicating that the inversion had successfully taken place. This post-selection measurement of the ancilla register requires the algorithm be run multiple times to acquire enough data to make a statistical analysis of how many times the amplification engine in the QAA needs to be applied. Clader proposed a solution to this problem by eliminating the use of the QAA and instead using the QAEA to deterministically evaluate the success probabilities of successful ancilla measurements which then factor into the calculation of the RCS value later [10].

The QLSA has been applied to several areas of study in quantum computing such as quantum machine learning [23] [24] [25] [26] [27], least-squares curve fitting [28], solving linear systems of differential equations [29] [30], estimating resistance of electrical networks [31], and solving Toeplitz systems [32]. A modern compendium of quantum algorithms and their recent developments are kept online at http://math.nist.gov/quantum/zoo/. Small experimental systems showcasing the QLSA are presented in [14] [15] [16].

It is the function of the conjugate gradient method—more thoroughly described in [33], to solve the system of equations containing these large matrices through the inversion of the A matrix classically in an iterative process. The development of the

QLSA was spurned by the applicable nature of the system of linear equations problem as well as the possibility of a speedup utilizing quantum systems.

The QLSA takes advantage of the ability to quantum mechanically invert a matrix via finding the eigenvalues by phase estimation and inverting those values. The solution state $|x\rangle$ however is not the same as the solution matrix in the classical process. Each individual solution in the solution state is contained as an amplitude of the wave function.

For small experimental demonstrations such as [15] a simplistic model of the QLSA is sufficient:

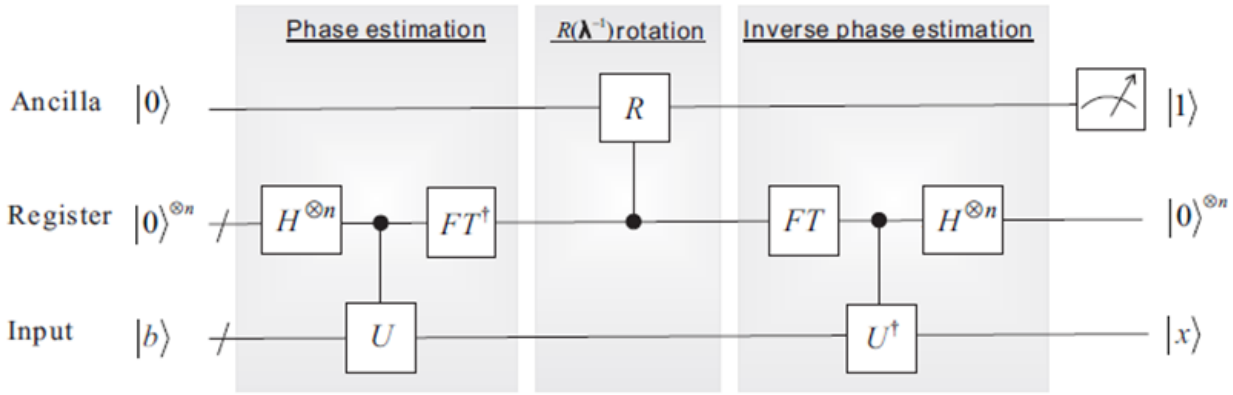


Figure 8. Stages of QLSA.

The simple formulation of the QLSA needs only three registers. The first is an ancillary qubit on which the rotation about the inverse of the eigenvalue will be performed. The second register is the workspace which needs to be able to hold the eigenvalues in superposition, and the third register is the quantum state of the eigenvector. The preparation of the quantum state of the eigenvector $|b\rangle$ is assumed to be

an efficient black-box procedure, which is addressed in the discussion of quantum state preparation.

The QLSA first proceeds through the QPEA, preparing the quantum state of the eigenvalues of the matrix A. These eigenvalues are contained in the second register while the third register remains in the state containing the eigenvector. After the QPEA, the total quantum state of the system is:

$$|\psi\rangle = |0\rangle|\tilde{\lambda}\rangle|b\rangle \qquad \text{Eq. 2.7}$$

The ancilla needs to be rotated about the value of the inverse of the eigenvalues. Smaller experiments provide easily calculated, thus known eigenvalues, and therefore constructing the rotation is much easier. For larger problems, unknown eigenvalues will complicate the process.

A method to perform the rotation about the inverse of the eigenvalues is presented in [34] and plays a role in the addition of a quantum register to the QLSA. The rotation is performed conditional on the probability of the rotation bit being measured in the state $|1\rangle$. Mathematically this is represented by the form:

$$|\psi\rangle = \sqrt{1 - \frac{C^2}{\lambda_j^2}}|0\rangle + \frac{C}{\lambda_j}|1\rangle \qquad \text{Eq. 2.8}$$

The wave vector in this form represents only the state of the rotation bit, and if the bit is measured in the $|1\rangle$ state, then the intended rotation (inverting the eigenvalues) has happened successfully.

Using a rotation controlled by the inverse of the eigenvalues the state of the system becomes:

$$|\psi\rangle = \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}}|0\rangle + \frac{C}{\lambda_j}|1\rangle\right)|\hat{\lambda}\rangle|b\rangle \qquad \text{Eq. 2.9}$$

The conditional about the ancillary qubit has been applied to the system, enforcing the condition that when the ancilla is in the $|1\rangle$ state, the correct rotation has been applied. In this way it is possible to force this condition with a measurement of the ancillary register, which occurs at the end of the original design of the QLSA.

The inverse QPEA part of the algorithm—or the "uncomputation", resets the second register containing the eigenvalues back to an initial state so that the whole quantum system returns to:

$$|\psi\rangle = \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}}|0\rangle + \frac{C}{\lambda_j}|1\rangle\right)|0\rangle|b\rangle \qquad \text{Eq. 2.10}$$

After the uncomputation, the correct rotation can then be enforced by a measurement of the ancillary register. If the measurement of the ancilla turns out to be $|0\rangle$, the algorithm needs to be repeated until the correct output appears. The repetition of the algorithm can be reduced significantly by the application of QAA (Quantum Amplitude Amplification). However, in the application of the QLSA to solving RCS problems, the QAA and the post-selection measurement of the ancilla register can be neglected yielding a more efficient algorithm utilizing another quantum subroutine—the QAEA.

This suffices for an introduction into the QLSA algorithm, the main algorithm of the matrix inversion process used in the calculation of the RCS value. Although the idea is simplistic, the execution and specifically the constraints on the register sizes drive the resource requirements of successful implementation. There exist assumptions in the

QLSA which need to be addressed as well including state preparation and rotation issues which are complex and require their own additions to the process, thus additional resource requirements to the QLSA.

State Preparation

Clader attempted to remedy a few of the problems associated with the QLSA, one of which being the state preparation of the eigenvector $|b\rangle$ [10]. This same process will also apply to the preparation of the state $|R\rangle$ later when this algorithm is configured for EM scattering. First, it is noteworthy to add that the prepared state will be conditional on a rotation qubit, similar to the inversion in the QLSA. This conditional rotation provides a means to effectively judge the preparation of the quantum state, and later to evaluate the calculated RCS value.

The QSPA proposal does add three new registers, and while theoretically this is unimportant, for the physical implementation this may constitute technological leaps in the development of quantum computer technology. Mathematically, the prepared state will take the form:

$$\left|b\right\rangle = \cos(\phi_b)\left|\hat{b}\right\rangle\left|0\right\rangle\left|0\right\rangle|0\rangle + \sin(\phi_b)|b\rangle|0\rangle|0\rangle|1\rangle \qquad \text{Eq. 2.11}$$

Interpreting this quantum state, there is an associated probability of each state being created successfully, the first associated with some failed creation adjoined to the $|0\rangle$ state in the ancillary register. The second is the successful creation of the state adjoined to the $|1\rangle$ state in the ancillary register. The prepared register is noted first,

while the ancilla register is last; the two other registers are used in the computation of the prepared state and return to their initial states at the end of the QSPA.

The overall process of the state preparation procedure can be summarized in the quantum circuit diagram of Figure 9.

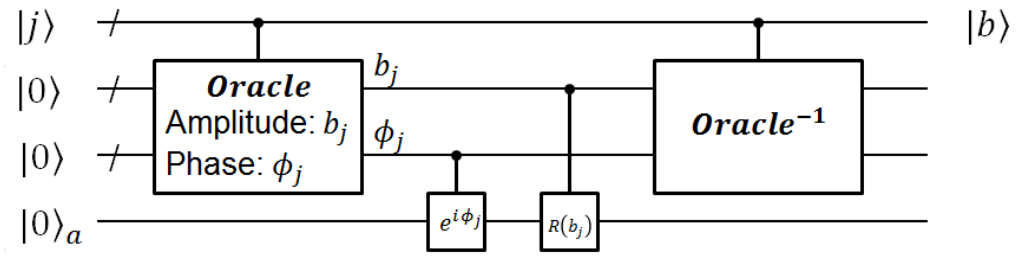


Figure 9. The quantum state preparation subroutine as abstracted from [10].

The desired state at the end of the algorithm is:

$$|b\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} b_j e^{i\phi_j} |j\rangle \qquad \text{Eq. 2.12}$$

The first step in the QSPA is to initialize the four required registers and apply the black-box oracle to the second two registers controlled off the first. This black-box oracle is assumed to be an efficient oracle which can calculate the phase and amplitude components of the desired quantum state $|b\rangle$ from the register initially in the state $|j\rangle$.

The phase and amplitude components, ϕ_j and b_j respectively, are stored in the second and third registers. The second and third registers are then used as a control for a phase shift and a rotation based off the amplitude on the ancillary register. The quantum state at this point in the algorithm is:

$$|\psi\rangle = e^{i\phi_j}|j\rangle|b_j\rangle|\phi_j\rangle\left(\sqrt{1 - C_b^2 b_j^2}|0\rangle + C_b b_j|1\rangle\right) \quad \text{Eq. 2.13}$$

After this the inverse of the oracle is called to uncompute the second and third register. Note that this "uncomputation" is similar to the inverse QPEA in the QLSA. The successful implementation (phase shift and rotations by the oracle) is dependent on the probability of a $|1\rangle$ in the ancillary register:

$$e^{i\phi_j}|j\rangle|0\rangle|0\rangle\left(\sqrt{1 - C_b^2 b_j^2}|0\rangle + C_b b_j|1\rangle\right) \quad \text{Eq. 2.14}$$

The successful implementation of the algorithm yields the approximate preparation of the quantum state $|b\rangle$ in the register originally assigned as $|j\rangle$ with a constant C_b:

$$C_b b_j e^{i\phi_j}|j\rangle|0\rangle|0\rangle|1\rangle = C_b|b\rangle|0\rangle|0\rangle|1\rangle \quad \text{Eq. 2.15}$$

Rotation

The eigenvalues stored in the working register of the QPEA need to be inverted for the matrix inversion to succeed, and the inverse eigenvalues need to be stored in another register so that the rotation about the inverse eigenvalues can be accomplished. Smaller experiments [14] [15] [16] provide easily calculated, thus known eigenvalues, and therefore constructing the rotation is much easier. For larger problems, unknown eigenvalues complicate the process.

A method to accomplish this rotation for larger problems is presented in [34], and the circuit diagram notation is shown in Figure 10. It is important that the uncomputation after the rotation of the QLSA still needs to occur with the newly included operation for the rotation.

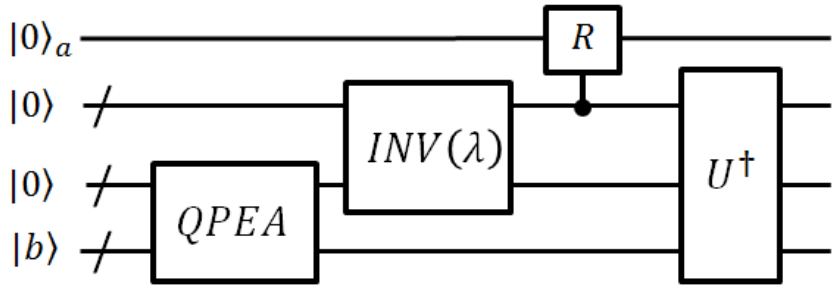


Figure 10. The circuit design for the inversion of the eigenvalues.

The method used to invert the eigenvalues is Newton iteration, and it is this method that requires a particular size of register needed to hold the inverted eigenvalues. The large size of the register needed to hold the inverted eigenvalues is important that it succeeds in the Newton iteration process with at least precision ϵ_{inv}. The classical process of Newton iteration on the binary register (albeit performed quantum mechanically) is the limiting factor in this step of the algorithm.

Swap Test

The quantum swap test, originally introduced as quantum "fingerprinting" in [35] is a test of the similarity of quantum states. The swap test is a conditional swap of two quantum states, akin to a dot product of two geometric vectors. The swap happens with a probability associated with the relative overlap of the two states. The quantum circuit diagram shown in Figure 11. demonstrates the simplistic construct of the swap test.

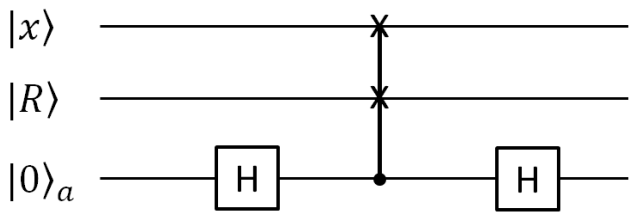


Figure 11. The quantum swap test on two quantum states, $|x\rangle$ and $|R\rangle$.

The quantum swap test starts with a Hadamard operation on an ancilla; this ancilla will be used as the conditional for the swap and will subsequently store the relational data of the overlap if it is not directly measured. The state of the system progresses according to Table 3.

Table 3. The state of the quantum system through the swap test.

Step	States	Description
1.	$\|x\rangle\|R\rangle\|0\rangle$	Initial state
2.	$\|x\rangle\|R\rangle\frac{\|0\rangle+\|1\rangle}{\sqrt{2}}$	After first Hadamard
3.	$\frac{\|x\rangle\|R\rangle\|0\rangle+\|R\rangle\|x\rangle\|1\rangle}{\sqrt{2}}$	After the conditional swap (CSWAP)
4.	$\frac{1}{2}\|0\rangle[\|x\rangle\|R\rangle+\|R\rangle\|x\rangle]+\frac{1}{2}\|1\rangle[\|x\rangle\|R\rangle-\|R\rangle\|x\rangle]$	After second Hadamard

Nominally, the swap test calls for a measurement of the ancilla qubit after the operations to determine whether or not the states were indeed different. If the states are equal, the outcome will be $|0\rangle$ with probability $P = 1$, this is deemed a "pass". If the states are

different the outcome may be either $|0\rangle$ or $|1\rangle$. Because of this, if the outcome is $|1\rangle$ then the states were definitely different, and a $|1\rangle$ in the ancilla is deemed a "fail".

The probability of a pass in the swap test is given by $P = \frac{1+|\langle R|x\rangle|^2}{2}$. The probability of a "fail" in the swap test is given by $P = \frac{1+|\langle x|R\rangle|^2}{2}$. One would need to repeat the measurements enough times to complete a statistical analysis of the results in order to ascertain the amount of overlap between the states, however this would be detrimental to the process of calculating the RCS value, therefore the QAEA is used on the ancilla of the swap test to determine the associated amplitude, thus probability of successfully or unsuccessfully completing the test. From this probability, the value of the overlap (represented in the Maxwell equations by the dot product between R and x) can be determined.

Amplitude Amplification

In an effort the better understand Grover's "amplification engine" used in the QAEA, a discussion of the QAA algorithm will be helpful. Although the original QLSA calls for the use of the QAA on the ancillary qubit to ensure successful inversion, the application of the QLSA to the EM scattering problem does not. However, it is still useful to understand the QAA to better grasp how the QAEA works.

The QAA algorithm is a subroutine used to grow the amplitude of a specific value one wishes to measure. The most common application of this iterative technique is Grover's algorithm, introduced originally in [36], where an unsorted database is quickly searched for a specific result. It is useful to explain this subroutine in terms of Grover's

application and then abstract the method to the more generalized amplitude amplification process.

Probabilistically, by guessing randomly, each guess would have a $a=\frac{1}{N}$ chance at choosing the correct answer, where a is the probability of the correct answer and N is the total number of elements in the database.

The key subroutine involved in Grover's algorithm is known as the amplitude amplification "engine" and this subroutine is characterized in the form of quantum circuit notation as:

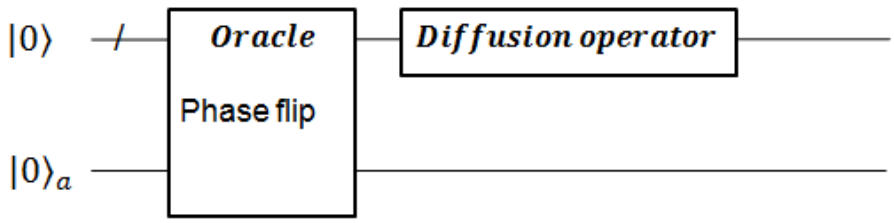


Figure 12. Amplitude amplification engine.

When using Grover's application of the QAA algorithm, the amplitude of the correct answer is grown through a process of inversion about the mean. More specifically, the correct answer is first identified by an oracle and that amplitude's sign is inverted. Represented graphically this makes more intuitive sense. Using a simple example of an unsorted database of 16 elements represented with equal probability of choosing each one (i.e. a four qubit register in perfect superposition), the oracle flips the phase of the desired answer in Figure 13. Note that the phase flip does not affect the answer itself, rather only the amplitude associated with that answer.

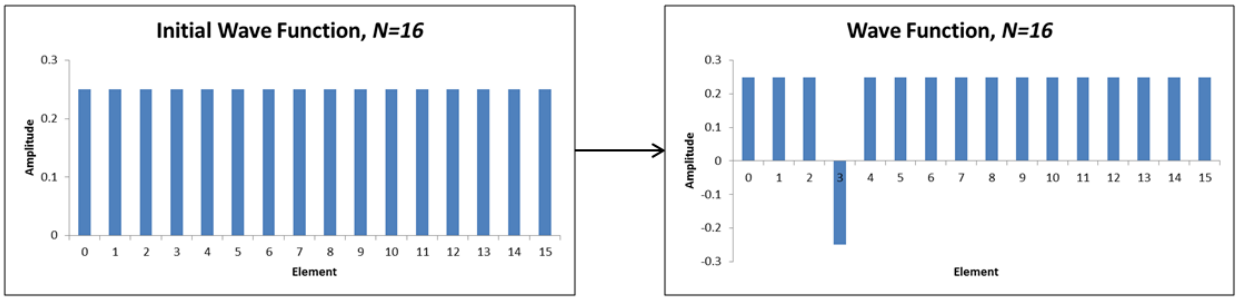


Figure 13. The wave function before and after the oracle.

After the sign flip, the main part of the QAA algorithm inverts the values in the register about the mean, which has been lowered slightly in this case due to the sign flip of the particular value. This inversion about the mean is the most important step because it changes the magnitude of the amplitudes in the register, most importantly, the amplitude of the desired solution (and thus the probability of successfully measuring that answer). This operation is characterized by the "diffusion operator" in Figure 12. Graphically, the iterative requirement of the algorithm becomes clearer:

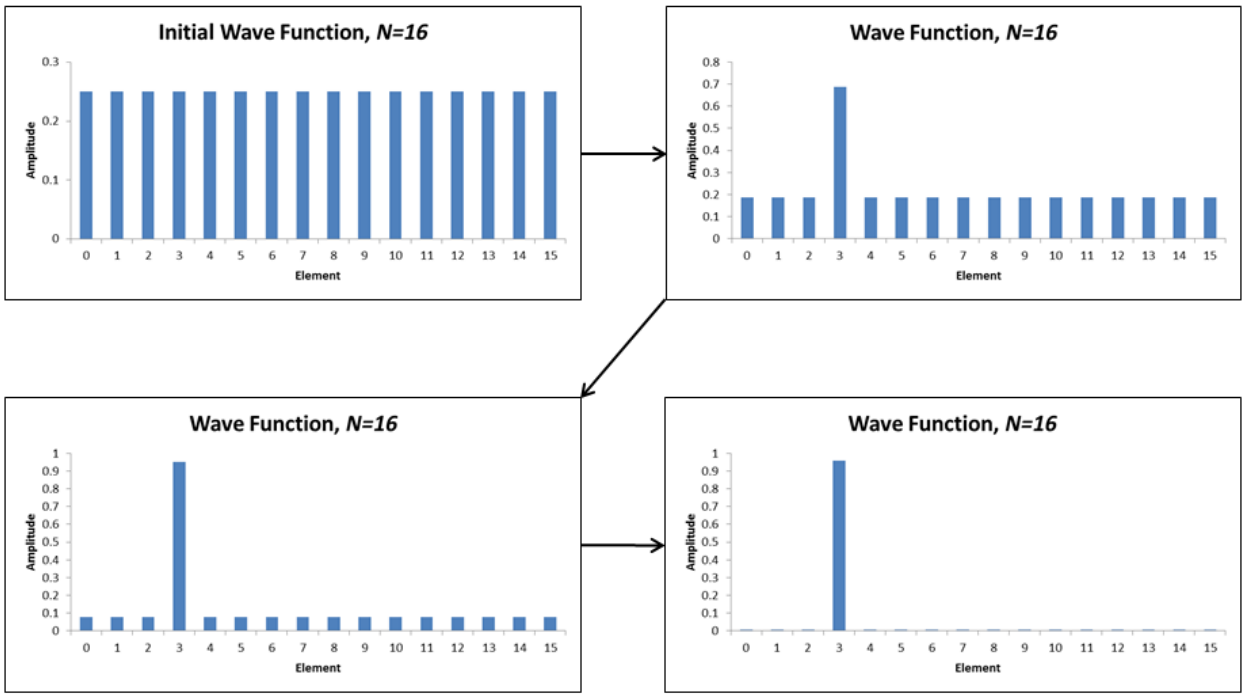


Figure 14. The wave function after each successive application of the amplification "engine".

This process is crucial to the function of the subroutine; it is cyclical—meaning that continuous application of the subroutine will maximize, and then minimize the amplitude of the desired state. This is an important fact for this algorithm, so that there exists a specific number of iterations which are optimal. The cyclic nature of the amplitude amplification is shown by the relative probability (square of the amplitude) of the desired state over successive iterations in Figure 15.

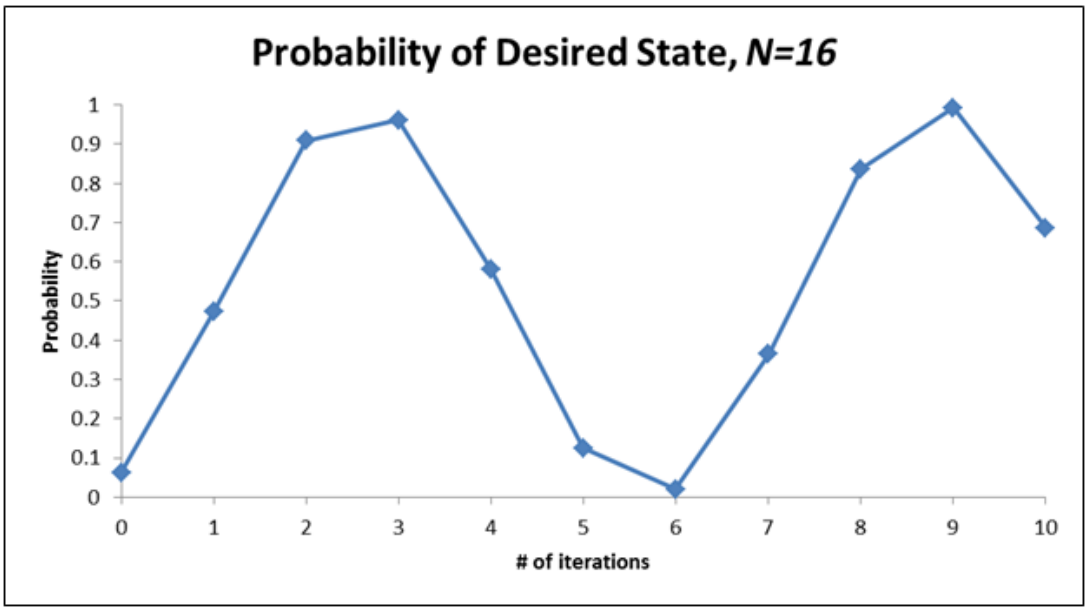


Figure 15. The cyclic nature of the amplitude amplification engine.

The use of this algorithm involves an extra work qubit or "scratch" qubit. This extra ancillary qubit is used for the sign flip process by the oracle and is often left unmentioned in the discussion of the algorithm. While in the application of Grover's algorithm the extra singular qubit may be of little importance, in the construction of the useful application of the QLSA algorithm, the scratch qubit becomes important in the QAEA and constitutes more logical resources.

The QAA algorithm can amplify the probability of success of operations such as state preparation, rotation, and later the swap test of two quantum registers. However, the nature of using the QAA is tied to the measurement of the ancilla registers, which requires iterations of the whole QLSA algorithm and is counter to implementing a faster algorithm. The use of the QAEA does not force one to stop and restart the algorithm if a conditional qubit fails—rather it allows one to determine the probability of the correct

answer forming later in the algorithm—as indicated by the probability of the conditional qubit in a particular "success" state. The probability of a "success" in the conditional qubit is then used in the computation of the RCS value. However, in order to create a statistical confidence interval for the RCS calculation, repetition of the algorithm as a whole is required.

Amplitude Estimation

The amplitude estimation subroutine of [37] follows from the discussion of the QAA algorithm in that the QAEA subroutine utilizes the same iterative engine involved in the QAA. However, this QAEA subroutine is able to apply different numbers of iterations to a specific register "in parallel". By this method the QAEA creates a superposition of all the values of the number of iterations which created good amplitude amplification.

The circuit design for the application of this algorithm introduces another working register with which to store the superposition of the iterations:

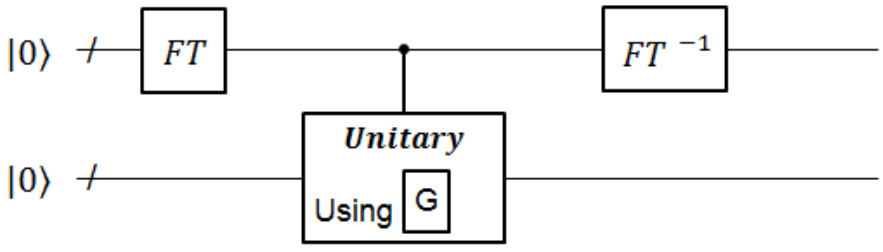


Figure 16. Quantum circuit for amplitude estimation.

The first QFT on the working register in this algorithm may be comprised of Hadamard transforms if the register is of size 2^M, where M represents the Hilbert space of the register, since a single qubit Fourier transform is simply a Walsh-Hadamard transform when the size of the register is a power of 2. Viewed this way the QAEA subroutine is the QPEA subroutine in disguise.

The phase in which this algorithm finds can be decomposed with a little post-processing into a good estimate of the amplitude per Theorem 12 in [37], and the accuracy of this estimation is determined largely by the size of the extra working register. The amplitudes of the quantum state are used later in the calculation of the RCS value, more specifically the amplitudes of the conditional qubits being successful or amplitudes associated with the $|1\rangle$ state.

RCS General Process

The calculation of the RCS value aids in the determination of design parameters for constructing systems intended to have more "stealth", or be less detectable by scattering electromagnetic wave detection systems such as radar. Designs which utilize stealth include drones, planes, ships, vehicles, installations, and many others. The process of designing these systems includes evaluating the current models and making geometric and material changes to maximize the benefits of building stealthier systems.

Included in the optimization of the stealth characteristics is often the computer-aided simulation of the EM scattering, which is far cheaper than building scale models and performing physical tests. These simulated models are commonly composed of large FEM grids which are used to model the EM waves and their interaction with the

boundary of the model. As a result of the FEM expansion and the mesh construction, the A matrix is very sparse, which is a requirement for the QLSA; further decomposition into 1-sparse submatrices is addressed later to ensure speedup of the algorithm.

These grids, and subsequently matrices contain large numbers of data points. In the 2-D example used in [8] the problem size was $N = 332{,}020{,}680$. This problem size corresponds to the number of grid points generated to model the simulation to the desired accuracy. In a classical computation, this would mean the inversion of the $N \times N$ matrix containing millions or billions of values. Notably, the problem size in that example corresponded to a square model and more practical problems may be composed of even larger meshes.

RCS Set-up

There are a number of classical processes which are involved in the set-up of the EM scattering problem, which happen before the quantum algorithm is taken into account. One process includes the creation of the finite element model, specifically the construction of a grid of the model which will be able to produce a sparse A matrix. The QLSA process requires that the matrix elements be efficiently row-computable, limiting the finite element meshes that can be used in the set-up of the problem. The work in [8] used a square mesh to limit the sparsity of the resulting A matrix. By using a rectangular 2-D mesh design, the resulting A matrix was limited to a maximum of seven non-zero elements per row and a total of nine bands.

For the analysis used in this thesis, a similar method is utilized. A 2-D grid of square elements is used for the resource evaluation. The nature of the resulting A matrix

means that in order to use it in the Hamiltonian Simulation, then certain steps detailed later will need to be taken, increasing the size of the matrix and the number of bands, and thus the computational resources required.

The preconditioning of the linear systems proposed by Clader involves the creation of another matrix M which then factors into the general linear systems equation:

$$MAx = Mb \qquad \text{Eq. 2.16}$$

The function of this additional matrix M is to force the A matrix to be better conditioned. This matrix is created by a SPAI (Sparse Approximate Inverse) preconditioner, and although the best M matrix would be A^{-1}, this would essentially solve the problem, so a quickly computable substitute is used instead.

Summary

The current research on the application of the QLSA to the RCS problem specifically is quite limited, particularly it has been approached in two papers to the knowledge of the author: [10] and [8]. The resource analysis of such an application is even smaller, limited to the latter paper. Much research has been completed in the field of quantum algorithms with respect to the QLSA and a proposed modification to the rotation step solves one of the prior assumptions. The QSPA by Clader attempts to solve another assumption of the original QLSA but introduces yet another "black-box" oracle. The swap test is a relatively simple construct, and the QAEA has been approached thoroughly. With all the necessary quantum subroutines in place it is time to proceed to the resource estimation methodology.

III. Methodology

Chapter Overview

This chapter describes the approach to researching quantum algorithms from the standpoint of space complexity. More often, quantum algorithms are researched with computational complexity in mind, and the resource requirements are left in big "O" notation or contained within the error analysis. First, an explanation of where to find relevant information about quantum algorithms is introduced, followed by the acquisition/creation of usable data for the resource estimation. Lastly, the methods used to create the final resource estimation are described which involve the use of resource leveling techniques.

The case study used in this thesis is that of a scalable grid generation for the EM scattering of an aerodynamic cone. Although a complex problem in itself, only part of the problem is used in the resource analysis.

Research Methodology

The method used to approach the resource requirements analysis is the acquisition of the general knowledge of the function of several quantum algorithms to include Simon's algorithm, Shor's algorithm (QPEA), the QLSA, Clader's proposed steps for preconditioning and application [10], quantum eigenvalue inversion, Grover's algorithm (QAA), QAEA, and the swap test (quantum fingerprinting). The understanding of these algorithms is important from a general quantum computation standpoint but also for understanding the critical assumptions for each of the algorithms. Inherent in

understanding the algorithms is the knowledge of often unstated resource requirements such as ancillary qubits and/or determinations of working register sizes based on precision or probability of success.

This knowledge is important in that one doesn't get lost in the promises of exponential speedup without first understanding the limitations. The runtime of the algorithms is necessary to ensure that with the integration of reliable qubits and sufficient error correction a promised speedup is still viable. Experimental research in this area is quite limited as small toy problems are able to shortcut processes that may be needed to be implemented in full in the final construction of an applicable algorithm [38] and successful implementation is the focus rather than scalability.

The logical resource requirement for each of the quantum subroutines is the main component of this work, in understanding the physical requirements for the practical application of the algorithm—more specifically, the application of the QLSA to the calculation of the RCS of a 2-D mesh. This analysis requires the aforementioned intimate knowledge of the quantum algorithms, notably the effect of entanglement on the amount of required resources, and the post-measurement ability to re-use resources for later processes in the overall algorithm.

While the re-use of logical resources is not required for the resource evaluation, it is a critical component in the construction of more optimal estimates and should be used as a common approach to these problems. It is a specific emphasis of this work because of the limited nature of current quantum computer technology and lends itself to practical application.

The determination of the logical resource requirements often stems from the error analysis of particular quantum algorithms. The error analysis is important when working to solve problems with unknown parameters, such as the eigenvalues in the QLSA. The size of the registers is determined by the desired precision, probability of success, and parameters of the problem—in this particular case, characteristics of the A matrix. The A matrix stems directly from the creation of a grid in the FEM process. While the runtime of the algorithm isn't the specific focus of this work, it is worth mentioning that the runtime of the QLSA depends heavily on the characteristics of A such as the condition number κ.

Acquisition of Real Data

The next stage of this work is the acquisition of real data concerning the calculations of an RCS value for a 2-D aerodynamic cone—specifically, the raw data concerning the original systems of equations $Ax = b$. Although the calculation of the data or even the data itself is not necessarily the goal, the metadata inherent are important for the resource analysis. Knowing the values of $|b\rangle_{max}$, $|R\rangle_{max}$, and properties of the A matrix to include d, N_b (number of bands), N, and the desired precision ϵ, as well as the desired probability of success of both the QLSA and the QAEA should one chose to implement the calculation of the RCS value via quantum algorithms is critical.

Resource Evaluation

It is from these variables that reliable size estimates of quantum registers can be constructed which will be able to adequately hold the problem values and gain the desired

precision for the RCS calculation. The resource evaluation progresses from the main algorithm to additional subroutines required for the specific problem and then recombines into a large and complex algorithm which requires many registers of various sizes.

The resource pool needed for the algorithm as a whole can be reduced somewhat when re-use of specific registers is included. Registers that are intermediately measured and thus can be reset, and allow for a smaller overall resource pool for the problem. This is a critical assumption of the problem, as some implementations of quantum bits do not allow for re-use (e.g. photonic systems) and instead require regeneration of qubits. A time-based analysis of the resource pool size and a subsequent verification that the speedup of the algorithm is not affected by decisions regarding processes in parallel or sequence are required. This method constitutes a resource leveling of the available quantum bits over the runtime of the algorithm.

Analysis of Scaling of Problems

A brief foray into the scaling of the RCS problem sizes is also applicable to the algorithms resources. The scaling of the 2-D aerodynamic cone problem with regard to the density of the FEM grid is considered. Larger and much more complex problems such as RCS evaluations of aircraft or ships may also pose challenges for the practical application of quantum algorithms.

Description of Dependent and Independent Variables

There are several variables which play a direct role in the resource estimation including N, ϵ_{amp}, ϵ_{phase}, ϵ_{λ}, ϵ_{inv}, $P_{err\ QLSA}$, ϵ_{QAEA}, a, and $P_{err\ QAEA}$. These variables

have an impact on the size of the registers and are thus included in the estimation; other variables of the problem may have an effect on the time complexity and invertability of the matrix but are not included.

Experimental Design/Description of Data Set and Sources

The example problem used in the resource estimation is that of a grid generated for the EM scattering of an aerodynamic cone. The mesh characteristics include square finite elements which make the application of the QLSA easier via the byproduct of a banded A matrix. The grids used for the scaling of the problem included a 50×50, 100×100, 200×200, and a 400×400 grid. The meshes are square areas with square finite elements. Figures 17-21 show the part of the mesh which overlap the aerodynamic cone; in which increasing the number of grid generation points allows for a more accurate analysis of the EM scattering in both the classical computation as well as the application of the quantum algorithm.

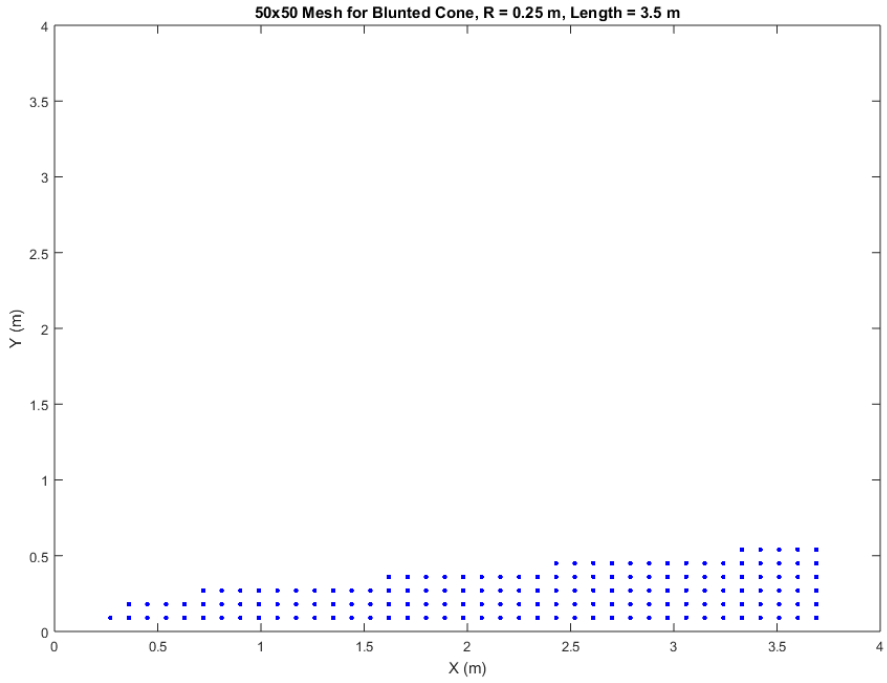


Figure 17. 50 × 50 mesh generation area.

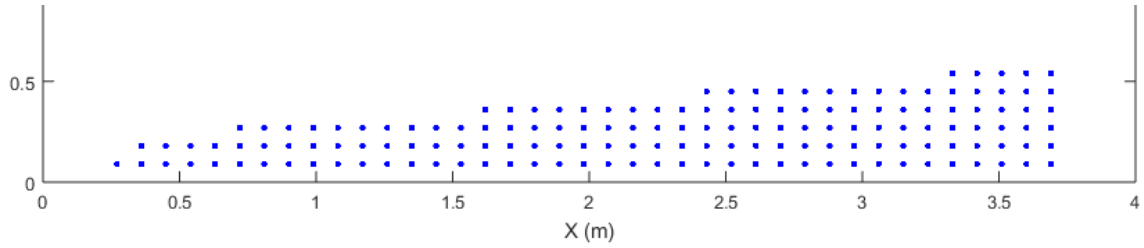


Figure 18. 50 × 50 mesh generation over cone.

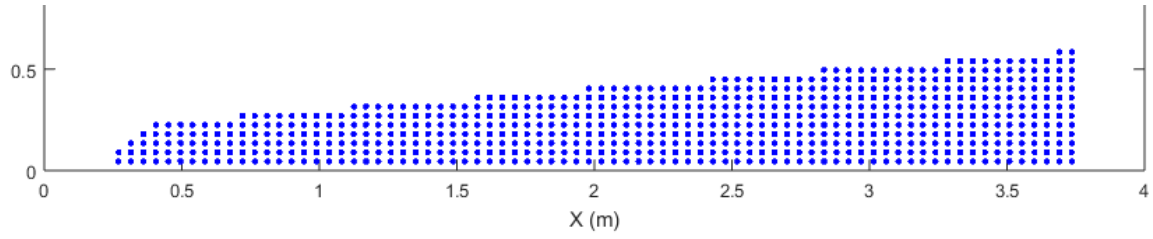


Figure 19. 100 × 100 mesh generation over cone.

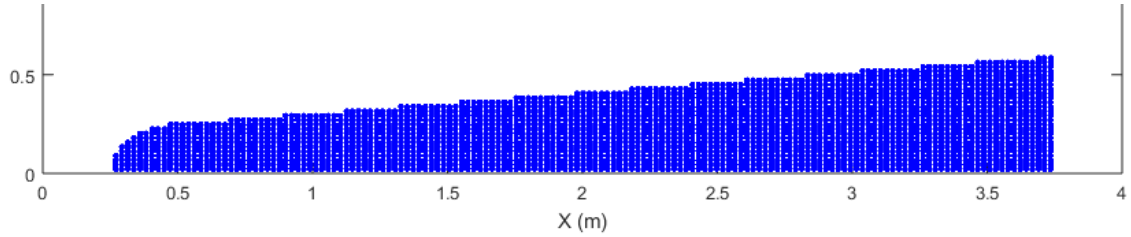


Figure 20. 200×200 mesh generation over cone.

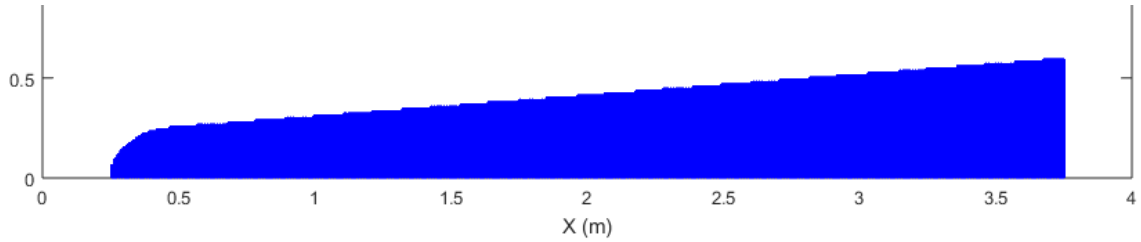


Figure 21. 400×400 mesh generation over cone.

Assumptions

The assumptions made during the resource evaluation are critical to the real world evaluation of such a problem. The quantum bits deemed re-usable excludes such methods for a general quantum computer such as photons – where new qubits need to be generated after measurements. The assumption for the QSPA is that an efficient oracle exists which can compute the required phase and amplitude of desired values. The assumption for the application of the QLSA are that the A matrix be Hermitian, which for a non-Hermitian case can be fixed, and that a quantum state $|b\rangle$ be available. Although the algorithms themselves are scalable for larger problem sizes, the time complexity of the algorithms used may exceed the current lifetime of a coherent qubit.

Description of How to Perform Analyses

As the analysis is the main focus of the work, it is derived from the research of the particular sub-algorithms as well as the problem parameters. Due to the inherent nature of quantum computation, it is not currently feasible to simulate this problem in even the smallest size (the 50×50 grid) with classical resources. The resource analysis is approached through the theoretical construction of the quantum circuits. The analysis also consists of resource leveling in which quantum registers are re-used post-measurement in an effort to reduce the logical resource requirements, which also highlights the main resource intensive sub-algorithms and scaling effects.

Summary

The method of approach is largely a research endeavor into the error analysis of several papers, including those of which the particular sub-algorithms for the problem are used. Understanding the algorithms is necessary in order to evaluate which registers can be re-used as well as the particular size requirements for the quantum registers. Basic research into the construction of a linear system problem for finite method electromagnetic scattering is sufficient for the problem construction. The resource analysis and resource leveling will comprise the main results of the work. The next two chapters in this thesis are an introductory paper into quantum algorithms, namely a case study using the infamous Shor's algorithm and a journal article which comprises the particular resource requirements of the EM scattering for an aerodynamic cone problem.

IV. Conference Paper

Publication Details

Title: Understanding Quantum Computing: A Case Study Using Shor's Algorithm

Publication: Proceedings of the International Conference on Foundations of Computer Science (FCS)

Date: July 2016

The topic of the conference paper is how to approach the basics of quantum algorithms through the illustration of Shor's quantum factoring algorithm. The paper introduces fundamental quantum mechanics as they apply to quantum computing as well as the hard factoring problem which is integral to much of modern public key security. Both are necessary in understanding the application of Shor's algorithm. In the process of creating a resource analysis of quantum algorithms it becomes necessary to understand how the algorithms function and what parameters allow them to succeed; often this includes the understanding of classical computational approaches as well. In the space complexity analysis of quantum algorithms applied to EM scattering, the understanding of many quantum subroutines is vital and is approached in a similar way.

Understanding Quantum Computing: A Case Study Using Shor's Algorithm

Casey J. Riggs, Charlton D. Lewis, Logan O. Mailloux, Michael Grimaila
Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio 45433, United States
{Casey.Riggs, Charlton.Lewis, Logan.Mailloux, Michael.Grimaila}@afit.edu

***Abstract*—Quantum computing is an exciting technology which utilizes the unique properties of quantum mechanics to increase the speed of classical computational operations in certain cases. However, understanding quantum computing requires knowledge of both computer science and quantum mechanics in order to develop and employ quantum algorithms. Thus, this paper provides an understandable introduction to quantum computing, and more specifically, quantum algorithms for computer scientists and practitioners. First, a number of foundational topics such as quantum measurement, RSA security, and Simon's algorithm are discussed. Next, a detailed case study of Shor's algorithm is presented as an example of how quantum algorithms can be utilized to solve computationally difficult problems.**

***Keywords*—Quantum Computing; Quantum Algorithms; Shor's Algorithm; Simon's Algorithm; RSA Encryption**

I. INTRODUCTION

Currently RSA encryption is widely employed to protect digital information including e-mails, bank transactions, and even things as simple as text messages. The security of RSA is typically measured in the amount of time it would take to break the scheme and decrypt the data. Because the decryption process is relatively quick once the scheme is broken, the inherent strength of RSA relies on the tedious nature of finding prime factors to large numbers.

Shor's algorithm grants the ability to find these prime numbers much faster than current methods. It is because the current encryption scheme is relied on so heavily by both the private and government sectors—to include the military, which drives a new field of study dubbed "post-quantum cryptography". This field is concentrated on what to do after the physical implementation of sufficiently large quantum computers and the realization of Shor's algorithm.

In 1994 Peter Shor developed a quantum algorithm (i.e., a mathematical or quantum mechanical algorithm to be executed on quantum computer) to factor large numbers with prime factors extremely quickly [11]. This discovery threatens the security of RSA encryption directly. Although a large part of the algorithm is run on a classical computer, the key component that allows Shor's algorithm to be so effective relies on quantum computing technology. Although quantum computing is still in nascent stages, researchers at MIT and the University of Innsbruck in Austria have published findings for a scalable architecture to execute Shor's algorithm [1]. Although there are challenges associated with scaling this architecture to solve larger problems, this breakthrough is instrumental in the downfall of the RSA encryption scheme [13], [21], [22], [23].

Shor's algorithm incorporates several quantum phenomena which are fundamental to quantum mechanics. It is vital to understand these quantum properties and effects before studying Shor's quantum algorithm. Additionally, Simon's quantum algorithm is also useful to understand before approaching Shor's work because it is a much more simplified period finding algorithm. A brief introduction to quantum phenomena and an abbreviated RSA encryption overview will give us the background needed to approach both Simon's then Shor's algorithm in detail.

II. QUANTUM PHENOMENA

Quantum computing offers the ability to solve relational problems rather than execute set processes. Extracting this relational information is at the heart of quantum computing. In this section, we introduce several areas of quantum mechanics necessary for understanding quantum algorithms.

A. Quantum Bits

A classical bit is restricted to existing in one of two states (either a 0 or a 1), while a quantum bit or "qubit" is a quantum-mechanical system that exists in a superposition of states (a continuum between 0 and 1). These qubits differ significantly from classical bits and because of the qubit's unique properties (i.e., the ability to put qubits into a superposition of states and entangle them with each other) means that qubits can interact naturally, and in these interactions is where large amounts of relational information is stored [17].

ISBN: 1-60132-434-0, CSREA Press ©

With regard to the Bloch Sphere in Figure 1, classical bits can exist as a unit vector in the z-direction, straight up or down. These two states can also be described in a 2-dimensional vector space as two orthonormal vectors $|0\rangle$ and $|1\rangle$. Qubits on the other hand, are able to exist in a linear combination (superposition) of these two states [16]. This is best illustrated as the state of a qubit which can exist as any unit vector in the Bloch Sphere $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, subject to the constraint $|\alpha^2| + |\beta^2| = 1$. The key difference is that the classical bit is restricted to existing solely in the direction of the unit vectors $|0\rangle$ and $|1\rangle$, while the qubit can exist in any combination of $|0\rangle$ and $|1\rangle$. This means, the qubit can exist in an infinite number of states.

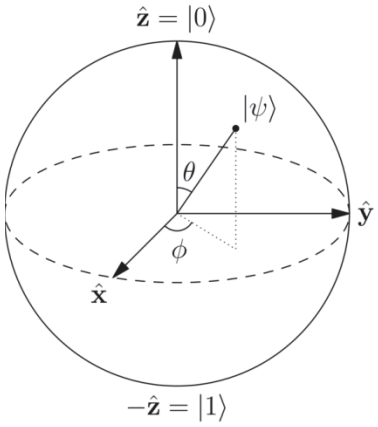


Figure 1. Bloch Sphere [9].

Many options are being considered for physical implementation of qubits including photons, trapped ions, electrons, superconducting materials, and atomic nuclei [2], [14], [15], [18], [19], [20].

B. Hadamard Gate

The Hadamard gate is often one of the first operations in a quantum circuit model, as the ability to leverage the superposition principle of the qubit is what gives a quantum computer its power. The Hadamard gate, when used to operate on a qubit, maps a single qubit into a superposition of $|0\rangle$ and $|1\rangle$ basis vectors with equal weight $|\psi\rangle = 1/\sqrt{2}\,(|0\rangle + 1/\sqrt{2}\,|1\rangle)$ where $\left|\left(\frac{1}{\sqrt{2}}\right)^2\right| + \left|\left(\frac{1}{\sqrt{2}}\right)^2\right| = 1$. This is best described as a horizontal unit vector (perpendicular to the z-vector, $\theta = 90°$) in the Bloch Sphere—a superposition of both states $|0\rangle$ and $|1\rangle$. For example, if there are 100 qubits in the model, and each is acted upon by a Hadamard gate, there now exists a superposition of all 2^{100} possible solutions within the model. However, it is not possible to measure all these solutions. In a quantum system it is only possible to measure each qubit once, and thus, obtain a single solution.

C. Measuring Qubits

In a classical computer, bits can be measured and then remain in the same state afterwards; in a quantum computer, measuring the qubits forces the qubits to collapse into a particular state of the measurement basis (e.g., either $|0\rangle$ or $|1\rangle$) [16]. Any superposition, which is where relational data is held, disappears once the qubit has been measured. This phenomenon is called the "collapse" of the qubit. It is important to note that no further data from the quantum system can be taken from the qubit after the measurement is performed, it is an irreversible process.

D. Qubit Decoherence

While purposefully measuring a qubit causes it to collapse, outside factors such as environmental noise (e.g., errant electro-magnetic waves) may also cause the quantum system to collapse before a proper measurement can be taken [8]. Quantum computing requires precisely controlled conditions in order for qubits to maintain superposition and become entangled (that the state of one qubit is dependent on the state of one or more other qubits) [17]. For example, the qubits maintained in D-wave's adiabatic quantum computer must be kept at near absolute zero in order to effectively function in superposition [12]. Whether it be isolation from electro-magnetic waves, extreme temperatures, or other unknown factors, decoherence can cause major problems with the integrity of the data stored in the qubits. Solutions to this problem include isolation from environmental factors (e.g., controlled environments and shielding), as well as quantum error correction techniques to mitigate the effects of decoherence.

E. Quantum Error Correction

In a classical computer, in order to reliably store information for long periods of time, bits can be copied, re-copied, and stored redundantly. However, in a quantum computer, it is not possible to perfectly clone an unknown quantum state [6]. This is because the measurement inherently affects the qubit you wish to copy. However, it is possible to create a series of entangled qubits and use that series as a representation of a single qubit of information, this is called a "logical qubit" [3]. If one or a few of those entangled qubits erroneously change state due to decoherence it can be corrected by assessing its conformity with the other qubits within the logical qubit.

III. RSA ENCRYPTION

Modern computer systems use public-key cryptography such as RSA which relies on the difficulty of factoring the product of two large prime numbers. For most computer systems the time it would take to factor these large numbers becomes unreasonable, and therefore public key cryptography is able to provide strong security [10].

A. Key Creation

The architecture of the RSA schema is comprised of three parts: a private key d, a public key c, and a publicly available very large number N. The process of creating these keys starts with picking two very large prime numbers; typically called p and q. Next, these numbers are multiplied to create a very large number N:

$$N = pq \tag{1}$$

After the creation of N, Euler's totient of N is created, which is the total number of integers less than N which are relatively prime to N (i.e., all the integers in the totient and N have a greatest common divisor of 1). Because Euler's totient is multiplicative we know:

$$\varphi(N) = \varphi(p)\varphi(q) \tag{2}$$

Also, because we chose p and q as prime numbers, we know $\varphi(p) = p - 1$ and $\varphi(q) = q - 1$. This allows us to create the totient of N:

$$\varphi(N) = (p-1)(q-1) \tag{3}$$

We now choose the public key c which is relatively prime to the totient of N, meaning the greatest common divisor of the totient of N and c is 1. The fastest way to know if a chosen number and the totient of N are relatively prime is by using the Euclidian algorithm to calculate the greatest common divisor and check if it really is 1:

$$\gcd\big(c, \varphi(N)\big) = 1 \tag{4}$$

Next, in order to calculate the private key, d, we need to calculate the modular inverse of our public key c. This is done by using the extended Euclidian algorithm. This process solves the following equation for d [2]:

$$cd = 1 mod(\varphi(N)) \tag{5}$$

After the creation of the private key d, the cryptosystem is complete and the encryption/decryption process can begin. At this point it is important to understand that only the large number N and the public key c are publicly available. The private key d is only known by the individual to whom it belongs and the totient $\varphi(N)$ is discarded.

B. Encrypting/Decrypting with RSA

Once the private-public key pairs are created and appropriate distribution techniques are established, the encryption process is relatively straightforward. To encrypt the message a Bob wants to send to Alice, it is first encrypted using both Alice's public key, c, and the large number N which are available to Bob because they are public knowledge. The encrypted message is denoted by the letter b:

$$b = a^c mod(N) \tag{6}$$

When Alice receives the encrypted message b she is able to decrypt the message using her private key:

$$a = b^d mod(N) \tag{7}$$

A simple overview of the public key encryption scheme is provided in Figure 2.

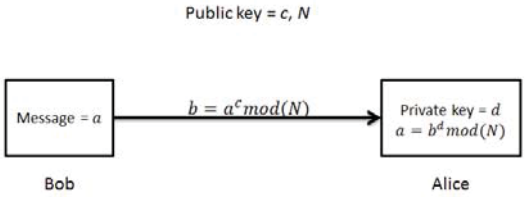


Figure 2. Illustration of public key cryptography.

IV. Understanding Quantum Algorithms

Before moving on to a complex quantum algorithm such as Shor's algorithm, understanding another—Simon's algorithm makes the approach significantly easier. As Shor's algorithm is a specific implementation of Simon's algorithm, an overview of Simon's period finding algorithm is useful. The quantum Fourier transform will be introduced later because it is used in Shor's algorithm to speed up the period finding process.

A. Simon's Algorithm

In 1997, Daniel Simon introduced a quantum algorithm to reduce the number of measurements required to solve an unknown period problem [5]. In a classical computer, finding an unknown period a takes order $O(2^{n/2})$ measurements, while Simon's technique only requires $O(n)$ measurements where n is the number of bits needed to represent the period in base 2 [3]. The classical method is akin to a guess and check until the unknown period is found and as the size of the period a grows, the number of measurements grows exponentially along with it. Using Simon's algorithm, as the size of the period a grows, the number of measurements only grows linearly with n.

Simon's algorithm works through a series of quantum operations and measurements. First, the input and output registers must be initialized, which is by convention done in the state $|0\rangle$. Next, each qubit in the input register is operated on by a Hadamard transformation, putting the qubits into a state of equal superposition of all possible combinations. The state of the system is described as [4]:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle\,|0\rangle \tag{8}$$

where $|x\rangle$ represents the input register after the Hadamard transformation such that $|x\rangle$ is in a superposition state and $|0\rangle$ represents the output

register still in its initialization state. Next, the unitary transform $\hat{U}_f$ is applied to the superposition state of the input register $|x\rangle$ and stored in the output register, the new state of the system becomes [3]:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle \quad (9)$$

After the unitary transform $\hat{U}_f$ operates, the output register holds the results of the function $|f(x)\rangle$, while the input register $|x\rangle$ is still in a state of superposition. Now suppose a measurement of the output register $|f(x)\rangle$ is taken, and thus, collapses both the output and input registers. The output register collapses to a random evaluation of x called $f(x_0)$. The input register can now only exist in one of two states: $|x_0\rangle$ or $|x_0 \oplus a\rangle$ according to the generalized Born rule [7]. This is because the function (the unitary transform $\hat{U}_f$) is defined as having the same result for two specific inputs (i.e., the function is periodic under bitwise modulo-2 addition) and $f(x) = f(x \oplus a)$. The resulting state of the input register is [3]:

$$\frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle) \quad (10)$$

The input register, although it now contains valuable information (i.e., we can solve for a given both states), is not as useful as it seems because the register can only be measured one time. Successive trials would yield more random values for $|x_0\rangle$ and $|x_0 \oplus a\rangle$ satisfying different measured outputs, which would not help solve for the unknown period a efficiently.

The next step in this process is to again apply the Hadamard transformation to the input register $|x_0\rangle + |x_0 \oplus a\rangle$, and the state of the quantum system becomes [3]:

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^{n-1}} |(-1)^{x_0 \cdot y} + (-1)^{(x \oplus a) \cdot y}\rangle |y\rangle \quad (11)$$

where $|y\rangle$ represents the output register. More simply, the input register can be interpreted as the expansion coefficient of the output register ($|(-1)^{x_0 \cdot y} + (-1)^{(x \oplus a) \cdot y}\rangle$ becomes a_y) and Eq. (11) simplifies to [4]:

$$\sum_{y=0}^{2^{n-1}} a_y |y\rangle \quad (12)$$

From Eqs. (11) and (12), we know that the coefficient of the output register $|y\rangle$ will be 0 if $a \cdot y = 1$. Because the probability of a measurement is represented by the absolute value squared of the expansion coefficient, $|a_y|^2$, this means the probability of measuring a solution in which $a \cdot y = 1$ is 0. Thus, the output register $|y\rangle$ is limited only to solutions in which $a \cdot y = 0$.

For this reason, any measurement of Eq. (12) must yield a random y in which $a \cdot y = 0$, where each y value obtained reduces the possible choices for the period a by half. This allows the unknown period a to be found in only $O(n)$ invocations of Simon's algorithm by the creation of a system of equations for a which is comprised of n equations.

B. Quantum Fourier Transform

The quantum Fourier transform (QFT) is an important part of Shor's algorithm because when introduced, it emphasizes a relationship between the states of an input register, the period of the function, and the total size of the register. The QFT (denoted as U_{FT}) like all other valid quantum operations is a linear, unitary operator. The QFT maps n qubits to n qubits (the output size of the QFT is the same as the input size in terms of number of qubits), and the effect of the QFT on a register is [3]:

$$U_{FT}|x\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^{n-1}} e^{2\pi i x y / 2^n} |y\rangle_n \quad (13)$$

The QFT operates on the input register $|x\rangle$ to create a set of states in the output register $|y\rangle$ with the probabilities of measurement of $\left|e^{2\pi i x y / 2^n}\right|^2$ for each state. The QFT, like the other operators, can also operate on a superposition of states which is invaluable for Shor's algorithm.

V. SHOR'S ALGORITHM

Introduced in 1994, Shor's algorithm is a quantum algorithm designed to quickly solve prime factors of a given number which is of great concern in modern cryptography—specifically the RSA public key cryptography [11]. The method Shor created to solve these prime factors utilizes a number of classical computing processes and only leverages quantum computing to solve one aspect of the problem—finding the period. This piece of Shor's algorithm is a specific realization of Simon's algorithm.

As shown in Table 1, Shor's factoring process can be summarized in five steps, of which only the fourth step is quantum in nature—the very same step is the most computationally intensive part of the process [4].

Table 1. A summary of the factoring process [4].

1. If N is even, return a factor of 2. Otherwise, continue to the next step.
2. Check whether $N = a^b$ for integers a and b such that $a \geq 1$ and $b \geq 2$. If $N = a^b$ then return the factor a.
3. Randomly choose an integer $x \in \{2,3,\cdots, N-1\}$ and compute $\gcd(x, N)$. If $\gcd(x, N) > 1$ then return the factor $\gcd(x, N)$. If $\gcd(x, N) = 1$ (i.e., if x is a coprime of N) then continue to the next step.
4. Find out the order r [period] of $x^r mod N$. If r is even and $x^{r/2} mod N \neq -1$ then continue to the next step. Otherwise, restart from Step 3 with a different x.
5. Compute $gcd(x^{r/2} \pm 1, N)$ and check whether one of them is (or both of them are) nontrivial factor (factors) of N. If so, then return the factor (factors). Otherwise, restart from Step 3 with a different x.

The remainder of the paper focuses specifically on understanding Shor's quantum algorithm contribution as described in step 4, where the order r is the period of the function which needs to be found. Just as in Simon's algorithm, we will consider both an input and an output register throughout each step.

A. *Understanding Shor's Quantum Algorithm*

The output register must be able to hold N, in binary form. This means, for example, if $N = 64$ the output register must contain at least 6 qubits because 64 is represented within 6 binary digits ($2^6 = 64$). The size of the output register (the number of qubits required), will be denoted as l.

The input register generally needs to have twice as many qubits as the output register ($2l$). This configuration is desirable so that the input register can contain at least N different states that produce the same output — this gives us more "workspace" with which to capture the period of the function. The size of the input register is denoted as t.

Entering step 4 of the process, we know that the number to be factored is N and we have already chosen an x which is coprime to N. First, the input and output registers must be initialized to a known value (typically $|0\rangle$):

$$|\psi\rangle_0 = |0\rangle_t |0\rangle_l \tag{14}$$

The quantum system or total wave function of the system is written as $|\psi\rangle$ at step 0 with the input and output registers (t and l, respectively) initialized to $|0\rangle$. Next, the input register is put through t Hadamard gates, placing the input register $|0\rangle_t$ into a state of superposition represented as $|k\rangle$ [4]:

$$|\psi\rangle_1 = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |k\rangle |0\rangle \tag{15}$$

Next, the superposition state $|k\rangle$ is operated on by a modular exponent and the result is stored in the output register [4]:

$$|\psi\rangle_2 = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |k\rangle |x^k mod(N)\rangle \tag{16}$$

Notice the similarity with Simon's problem with this method. Next, a measurement of the output register, yields a random value of $x^k mod(N)$ called z_0. This measurement forces the input register into a state of superposition of all the possible inputs that would yield the measured value z_0, satisfying the generalized Born rule [7]. The total number of valid input states is represented as M. The function is periodic so we know that the valid inputs for a particular solution are $f(d + mr) = z_0$, where the value of d is the smallest possible input for this function that yields z_0 and any multiple m of the period r added to the smallest value d will yield the same z_0.

Focusing on the input register, which now contains the values of interest, and temporarily disregarding the output register, the total wave function at step 3, without the output register is now [4]:

$$|\psi\rangle_3 = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} |d + mr\rangle \tag{17}$$

Similar to Simon's problem, valuable information is stored in the input register and if it was possible to make a copy of it, the period r could be found in a small number of measurements. However, only one measurement yielding a random number can be taken and successive measurements would yield more random numbers for different measured outputs.

Since, the number of qubits in the input register is double the output register, the number of solutions that can simultaneously exist in the input register satisfying $|d + mr\rangle$ is large. Thus, the next step is to apply a quantum Fourier transform to the input register yielding [4]:

$$|\psi\rangle_4 = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} \frac{1}{\sqrt{2^t}} \sum_{y=0}^{2^t-1} e^{2i\pi(d+mr)y/2^t} |y\rangle \tag{18}$$

where the input register is now represented as $|y\rangle$ and useful information can now be measured.

B. *Finding the Period*

Simplifying Eq. (18) and using the substitution $\xi = e^{2i\pi\frac{yr}{2^t}}$ gives a wave function of the input register [4]:

$$|\psi\rangle_4 = \frac{1}{\sqrt{2^t M}} \sum_{y=0}^{2^t-1} e^{2i\pi dy/2^t} \left(\sum_{m=0}^{M-1} \xi^m |y\rangle \right) \tag{19}$$

From this wave function, the probability of measuring any particular $|y\rangle$ is given by [4]:

$$\frac{1}{2^t M}\left|\sum_{m=0}^{M-1}\xi^m\right|^2 \quad (20)$$

This means the inputs will constructively interfere when $\frac{yr}{2^t}$ is close to an integer and destructively interfere when $\frac{yr}{2^t}$ is otherwise. This raises the probability of measuring a particular input y that, if C is an integer, satisfies $\frac{yr}{2^t} \approx C$. Moreover, if this value of y is close to an integer, we know that $\xi \approx 1$, and therefore the probability of measurement is [4]:

$$\frac{1}{2^t M}\left|\sum_{m=0}^{M-1}\xi^m\right|^2 = \frac{M^2}{2^t M} = \frac{M}{2^t} \approx \frac{1}{r} \quad (21)$$

Thus, the probability of measuring a specific value in the input register $|y\rangle$ that satisfies $\frac{yr}{2^t} \approx C$ is approximately $\frac{1}{r}$, which is much higher than the values in the input register which destructively interfere.

The final quantum step of Shor's algorithm is to measure the input register $|y\rangle$. The result of this measurement is assumed to follow the high likelihood that $\frac{yr}{2^t} \approx C$. Assuming this is true, we can rearrange the equation to understand the relationship better [4]:

$$\frac{y}{2^t} \approx \frac{C}{r} \quad (22)$$

The quantum part of Shor's algorithm is now complete and the rest can be handled by a classical computer. The quantum aspects of Shor's algorithm result in a high likelihood of a solution which satisfies a relationship between the period r, the solution space 2^t, an integer C, and the measured result y. Since the result y and solution space 2^t are known, we can solve for the left half of Eq. (22) and find an equivalent integer fraction to solve the right hand side. More specifically, the continued fraction method is used to solve for the period r.

Since we know that C is likely an integer, thus $2C$, $3C$,…,etc. are also likely integers. This means that when we find the equivalent fraction for the right hand side we must also consider that $\frac{2C}{r} = \frac{C}{r/2}$ and $\frac{3C}{r} = \frac{C}{r/3}$ and so on, are valid solutions as well. Using the number of steps to convergence in the continued fraction, an initial value for the period r is generated. The initial period r must be double checked by substituting the value r back into the original equation we are trying to solve:

$$x^r mod(N) = 1 \quad (23)$$

If the statement is incorrect, then small multiples of r can be tried, since C, $2C$, $3C$,…,etc. are all integers. This process is used to find the smallest period r that satisfies Eq. (23).

Lastly, the value r must also be even and satisfy the condition $x^{r/2} mod N \neq -1$. If r does not satisfy these conditions, the quantum algorithm must be re-accomplished with a new value for our initial coprime number x. Once this step has been accomplished successfully and one or both prime factors of N has been found—the factoring process would be complete. If only one prime factor is found, simple division of N by the known value would yield the other prime factor. Knowing the prime factors to N would effectively break the RSA encryption because once the prime factors are known the private key can be computed easily.

C. Breaking RSA

To break the RSA encryption an alternate step may also be used. An overview of this attack on RSA public-key encryption is provided in Figure 3.

After finding the period r, a pseudo-private key d' can be created satisfying [3]:

$$cd' = 1 mod(r) \quad (24)$$

Using this value for d', the original content of the encrypted message b can be easily decrypted [3]:

$$a = b^{d'} mod(N) \quad (25)$$

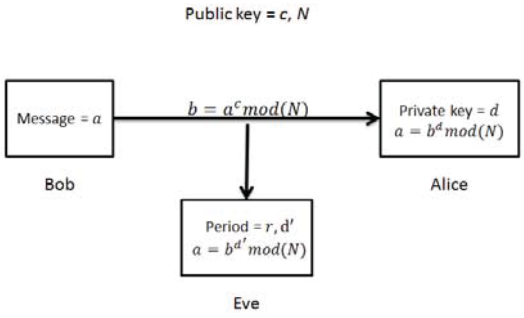


Figure 3. An overview of the alternate method to break public key encryption using the period.

VI. CONCLUSIONS

Peter Shor made a very important contribution to the field of quantum algorithms with his realization of quantum period finding—its relation to the RSA encryption scheme has drawn international acclaim and notoriety from renowned security specialists. However, there have been many other discoveries as to the types of computations quantum computers can perform. Currently, three classes of algorithms: (i) algebraic and number theoretic; (ii) oracular; and (iii) approximation and simulation are highlighted in the "quantum zoo," the most complete compendium of quantum algorithms available [24]. Unfortunately, each of these algorithms needs to be further studied

and expanded upon as they wait to be applied on a quantum computer.

Further study of this area needs to run parallel with the kinds of difficult problems we are facing using classical computers to determine how we can leverage the strengths of quantum computing. In this work, we have built a foundation for understanding quantum algorithms by first understanding the quantum phenomena necessary for quantum computing and then demonstrated the importance of applying quantum algorithms by using Shor's algorithm. This work provides a starting point for those interested in quantum computing and quantum algorithms.

DISCLAIMER

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

REFERENCES

[1] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt. "Realization of a scalable Shor algorithm." arXiv preprint arXiv:1507.08852 (2015).

[2] M. A. Nielsen., and I. L. Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[3] N. D. Mermin. *Quantum computer science: an introduction*. Cambridge University Press, 2007.

[4] A. Pathak. *Elements of quantum computation and quantum communication*. Taylor & Francis, 2013.

[5] D. R. Simon. "On the power of quantum computation." *SIAM journal on computing* 26, no. 5 (1997): 1474-1483.

[6] W. K. Wootters, and W. H. Zurek. "A single quantum cannot be cloned." *Nature* 299, no. 5886 (1982): 802-803.

[7] Born, Max. "Quantenmechanik der stoßvorgänge." *Zeitschrift für Physik* 38, no. 11-12 (1926): 803-827.

[8] M. Schlosshauer. "Decoherence, the measurement problem, and interpretations of quantum mechanics." *Reviews of Modern Physics* 76, no. 4 (2005): 1267.

[9] [Bloch Sphere]. Retrieved February 25, 2016 from https://upload.wikimedia.org/wikipedia/commons/thumb/f/f4/Bloch_Sphere.svg/2000px-Bloch_Sphere.svg.png

[10] R. L. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems." Communications of the ACM 21, no. 2 (1978): 120-126.

[11] P. W. Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." SIAM review 41, no. 2 (1999): 303-332.

[12] R. Harris, J. Johansson, A. J. Berkley, M. W. Johnson, T. Lanting, S. Han, P. Bunyk et al. "Experimental demonstration of a robust and scalable flux qubit." Physical Review B 81, no. 13 (2010): 134510.

[13] R. Van Meter, and C. Horsman. "A blueprint for building a quantum computer." Communications of the ACM 56, no. 10 (2013): 84-93.

[14] L. M. K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M. H. Sherwood, and I. L. Chuang. "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance." Nature 414, no. 6866 (2001): 883-887.

[15] D.Kielpinski, C. Monroe, and D. J. Wineland. "Architecture for a large-scale ion-trap quantum computer." Nature 417, no. 6890 (2002): 709-711.

[16] E. Schrödinger. "Discussion of probability relations between separated systems." In Mathematical Proceedings of the Cambridge Philosophical Society, vol. 31, no. 04, pp. 555-563. Cambridge University Press, 1935.

[17] A. Einstein, B. Podolsky, and N. Rosen. "Can quantum-mechanical description of physical reality be considered complete?." Physical review 47, no. 10 (1935): 777.

[18] C. Lu, D. E. Browne, T. Yang, and J. Pan. "Demonstration of a compiled version of shor's quantum factoring algorithm using photonic qubits." Physical Review Letters 99, no. 25 (2007): 250504.

[19] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White. "Experimental demonstration of a compiled version of shor's algorithm with quantum entanglement." Physical Review Letters 99, no. 25 (2007): 250505.

[20] A. Politi, J. C.F. Matthews, and J. L. O'brien. "Shor's quantum factoring algorithm on a photonic chip." Science 325, no. 5945 (2009): 1221-1221.

[21] I. Chuang, R. Laflamme, P. Shor, and W. Zurek. "Quantum computers, factoring, and decoherence." arXiv preprint quant-ph/9503007 (1995).

[22] R. Landauer. Is quantum mechanically coherent computation useful?. IBM Thomas J. Watson Research Division, 1994.

[23] W. G. Unruh. "Maintaining coherence in quantum computers." Physical Review A 51, no. 2 (1995): 992.

[24] National Institute of Standards and Technology. "The Quantum Zoo." Available at: http://math.nist.gov/quantum/zoo/.

V. Journal Paper

Publication Details

Title: Resource Evaluation of Quantum Linear Systems Algorithm for Application to Electromagnetic Scattering

Publication: TBD

Date: TBD

Resource Evaluation of Quantum Linear Systems Algorithm for Application to Electromagnetic Scattering

Casey J. R. Riggs, Charlton D. Lewis, Logan O. Mailloux
Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio 45433, United States
{Casey.Riggs, Charlton.Lewis, Logan.Mailloux}@afit.edu, odparek@sandia.gov

Abstract*—A decomposition of the logical resource requirement for the application of the Quantum Linear Systems Algorithm to the calculation of an electromagnetic scattering cross-section value for an aerodynamic cone is presented. The generalized space complexity requirements for the Quantum State Preparation Algorithm, Quantum Linear Systems Algorithm, Swap test, and Quantum Amplitude Estimation Algorithm are simplified according to electromagnetic scattering finite element method variables, desired precision, and probability of success. Resource optimization is approached through the re-use of quantum resources post-measurement via resource-leveling techniques. Varying bit precisions independently between 8-bit and 128-bit created resource pools of 133 to 853 logical qubits, while varying the problem size independently from $\mathbf{50 \times 50}$ *to* $\mathbf{400 \times 400}$ *grid meshes created resource pools of 453 to 477 logical qubits.

Keywords*—Quantum Linear System Algorithm; Quantum Computing; Linear Systems; Electromagnetic Scattering

I. INTRODUCTION

The Quantum Linear Systems Algorithm (QLSA) has been applied to several areas of study in quantum computing such as quantum machine learning [1] [2] [3] [4] [5], least-squares curve fitting [6], solving linear systems of differential equations [7] [8], estimating resistance of electrical networks [9], and solving Toeplitz systems [10]. Small experimental systems showcasing the QLSA are presented in [11] [12] [13]. The QLSA requires that in a linear system of equations in the form of $Ax = b$ be prepared such that the A matrix to be inverted is sparse and that the vector b be prepared as a quantum vector, in an effort to solve for $x = A^{-1}b$.

Certain finite element applications of Maxwell's equations can be configured to a linear system of equations. The most computationally intensive step in solving an electromagnetic (EM) scattering problem prepared in this way is the inversion of a large matrix. This large matrix is constructed as a system of equations from the Finite Element Method (FEM). By using standard methods for the construction of the mesh and the resulting matrices [14] one can construct a system of equations fitting the requirements of the QLSA.

EM scattering systems of equations grow in size considerably as the mesh used in the construction becomes more and more refined. Thus, these problems are prime candidates for a quantum mechanical speedup. Real world applications of EM scattering problems include antenna design, predicting radar signals for intelligence applications, remote sensing, and radar cross-section analysis for aircraft design.

The need to calculate a space complexity requirement for applications of quantum algorithms stems from the limited available quantum resources. The circuit construction of the various registers on which to perform four quantum algorithms is briefly summarized, followed by a resource evaluation of each construct. Lastly, the generalized resource requirement is applied to a particular EM scattering problem and a brief discussion of scaling with regard to edge count and bit precision is presented.

II. THEORY

The construction of the four quantum algorithm circuits, including quantum state preparation, the quantum linear system algorithm, the swap test, and the amplitude estimation are detailed briefly in the following sections. These four quantum subroutines are required for the calculation of an EM scattering cross-section value [15].

A. State Preparation

The overall process of the Quantum State Preparation Algorithm (QSPA) is summarized in the quantum circuit diagram of Fig. 1.

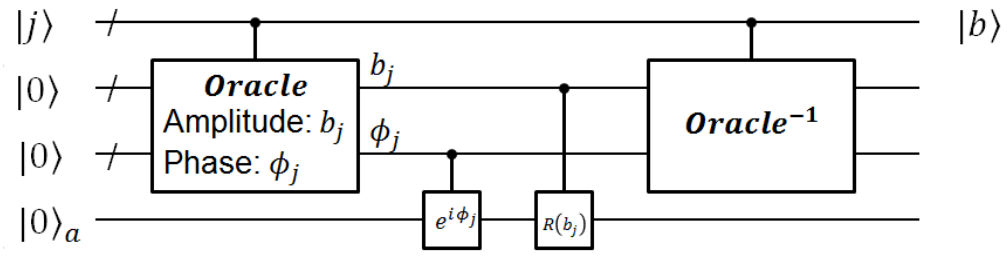


Figure 1. The state preparation algorithm from [15].

The successful implementation of the algorithm yields the approximate preparation of the quantum state $|b\rangle$ in the register originally assigned as $|j\rangle$ where $|b\rangle = C_b b_j e^{i\phi_j}|j\rangle$ where C_b is a constant. The oracles in this construction are not assumed to add any additional resource costs. The final prepared quantum state is dependent on the probability of a $|1\rangle$ in the ancillary register.

B. Quanutm Linear Systems Algorithm

The original design for the QLSA proposed in [16] solves a system of linear equations in the form $Ax = b$ quantum mechanically. The new system of linear equations can be summarized by $\hat{A}|x\rangle = |b\rangle$, where a given operator $\hat{A}$ is represented by a Hermitian $N \times N$ matrix, and $|x\rangle$ and $|b\rangle$ are represented by normalized vectors in a Hilbert space represented by a quantum superposition of values. The QLSA effectively inverts the matrix A, thereby creating the solution $|x\rangle = \hat{A}^{-1}|b\rangle$.

The two main subroutines of the QLSA are the Quantum Phase Estimation Algorithm (QPEA) which requires the use of a particular quantum oracle for Hamiltonian Simulation and an eigenvalue rotation algorithm involving an ancilla.

1) Phase Estimation

The QPEA uniquely identifies the eigenphase of an eigenvector of a particular unitary operator, also known as the Abelian stabilizer problem [17]. Although the QPEA is introduced in Shor's algorithm for quantum factoring, the generalization of the algorithm proper is presented by Kitaev [18].

If the matrix A can be constructed well enough to be a sufficiently sparse Hermitian matrix, then it can be applied as a unitary operator in the form of e^{iAt} by Hamiltonian simulation. There exists a method to convert a non-Hermitian matrix into a larger Hermitian matrix outlined in [16]. The Hamiltonian simulation paired with the application of the inverse quantum Fourier transform composes the main components of the phase estimation algorithm. The general quantum circuit diagram which illustrates the process is presented in Fig. 2.

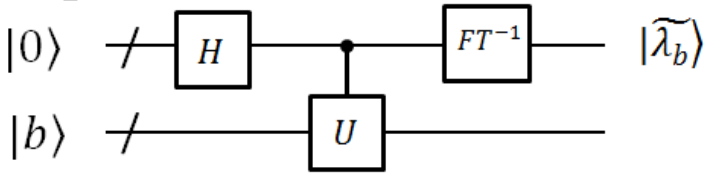


Figure 2. A simplified phase estimation circuit [17].

For the purposes of application, the QPEA will be used to extract the eigenvalues from the simulated Hermitian matrix A in an effort to invert the matrix. The QPEA produces the quantum state $|\widetilde{\lambda_b}\rangle|b\rangle$ with probability $1 - P_{err}$, where the approximated eigenvalues ($\tilde{\lambda}_b$) are stored in superposition in the first register.

In the application of the Hermitian unitary operator, we require an oracle with which to read and simulate A. Clader et al [15] give a rudimentary construction of such an oracle as a generalization of a quantum random walk [19] [20] [21] and is discussed in more detail next.

2) Oracle Construction

The generalized oracle used to simulate the unitary operator U in the phase estimation is given by [15] [22]. The method is used to simulate a given Hermitian matrix. Given a particular Hermitian matrix A, which can be subdivided into c 1-sparse sub-matrices (matrices with at most 1 non-zero element per row), as well as two specific unitaries: one to calculate the magnitude (U_m) and one to calculate the phase (U_p) of the particular elements of the 1-sparse sub-matrix A_c, one can perform a series of operations to simulate A.

The quantum circuit in Fig. 3 summarizes this oracle including both unitary operators, a phase shift ($P(\phi)$), and an operator similar to a quantum walk Hamiltonian (H_{rw}).

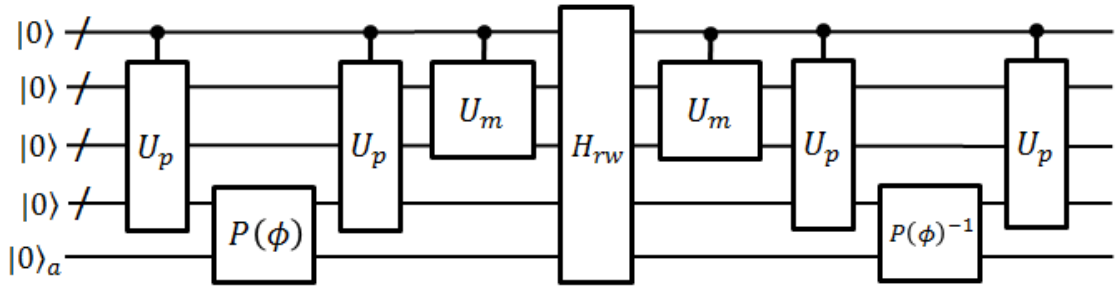


Figure 3. Circuit diagram of Oracle for Hamiltonian Simulation [15].

This particular construction may not be ideal, however further research into the field of more efficiently simulating Hamiltonians is ongoing. Notably, efforts have been made to reduce the time complexity of the operation in [23] [24] and [25].

3) Rotation

The ancilla qubit in the QLSA needs to be rotated about the values of the inverse of the eigenvalues. Smaller experiments [11] [12] [13] provide easily calculated, thus known eigenvalues (and known inverses), and therefore constructing the rotation is much easier. For larger problems, unknown eigenvalues complicate the process.

A method to accomplish this eigenvalue inversion for larger problems is presented in [26], and the circuit diagram notation is shown in Fig. 4. It is important that the uncomputation after the rotation of the QLSA still needs to occur with the newly included inversion operation. An extra register is used to store the inverted eigenvalues as a control for the rotation on the ancilla qubit.

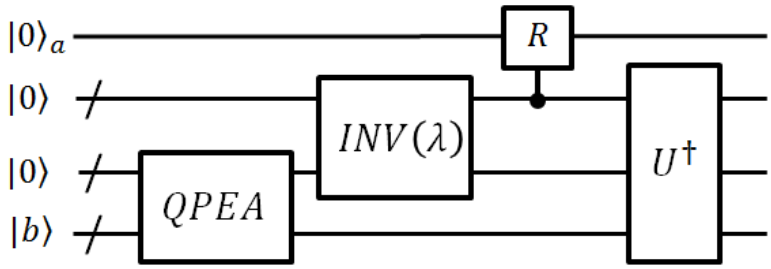


Figure 4. The circuit design for the inversion of the eigenvalues [26].

The details on the evolution of the quantum state through the circuit are detailed more thoroughly in [26], however it does include the use of Newton iteration on the working register of the QPEA to invert the eigenvalues. The size of the register needed to hold the inverted eigenvalue is directly a result of this method.

C. Swap Test

The Quantum Swap Test, originally introduced as quantum "fingerprinting" in [27] is a test of similarity of quantum states. The swap test is a conditional swap of two quantum states, akin to a dot product of two geometric vectors. The swap operation occurs with a probability associated with the relative overlap of the two states. With regard to the QLSA, the solution state $|x\rangle$ will be conditionally swapped with another vector $|R\rangle$; this is one way to utilize the output of the QLSA. The quantum circuit diagram shown in Fig. 5 demonstrates the simplistic construct of the swap test.

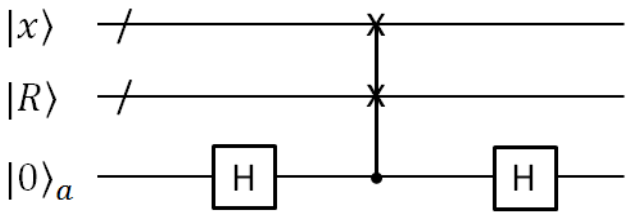


Figure 5. The quantum swap test on two quantum states, $|x\rangle$ and $|R\rangle$.

The quantum swap test involves the use of an ancilla; this ancilla will be used as the conditional for the swap and will subsequently store the relational data of the overlap if it is not directly measured.

Nominally, the swap test calls for a measurement of the ancilla qubit after the algorithm to determine whether or not the states were indeed different. If the states are equal, the outcome will be $|0\rangle$ with probability P=1, this is deemed a "pass". If the states are different the outcome may be either $|0\rangle$ or $|1\rangle$. Because of this, if the outcome is $|1\rangle$ then the states were definitely different, and a $|1\rangle$ in the ancilla is deemed a "fail".

The probability of a "fail" in the swap test is given by $P = \frac{1+|\langle x|R\rangle|^2}{2}$. One would need to repeat the measurements enough times to complete a statistical analysis of the results in order to ascertain the amount of overlap between the states. However, when paired with the Quantum Amplitude Estimation Algorithm, the repetitive measurements are not needed.

D. Amplitude Estmation

Grover's quantum search algorithm [28] represents an approach to search through an unsorted database for a specific value with the fewest number of guesses. The key subroutine involved in Grover's algorithm is known as the amplitude amplification "engine" and this subroutine is characterized in the form of quantum circuit notation as:

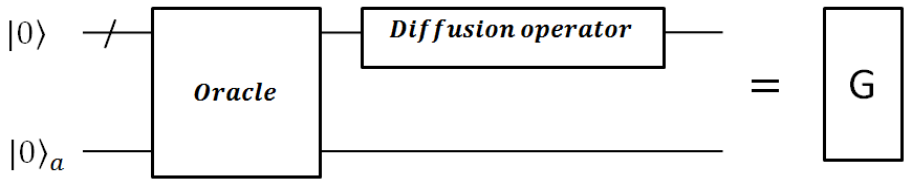


Figure 6. Amplitude amplification engine.

The Quantum Amplitude Estimation Algorithm (QAEA) of [29] utilizes the same iterative engine involved in Grover's search algorithm, however, this QAEA subroutine is able to apply different numbers of iterations to a specific register "in parallel". From this, the algorithm can create a superposition of all the values of the number of iterations which created good amplitude amplification. The circuit notation for the QAEA is given in Fig. 7 where Fig. 6 represents the amplification engine G.

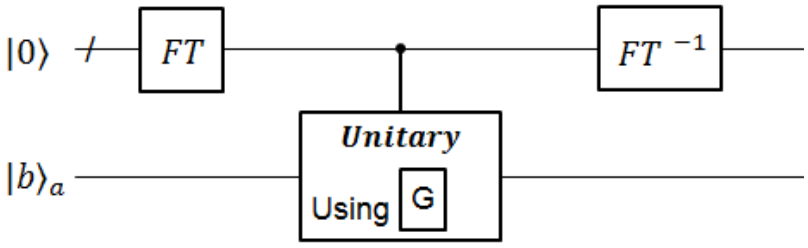


Figure 7. Circuit Diagram for the QAEA on the ancilla from the QSPA.

Amplitude amplification is naturally cyclical, and the inverse Fourier transform of the QAEA can be used to find the period of this sinusoidal function. This extraction of the period can then be used with a little post processing per Theorem 12 in [29] to estimate the value for the amplitude of the particular target value according to Eq. 2:

$$|\tilde{a} - a| \leq \frac{2k\pi\sqrt{a(1-a)}}{M} + \frac{k^2\pi^2}{M^2} \leq \epsilon a \qquad (2)$$

In this equation M represents the dimensionality of the Hilbert space spanned by the working register, ϵ represents the desired precision of the measurement to within $\pm\epsilon a$, a is the probability of the desired value, and k is related to the probability of succeeding in our amplitude estimation. The measurement has precision $\pm\epsilon a$ with probability of at least $\frac{8}{\pi^2}$ for $k = 1$ and probability greater than $1 - \frac{1}{2(k-1)}$ for $k \geq 2$ where a is the probability associated with a desired value. It is of note that if $a = 0$ then $\tilde{a} = 0$ with certainty and if $a = 1$, then $\tilde{a} = 1$ with certainty. Solving for the size of the Hilbert space M of the register yields:

$$M \geq \left\lceil \frac{k\pi}{\epsilon\sqrt{a}} \left(\sqrt{1-a} + \sqrt{1-a+\epsilon} \right) \right\rceil \qquad (3)$$

The probability of error in making a measurement of the desired probability is shown in [29] to be:

$$1 - \frac{1}{2(k-1)} \geq 1 - P_{err} \qquad (4)$$

By solving for the variable k, which gives $k \geq 1 + \frac{1}{2P_{err}}$ and substituting back into Eq. 3:

$$M \geq \left\lceil \frac{\pi}{\epsilon\sqrt{a}} \left(1 + \frac{1}{2P_{err}} \right) \left(\sqrt{1-a} + \sqrt{1-a+\epsilon} \right) \right\rceil \qquad (5)$$

III. RESOURCE EVALUATION

The resource estimation for the implementation of the quantum algorithm is constructed in terms of logical entities. The qubits which compose the registers and ancilla of the process are considered to be fault-tolerant. This is a critical assumption as the implementation of logical qubits requires scaling in the construction of the quantum bits. For example, a logical qubit composed of 9 individual qubits utilizing quantum error correction (QEC) algorithms (such as the Shor code [30]) would scale the physical requirements by a factor of 9. Here only logical resources will be evaluated.

The algorithm cost as a whole, is determined by the sum of the requirements of the individual subroutines, with some overlap between registers where registers such as those used to store the phase and amplitude components are reused. The register sizes described in this work are determined by many factors such as the properties of the matrix A, the desire to minimize error, and gain specific levels of precision.

A. QSPA

The QSPA resource cost is incurred by the use of four registers (see Fig. 1). The first register needed for the QSPA is the register which will contain the prepared quantum state $|b\rangle$. The size of this register is determined by the problem size N of the $N \times N$ matrix A.

Because there are two applications of the QSPA, one to prepare $|b\rangle$ and one to prepare $|R\rangle$, two registers of $log_2(2N)$ are needed. The register size is $log_2(2N)$ instead of $log_2(N)$ because the method used to create a Hermitian from the non-Hermitian A matrix is realistically assumed to be used and consequently the vectors double in size. The two registers need to be the same size for the application of the swap test later in the process.

The second and third registers are used to store the values for the amplitude and phase components for the desired quantum state. These registers need only to be able to store the largest value of amplitude or phase with the desired precision ϵ_α and ϵ_ϕ respectively. The precision of these registers is important as they function as inputs into the quantum system. These registers may be re-used to prepare both $|b\rangle$ and $|R\rangle$. It is important to note that these two registers are also used later in the construction of the oracle for the Hamiltonian simulation in the QLSA, for the purposes of storing the magnitude and phase of the elements in the submatrices of A.

The fourth register is an ancilla used for the conditional rotation and only incurs a cost of one extra quantum bit per QSPA. The QSPA is used twice—to prepare $|b\rangle$ and $|R\rangle$, so two ancilla are needed.

The cost of implementing the QSPA for both registers $|b\rangle$ and $|R\rangle$ in sequence, assuming re-use of the registers used to store the amplitude and phase components are:

$$max_{QSPA(S)} = 2 + 2log_2(2N) + (m + p) \qquad (6)$$

where the register used to input the phase component is denoted as $m = log_2\left(\frac{1}{\epsilon_\phi}\right)$ and the register for the input of the amplitude component $p = log_2\left(\frac{1}{\epsilon_\alpha}\right)$. When two QSPA's are performed in parallel with each other the resource cost is governed by:

$$max_{QSPA(P)} = 2 + 2log_2(2N) + 2(m + p) \qquad (7)$$

B. QLSA

The heart of the problem—the QLSA along with the modified circuit construction from [26] requires three registers (disregarding the register which holds the prepared quantum state $|b\rangle$ which is taken into account in the analysis of the QSPA). The first of these registers is the ancilla—on which the conditional rotation about the inverse of the eigenvalues is performed and incurs a cost of a single quantum bit.

The second register is used to store the values inversely proportional to the eigenvalues of matrix A. This register size is determined according to the Newton iteration used to perform the inversion operation with a desired precision ϵ_{inv}.

The determination the third register size is based on the n-bit precision (ϵ_λ) desired to hold the eigenvalues and the probability of the Hamiltonian Simulation being successfully implemented $P = 1 - P_{err}$, where P_{err} represents the probability of failure. The total number of qubits required for the register used to hold the eigenvalues is denoted as t [17].

$$t = log_2\left(\frac{1}{\epsilon_\lambda}\right) + log_2\left(2 + \frac{1}{2P_{err_{QLSA}}}\right) \qquad (8)$$

The cost of implementing the QLSA with the modified rotation is given by:

$$= 1 + 3log_2\left(\frac{1}{\epsilon_{inv}}\right) + log_2\left(\frac{1}{\epsilon_\lambda}\right) + log_2\left(2 + \frac{1}{2P_{err_{QLSA}}}\right) \qquad (9)$$

The construction of the oracle required to simulate A is composed of five quantum registers (see Fig. 3). The first register, used to store the node index of the 1-sparse Hamiltonian, requires a $log_2(2N)$ qubit register.

The second register in the oracle holds the notation for the desired submatrix, of which there can be at most $6d^2$ submatrices. For rectangular grids, this is limited to the maximum number of bands ($N_b = 9$). However, because of the swap operation inherent in the function of the oracle, this register also holds the same respective node index as the first register. For a large A matrix, $log_2(2N) > log_2(2N_b)$, thus the second register size is determined by the size of the first register for sufficiently large A matrices.

The third and fourth registers in the oracle hold the value of the calculated magnitude and phase component of the submatrix element respectively to desired precision. These two registers are assumed to be the same as those used in the QSPA, and are re-used for this oracle.

The fifth register is an ancilla used for the phase shift operation within the function of the oracle.

The cost of implementing the oracle is given by:

$$= 1 + 2log_2(2N) + (m + p) \qquad (10)$$

The summation for the maximum resources required during the QLSA is given by Eq. 11. This includes maintaining two prepared quantum states prepared earlier and their respective ancilla as well as the QLSA and oracle costs.

$$max_{QLSA} = 4 + 4log_2(2N) + m + p + t + 3log_2\left(\frac{1}{\epsilon_{inv}}\right) \qquad (11)$$

C. Swap Test

The quantum swap test is the most simplistic subroutine in the algorithm, requiring only one additional ancilla to perform (see Fig. 5). The swap test requires that both the registers for the prepared state $|R\rangle$ and the solution state $|x\rangle$ be composed of the same number of quantum bits—the register sizes for $|x\rangle$ and $|R\rangle$ need to be of size $log_2(2N)$.

The maximum total resources incurred during the swap test, including the preceding quantum vectors which need to be maintained ($|b\rangle \rightarrow |x\rangle$, $|R\rangle$) and their ancilla, while neglecting the registers which are no longer used becomes:

$$max_{QSWAP} = 4 + 2log_2(2N) \qquad (12)$$

D. QAEA

The QAEA estimates the amplitude associated with a particular value within a Hilbert space. When used on a conditional qubit the Hilbert space is spanned by two distinct states, so the estimation is performed on the particular amplitudes $|0\rangle$ or $|1\rangle$ in these ancillary registers.

The working register size for the QAEA subroutine grows as ϵ, a, and P_{err} become smaller and the cost of the working register for the QAEA subroutine is $log_2(M)$where M is given by Eq. 5.

In previous works approximations for sufficient register sizes were given based on reasonable assumptions. Clader [15] in his supplementary material uses the approximation of $M = 2^{(log_2(1/\epsilon))(log_2(\pi/2\epsilon+\pi))}$ and Scherer [31] $M = 2^{log_2(1/\epsilon^2)}$. These approximations are based on the assumption that $O(\epsilon) \sim O(P_{err})$ and that a is not too small.

Inherent in the operation of the QAEA is the amplitude amplification "engine", which requires an extra working ancilla qubit with which to perform the phase flip (see Fig. 6).

If applications of the QAEA are considered to be sequential, in which case the QAEA working register can be reused post-measurement, and the previous quantum vectors and ancilla from earlier are maintained, then the maximum resources required during this portion of the algorithm follow:

$$max_{QAEA(S)} = 5 + 2log_2(2N) + log_2(M) \qquad (13)$$

where M is given by Eq. 5.

If four QAEA operations are to be computed in parallel, multiple working registers need to be used which bring the maximum requirements for the QAEA phase to:

$$max_{QAEA(P)} = 8 + 2log_2(2N) + 4log_2(M) \qquad (14)$$

E. Bit Precision

Precision of a quantum register is denoted as the d-dimensional Hilbert space spanned by the register, where $d = 2^n$ of an n-bit register. However, decimal digit precision and bit precision are more commonly used where $n = log_2(10^x)$ such that x represents the desired decimal precision and n represents the bit precision. Alternatively, one can calculate the decimal precision by $x = log_{10}(2^n)$ of an n bit register if needed.

IV. APPLICATION

The application of the quantum algorithms requires that the EM scattering FEM be set up in a particular fashion. The resource analysis of the application is separated by algorithm and scaling of computational domain size as well as bit precision is evaluated. Lastly, some considerations regarding error analysis and variable selection are given.

A. Problem Set-up

In order to satisfy the constraints of the QLSA—that the matrix to be inverted must be a sparse linear matrix, there are certain techniques which can be used in the construction of the EM scattering problem. First is the use of rectangular finite elements in the construction of the FEM mesh.

While the use of unstructured grids is more common, using a rectangular mesh will force the sparsity to be at most 7 non-zero elements per row ($d = 7$). It will also force the A matrix to have a maximum of 9 total bands, which can then be deconstructed into 1-sparse linear submatrices and simulated according the method outlined in [22].

Because the locations of the bands are based on the numbering scheme used in the construction of the FEM mesh, they are known, and the A matrix can be efficiently decomposed into at most 9 submatrices. This aids in reducing the time complexity of the Hamiltonian Simulation.

If the A matrix needs to be forced into a Hermitian form, then it will contain a maximum of 7 non-zero elements per row ($d = 7$) and 18 bands — ergo 18 submatrices; the Hamiltonian simulation can be performed using the same methods.

B. Electromagnetic Scattering Problem

The example problem used for the resource analysis will be the EM scattering of the nose cone of a generic aerodynamic cone. While the boundary conditions regarding the problem may be complicated, the construction of the linear system of equations is relatively straightforward.

The mesh used to evaluate the problem is constructed of square elements which compose the computational space of and around the nose cone. The visualization of the mesh over the cone portion used for the calculation can be seen in Fig. 8, and a more refined mesh of the same space in Fig. 9 which scales with the edge count N.

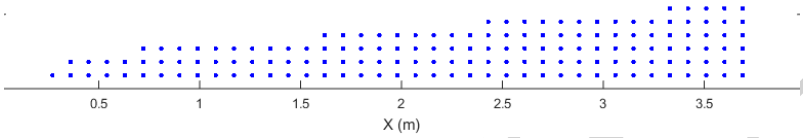


Figure 8. FEM grid points over cone for 50×50 region.

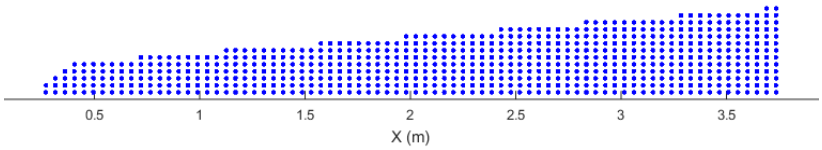


Figure 9. FEM grid points over cone for 100×100 region.

The respective edge counts for the 2-D computational regions are governed by the number of grid points in the x and y directions (n_x and n_y respectively):

$$N = n_x(n_y - 1) + n_y(n_x - 1) \qquad (15)$$

For a square computational region divided by n_i nodes in the x and y direction Eq. 15 can be simplified to $2(n_i^2 - n_i)$. This edge count will be used as the variable N in the resource analysis.

C. Resource analysis

The resource analysis is a function of variables including the edge count of the 200×200 region ($N = 79600$), precision for the registers containing amplitude, phase, eigenvalue, eigenvalue inversion, and amplitude estimation are chosen as 64 bit precision ($\epsilon = 1/2^{64}$). The error probability associated with the QLSA process is chosen as 0.001, the lowest expected amplitude for the QAEA is estimated at $\alpha = 0.01$, and the error associated with the QAEA process is set at $P_{err_{QAEA}} = 0.001$. These parameters are summarized in Table 1.

Table 1. Resource Estimation Variables for the EM Scattering Problem.

Variable	Value
N	79600
ϵ_ϕ	64-bit
ϵ_α	64-bit
ϵ_λ	64-bit
$P_{err_{QLSA}}$	0.001
ϵ_{inv}	64-bit
ϵ_{QAEA}	64-bit
α	0.01
$P_{err_{QAEA}}$	0.001

Given these variables, the maximum numbers of required qubits during each phase of the process are given in Fig. 10 and Table 2. Note, if both the QSPA and QAEA phases of the algorithm are run in parallel, the qubit requirement for the problem does not grow, because the maximal requirement for the QLSA utilizing the oracle supersedes both of the other phases as shown in Table 2. Note that in Fig. 10, the QLSA and swap test are not viable for parallel computation.

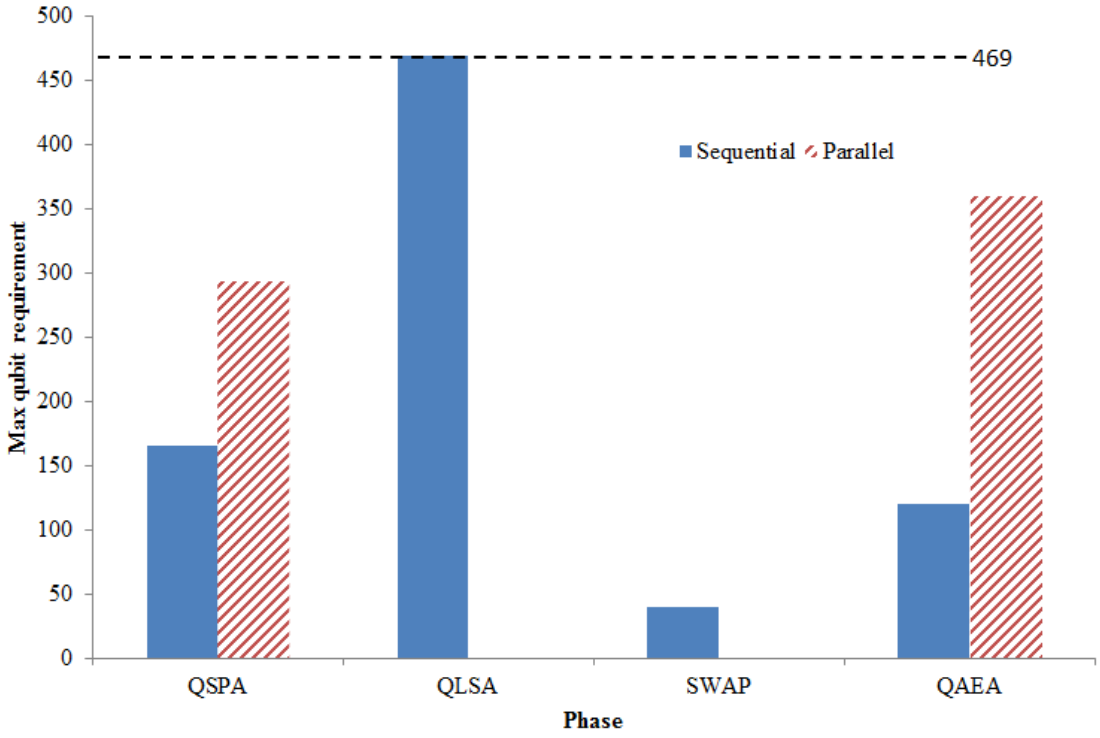


Figure 10. Maximum required resources by algorithm for sequential and parallel computation.

Table 2. Maximum Resource Requirement.

Phase	Max Qubit Requirement	
	Sequence	Parallel
QSPA	166	294
QLSA	469	N/A
SWAP	40	N/A
QAEA	120	360

D. Scaling requirements

When considering the process run in sequence, the variables N, ϵ_ϕ, ϵ_α, ϵ_λ, ϵ_{inv}, and $P_{err_{QLSA}}$ in Eq. 11 become the most relevant. When the QSPA and QAEA process are run in parallel, Eq. 11 still dominates the resource pool, unless the variables ϵ_{QAEA}, α, and $P_{err_{QAEA}}$ in the QAEA specifically are very small.

Considering Eq. 11, as the problem size alone increases, the qubit requirements scales with $4log_2(2N)$. The effect of this scaling is shown in Fig. 11 with the computational regions 50×50, 100×100, 200×200, and 400×400 and their corresponding edge counts according to Eq. 15: 4900, 19800, 79600, and 319200, respectively. Note that the edge count on the x-axis in Fig. 11 is logarithmic in N.

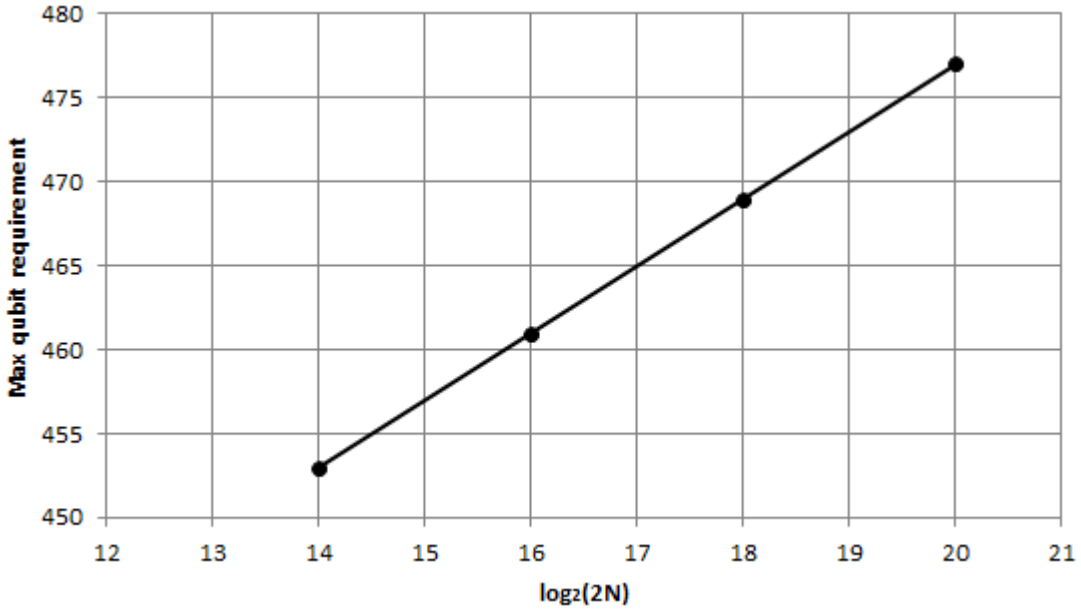


Figure 11. Scaling with edge count $log_2(2N)$.

The resource pool scales directly with the bit precision by a factor of 6. Due to logarithmic compression of the problem size (as related to N), the direct scaling with bit precision has a larger effect on the overall resource pool for the problem. Scaling with common bit precision values 8, 16, 32, 64 and 128 is shown in Fig. 12.

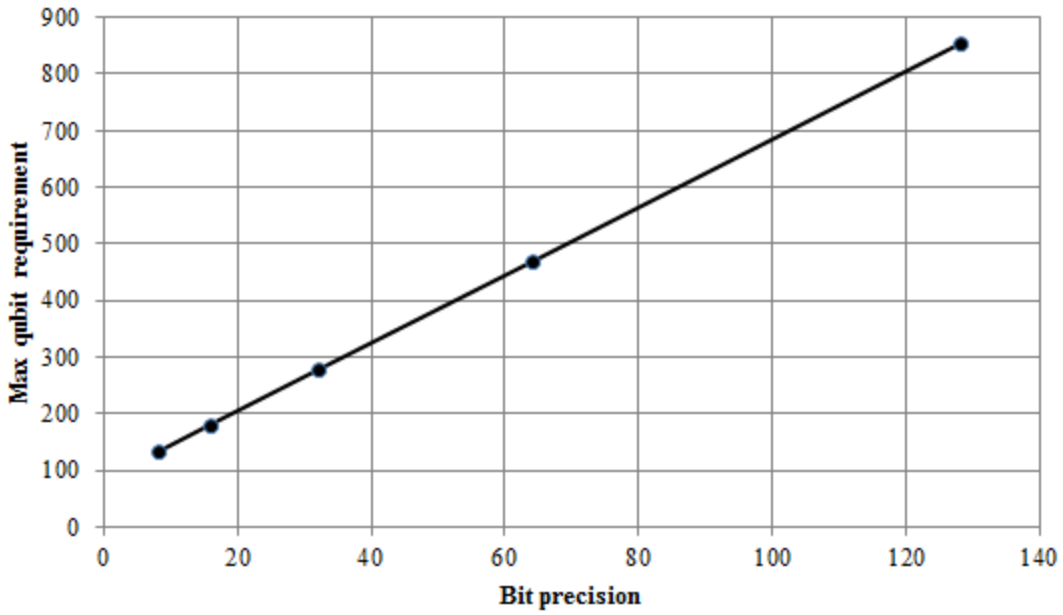


Figure 12. Scaling with bit precision.

E. Error Analysis

In the final calculation of the RCS value from the four applications of the QAEA as given by:

$$\sigma_{RCS} = \frac{1}{4\pi}\frac{N^2}{C_b^2 C_r^2}\frac{sin^2\phi_b}{sin^2\phi_x}(sin^2\phi_{r0} - sin^2\phi_{r1}) \qquad (16)$$

where $sin^2\phi_{r0} \coloneqq P_{1110}^{\frac{1}{2}} sin\phi_r$ and $sin^2\phi_{r1} \coloneqq P_{1111}^{\frac{1}{2}} sin\phi_r$. The constants C_b and C_r are constants associated with the creation of the $|b\rangle$ and $|R\rangle$ vectors, and the sin^2 terms are the probabilities of the states $|b\rangle$, $|R\rangle$, and $|x\rangle$ being created successfully, which are stored in the ancilla of the 2 applications of the QSPA and the rotation ancilla in the QLSA respectively. The terms P_{1110} and P_{1111} are the probabilities that the three states are created successfully and the probability of the swap test ancilla producing a $|0\rangle$ or $|1\rangle$.

If each of the four QAEA measurements are accurate to within $\pm\epsilon a$, the respective values for measured probabilities can be represented by $\tilde{a}_i = a_i \pm \epsilon a_i$. With constants removed, the error in the RCS calculation is given as $\frac{\tilde{a}_1}{\tilde{a}_2}(\tilde{a}_3 - \tilde{a}_4)$ which means the estimate for the RCS calculation follows $|\tilde{\sigma}_{RCS} - \sigma_{RCS}| \leq 3\epsilon_{QAEA}\sigma_{RCS}$ with probability $P_{RCS} \geq \left(1 - P_{err_{QAEA}}\right)^4$. Note the error in the final calculation is not simply the desired precision of a single application of the QAEA [31].

Thus, the precision of the QAEA subroutine should be chosen such that the final error in the RCS calculation is achieved with precision $\epsilon\sigma_{RCS}$ according to $\epsilon_{QAEA} = \frac{\epsilon_{RCS}}{3}$. The desired probability of the success of the QAEA subroutines should be chosen such that $P_{err_{QAEA}} = 1 - \sqrt[4]{P_{RCS}}$ where P_{RCS} is the probability of the RCS calculation being computed from four successful applications of the QAEA.

V. CONCLUSIONS

The results of the study determined that the resource requirement for the application of quantum algorithms for EM scattering is dominated by the QLSA, specifically by the use of 2 working registers needed to create a high probability of extracting the eigenvalues to precision and inverting them, followed by the use of registers needed to store values in a binary precision manner (such as the phase and amplitude). The most efficient resource saving technique is to relax the bit precision, due to the 1: 6 scaling of bit precision to maximum resource pool size.

For problems of applicable size related to calculating EM scattering cross-sections, these same methods can be used in the resource analysis of 3-D EM scattering problems utilizing similar FEM mesh construction and linear systems of equations.

Further research needs to be conducted to reduce the resource requirements of the QLSA and the QSPA. Currently, much of the focus in the field of quantum algorithms is focused at reducing the time complexity of these algorithms under the assumption that the volume of quantum bits will be available eventually. While this assumption allows for creative problem solving, resources are finite, and very small numbers of quantum bits are available today. We suggest that quantum algorithms be developed with resource requirements in mind, so that a balance between space complexity and time complexity can be achieved for practical application sooner rather than later. For fixed numbers of available qubits, a space complexity analysis is necessary to ensure the highest problem accuracy and maximum utilization of resources, notably with respect to quantum subroutines run in parallel.

DISCLAIMER

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

REFERENCES

[1] S. Lloyd, M. Mohseni and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," *arXiv:1307.0411,* 2013.

[2] S. Lloyd, S. Garnerone and P. Zanardi, "Quantum algorithms for topological and geometric analysis of data," *Nature communications,* vol. 7, 2016.

[3] S. Lloyd, M. Mohseni and P. Rebentrost, "Quantum principal component analysis," *Nature Physics,* vol. 10, no. 9, pp. 631-633, 2014.

[4] P. Rebentrost, M. Mohseni and S. Lloyd, "Quantum support vector machine for big data classification," *Physical review letters,* vol. 113, no. 13, 2014.

[5] I. Kerenidis and A. Prakash, "Quantum Recommendation Systems," *arXiv:1603.08675,* 2016.

[6] N. Wiebe, D. Braun and S. Lloyd, "Quantum Data-Fitting," *Physical review letters,* vol. 109, no. 5, 2012.

[7] D. Berry, "High-order quantum algorithm for solving linear differential equations," *Journal of Physics A: Mathematical and Theoretical,* vol. 47, no. 10, 2014.

[8] A. Montanaro and S. Pallister, "Quantum algorithms and the finite element method," *Physical Review A,* vol. 93, no. 3, 2016.

[9] G. Wang, "Quantum Algorithms for Approximating the Effective Resistances in Electrical Networks," *arXiv:1311.1851,* 2013.

[10] L.-C. Wan, C.-H. Yu, S.-J. Pan, F. Gao and Q.-Y. Wen, "Quantum Algorithm for the Toeplitz Systems," *arXiv:1608.02184,* 2016.

[11] S. Barz, I. Kassal, M. Ringbauer, Y. O. Lipp, B. Dakic, A. Aspuru-Guzik and P. Walther, "A two-qubit photonic quantum processor and its application to solving systems of linear equations," *Scientific Reports,* no. 4, 2014.

[12] X. D. Cai, C. Weedbrook, Z. E. Su, M. C. Chen, M. Gu, M. J. Zhu, L. Li, N.-L. Liu, C.-Y. Lu and J.-W. Pan, "Experimental quantum computing to solve systems of linear equations," *Physical review letters,* vol. 110, no. 23, 2013.

[13] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, X. Peng, S. Kais and J. Du, "Experimental realization of quantum algorithm for solving linear systems of equations," *Physical Review A,* vol. 89, no. 2, 2014.

[14] J. Jin, The Finite Element Method in Electromagnetics, 2nd ed., New York: John Wiley & Sons, Inc., 2002.

[15] B. D. Clader, B. C. Jacobs and C. R. Sprouse, "Preconditioned quantum linear system algorithm," *Physical review letters,* vol. 110, no. 25, 2013.

[16] A. W. Harrow, A. Hassidim and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical review letters,* vol. 103, no. 15, 2009.

[17] M. A. Nielson and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University

Press, 2011.

[18] A. Y. Kitaev, "Quantum measurements and the Abelian stabilizer problem," *arXiv:quant-ph/9511026*, 1995.

[19] D. Aharonov and A. Ta-Shma, "Adiabatic quantum state generation and statistical zero knowledge," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003.

[20] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann and D. A. Spielman, "Exponential algorithmic speedup by a quantum walk," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003.

[21] N. Wiebe, D. W. Berry, P. Høyer and B. C. Sanders, "Simulating quantum dynamics on a quantum computer," *Journal of Physics A: Mathematical and Theoretical*, vol. 44, no. 44, 2011.

[22] D. W. Berry, . G. Ahokas, R. Cleve and B. C. Sanders, "Efficient quantum algorithms for simulating sparse Hamiltonians," *Communications in Mathematical Physics*, vol. 270, no. 2, pp. 359-371, 2007.

[23] D. W. Berry, R. Cleve and R. D. Somma, "Exponential improvement in precision for Hamiltonian-evolution simulation," *arXiv:1308.5424*, 2013.

[24] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari and R. D. Somma, "Exponential improvement in precision for simulating sparse Hamiltonians," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, 2014.

[25] D. W. Berry, A. M. Childs and R. Kothari, "Hamiltonian simulation with nearly optimal dependence on all parameters," in *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, 2015.

[26] Y. Cao, A. Papageorgiou, I. Petras, J. Traub and S. Kais, "Quantum algorithm and circuit design solving the Poisson equation," *New Journal of Physics*, vol. 15, no. 1, p. 013021, 2013.

[27] H. Buhrman, R. Cleve, J. Watrous and R. de Wolf, "Quantum fingerprinting," *Physical Review Letters*, vol. 87, no. 16, 2001.

[28] L. K. Grover, "A fast quantum mechanical algorithm for database search," *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212-219, 1996.

[29] G. Brassard, P. Hoyer, M. Mosca and A. Tapp, "Quantum amplitude amplification and estimation," *Contemporary Mathematics*, vol. 305, pp. 53-74, 2002.

[30] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical review A*, vol. 52, no. 4, 1995.

[31] A. Scherer, B. Valiron, S.-C. Mau, S. Alexander, E. van den Berg and T. E. Chapuran, "Concrete resource analysis of the quantum linear system algorithm used to compute the electromagnetic scattering cross section of a 2D target," *arXiv:1505.06552 [quant-ph]*, 2015.

[32] Y. Cao, A. Daskin, S. Frankel and S. Kais, "Quantum circuit design for solving linear systems of equations," *Molecular Physics*, vol. 110, no. 15-16, pp. 1675-1680, 2012.

[33] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM review*, vol. 41, no. 2, pp. 303-332, 1999.

VI. Conclusions and Recommendations

Summary of Research Gap, Research Questions

The research gap for this thesis is the lack of current information regarding space complexity of applied quantum algorithms, notably in the application of EM scattering, which is a problem of interest as stated by the Scientific Review Board of the United States Air Force [9]. The research in this thesis used an example problem in EM scattering with the correct FEM parameters to detail a resource evaluation of the algorithms proper. The four algorithms included are the QSPA, QLSA, Quantum Swap test, and the QAEA. The three research questions posed for this work are concisely answered in the next paragraphs.

Answer to Research-Question 1

The quantum phenomena which need to be understood to study quantum algorithms are much simpler than they might appear. Three main principles are necessary when considering quantum algorithms: superposition, entanglement, and measurement. The ability of a quantum system such as a singular qubit to be able to exist in a state of superposition between two orthonormal vectors (such as the standard computational binary representation of 0 or 1) is the most important property. In order to begin thinking of quantum systems, a singular qubit being able to represent any number between 0 and 1 is important.

The next quantum phenomenon is that of entanglement. This plays a part in the creation of quantum registers (i.e. larger quantum systems). The premise that two or more quantum bits can be entangled to create a register of quantum bits in order to expand the

quantum space from 2^1 to 2^n is vital to understanding the function of a quantum register to store binary values. Although the space of one qubit is technically an infinite superposition between two orthonormal vectors, the measurement basis only allows for two measureable states. The space of n qubits is also infinite, however has 2^n measurable states.

The last phenomena is quantum measurement (or "collapse"), which refers to the collapse of the quantum state from a state of superposition to that of a measurement basis. In the case of a singular qubit, this would mean that when the qubit is measured, it must collapse into one of the two orthonormal vectors $|0\rangle$ or $|1\rangle$. For a register of qubits, the operation is the same and collapses the register from a state of superposition into one of the measurement basis (in this case a binary string). In the measurement of a wave function, any relational information stored in the superposition is lost, so it is the focus of quantum algorithms to save the measurement of the qubit for the last possible step.

Answer to Research-Question 2

In order to prepare an implementation of the QLSA for an EM scattering problem, first there needs to be certain conditions imposed on the creation of the EM scattering model, specifically that of the construction of the A matrix in the system of equations. Given a rectangular FEM grid and the method used to create a Hermitian out of the matrix A, a banded and relatively sparse matrix can be created for the Hamiltonian Simulation.

In order to implement the quantum algorithm, there needs to be a sufficient number of error tolerant quantum bits available. Not only does there need to be a

sufficient number of qubits, but they must be able to maintain quantum states throughout the processing of the algorithm. For the specific application of the QLSA for the calculation of a RCS value, there are other quantum algorithms which are required including the QSPA, the quantum swap test, and the QAEA. Sufficient resources for the implementation of the problem are dependent primarily on the QLSA. The types of quantum bits used in the process described in this thesis are assumed to be reusable, excluding things such as photonic systems.

Answer to Research-Question 3

The resources required for the application of the QLSA to the calculation of an EM scattering cross-section are the maximum of four different quantum algorithms. The maximum, barring any astronomical problem variables in the QAEA, occurs within the QLSA. The QLSA with the completed inversion and rotation algorithm as well as the quantum oracle needed for the Hamiltonian Simulation surmount to the maximum number of resources. For the problem sizes described in this thesis, this amounts to a resource pool of quantum resources in the hundreds of logical qubits.

More detail on the factors contributing to the size of the registers can be found in the journal paper provided in chapter V. Of interest are the reductions in the resource pool by reducing the bit precision of the input and stored values. This dramatically reduces the resource pool as the bit precision is directly related to resource pool size, next is the reduction in the problem size via edge count reduction. Reducing the computational space in which to be simulated, the resource pool also shrinks.

Study Limitations

The scope of this thesis did not allow for an in-depth analysis of logical quantum resources per quantum error correction techniques and assumed that logical entities were available. This work also did not delve into the time complexity of the algorithms, rather the raw number of quantum systems required to perform operations on. The optimization of resources was approached heuristically, and not strictly mathematically.

Recommendations for Future Research

A more thorough analysis of the optimization of the resources necessary for the application of the four algorithms discussed in this work could also be accomplished. The application of quantum algorithms to other real world problems is another area which is ongoing and is receiving attention from the community and could benefit from more space complexity studies such as this one. Thus, resource analysis of other quantum algorithms could be accomplished in a similar manner to the one presented in this work.

Summary

In this thesis the most current and up to date resource analysis of quantum algorithms as applied to a real EM scattering problem size was performed. The results of the analysis show that using current methods allows for the computation of an EM scattering cross-section with a resource pool in the hundreds. As there were many registers and variables involved it became clear that the QLSA became the most resource heavy. The results also showed that the problem size had far less to do with the resource requirements than did the precision of multiple registers needed to store interim values such as those which store the eigenvalues and their inverses, as well as the quantum

registers which are effectively inputs into the system. Dramatic reduction in resources can be accomplished by accepting lower standards of bit precision.

It is the current limitations on quantum computer availability—specifically that of sufficient numbers of quantum bits, which drives the space complexity analysis, and thus the draw to minimize the resource pool for the applications of quantum algorithms. In order to proceed into the era of quantum computing, it is necessary not only to reduce the time complexity of the algorithms themselves (as is the focus of most of the work in this field) but it is also necessary to develop quantum algorithms with space complexity in mind. While it is not clear that minimizing the resource requirements has a detrimental effect on the algorithms' time complexity, in the case that it does—it is still important to strike a balance between register sizes and the run-time associated with using a finite set of quantum resources.

Bibliography

[1] C. J. Riggs, C. D. Lewis, L. O. Mailloux and M. Grimaila, "Understanding quantum computing: a case study using Shor's algorithm," in *Proceedings of the International Conference on Foundations of Computer Science (FCS)*, 2016.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review,* vol. 41, no. 2, pp. 303-332, 1999.

[3] M. A. Nielson and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2011.

[4] D. Mermin, Quantum Computer Science: An Introduction, Cambridge University Press, 2008.

[5] R. J. Lipton and K. W. Regan, Quantum Algorithms via Linear Algebra, The MIT Press, 2014.

[6] A. Pathak, Elements of Quantum Computation and Quantum Communication, Taylor & Francis Inc, 2013.

[7] A. W. Harrow, A. Hassidim and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters,* vol. 103, no. 15, 2009.

[8] A. Scherer, B. Valiron, S.-C. Mau, S. Alexander, E. van den Berg and T. E. Chapuran, "Concrete resource analysis of the quantum linear system algorithm used to compute the electromagnetic scattering cross section of a 2D target," *arXiv:1505.06552 [quant-ph]*, 2015.

[9] "Scientific Advisory Board," [Online]. Available: http://www.scientificadvisoryboard.af.mil/Portals/73/documents/AFD-151214-041.pdf?ver=2016-08-19-101445-230. [Accessed 12 February 2017].

[10] B. D. Clader, B. C. Jacobs and C. R. Sprouse, "Preconditioned quantum linear system algorithm," *Physical Review Letters,* vol. 110, no. 25, 2013.

[11] A. Einstein, B. Podolsky and N. Rosen, "Can quantum-mechanical description of physical reality be considered complete?," *Physical Review,* vol. 47, no. 10, p. 777, 1935.

[12] A. Daskin and S. Kais, "Decomposition of unitary matrices for finding quantum circuits: application to molecular Hamiltonians," *The Journal of Chemical Physics,* vol. 134, no. 14, 2011.

[13] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical Review A,* vol. 52, no. 4, 1995.

[14] S. Barz, I. Kassal, M. Ringbauer, Y. O. Lipp, B. Dakic, A. Aspuru-Guzik and P. Walther, "A two-qubit photonic quantum processor and its application to solving systems of linear equations," *Scientific Reports,* no. 4, 2014.

[15] X. D. Cai, C. Weedbrook, Z. E. Su, M. C. Chen, M. Gu, M. J. Zhu, L. Li, N.-L. Liu, C.-Y. Lu and J.-W. Pan, "Experimental quantum computing to solve systems of linear equations," *Physical Review Letters,* vol. 110, no. 23, 2013.

[16] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, X. Peng, S. Kais and J. Du, "Experimental realization of quantum algorithm for solving linear systems of equations," *Physical Review A,* vol. 89, no. 2, 2014.

[17] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol and D. M. Lucas, "High-fidelity quantum logic gates using trapped-ion hyperfine qubits," *Physical Review Letters,* vol. 117, no. 6, 2016.

[18] A. Y. Kitaev, "Quantum measurements and the Abelian stabilizer problem," *arXiv:quant-ph/9511026,* 1995.

[19] D. W. Berry, . G. Ahokas, R. Cleve and B. C. Sanders, "Efficient quantum algorithms for simulating sparse Hamiltonians," *Communications in Mathematical Physics,* vol. 270, no. 2, pp. 359-371, 2007.

[20] D. W. Berry, R. Cleve and R. D. Somma, "Exponential improvement in precision for Hamiltonian-evolution simulation," *arXiv:1308.5424,* 2013.

[21] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari and R. D. Somma, "Exponential improvement in precision for simulating sparse Hamiltonians," in *Proceedings of*

the 46th Annual ACM Symposium on Theory of Computing, 2014.

[22] D. W. Berry, A. M. Childs and R. Kothari, "Hamiltonian simulation with nearly optimal dependence on all parameters," in *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, 2015.

[23] S. Lloyd, M. Mohseni and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," *arXiv:1307.0411*, 2013.

[24] S. Lloyd, S. Garnerone and P. Zanardi, "Quantum algorithms for topological and geometric analysis of data," *Nature Communications*, vol. 7, 2016.

[25] S. Lloyd, M. Mohseni and P. Rebentrost, "Quantum principal component analysis," *Nature Physics*, vol. 10, no. 9, pp. 631-633, 2014.

[26] P. Rebentrost, M. Mohseni and S. Lloyd, "Quantum support vector machine for big data classification," *Physical Review Letters*, vol. 113, no. 13, 2014.

[27] I. Kerenidis and A. Prakash, "Quantum recommendation systems," *arXiv:1603.08675*, 2016.

[28] N. Wiebe, D. Braun and S. Lloyd, "Quantum data-fitting," *Physical Review Letters*, vol. 109, no. 5, 2012.

[29] D. Berry, "High-order quantum algorithm for solving linear differential equations," *Journal of Physics A: Mathematical and Theoretical*, vol. 47, no. 10, 2014.

[30] A. Montanaro and S. Pallister, "Quantum algorithms and the finite element method," *Physical Review A*, vol. 93, no. 3, 2016.

[31] G. Wang, "Quantum algorithms for approximating the effective resistances in electrical networks," *arXiv:1311.1851*, 2013.

[32] L.-C. Wan, C.-H. Yu, S.-J. Pan, F. Gao and Q.-Y. Wen, "Quantum algorithm for the Toeplitz systems," *arXiv:1608.02184*, 2016.

[33] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear

systems," *NBS,* vol. 49, p. 1, 1952.

[34] Y. Cao, A. Papageorgiou, I. Petras, J. Traub and S. Kais, "Quantum algorithm and circuit design solving the Poisson equation," *New Journal of Physics,* vol. 15, no. 1, p. 013021, 2013.

[35] H. Buhrman, R. Cleve, J. Watrous and R. de Wolf, "Quantum fingerprinting," *Physical Review Letters,* vol. 87, no. 16, 2001.

[36] L. K. Grover, "A fast quantum mechanical algorithm for database search," *Proceedings of the twenty-eighth annual ACM symposium on theory of computing,* pp. 212-219, 1996.

[37] G. Brassard, P. Hoyer, M. Mosca and A. Tapp, "Quantum amplitude amplification and estimation," *Contemporary Mathematics,* vol. 305, pp. 53-74, 2002.

[38] Y. Cao, A. Daskin, S. Frankel and S. Kais, "Quantum circuit design for solving linear systems of equations," *Molecular Physics,* vol. 110, no. 15-16, pp. 1675-1680, 2012.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
23-03-2017	Master's Thesis	March 2016 – March 2017

4. TITLE AND SUBTITLE

Resource Evaluation of Quantum Linear Systems Algorithm for Application to Electromagnetic Scattering Problems

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Riggs, Casey J.R., 2nd Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENV-MS-17-M-216

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. Ojas D. Parekh
Sandia National Laboratories
1515 Eubank
Albuquerque, NM 87123
odparek@sandia.gov
(505)844-0287

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Current limitations in quantum computing technology does not allow for very large applications of quantum algorithms, and it is the nature of quantum algorithms not only to be able to solve problems of interest much more quickly than classical means but also to do so with less resources which makes them so promising! One such problem of interest is the application of the Quantum Linear Systems Algorithm, along with a few other subroutines, to the calculation of an electromagnetic scattering cross-section via finite element methods. This work composes a resource analysis of the algorithm as well as required subroutines and details the primary contributors to the resources involved as well as methods to decrease these resource requirements.

15. SUBJECT TERMS

Quantum Linear Systems Algorithm, Resource Analysis, Electromagnetic Scattering, Quantum Algorithms

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Major Logan O. Mailloux, AFIT/ENV
U	**U**	**U**	**UU**	93	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext. 3348 Logan.Mailloux@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18