Actuarial Innovation and Technology

Quantum Computing and its Impact on Actuarial Modeling

October | 2023

SOA Research INSTITUTE

Quantum Computing and its Impact on Actuarial Modeling

AUTHORS Kailan Shang, FSA, CFA, PRM, SCJP

SPONSOR Actuarial Innovation and Technology Strategic Research Program Steering Committee

CONTENTS

Quantum Computing and its Impact on Actuarial Modeling

Executive Summary

The complexity of actuarial modeling keeps growing given the broad guarantees provided by insurers, the heightened market volatility and correlations, and the regulatory reforms that require more comprehensive, and usually stochastic, analysis. On the other end, insurers are dealing with millions of policies with different features and possible paths given both financial risk and insurance risk. Although techniques such as cloud-based solutions and parallel computing are used to meet the requirements, usually approximation is needed to perform the analysis in a meaningful timeline. Those approximations may only be assessed once at an initial stage to validate their appropriateness but, in many cases, they become inappropriate without proper reassessments, which can be too time-consuming to conduct with existing technologies.

The fundamental breakthrough of computing technology, such as quantum computers, has a potential to solve many of the computationally intensive issues faced by actuaries. Quantum computers use qubit, an innovative way of representing data to allow the uncertainty of outcomes. Unlike a bit with two possible values (0 and 1) in a classical computer, a qubit in a quantum computer can exist in multiple states simultaneously at any given time, like flipping a coin. Until the coin finally lands on an object, the outcome is unknown. This design, together with quantum algorithms, allows speedup, even exponentially, in solving issues in different fields, such as cryptography, medical research, material science, and finance. In addition to cyber security, which basically affects almost all industries, optimization, simulation, and machine learning are three major areas where the insurance industry could potentially benefit from quantum computing.

- The design of quantum computing and quantum computers makes them suitable for Monte Carlo simulation, which is a key component in many actuarial works, and is computationally intensive. For many options offered in insurance products, given the path dependent payoff functions and the dynamic interaction with policyholder behaviors and management actions, Monte Carlo simulation is considered an appropriate approach to evaluate the risks without approximation. Quantum computing has a quadratic speedup compared to classical algorithms in Monte Carlo simulation. It can potentially be used for embedded option pricing to get the fair value that is expected to match the market price assuming no arbitrage opportunities, reserve calculation and capital management based on value at risk (VaR) and conditional tail expectation (CTE), asset valuation, hedging and risk management.

- Some optimization problems in the insurance fields are multiperiod high-dimensional problems that require long optimization time and approximation to reduce the complexity. Optimization using quantum algorithms can potentially have a polynomial speedup for certain optimization problems. Strategic asset allocation and asset liability management may benefit from quantum computing. Strategic asset allocation optimization to support an underlying liability portfolio can be computationally intensive. The number of possible asset allocation plans can be too large to test in a reasonable time framework. Quantum computing can be potentially used for both simulation and optimization in this process to take advantage of the speedup in both areas. In asset liability management, liability may be replicated using a pool of assets that can match the value, cash flows,

and sensitivities to key risks, such as interest rate risk, equity risk, and credit risk. Ideally, the full distribution can be taken into consideration in the replicating process. With the help of quantum optimization, more matching targets, including the results under each individual scenario, can be included to improve the liability replicating portfolio.

- Quantum machine learning is expected to be suitable for predictive modeling tasks involving large amount of data in terms of not only number of records, but also number of variables. Some quantum algorithms, such as quantum k-means clustering, already showed speed advantages compared to classical algorithms. In the insurance industry, given the increasing volume of available data and data type, such as social media data, efficient predictive modeling algorithms can be used to increase speed of model training, improve model accuracy, and enhance scalability. The areas in which quantum algorithms can undergo significant improvement usually involve data and decision-making at the individual policy level, such as insurance underwriting, pricing, marketing, valuation, risk management and data compression.

Although quantum computing has great potential, it is still in the stages of development and adoption, and there are currently only a few companies and research institutions that have the capability to develop and use quantum computers. The timeline for the application of quantum computing to the insurance industry is difficult to predict with certainty, as it depends on the pace of technological development, as well as the adoption and implementation of these technologies by the industry, but it is certain that the technology has the potential to significantly impact the industry, and continued interest and exploration are expected in this area in the coming years at an accelerated rate. Some predicted that quantum computing was only a few years away from use in markets. Asset pricing, asset allocation, and risk management are the insurance related areas in which applications of quantum computing exist.

While insurance professionals depend on quantum technology to continue advancing, they can embrace the future through identifying areas for applications of quantum computing, implementing quantum algorithms to insurance applications and, perhaps, contributing to developing new algorithms run on quantum computers to solve insurance related problems.

Section 1: Introduction

Quantum computing has been gradually applied to the finance field, with applications to portfolio optimization and diversification, option pricing, and credit risk analysis. On the other hand, the complexity of insurance products justifies the applications of quantum computing in many areas where conventional computers with CPUs and GPUs cannot guarantee a full exploration of the entire space of strategies. Experiments on insurance applications only started recently. Berry and Sharpe (2021) demonstrated the use of quantum computers for asset liability management (ALM) and concluded that the discovery of the strange world of quantum mechanics has the potential to create investment management efficiencies. However, direct applications to the insurance and actuarial field are rare, if any. Many problems in the insurance field are solved with shortcuts using conventional computers that very likely lead to suboptimal solutions. With the aid of quantum computing, improvements in optimization of strategies may be found, which have the potential to reduce capital requirements, enhance risk-adjusted investment performance, lower premiums, and increase retirement incomes.

To narrow the knowledge gap of insurance professionals, especially actuaries, this research tries to introduce quantum computing to actuaries. Ideally, it will build the bridge between actuarial knowledge and quantum computing and explain the important concepts of quantum computing using a numerical example that actuaries are familiar with. In addition, this research will assess and identify areas in the insurance field where quantum computing can make a difference, accompanied with suggested algorithms. Although quantum computers may not be ready for large-scale commercial use yet, it is beneficial to have actuaries get involved in quantum computing as soon and easy as possible because they are the implementers of many complex calculations in the insurance industry for which quantum computing can help.

An example where quantum computing can help is liability-driven investment (LDI) optimization. A common issue in dynamic LDI optimization is the curse of dimensionality. For example, in a four-asset-class space, the construction of an efficient frontier can be challenging. With 51 possible dividers between 0% and 100%, using an increment of 2%, it requires 23,426 static asset plans to be tested to get the full efficient frontiers. And, in the dynamic world, the asset allocation path is considered so that asset mix can be nonconstant, $23{,}426^{40}$ asset allocation paths need to be tested to determine the best one using the grid-searching approach, assuming quarterly rebalancing for 10 years. This is out of reach for conventional computers, but can be handled in a more efficient way using quantum computing.

We proceed as follows:

- Section 2 (Introduction of Quantum Computing) provides a general overview of quantum computing, including basic concepts such as qubits, quantum states, entanglement, quantum circuits, and measurement. It also discusses the differences among quantum computers, CPUs, and GPUs, emphasizing the type of problems for which each computing technique excels.
- Section 3 (Quantum Computing Insurance Applications) explores different areas of the actuarial/insurance field and identifies possible usages of quantum computing in the future.
- Section 4 (Road to Large-Scale Applications) studies promising real-world applications, existing activities in the banking and insurance industry related to quantum computing, and a possible timeline for large-scale applications. It also discusses other promising alternatives to quantum computers.
- Section 5 (Further Developments) discusses potential extensions of this research to reach the goal of applying quantum computing to the insurance industry.
- Section 6 (Conclusion) summarizes the key points of this research and concludes the main body of the report.

- Appendix (Quantum Computing Example: European Equity Index Option Pricing) provides details on pricing a European equity index call option using quantum algorithm amplitude estimation. The goal is to show how to implement quantum computing leveraging on existing technologies and tools. Open-Source codes are included and documented for illustration purpose.
- The end notes provide more numerical details of the examples used in section 2.

Section 2: Introduction to Quantum Computing

Quantum computing is a field of computing that uses quantum-mechanical phenomena to perform operations on data. In classical computing, data is represented in bits that are either 0 or 1, whereas, in quantum computing, data is represented in quantum bits or qubits, which can be in a superposition of states and can be entangled with other qubits. Properties of quantum mechanics, such as superposition and entanglement, allow quantum computers to perform certain operations faster than classical computers. For example, a quantum computer can factor large numbers exponentially faster than a classical computer, which is important for applications in cryptography.

Before introducing the key concepts used in quantum computing, it is helpful to compare quantum computers with two currently popular computing technologies: Central Processing Unit (CPU) and Graphics Processing Unit (GPU).

2.1 COMPUTING ARCHITECTURES

CPU, GPU, and quantum computers are different types of computing architectures, and they differ in how they process information and the types of problems they can solve.

- A CPU is a general-purpose processor designed to perform a wide range of tasks, such as running applications, managing the operating system, and executing instructions. It is optimized for sequential processing and typically has a small number of cores with high operating speeds.
- A GPU is a specialized processor designed to handle complex graphical operations, such as rendering 3D graphics and video processing. It is optimized for parallel processing and typically has thousands of small processing cores that can perform multiple calculations simultaneously.
- A quantum computer is a type of computer that uses quantum mechanics to process information. Unlike classical computers, which process information in binary digits (bits), quantum computers use quantum bits (qubits) that can exist in multiple states simultaneously. This property allows quantum computers to perform certain computations much faster than classical computers, particularly for problems involving large datasets or complex calculations.

On the algorithm side, CPUs and GPUs are based on classical computing principles and are designed to solve problems using classical algorithms, whereas quantum computers are based on quantum mechanics and are designed to solve problems using quantum algorithms.

In terms of performance, quantum computers have the potential to perform certain calculations exponentially faster than classical computers, while GPUs excel at parallel processing and can perform certain calculations faster than CPUs. Quantum computing has the potential to solve certain problems much faster than classical computers, particularly those involving large datasets or complex calculations that are difficult or impossible for classical computers to solve. Some of the areas where quantum computing is expected to have the biggest impact compared to classical computing architectures include:

- Cryptography: Quantum computers are expected to be able to break many of the encryption schemes currently used to protect sensitive data, such as credit card information, online communications, and financial transactions. At the same time, quantum computers can also be used to create new forms of encryption that are more secure than classical encryption.
- Optimization: Many real-world problems, such as logistics and scheduling, involve finding the best possible solution from a large number of possible options. Quantum computers are expected to be able to solve optimization problems much faster than classical computers, which could have significant applications in fields such as finance, transportation, and logistics.

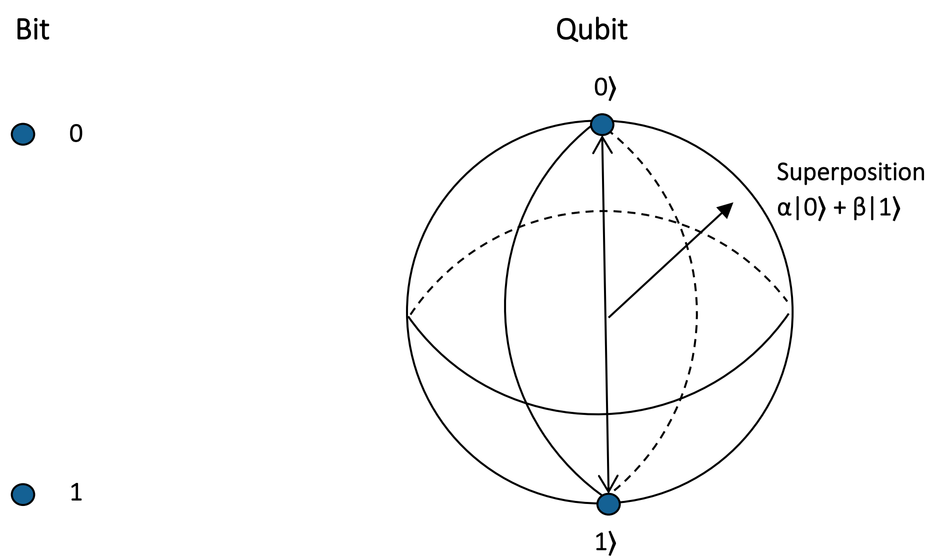
- Simulation: Quantum computers can simulate quantum systems much more efficiently than classical computers, which could have significant applications in fields such as chemistry, materials science, and drug discovery. By simulating the behavior of molecules and materials at the quantum level, quantum computers could help scientists develop new materials, drugs, and chemicals much faster than with classical computers.
- Machine learning: Quantum computers can be used to speed up certain types of machine learning algorithms, which could have significant applications in fields such as natural language processing, computer vision, and speech recognition.

Quantum computers are still in the early stages of development and are not yet widely available for practical use, but they have the potential to revolutionize many areas in the future. It is also noted that, to solve real-world problems, Quantum computers may be used together with CPUs to make the best of the two architectures.

2.2 QUBITS

A quantum bit, known as a qubit, is a basic unit of information used in quantum computing. Unlike classical bits, which can only be in one of two states (0 or 1) at any given time, qubits can exist in multiple states simultaneously, a property known as superposition, as shown in figure 1.

Figure 1
BIT VS. QUBIT



Note:

α and β: the amplitudes of the qubit being in the 0 or 1 state, respectively. They can be real numbers or complex numbers with both real and imaginary components. In quantum mechanics, the probability of measuring a particular outcome is given by the square of the absolute value of the amplitude corresponding to that outcome. The probability of measuring the qubit in the state $|0\rangle$ is given by $|\alpha|^2$, and the probability of measuring the qubit in the state $|1\rangle$ is given by $|\beta|^2$.

Superposition is a concept that means when objects, such as waves and particles, meet each other and interact, different results may occur based on the circumstances. It is related to the famous Schrödinger's Cat paradox that a hypothetical cat may be considered simultaneously both alive and dead (state 0 or 1) when it is in a closed box and cannot be observed. The final result depends on a random event that may or may not occur. The probability of being alive and dead may be known depending on the setup in the closed box, but the outcome is uncertain and unknown until the box is opened.

You may consider superposition as a special case of a linear combination in algebra. In quantum mechanics, a physical system can be in a state that is a linear combination of two or more basis states. This is known as a superposition state. For example, a qubit can be in a superposition state of both 0 and 1 at the same time[1]. Using bits, which can only have one definite value (0 or 1), a traditional computer solves a problem by sequential calculations with a definite path, until it restarts with another sequence of calculations and the solution is found. With qubits and superposition, quantum computers can test all the paths at once and find the shortcut to the solutions. With qubits, parallelism is performed by design. Assume that we have two bits in our computer. They can be in four possible states (00, 01, 10, or 11), but in only one of them at any given time in classical computers.

In a quantum computer, two qubits can also represent the exact same four states (00, 01, 10, or 11). However, the qubits can represent all four states at the same time because of superposition. For many calculations, it is similar to having four regular computers running side-by-side.

The benefit of using qubits increases exponentially with the number of qubits. When n bits are used by a classical computer, it can still only deal with one state at a time for each bit and the benefit grows linearly. However, n qubits are available in a quantum computer, so 2n states can be represented simultaneously. The largest quantum computer built so far, named Dubbed Osprey, has 433 qubits. With 2433 possible states in parallel, it represents 2.2 X 10130 possibilities.

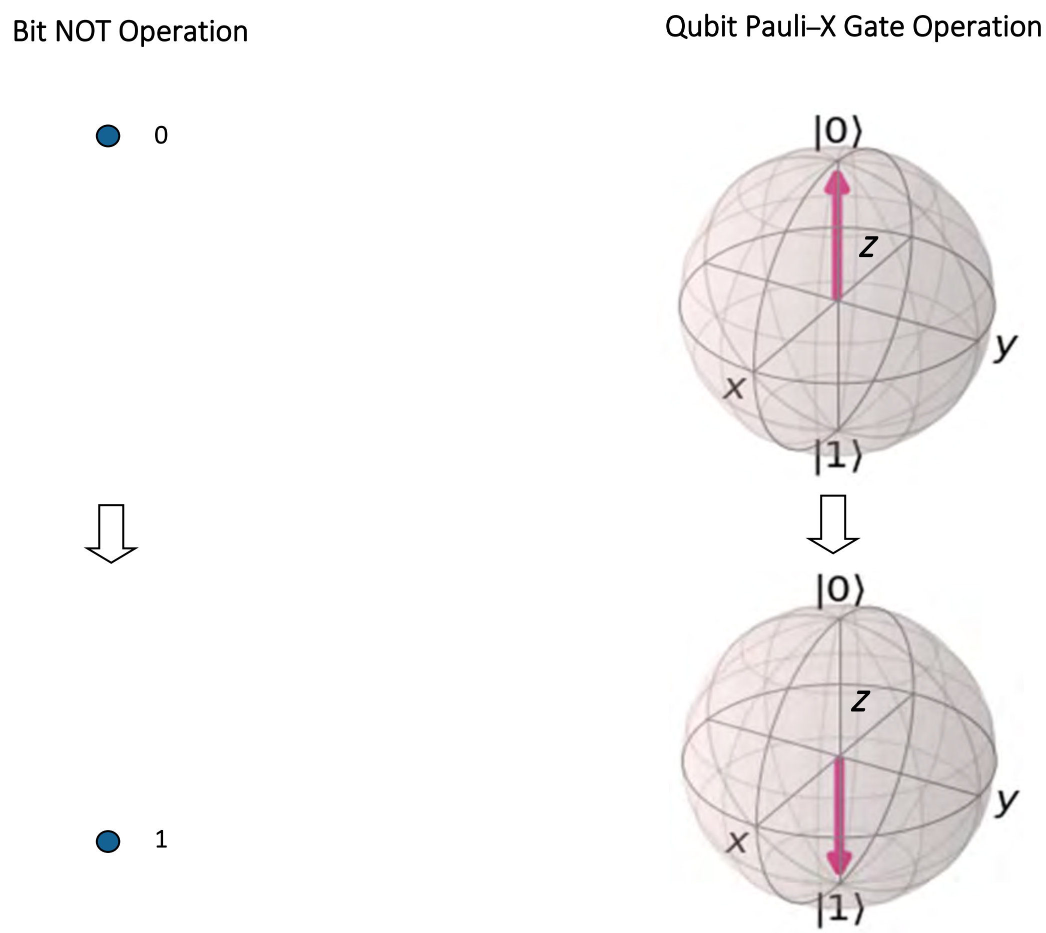
2.3 OPERATION, MEASUREMENT AND CIRCUIT

In quantum computing, operations and measurements are two important types of quantum processes that are used to manipulate and extract information from quantum states.

A quantum operation, also known as a quantum gate, is a mathematical transformation that acts on a quantum state to modify it in a specific way. Quantum gates are used to manipulate the quantum state of qubits, and they can be used to perform various tasks, such as creating superposition states, entangling qubits, or implementing quantum algorithms.

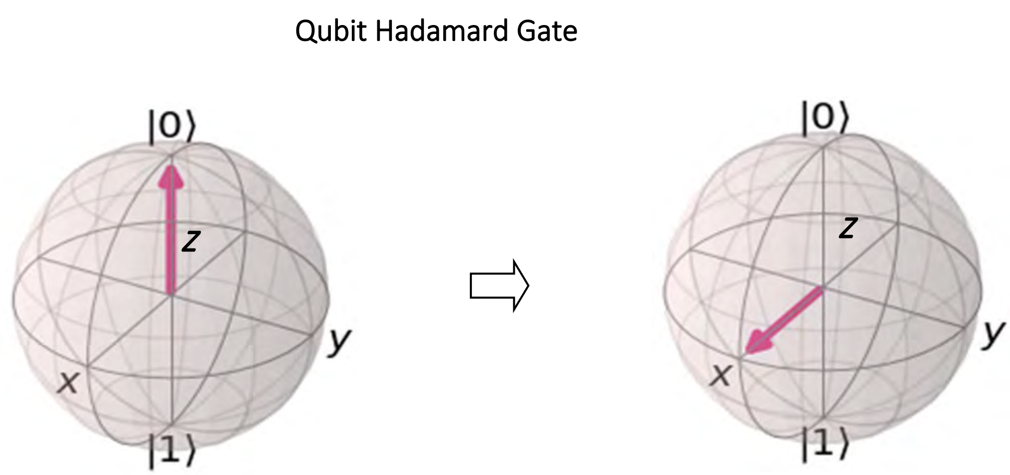
Figure 2 illustrates the Pauli-X gate[2] operation, which is similar to the NOT operation in classic systems that transforms the state of 0 to the state of 1.

Figure 2
BIT VS. QUBIT NOT OPERATION



If qubits are considered as vectors, a quantum gate can be considered as matrices that will be multiplied by the vectors to transform the qubits to a new superposition state. Other more complicated gates, such as Hadamard gate[3], change a basic state to a superposition state. Figure 3 shows the resulting state after applying Hadamard gate to the basic state 0⟩.

Figure 3
QUBIT HADAMARD GATE



Another important gate in quantum computing is Controlled NOT gate[4], also known as CNOT gate. It involves two qubits, with the first qubit as a target qubit whose state will be changed based on the second qubit, which is a control qubit. The CNOT gate is a crucial gate in quantum computing, as it enables the creation of entangled states and the implementation of various quantum algorithms, such as quantum error correction.

There are many other types of quantum gates as well, each with its own specific operation on qubits. Quantum gates are important in quantum computing because they allow us to manipulate the state of qubits and perform operations on quantum systems, which is the basis for quantum algorithms and quantum computation.

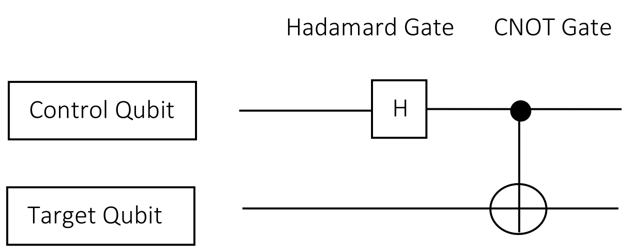
After quantum operations, qubits are still in superposition with unknown outcomes until they are measured. A quantum measurement is a process by which information is extracted from a quantum state. When a quantum system is measured, it collapses into one of its basis states ($|0\rangle$ and $|1\rangle$ for a single qubit) with a probability that is proportional to the square of the amplitude of that state (α and β are the amplitudes for $|0\rangle$ and $|1\rangle$, respectively). Unlike quantum gates that transform the state of qubits, measurement is a destructive process, meaning that the quantum state is irreversibly altered by the measurement operation. In quantum computing, measurements are used to extract classical information from quantum states, such as the outcome of a quantum algorithm or the value of a qubit in a particular basis state. Measurements are also used to perform error correction and to verify the correctness of quantum computations.

Quantum operations and measurements are both important tools for manipulating and extracting information from quantum states in quantum computing. Quantum operations are used to manipulate the quantum state in specific ways, while measurements are used to extract information from the quantum state by collapsing it into a classical state.

Based on quantum operations, a quantum circuit is a mathematical model used to describe and analyze the behavior of a quantum system, such as a quantum computer. It is composed of a set of quantum gates, which are operations that act on qubits to manipulate their quantum state. Quantum circuits define the quantum computation, which involves not only the quantum gates, but also measurement and the initialization of qubits.

In a quantum circuit, qubits are represented as lines, and quantum gates are represented as boxes that act on one or more qubits. The state of the qubits is represented as a vector in a high-dimensional complex space, known as the Hilbert space, with examples of both a single qubit and two qubits as described earlier. The quantum gates act on the qubits by multiplying the state vector with a unitary matrix that corresponds to the gate operation. An example of a quantum circuit is provided below to describe a Bell state. The Bell state is a maximally entangled two-qubit state in quantum computing, and it can be created using a circuit that consists of a Hadamard gate and a CNOT gate, as shown in figure 4.

Figure 4
SAMPLE QUANTUM CIRCUIT FOR BELL STATE



We start with two qubits, labeled 'control qubit' and 'target qubit,' both initially in the $|0\rangle$ state. A Hadamard gate is applied to the control qubit. This creates a superposition of the $|0\rangle$ and $|1\rangle$ states on the first qubit. A CNOT gate is then applied to the control qubit and target qubit. This creates the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$[5], which is a maximally entangled state of two qubits, as explained in the following section.

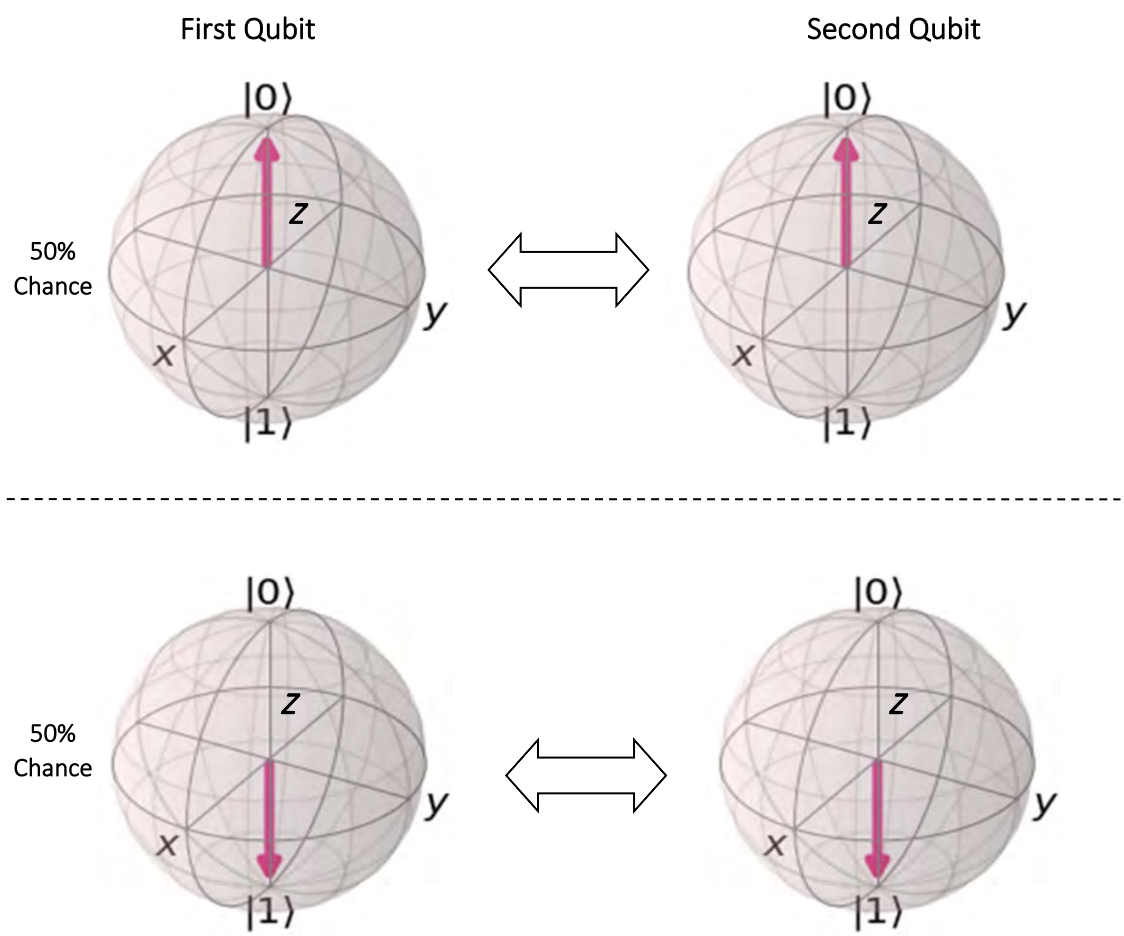
Quantum circuits can be used to implement quantum algorithms, which are sets of instructions that perform a specific computation using a quantum computer. Quantum algorithms typically involve a sequence of quantum gates that manipulate the quantum state of the qubits in a specific way, allowing the quantum computer to solve a particular problem faster than a classical computer.

Quantum circuits are also an important tool for understanding the behavior of quantum systems and developing new quantum algorithms. They are also used in the design and optimization of quantum hardware, as the implementation of quantum gates is a crucial component of building a functional quantum computer.

2.4 ENTANGLEMENT

In addition to parallelism of qubits discussed in section 2.2, entanglement is another fundamental, but also mystic, concept in quantum computing. It describes a phenomenon where two or more qubits can become correlated in such a way that their states are no longer independent of each other. In an entangled system, the state of one qubit cannot be described independently of the state of the other qubits in the system. This means that if one qubit is measured, the state of the other qubits can be determined with a high degree of accuracy, even if they are far apart from each other. An example of entangled states is the Bell state described in the previous section. In that example, the superposition of the state is given as $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. As illustrated in figure 5, for the first qubit, it has an equal probability of being measured as |0⟩ or |1⟩. However, if the first qubit is measured as |0⟩, then the second qubit will also be |0⟩. Similarly, if the first qubit is measure as |1⟩, the second qubit will have the same measured results. This is because only two possible states are available in this Bell state: |00⟩ and |11⟩.

Figure 5
SAMPLE BELL STATE



Despite its many useful applications, entanglement can be a challenging concept to understand, as it defies our intuition and violates some of our everyday understanding of causality and locality. You may think about it as two variables that are perfectly correlated to each other. If one variable has a definite value, the value of the other variable will be known for sure. However, two things that make it difficult to understand are:

- The variables are random variables. Before they are measured, their values are unknown and random.
- The variables are far away from each other with no communication and hidden rules among them.

For simplicity, we may consider entanglement as two random variables that are perfectly correlated, knowing that this feature can be used in quantum computing. Entanglement is a powerful tool in quantum computing because it allows for the construction of quantum circuits that can perform certain operations much faster than classical circuits. Entanglement can help speed up quantum computation with increased parallelism. Entanglement allows for a type of parallelism that is not possible in classical computing. In classical computing, each bit can only represent a 0 or a 1 at any given time. However, in quantum computing, multiple qubits can exist in a superposition of states simultaneously. When these qubits are entangled, the state of one qubit can influence the state of another, allowing for a much greater degree of parallelism. Entanglement can also reduce the computational complexity of certain types of problems. Entanglement can allow quantum computers to explore multiple paths simultaneously, enabling them to find the solution to a problem more quickly than classical computers.

One well-known example of how entanglement can speed up quantum computation is in the use of Grover's algorithm for searching an unsorted database, as discussed in section 2.6. Grover's algorithm uses entanglement to amplify the probability of finding the correct answer to a search problem. This algorithm can provide a quadratic speedup over classical algorithms, meaning that it can find the answer to a search problem in a small portion of the time it would take a classical computer.

2.5 SPEED ADVANTAGE

Quantum computing can speed up certain types of calculations because it can perform certain operations much faster than classical computers. One of the fundamental differences between classical and quantum computing is that classical computers process information in a binary format (0s and 1s), whereas quantum computers use quantum bits, or qubits, which can exist in a superposition of states. This means that a qubit can represent both 0 and 1 at the same time, allowing quantum computers to perform multiple calculations simultaneously.

In addition to this parallelism, quantum computing can also leverage quantum entanglement, a phenomenon where two or more qubits become correlated in such a way that the state of one qubit affects the state of the others, as explained in section 2.4. This allows quantum computers to perform certain calculations that are difficult or impossible for classical computers, such as factoring large numbers, searching large databases, and simulating quantum systems.

For example, the well-known Shor's algorithm, as explained in section 2.6, which is a quantum algorithm for factoring large numbers, can solve this problem exponentially faster than the best-known classical algorithms. This has important implications for cryptography, as many encryption schemes rely on the difficulty of factoring large numbers.

However, not all problems can be sped up using quantum computing and the level of speedup is not always exponential. With new progress in developing quantum algorithms and improving the performance of quantum hardware, more areas may benefit from the speedup in the future.

2.6 KEY ALGORITHMS

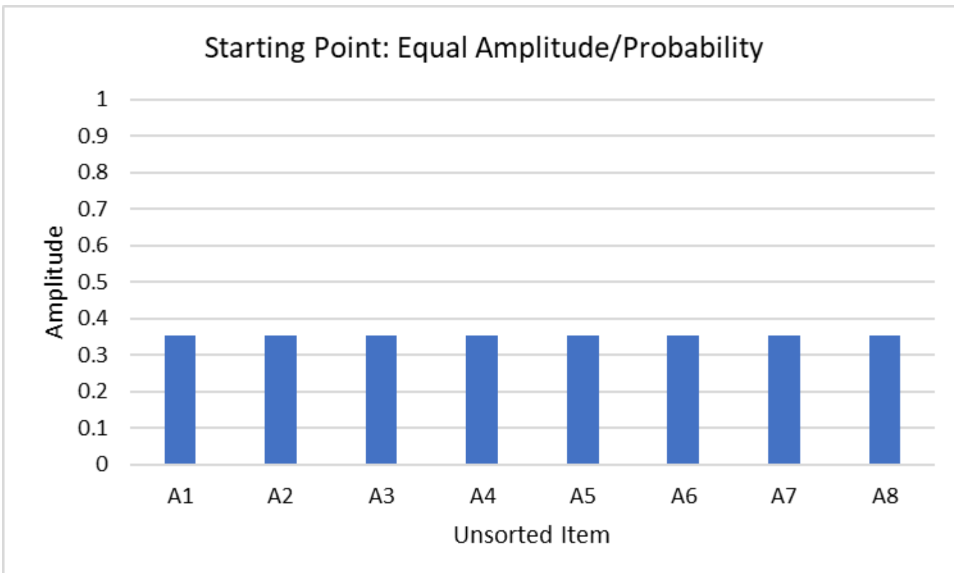
The success of quantum computing is based on not only quantum computers, but also quantum algorithms that utilize the hardware in an efficient way. Several key algorithms in quantum computing have been developed to achieve superior speed compared to classical algorithms.

- **Grover's algorithm**: Assume that we want to search among N unsorted items for those with a specific value. In a classical computer with bits, the time required grows linearly with N using classical algorithms by comparing each item with the desired value. In other words, the task can be finished in $O(N)$ time. Here the big O notation represents the time complexity of the algorithm. In a quantum computer, searching an unsorted database of N elements can be achieved in $O(\sqrt{N})$ time, which is quadratically faster than the best classical algorithms. This quantum algorithm for searching an unsorted database was developed by Grover (1996).

The algorithm works by using a quantum operation to assign equal amplitude and, therefore, probability in the unsorted database, as shown in figure 6. It then uses an oracle operator to mark the desired item and change the sign of its amplitude, as shown in figure 7. In this example, item *A5* has the desired value. The average of the amplitudes is reduced due to the flipping of the sign. The next step is to apply a quantum operator called the Grover diffusion operator, which increases the amplitude of the desired item and reduces the amplitudes of the other items. The new amplitude of each item is the reflection of the current amplitude against the average amplitude calculated in the previous step, as shown in figure 8.

Figure 6

GROVER'S ALGORITHM ILLUSTRATION: STARTING POINT



Note:

Probability = Amplitude^2

Figure 7

GROVER'S ALGORITHM ILLUSTRATION: SIGN CHANGE OF TARGET ITEM (FIRST ITERATION)

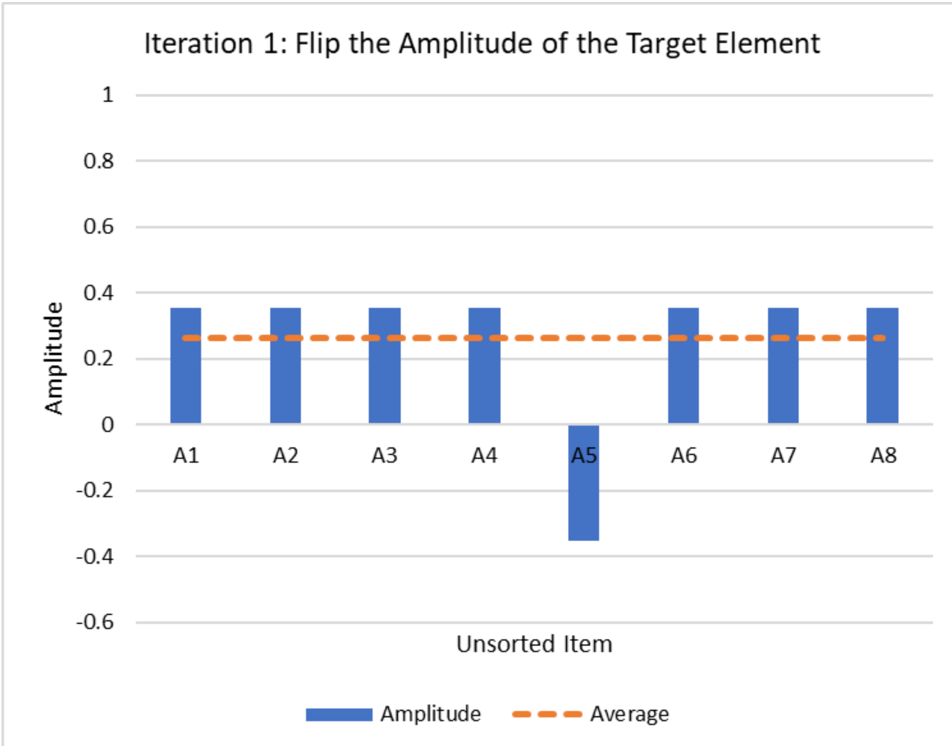
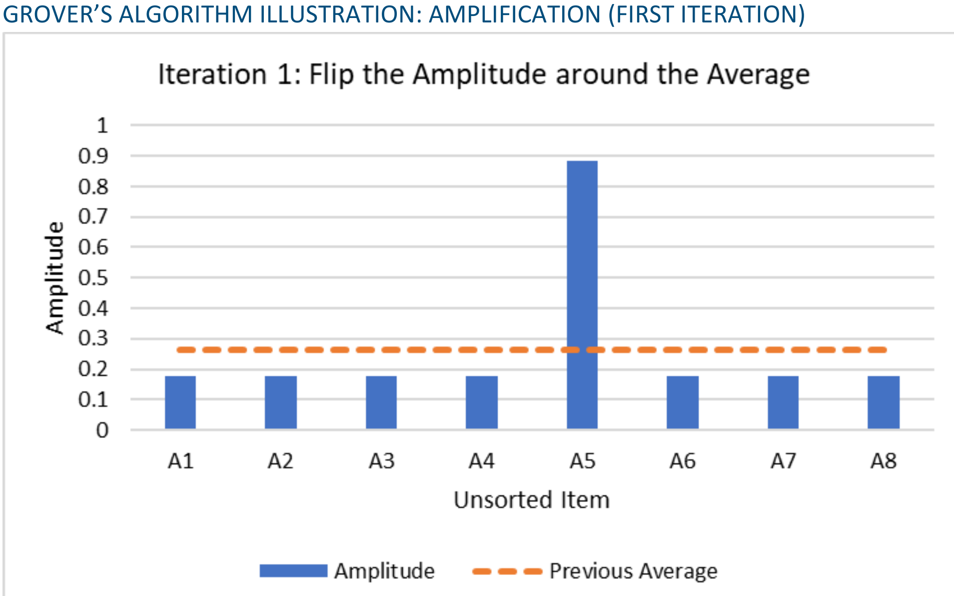


Figure 8

GROVER'S ALGORITHM ILLUSTRATION: AMPLIFICATION (FIRST ITERATION)



The process, including sign flipping and amplification, can be repeated $O(\sqrt{N})$ times to have a high amplitude for the desired item(s) and dampened amplitudes for others, allowing the desired item to be identified with high probability. The second iteration is illustrated in figures 9 and 10. More details of this example can be found in the end note[6].

Figure 9

GROVER'S ALGORITHM ILLUSTRATION: SIGN CHANGE OF TARGET ITEM (SECOND ITERATION)

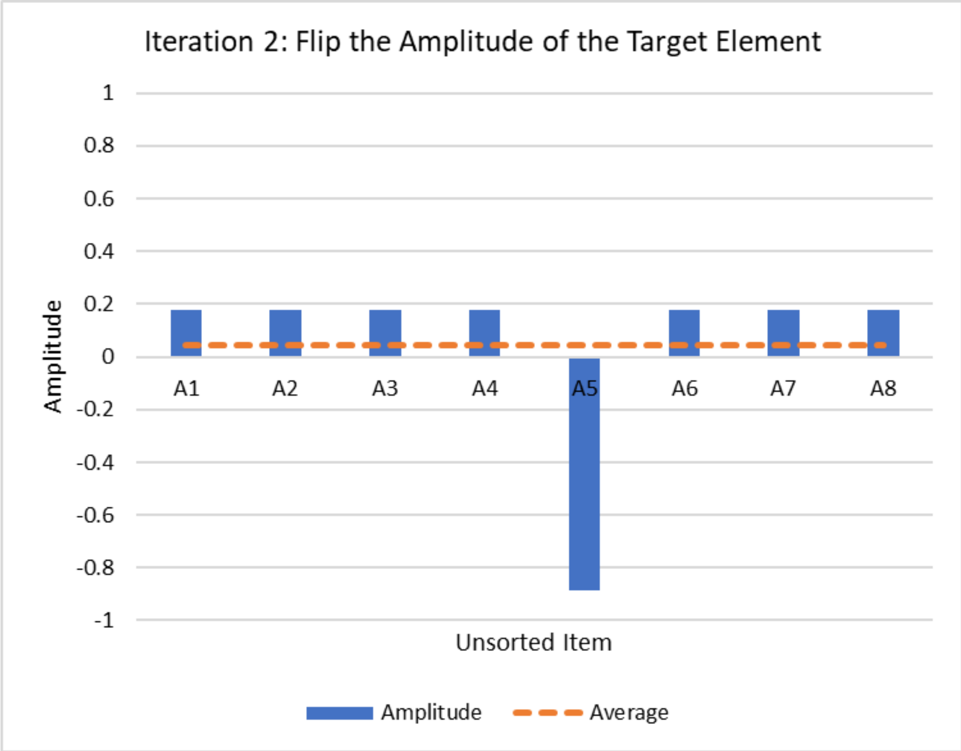
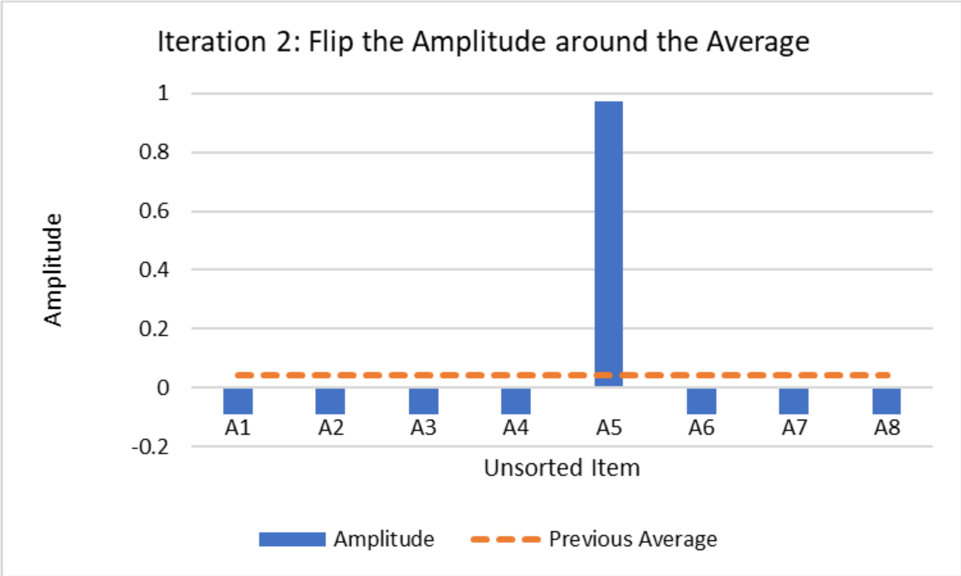


Figure 10

GROVER'S ALGORITHM ILLUSTRATION: AMPLIFICATION (SECOND ITERATION)

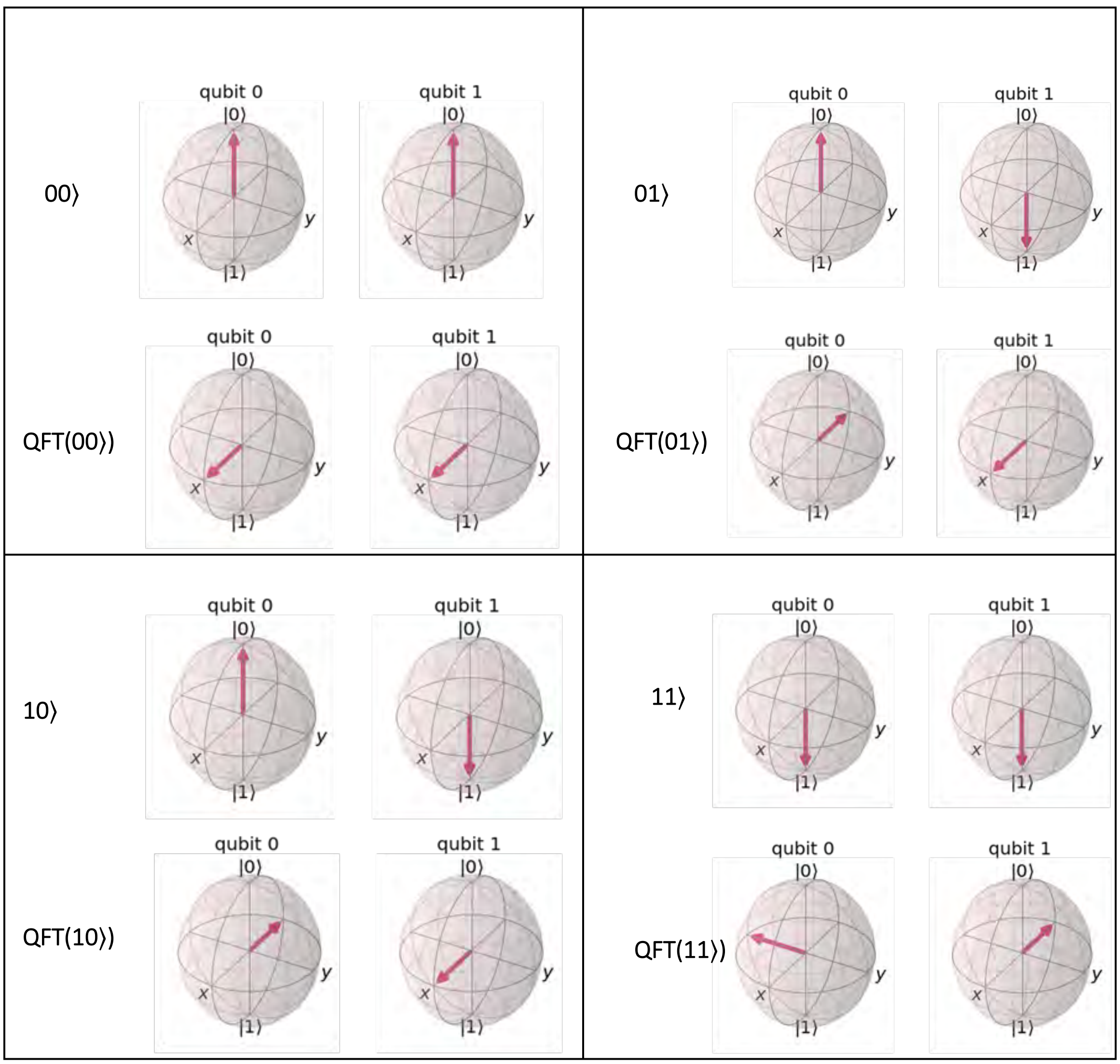


The entanglement between the qubits is critical to the success of the algorithm. By entangling the qubits, the algorithm is able to amplify the probability of finding the correct answer by constructive interference, while suppressing the probability of finding incorrect answers by destructive interference. In particular, the oracle and reflection operations are designed to create entanglement between the marked state and the other states in the superposition. As the algorithm iterates, the entanglement between the marked state and the other states becomes stronger, leading to a higher probability of measuring the marked state. Grover's algorithm has applications in database searches, optimization, and cryptography, all of which have potential impact on solving insurance related problems.

- **Quantum Fourier transform**: This algorithm is a quantum version of the classical Fourier transform, which is used to convert signals to frequencies and analyze periodic functions. The quantum Fourier transform can be considered as mapping from qubit states and the Fourier basis, as constructed by the Hadamard gate illustrated in figure 3. In figure 11, a 2-qubit example is used to show the results of applying quantum Fourier transfer to four different states 00⟩, 01⟩, 10⟩, and 11⟩.

Figure 11

QUANTUM FOURIER TRANSFORM ILLUSTRATION: 2-QUBIT EXAMPLE



Note:

QTF: quantum Fourier transform

To interpret it, you may consider the Z-axis represents the signal domain, while the X- and Y-axes represent the frequency domain. Although it is not obvious how the mapping of qubit states to other basis is beneficial, it enables quantum algorithms to efficiently process large amounts of data to amplify desired information, suppress undesired information, and find the period in periodic functions. The algorithm can be implemented efficiently on a quantum computer using a network of quantum gates. It is used in many quantum algorithms, including Shor's algorithm and the quantum phase estimation algorithm.

- **Shor's algorithm**: Created by Shor (1994), this algorithm is a quantum algorithm for factoring large numbers into their prime factors, a problem that is considered to be very difficult for classical computers and the basis of encryption technologies such as the public key cryptosystem RSA. Usually, the trial-and-error method is used. Shor's algorithm uses quantum computers to perform the factorization process much faster than classical computers. It has the potential to break many commonly used cryptographic systems. The algorithm works by using a quantum Fourier transform to find the period of a periodic function, which is related to the factors of the integer being factorized. By finding the period of the function, the algorithm is able to determine the factors of the integer in polynomial time, which is exponentially faster than the classical algorithms. A numerical example is given in the end notes[7].

- **Quantum phase estimation algorithm**: This algorithm is used to estimate the eigenvalues of a unitary operator in a way that is exponentially faster than classical algorithms, introduced by Kitaev (1995). With a unitary operator and a quantum state, the algorithm generates an estimate of the eigenvalue, also known as phase, of the unitary operator corresponding to the quantum state. The quantum phase estimation algorithm is used in quantum algorithms for solving linear systems of equations ($Ax = b$) and also quantum amplitude estimation, which is key to financial applications such as the pricing of options. The algorithm works by using a controlled version of the operator to create a superposition of eigenstates of the operator, and then applying a series of controlled rotations to estimate the phase information of the eigenstates. Eigenstates can be considered as eigenvectors in classical linear algebra. The number of rotations required depends on the desired accuracy of the estimation, but it grows logarithmically with the dimension of the Hilbert space, which is the length of an eigenvector in classical linear algebra. This makes the algorithm exponentially faster than classical algorithms.

- **Amplitude Estimation**: It is a quantum algorithm that allows for the efficient estimation of the amplitude of a specific quantum state in a superposition. If we consider a quantum state as a risk event we are interested in, knowing its amplitude means knowing its probability, because its probability is the square of its amplitude. The algorithm was introduced and created by Brassard, et al. (2002). The algorithm uses the quantum phase estimation algorithm, which is a quantum algorithm used to estimate the eigenvalues of a unitary operator as explained above. The phase estimation algorithm is applied to magnify the amplitude of the target state. Variations of the algorithm exist, such as using Grover’s algorithm for amplitude amplification, as discussed in Zhao, et al. (2022). By repeating this process multiple times and taking the average, the amplitude, or the probability of the target state, can be estimated. The amplitude estimation algorithm has important applications in a variety of areas, including optimization, predictive modeling, and cryptography. It can be used to speed up classical algorithms for solving linear systems of equations and for performing principal component analysis. It can also be used to speed up the evaluation of certain functions, such as the distance between two points in high-dimensional space, which may be needed in unsupervised learning such as clustering. On the financial and investment side, it can help estimate the probability of certain financial outcomes. It is also used in predictive modeling and, specifically, the quantum support vector machine that has the potential exponential speedup in certain cases, such as credit scoring, allowing for faster and more efficient classification of large datasets.

- **Quantum annealing**: It is a quantum version of stochastic annealing that is used to approximate the global optimum of a given function in a large search space for an optimization problem. It can potentially improve the optimization process to find global maxima or minima.

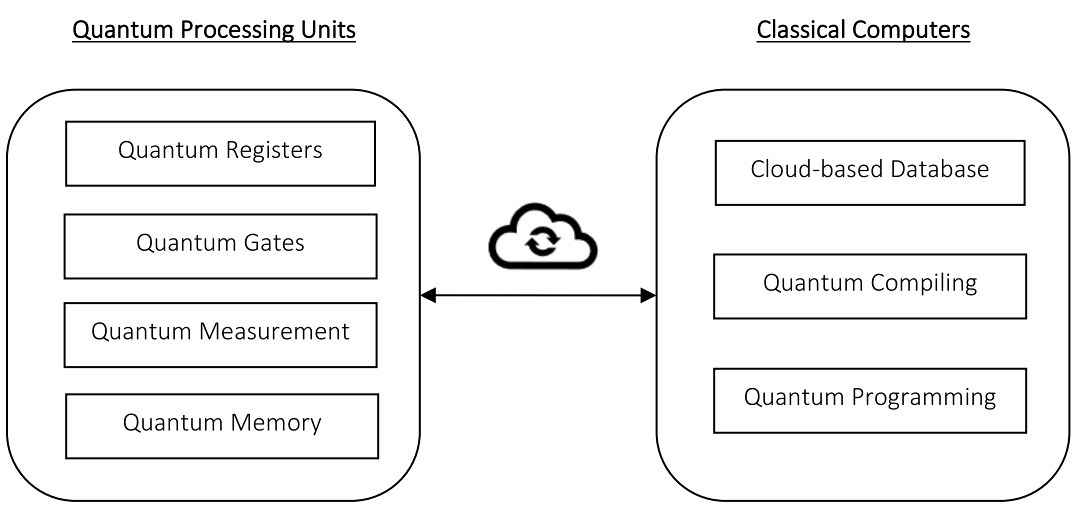
Many of these algorithms were developed in the last century, but tested recently with the advancements on quantum computers to potentially revolutionize the financial and insurance industries by providing new ways to perform financial modeling, optimization, and risk analysis. The field is still in its early stages, and much more research and development are needed before the full power of quantum computing can be fully realized.

An example of applying amplitude estimation and quantum phase estimation algorithm to equity index option pricing is provided in appendix A.

2.7 QUANTUM COMPUTERS

To achieve the benefits of quantum computing, quantum computers need to be built to store information using qubits, which can be carried as photons, ions, electrons, and so on. Figure 12 illustrates the key components of quantum processing units (QPU) and how they can be integrated with classical computing techniques.

Figure 12
QUANTUM COMPUTER DESIGN



As part of a QPU, a quantum register contains multiple connected qubits. A QPU can perform operations on qubits through quantum gates and can be instructed to measure the quantum system. Quantum memory is needed to store the data during quantum computing. It is important to know that qubits are difficult to store and their superposition and entangled states usually only last for a few seconds before decaying or collapsing at the current stage. This is why quantum computers need to be used together with classical computers for completing the computation tasks. Quantum computers have been built using different approaches, including, but not limited to, the following:

- **Superconducting materials**. Qubits are kept inside a cold and controlled environment and manipulated by electrical signals using tiny electrical circuits made from superconducting materials. Qubits can be operated and measured quickly and correctly. However, quantum computers need to be stored in a cold environment, such as a refrigerator with temperature close to absolute zero (-459 degrees Fahrenheit), to avoid the quantum states being distorted. Large technology companies, such as Google, IBM, and Intel, have built quantum computers using superconducting materials. The largest quantum computer built so far is the Dubbed Osprey with 433 qubits by IBM.
- **Trapped ions**. Ions are atoms or molecules with a net electrical charge. When trapped in free space using electric and magnetic fields, they can be used to store qubits. Optical signals, such as lasers, can be used to perform operations on qubit states. Compared to using superconducting materials, trapped ions may be more suitable for applications that require high levels of stability and control. Quantum computers built with trapped ions can be operated at room temperature. However, speed and scalability are currently limited. Both Honeywell and IonQ are leaders in this area, with the largest one built containing 32 qubits.
- **Cold atoms**. Cold atoms, also known as neutral atoms, are used to store qubits with the operations usually performed through optical signals. Cold atom quantum computers are less sensitive to electric fields and potentially better at error correction. Like quantum computers using superconducting materials, cold atoms operate near absolute zero temperature, as well. The largest cold atom quantum computer built so far is Aquila, the 256-qubit quantum computer by QuEra.

Based on the survey conducted by Mosca and Piani (2022), superconducting systems and trapped ions are considered the leading platforms, with cold atoms a promising candidate.

With available quantum computers, classical computers are usually used to develop quantum computing programs, deploy them to quantum computers, prepare data inputs for the computations, and collect results for analysis and integration with other tasks in the whole process. Given the requirements of storing quantum computers, users usually will access quantum computers through a cloud-based infrastructure, with quantum computers managed by large technology companies. Nowadays, users can access quantum computers for research purposes using the cloud-based approach, like an existing cloud-computing platform. Quantum programming is also performed on classical computers using special development kits, but still relies on or is related to existing programming languages such as Python or C#.

In the near future, it is expected that the number of qubits in a single quantum computer will keep increasing, with IBM planning to have a 1121-qubit quantum computer developed in 2023. In addition, the connection among quantum computers may see a breakthrough with the upcoming 133-qubit IBM Heron by the end of 2023. This is essential for realizing the scalability of quantum computing.

For quantum computers to be ready for large-scale commercial usage, in addition to necessary hardware advancements, the price may be another obstacle. However, given that quantum computers are likely to be used for the issues they are good at solving, cloud computing is likely to be used, which will improve the usage efficiency and, therefore, the cost.

2.8 ERROR CORRECTION

While most of the computers we use nowadays generate stable results, they can still encounter errors. You may experience blue screens showing memory errors. For GPUs, one or two pixels on your screen may have wrong colors, which may not be detected by human eyes. Quantum computers are also susceptible to errors due to a phenomenon called quantum decoherence. Decoherence occurs when a quantum system interacts with its environment, causing the system to lose its coherence and become entangled with the environment. Quantum computers are sensitive to electromagnetic fields, air, and heat, which can cause a qubit to lose its properties. This can lead to errors in quantum computations. In addition to quantum decoherence, there are other sources of error in quantum computing, such as imperfect gates, imperfect measurements, and environmental noise. Quantum computers are kept in extremely cold environments, as subatomic particles must be as close as possible to a stationary state.

At the current stage, quantum computers are error prone. Quantum error correction, which involves encoding quantum information in such a way that errors can be detected and corrected, is used to mitigate these errors. New techniques and algorithms are also being developed to improve their reliability and reduce the impact of errors on quantum computations. It is essential for building practical quantum computers that can keep the error rate low.

Section 3: Quantum Computing Insurance Applications

Quantum computing is suitable for solving high-dimensional problems that typically require optimization and simulation. Such problems exist in many fields, including artificial intelligence, drug development, computational chemistry, weather forecasts, logistics, cryptography, and finance.

In the insurance field, especially the actuarial field, simulation and optimization are often needed to manage the risks embedded in the insurance products that offer complicated guarantees on investment and insurance risks. Insurance companies also deal with data of millions of policyholders. The combination of big data volume and model complexity demands large computing capacity in a timely manner. Approximations and workarounds are usually needed with existing technologies used by the insurance industry. In this section, such areas are explored where quantum computing can make a material improvement.

3.1 POST-QUANTUM CYBER SECURITY AND CYBER INSURANCE

Quantum computing has the potential to greatly impact the field of cybersecurity, including the area of cyber insurance. Quantum computers can perform certain types of computations much faster than classical computers, which could make it easier for attackers to break encryption and access sensitive data. Current public-key encryption technology, such as RSA, will be powerless when faced with functioning quantum computers. Quantum computing can be used to invalidate the foundation powering current encryption standards. A hacker who can exfiltrate encrypted data successfully will be able to decode the data using quantum computers.

Figure 13

PUBLIC-KEY CRYPTOGRAPHY ILLUSTRATION

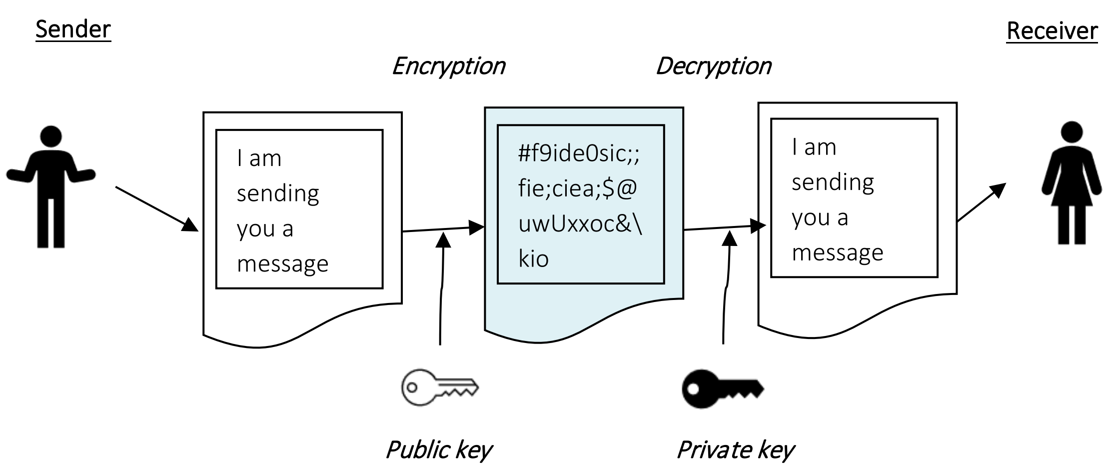


Figure 13 illustrates the public key cryptography that utilizes two security keys for encryption and decryption. The two keys are mathematically related and impossible to be cracked using conventional computers. However, as discussed in section 2.6, Shor's algorithm with quantum computers can derive the private key based on the public key in a short time. It may be wondered why a pair of keys is needed in the first place, instead of using the same security key in both the encryption and decryption phase. If a single key is used, it relies on a secure third party to store the key, which can be hacked as well.

Mosca and Piani (2022) performed a survey on the likelihood of a quantum threat to current public-key cryptosystems. A group of experts in the quantum computing fields, including academics and practitioners, participated in the survey. The majority of the respondents thought such a threat would be unlikely in the next five years, with 68% of respondents believing the chance was less than 1%. However, in a 10-year time horizon, 50% of respondents thought the threat would be more than 5% likely. Twenty-two and a half percent of respondents thought the chance would be greater than 50%. It is likely only a matter of time that current encryption will be vulnerable to quantum computers in the real world. New encryption methods that are quantum-resistant need to be used in the world of quantum computing in the future. Some quantum-resistant algorithms are available and new algorithms are being developed by researchers. A few promising quantum-resistant methods, using either quantum physics or more difficult math problems, are introduced as follows:

- **Quantum key distribution (QKD)** uses quantum mechanics to distribute cryptographic keys. As discussed in section 2.3, if a hacker wants to intercept the key, it will need to measure a quantum system. This will be known to other parties as well. QKD uses quantum physics, rather than mathematics, to enable encryption.

- **Multivariate cryptography** relies on solving systems of polynomial equations to get the private key for decryption. It is computationally difficult to solve the equations.

- **Lattice-based cryptography** uses lattices in math for encryption. The problem to be solved to get the private key is to find points in the lattice that are close to some chosen points. The problem becomes incredibly difficult with multiple dimensions as high as a few hundred.

The risks associated with cyberattacks may increase and evolve as quantum computing technology advances. These may have two types of impacts on the insurance industry:

- Insurance companies usually collect and hold sensitive data of existing and new clients to conduct business. They are popular targets of cyberattacks, which are potentially enabled by quantum computing. If the insurance companies do not stay on top of the technology and upgrade their cyber security, there can be disastrous outcomes. The National Institute of Standards and Technology (NIST) selected a few quantum-resistant algorithms in 2022[8]. However, it is also important to understand the difference between being available and being adopted. It is important that the insurance industry gets involved in quantum computing and is prepared for the threat in 5 to 10 years. In addition to using the latest technology to guard the data, insurers may also want to collect and keep data on a minimum basis.

- The development of quantum computing could potentially impact the pricing and availability of cyber insurance, as insurers may need to adjust their risk models to account for the increased threat of quantum-based attacks. Insurers may also need to consider the development of quantum-safe encryption methods in order to provide coverage against future attacks. In addition, insurers may need to provide education and training for their policyholders on how to mitigate the risks associated with quantum computing and cyberattacks.

The impact of quantum computing on cyber security and cyber insurance is still a developing area, and the full extent of its effect on the industry remains to be seen. However, it is clear that the emergence of quantum computing will have significant implications for cybersecurity in the insurance industry.

3.2 SIMULATION

The design of quantum computing and quantum computers make them suitable for Monte Carlo simulation, which is a key component in much actuarial work and computationally intensive. For many options offered in insurance products, given the path dependent payoff functions and the dynamic interaction with policyholder behaviors and management actions, Monte Carlo simulation is considered as an appropriate approach to evaluate the risks without approximation. Monte Carlo simulations involve generating random numbers to simulate the behavior of a system or process. However, as the size of the system being modeled increases, Monte Carlo simulations become increasingly computationally expensive.

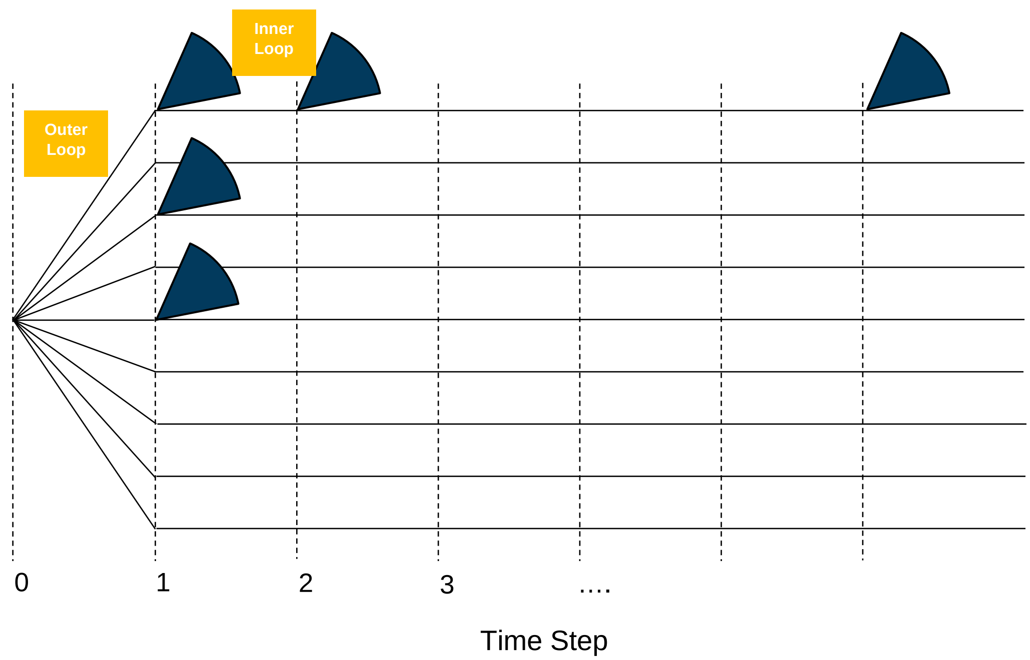
Quantum computing has a quadratic speedup compared to classical algorithms in Monte Carlo simulation. The time required to perform a Monte Carlo simulation on a quantum computer scales quadratically with the input size while, on a classical computer, the time required scales exponentially with the input size. Quantum computing offers the potential to speed up Monte Carlo simulations by exploiting the quantum parallelism of certain algorithms. In particular, certain quantum Monte Carlo algorithms can achieve a quadratic speedup over classical Monte Carlo methods, allowing for simulations of larger systems in less time.

Monte Carlo simulations usually need to be performed at a seriatim level in the following actuarial work:

- Embedded option pricing to get the fair value that is expected to match the market price assuming no arbitrage opportunities. Examples of embedded options are minimum credit interest rate guarantees in universal life products, principal protection in fixed indexed annuities, and secondary guarantees in variable annuity products, such as guaranteed minimum death benefits, guaranteed minimum income benefits, guaranteed minimum withdrawal benefits, and guaranteed minimum accumulation benefits.
- Reserve calculation with a specified confidence level. For example, for a statutory reserve calculation following the VM21 in the U.S., conditional tail expectation (CTE) with a confidence level of 70% is used. Instead of using the average of values under risk neutral scenarios, the average of results at the 70th percentile and above are used based on real-world scenarios. In many cases, thousands and millions of policy records need to be simulated individually given their different behaviors. Data compression may have to be used instead of seriatim calculations, but the impact of compression can be material and uncertain.
- Similar to reserve calculation, capital management needs to analyze results at an even higher confidence level using risk measures such as CTE and value at risk (VaR).
- Asset valuation and risk analysis. Some asset types have embedded options, such as callable bonds, puttable bonds, and mortgage-backed securities with prepayment options. These options can be valued using Monte Carlo simulations. For assets subject to credit risk, Monte Carlo simulation of credit risks (defaults, rating migration, and credit spreads) helps provide more insights into the credit risk analysis. These calculations are usually needed at individual asset holding level.
- Hedging and risk management. With both assets and liabilities analyzed using the stochastic approach, hedging and risk management that focus on reducing the exposure of the net position to different risks also use Monte Carlo simulations to get a holistic view.
- Monte Carlo simulations are not confined to evaluating financial risks. Insurance risks, such as mortality, morbidity, disability, and claim frequency and severity, may also be modeled in a stochastic way if the exposure to these risks is high.

In some actuarial analysis, nested stochastic, also known as stochastic on stochastic, calculations are needed for financial projection. A typical example is the pricing analysis of long-term insurance products that have secondary guarantees, such as guaranteed minimum income benefits and guaranteed minimum maturity benefits. As shown in figure 14, to evaluate the profitability of a product, thousands of outer loops real-world economic and/or insurance scenarios are used to evaluate the possible financial outcomes of the product. Analysis under each outer loop scenario requires projection until the expiry of all policies. At each time step in the projection, liability values in terms of fair value or reserve need to be calculated, in addition to required capital, to be able to estimate profit measures, such as embedded values, profit margin, and return on capital. Any hedging strategies also need to be reflected at each node to evaluate changes in hedging position. Fair values, reserves and capital requirements need to be calculated using another layer of stochastic analysis using inner loop scenarios, which depend on the state of the associated outer loop scenarios at the specific time point.

Figure 14
STOCHASTIC ON STOCHASTIC FRAMEWORK



A full-blown stochastic-on-stochastic (SoS) analysis can be challenging given the amount of computation it requires. However, being able to project liability values and capital requirements are important for pricing. Quantum computing has a potential to enable SoS analysis without undesired approximations in a timely manner.

3.3 OPTIMIZATION

Optimization using quantum linear system algorithms can have a polynomial speedup for certain convex optimization problems, as discussed in Aaronson (2015). Some optimization problems in the insurance fields are multi-period high-dimensional problems that require long optimization time and approximation to reduce the complexity. Two examples are given below:

<u>Strategic Asset Allocation</u>

Strategic asset allocation optimization to support an underlying liability portfolio can be computationally intensive. The number of possible asset allocation plans can be too large to test in a reasonable time period. For example, in a four-asset-class space, the construction of an efficient frontier can be challenging. The number of possible asset mixes can be calculated as combinations with replacement. With 51 possible dividers within the range of 0% and 100%, with an increment of 2%, three dividers need to be chosen to determine the portion assigned to the four asset classes. The chosen dividers can be the same, which means one or more asset classes have zero allocation.

$$C^R(n,r) = \frac{(n+r-1)!}{r!(n-1)!}$$

Where

n: the number of possible dividers

r: the number of chosen dividers, which can be repetitive.

This requires 23,426 static asset plans be tested to get the efficient frontiers similar to those shown before. If asset allocation path is considered so that static asset plans can be non-constant, 23426^{20} asset allocation paths need to be tested to figure out the best one using the grid searching approach, assuming a quarterly asset allocation adjustment with a time horizon of five years.

Quantum computing can be potentially used for both simulation and optimization in this process to take advantage of the speedup in both areas.

<u>Liability Replicating Portfolio</u>

In some cases, liability is replicated using a pool of assets that can match the value, cash flows, and sensitivities to key risks, such as interest rate risk, equity risk, and credit risk. Ideally, the full distribution can be taken into consideration in the replicating process. With the help of quantum optimization, more matching targets, including the results under each individual scenario, can be included to improve the liability replicating portfolio.

3.4 PREDICTIVE MODELING

Quantum machine learning is expected to be suitable for predictive modeling tasks involving large amounts of data in terms of not only number of records, but also number of variables. Quantum machine learning algorithms have been developed to handle both supervised and unsupervised learning. On the supervised learning side, examples of using quantum machine learning include regression with least-squares linear regression by Date (2021) and feedforward neural networks by Mitarai, et al. (2018), and classification with k-nearest neighbors by Basher, et al. (2020) and support vector machine by Kerenidis, et al. (2021). On the unsupervised learning side, quantum algorithms are implemented, such as k-means clustering by Lloyd, et al. (2013), principal component analysis by Lloyd, et al. (2014), and Bayesian networks by Borujeni (2021).

Some quantum algorithms already showed speed advantages compared to classical algorithms. For example, quantum k-means clustering can estimate the distance between data records to centroids with $\mathrm{O}(M\log(MN))$, compared to classical algorithms in $\mathrm{O}(M^2N)$. Here the big *O* notation represents the time complexity of the algorithm. It can be considered as the upper bound of the time needed to perform the calculations given the input size. *M* is the number of data records and *N* is the number of variables. K-means clustering can be used in many areas, such as data compression and customer profiling, where millions of records are available.

In the insurance industry, given the increasing volume of available data and data types, such as social media data, efficient predictive modeling algorithms can be used to increase speed of model training, improve model accuracy, and enhance scalability. The areas in which quantum algorithms can have a significant improvement usually involve data and decision-making at the individual policy level.

3.4.1 UNDERWRITING

Underwriting is the task of assessing the risk of insurance applications and assigning appropriate risk ratings to individual applications, which will be used to determine whether the risk is acceptable and the premium rate to charge. On the life insurance side, key variables, such as age, gender, smoking status, address, occupation and health conditions, can be used to predict the mortality and morbidity risk levels. For property and casualty (P&C) insurance, explanatory variables also contain information on the insured properties. For example, auto insurance underwriting uses additional information, such as the vehicle model, usage, past claims, and so on.

Table 1
EXAMPLE OF PREDICTIVE ANALYTICS FOR UNDERWRITING

Explanatory Variables	Target Variables
• Demographic information • Health information • Financial information • Insured property information • Social media data	• Risk rating • Underwriting decision: accept/reject

3.4.2 PRICING

While underwriting can provide insights on certain risk types, pricing has a more comprehensive list of risk factors to consider, as listed in table 2.

Table 2
EXAMPLE OF PREDICTIVE ANALYTICS FOR PRICING

Explanatory Variables	Target Variables
• Demographic information • Policy information • Account value • Guaranteed value • Economic conditions • Market competitiveness • Economic scenarios	• Dynamic lapses • Claim frequency and severity • Dynamic premium payments • Probability of option exercise • Probability of retaining a business

Dynamic Lapse Risk

Many long-term insurance products are sensitive to policy lapses before maturity. Predictive models allow a more refined way to predict lapses at the policy level, considering policy information, in addition to information at a much higher level, such as macroeconomic conditions, product type, pricing cohorts, etc. Essentially, policyholders are comparing the current surrender value (CSV) of the policy and the discounted guaranteed/expected value if held to maturity. The relationship between CSV and discounted maturity value changes over time and scenario. Representing the dynamic lapse with a predictive model can potentially improve the prediction accuracy in stochastic analysis and stress testing. Even though deterministic lapse rate assumptions may be correct at an aggregated level, removing the variation among policies during projection can underestimate the risk significantly.

Claim Frequency and Severity

For non-life insurance, such as auto insurance, predictive models can be used to estimate the probability of claim occurrence and the amount of the claim given the occurrence. The estimation can provide more tailored pure risk quantification, which can be used for rate setting at the policy level.

Policy Renewal

For insurance products with a short term, such as auto insurance, policy renewals can be challenging with the considerations of ratemaking and retaining existing business. Clearly, policy level experience, such as claim experience, is important and serves as a basis for rate adjustment. However, the degree of adjustment also depends on market competitiveness. Predictive models can help tell the expected profit and probability of retaining the client given a rate adjustment. A policy renewal decision can then be made to maximize the total profit.

Premium Persistency Risk

Long term products, such as Universal Life (UL), allow policyholders to pay renewal premiums different from the scheduled amount, subject to certain limits. The amount and timing of future premium payments can affect profitability. Similar to lapse risk, predictive models can help predict premium persistency at the policy level with available social and financial information.

Option Exercise

In many insurance products, policyholders are given options to choose at certain points during the life of the policy. For example, policyholders may choose when to retire and annuitize the account or guaranteed policy value. Policyholders may also choose the annuity type, such as life annuity or annuity with a certain number of payments. Policyholder behaviors can be studied and modeled using predictive models to estimate the cost of these options.

3.4.3 MARKETING

Dynamic and precise marketing are made possible by predictive modeling. Models can be used to segment the target market, predict life events of customers, and recommend appropriate products, as shown in table 3. The data volume is usually large and it takes a long time to process and update existing models. Quantum machine learning can help shorten the modeling process, as demonstrated in the speed improvement of k-means clustering using quantum computing.

Table 3
EXAMPLE OF PREDICTIVE ANALYTICS FOR MARKETING

Explanatory Variables	Target Variables
• Customer demographic information • Customer socioeconomic information • Transaction records • Social media data (if allowed by regulation)	• Customer profile • Probability and timing of major life events • Most likely next insurance purchase

<u>Market Segmentation</u>

The target market can be segmented using clustering analysis, which share some common features. These features are then used to design appropriate marketing strategies for each cluster.

<u>Life Event Prediction</u>

Information of existing and potential customers can be used to predict life events, such as education, marriage, property purchase, and retirement. Customized promotions can be delivered based on the predictions.

<u>Recommendation System</u>

A recommendation system can be built to recommend products that customers may need based on customers' demographic, socioeconomic, and past transaction information.

3.4.4 VALUATION

In addition to using predictive modeling to derive valuation assumptions, predictive modeling can be used in other areas for valuation, as summarized in table 4.

Table 4
EXAMPLE OF PREDICTIVE ANALYTICS FOR VALUATION

Explanatory Variables	Target Variables
• Policyholder reasonable expectation • Market conditions, such as interest rate and asset return • Competitors' (expected) management actions • Constraints on the magnitude of period-to-period change	• Reserve at claim level • Probability of reopening a claim • Proceeds from salvage and subrogation

Case Reserving

Predictive modeling can help automate claim processing and determine case reserve. With detailed claim information, models can be used to classify the claims and assign complicated claims to claim adjusters. For claims where the model has a high confidence in prediction accuracy, claim amount can be automatically calculated and paid. It can improve the efficiency of claim processing by allocating experts to the right cases. In addition to using numerical data, predictive analytics also uses non-numerical data, such as claim description, and image data, such as a picture of a damaged vehicle and house, to make claim decisions.

Claim Payment Pattern

Predictive modeling can also be used to predict claim development pattern and ultimate loss based on claim information to determine the amount of incurred but not enough reserved (IBNER). Using explanatory variables, such as claim type, development year, paid loss, outstanding loss, and policy information, future claim payments and settlements can be predicted at the individual claim level.

Claim Re-openess

Sometimes, claims may be reopened after settlement. Claim and policy information can be used to predict the probability of re-openess and the expected amount of adjustment to the claim amount. Claim re-openess can be considered as part of the incurred but not reported (IBNR) reserve.

Salvage and Subrogation

The proceeds from salvage and subrogation may be estimated by predictive models as well. Salvage means the recovered amount from damaged properties. Subrogation means the settlement amount with a negligent third-party through legal action or arbitration. Detailed claim information can be used as the explanatory variables.

3.4.5 RISK MANAGEMENT

Although risk management may have overlaps with other areas, such risk rating in underwriting, there are some unique examples of applying predictive modeling, as summarized in table 5.

Table 5
EXAMPLE OF PREDICTIVE ANALYTICS FOR RISK MANAGEMENT

Explanatory Variables	Target Variables
• Policy information • Claim information • Economic conditions/scenarios • Social media • Business and risk profile	• Probability of fraud at claim level • Impacts of extreme events

Fraud Detection

Detailed information about the claim and claim reporters can be used to estimate the probability of a false claim. Social network models can be used to provide insights of the clusters of claim reporters, which can be useful if it is a group activity. Then, cases with a high probability of fraud can be assigned to experts to further investigate and verify.

Risk Monitoring and Contingent Planning

Timely detection of extreme events and an estimation of their impact are important to act quickly to mitigate the adverse effects. Predictive models can be built to use various data sources, such as social media, together with insured businesses, to predict the severity of an extreme event and trigger disaster recovery programs to reduce the losses and disruption to business operations.

3.4.6 DATA COMPRESSION

In some cases, the data volume is too large to be used for calibration or projection purpose and needs to be compressed. Quantum machine learning can be used to speed up the compression process.

Model Point Compression

The insurance portfolio can be too large to perform seriatim calculations within a reasonable period of time. The policies can be compressed using predictive modeling to construct representative policies and maintain a satisfactory accuracy of estimation at the group level.

Transaction Data Feature Extraction

Transaction-type data, such as purchase history, bank truncations, and GPS data of driving, are difficult to be used directly by predictive models. Instead, data has to be compressed using feature engineering and unsupervised learning to a few features that will be used as explanatory variables. For example, the GPS data of a vehicle can be used to learn driving behaviors that can be described using speed, acceleration, deceleration, amount of driving, time of driving, and length of driving trips.

3.5 SYNTHETIC DATA GENERATION

Sometimes, insufficient data is a problem for predictive modeling and stochastic analysis. Quantum computing has been used to generate synthetic data to train artificial intelligence, which requires large amounts of high-dimensional data.

Similarly, for predictive modeling in insurance applications, data insufficiency can also be problematic. A common issue is imbalance. Imbalance means there is an unequal distribution of categories in your target variable. In the insurance field, claim frequency is a relevant example. In the insured cohort, only a small portion is expected to have a claim, while others are expected to have no claims. Compared to binary classification, multi-class classification is more prone to the issue of imbalance. Having five claims is rarer than having at least one claim. Other possible examples include fraud detection, individual default prediction, and option exercise prediction.

When imbalanced data is used for predictive modeling, it is challenging to predict the rare cases. The data records that belong to the rare classes may be insufficient using standard processes. The training algorithm may be overwhelmed by the common cases and provide little insight on the rare cases. In addition, statistical measures may indicate a high level of prediction accuracy, although rare cases are not predicted at all. For classification, precision, recall, and the F-measure are popular measures based on the confusion matrix, as shown in table 6.

Table 6

SAMPLE CONFUSION MATRIX

	Predicted: True	Predicted: False
Actual: True	True Positive	False Negative
Actual: False	False Positive	True Negative

Precision measures the Type I error[1] and recall measures the Type II error. F-measure (or F-score) is the harmonic average of precision and recall and may be used as a high-level measure to rank the performance of different models.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall\ (True\ Positive\ Rate) = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F - measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

If we use the common class to calculate these measures, the prediction results may look promising, as shown in table 7.

Table 7

CONFUSION MATRIX FOR THE COMMON CLASS

	Predicted: True	Predicted: False
Actual: True	95	0
Actual: False	5	0

Precision is 95%. Recall is 100%. F-measure is rounded to 97.4%. However, using the rare class, the same measures give us an opposite picture, as shown in table 8.

Table 8

CONFUSION MATRIX FOR THE RARE CLASS

	Predicted: True	Predicted: False
Actual: True	0	5
Actual: False	0	95

[1] Recall from classical statistics, a Type I error is a false positive where you reject a true hypothesis. A Type II error is a false negative and occurs when you fail to reject a false hypothesis.

Precision, recall and the F-measure become 0. It is clear that the rare class should be used to measure imbalanced classification. Otherwise, we can simply predict all cases belonging to the common class and still achieve high accuracy according to the measures.

To address the issue of imbalance, synthetic data can be generated by quantum computing to provide more training data with claims. A generative adversarial network consists of two neural networks, a generator and a discriminator. The generator creates a sample of synthetic data to deceive the discriminator. The discriminator tries to distinguish the synthetic data from real training data. With a successful training, the generated synthetic data cannot be distinguished from the real data by the discriminator. Quantum computing can be used to generate high-dimensional data.

Another potential usage of quantum synthetic data generation is economic scenario generation, which contains high-dimensional data covering multiple economies, multiple asset classes, multiple bond terms, and multiple options, such as prepayments for mortgage-backed securities, and debt to equity conversion for convertible debt.

Section 4: Road to Large-Scale Applications

Although quantum computing will benefit the areas discussed above in theory, it is not expected that any widespread commercial usages will happen anytime soon. The timeline for the application of quantum computing to the insurance industry is difficult to predict with certainty, as it depends on the pace of technological development, as well as the adoption and implementation of these technologies by the industry.

In general, quantum computing is still in its early stages of development and adoption, and there are currently only a handful of companies and research institutions that have the capability to develop and use quantum computers. In addition, the insurance industry could be cautious to adopt new technologies and processes due to its regulatory and risk-averse nature. Therefore, it may take several years or even decades before quantum computing becomes widely adopted within the industry.

However, some companies in and related to the insurance industry are already exploring the potential applications of quantum computing, particularly in the areas of risk pricing, credit scoring, and portfolio optimization. Recent breakthroughs made the applications of quantum computing to the finance and insurance industries more realistic.

4.1 PROMISING APPLICATIONS

Giurgica-Tiron, et al. (2021) showed that amplitude estimation can be realized in less demanding quantum computing hardware with good accuracy. An amplitude estimation algorithm is a quantum-computing algorithm for Monte Carlo simulation and enables quadratic speedup. Chakrabarti, et al. (2021) showed that quantum computing with sufficient qubits can perform Monte Carlo simulations in a quicker and more accurate way. Existing literature on quantum computing applications to the financial field, which is also relevant to the insurance industry, is summarized in three areas: asset pricing, asset allocation, and risk management.

Asset Pricing

Sakuma (2022) applied the deep quantum neural network to learn implied volatilities, option prices, and Greeks, such as delta and gamma. The deep quantum neural network is believed to be a promising technique for solving such numerical problems arising in finance efficiently.

Slate, et al. (2021) demonstrated good performance using a newly developed Quantum Walk Optimization Algorithm in finding high-quality solutions to the portfolio optimization problem, targeted at near-term noisy intermediate-scale quantum computers.

Stamatopoulos (2021) discussed the method to price options and portfolios of options on a gate-based quantum computer using amplitude estimation, covering vanilla options, multi-asset options and path-dependent options such as barrier options. The performance of option pricing circuits was also tested on quantum hardware using the IBM Q Tokyo quantum device.

Tang, et al. (2021) implemented two typical collateralized debt obligation (CDO) models, the single-factor Gaussian copula model and Normal Inverse Gaussian copula model in quantum circuits. Quantum amplitude estimation is used to estimate the price of CDOs.

<u>Asset Allocation</u>

Hodson, et al. (2019) evaluated the experimental performance of a discrete portfolio optimization problem using quantum computing, considering trading in discrete lots, non-linear trading costs, and the investment constraint.

Kalra (2019) used the D-Wave quantum annealing computer to study the problem of asset correlation identification for financial portfolio management using graphical models.

Rosenberg and Rounds (2020) discussed using the quantum annealing computer to perform portfolio optimization, allowing both long and short positions.

<u>Risk Management</u>

Herr, et al. (2021) developed a hybrid quantum-classical neural net that allows only high-dimensional classical data to enter the classical model and does not need to be prepared in a quantum circuit and demonstrate the effectiveness of this method on a credit card fraud dataset.

Milne, et al. (2017) studied quadratic unconstrained binary optimization using the German credit data, which results in a smaller feature subset with no loss of accuracy. Implemented on a quantum annealer, it has the potential to be faster than classical solvers.

As the technology continues to advance, we may see more companies experimenting with quantum computing in these and other areas of the industry sooner than expected.

4.2 ROADMAP

Business practitioners have been predicting the timeline of quantum computers being used for general purposes and, in general, expect 10 to 20 more years to pass before it happens. However, some are more optimistic. In 2021, Goldman Sachs predicted that quantum computing was only five years away from use in markets. Quantum computer producers, such as IBM, Google and Microsoft, also shared their roadmap to develop more advanced quantum computers and address some key issues discussed below that may block large-scale real-world applications in the upcoming years.

- Error correction. Hardware improvements are needed to reduce quantum hardware noise and, therefore, calculation errors, which are key for financial applications.
- Scalability. The number of qubits keeps increasing with new developments, which is key to quantum supremacy and solving large-scale real-world problems.
- Quantum and classical computers will require more integration to facilitate the analysis of quantum computing results and explore the most efficient way to use them together.

The timeline for the application of quantum computing to the insurance industry is uncertain and dependent on a variety of factors, including technology improvement and the speed of reaction of the insurance industry. However, it is clear the technology has the potential to greatly impact the industry, and continued interest and exploration are expected in this area in the coming years at an accelerated rate.

The banking industry moved quickly to prepare, research, and potentially adopt quantum computing to solve issues such as cyber security, trading, and portfolio optimization. A few out of many examples are provided as follows:

- J.P.Morgan Chase (JPMC) and QC Ware did a study of quantum "deep hedging," where quantum reinforcement learning methods based on policy-search and distributional actor-critic algorithms that use quantum neural network architectures were developed, as introduced in Cheratt, et al. (2023). In addition to quantum algorithms, JPMC also invested in post-quantum encryption techniques such as Quantum Key Distribution.

- HSBC has been participating in the Next Applications of Quantum Computing project that has diversified applications of quantum computing, including probabilistic risk assessment for energy infrastructures.

- Citigroup has been investing in quantum computing companies such as QC Ware, with senior management discussing the technology that will transform risk assessment and trading for the financial industry.

- Wells Fargo joined the IBM Quantum Network and worked with IBM and MIT to explore how quantum computing can help improve banking in terms of speed, security, and intelligence.

It is obvious that the banking industry is on board to embrace quantum computing technology. While acknowledging practical applications may still be 10 years away, the banking industry not only invests in quantum computing, but also actively participates in the research, which will be essential for any near-term relevant applications.

Some insurance companies also participated or expressed interest in quantum computing. For example, AVIVA (2021) discussed quantum computing in an article, mentioning its potential impact on risk management, cybersecurity, and so on. Munich Re (2021) is one of the founders of the German Quantum Technology and Application Consortium (QUTAC), with research on quantum technology applications for modern route planning. It is important to increase the awareness of quantum computing and its impact on cyber security in the insurance industry. Early involvement of the insurance industry will be the key to making the best use of the technology and also contribute to the development of the technology.

While insurance professionals depend on quantum technology to continue advancing, they can embrace the future through identifying areas for applications of quantum computing, implementing quantum algorithms to insurance applications, and perhaps contributing to developing new algorithms and software packages run on quantum computers to solve insurance-related problems.

4.3 ALTERNATIVES TO QUANTUM COMPUTERS

Quantum computers are expected to solve certain complex problems in a more efficient way than classical computers with CPUs and GPUs. However, in almost all cases, it is expected that both classical computers and quantum computers will be needed when classical computers interpret the outcomes from quantum computers. In addition, alternatives to quantum computers are being studied, which may have advantages compared to quantum computers in certain areas. A few examples are provided below:

- Optical computing, also known as photonic computing, uses photons, the particles of light, for the basic model of computation. For the normal computers we have, including superconducting quantum computers, they usually rely on electronic circuits to perform tasks. As light travels faster than electric currents, photonic computing can be efficient to perform numerical calculations, logical processing, data processing, data storage, data communications, and artificial intelligence.

- Biological computers use chemical inputs and biologically derived molecules, such as DNA and proteins, instead of electrical wiring and signaling, to perform calculations. Researchers have been able to get biological computers to complete a logic gate, but the potential is high. Biological computers may be able to solve complex problems using much less energy than electronic computers. Biological computers are believed to be more reliable, as well as they do not rely on functioning of all biological cells, while electronic computers may malfunction with one broken wire. Calculations are done simultaneously rather than sequentially in a biological computer. However, biological computing is still in its early stage, compared to quantum computing.

- Analog computing, compared to its counterpart, digital computing, is also a potential alternative to thrive in the future. Most of the current computing machines use digital computing that relies on digital signals, including both time and amplitude. On the other hand, analog computing uses a continuous variation of analog signals, such as electric, hydraulic, and mechanical signals, to solve the problem. A mechanical watch is an example of an analog computer with other examples such as flight simulators. Analog computing is error prone because analog signals are subject to noise and distortion. However, analog computing processes real-time analog signals and has the speed advantage, partially due to the capability of solving issues simultaneously rather than sequentially.

Although quantum computers may be the next innovative computing technology to be widely adopted, alternative computing technologies may potentially dominate in certain areas as well in the future, due to their specific advantages.

Section 5: Further Developments

This report introduces quantum computing with a focus on potential insurance applications. Although examples related to insurance are used, further research is needed to be able to apply quantum computing to solving a wide range of insurance problems.

- Nowadays, we may use existing software packages to perform difficult tasks, such as embedded option valuation, investment optimization, and predictive modeling on our computers using CPUs and/or GPUs. In the quantum computing field, although algorithms and programs have been developed to solve basic problems, significant efforts are required to develop the corresponding packages using quantum computing. This involves not only implementing methods using quantum algorithms, but also customization to meet the specific requirements of insurance applications.
- In most actuarial work, result reproduction is important from the perspective of auditing and regulatory requirements. Unlike classical computing systems, quantum computing systems cannot always reproduce the results given changing states. In order to attain a high degree of stability, further research is needed to quantify the measurement and sampling error of using quantum computing in a variety of potential insurance applications. A more advanced error correction technique is also needed to reach the goal of result reproduction.
- It is also important to analyze the entire calculation process and split the calculations among classical and quantum computing to optimize computing efficiency. This involves designs of data flow and workflow, as well as the communication between classical and quantum computers.

Many of the development items rely on further enhancements of quantum computing algorithms and hardware, but insurance domain knowledge is also critical in these areas to be able to identify key issues to work on, analyze the gaps, and potentially innovate by insurance practitioners to put quantum computing into practical use for the insurance industry.

Section 6: Conclusion

Quantum computing is a field of computing that uses quantum-mechanical phenomena to perform operations on data. It has the potential to solve complex issues in fields, such as cryptography, chemistry, physics, and finance, that are difficult or impossible for classical computers. When solving these issues, it usually takes much less time when exploring the entire decision space.

The complexity of insurance products justifies the applications of quantum computing in many areas where conventional computers with CPUs and GPUs cannot guarantee a full exploration of the entire space of strategies. Quantum computing can be used to enhance methods of simulation, optimization and predictive modeling, and synthetic data generation. These methods are widely used in different areas of insurance, such as underwriting, pricing, marketing, valuation, capital management, cyber insurance, and so on. With the enhanced methods, quantum computing may solve insurance-related issues that are currently solved with undesired shortcuts and approximations.

Although quantum computing has enormous potential benefits, it is still in the stage of technology development and far away from adoption by the insurance industry. Therefore, it is important to increase the awareness of quantum computing and, at a minimum, its impact on cyber security in the insurance industry. Early involvement of the insurance industry will be the key to making the best use of the technology and also contribute to the development of the technology. While actuaries depend on quantum technology to continue advancing, they can embrace the future through identifying areas for applications of quantum computing, implementing quantum algorithms to insurance applications, and, perhaps, contributing to developing new algorithms run on quantum computers to solve insurance-related problems. It is also beneficial to have actuaries get involved in quantum computing because they are the implementers of many complex calculations in the insurance industry where quantum computing can help.

Section 7: Acknowledgments

The authors would like to thank all members of the Project Oversight Group (POG) tasked with providing governance on this research project. This paper would not have attained its current level of relevance to practitioners without the POG's guidance, feedback and insightful input.

Project Oversight Group members:

Brent Carpenetti, FSA, MAAA, CERA

Lysanias Chikwanda

Tyler Doiron

David Moran, FSA, MAAA

Bryon Robidoux, FSA, MAAA, CERA

Sam Wehner, ASA, MAAA

At the Society of Actuaries Research Institute:

David Schraub, FSA, CERA, MAAA, AQ, Senior Practice Research Actuary

The authors would also like to thank Korrel Crawford for her effective coordination of this project, as well as the sponsorship and funding support of the Society of Actuaries.

Appendix A: Quantum Computing Example: European Equity Index Option Pricing

This appendix introduces how a European option can be valued using quantum computing. The goal is to provide concrete details on the application of quantum computing to a real problem.

An equity index option is financially linked to a specific equity index, such as S&P 500 index, and through which market risks can be traded in financial markets in their own right. A European equity index option is a contract that the option holder has the right, but not the obligation, to buy or sell an underlying instrument that is composed of an equity index at a specified strike price on the expiry date. The payoff of a European call option is given as follows and shown in figure A.1.

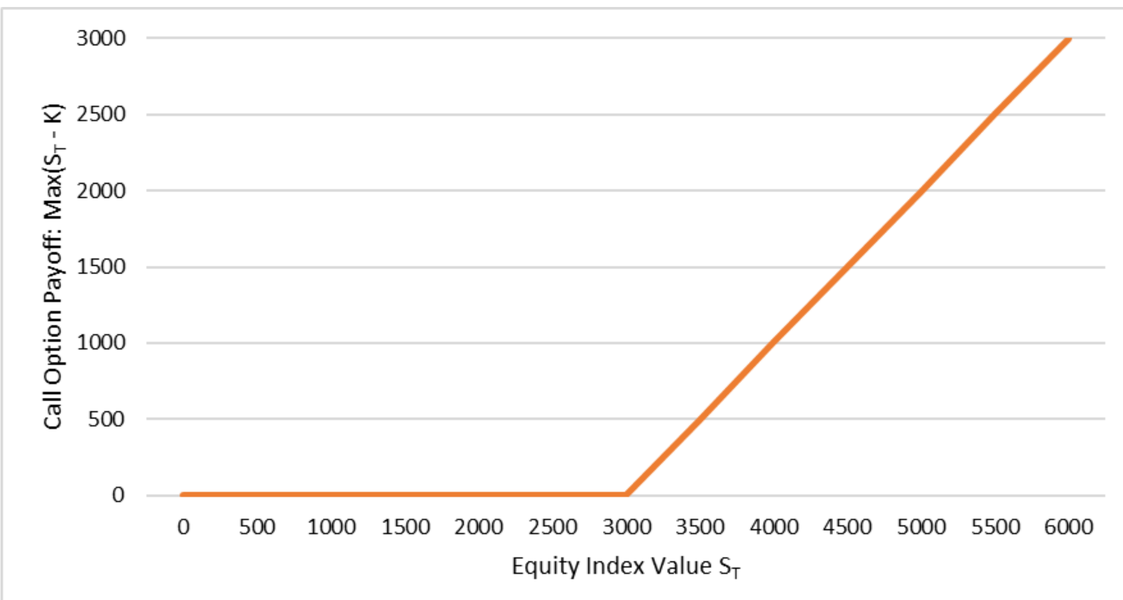
$$\max(S_T - K, 0)$$

where

> S_T: equity index value at time T.
>
> K: exercise price
>
> T: expiry time of the option contract

Figure A.1

EUROPEAN EQUITY INDEX CALL OPTION PAYOFF



Based on the payoff function of the European call option, the fair price of the option can be determined as the discounted value of the expected payoff.

$e^{-rT} \cdot \mathrm{E}[\max(S_T - K, 0)]$

where

r: discount rate

When the index value is assumed to follow the geometric Brownian motion (GBM) process, the fair value of the European option can be calculated using a closed-form solution, as the famous Black-Scholes formula, or a stochastic approach on which multiple paths of index values are simulated until the expiry date. The fair value is then calculated as the average of discounted payoffs under simulated paths. Although closed-form solutions exist for European-style equity index options, it is not true of many other applications that depend on not only the ending value, but also the path that leads to it. Based on the central limit theorem, the estimator that takes the average of the results of N paths has a convergence speed on the order of $1/\sqrt{N}$, according to the Berry–Esseen theorem discovered by Berry (1941) and Esseen independently.

The second approach is used in quantum computing because closed-form solutions are not available for many other applications. To implement it in quantum computing, the amplitude estimation algorithm created by Brassard, et al. (2002) is used, which can achieve quadratic speedup compared to classical algorithms of Monte Carlo simulations.

Amplitude estimation builds on the concept of amplitude magnification where, through a series of rotations of qubits, the probability of the given eigenstate being observed increases. Formally, operator $\mathcal{A}$ is defined as follows.

$$\mathcal{A}|0\rangle = \sqrt{1-a}|\Psi_0\rangle + \sqrt{a}|\Psi_1\rangle$$

Where

Ψ_0: state where the qubit will end up as 0 after superposition. You may consider this the state where the option will end up with no payoff.

Ψ_1: state where the qubit will end up as 1 after superposition. You may consider this the state where the option will end up with positive payoffs.

a: the amplitude of the state $|\Psi_1\rangle$, which is calculated as $|\langle\Psi_1|\Psi_1\rangle|^2$. The probability is that the qubit will end up in state Ψ_1.

To evaluate the fair price of the option, we can apply the operator to n+1 qubits. The first n qubits can be considered as the simulation part, while the last qubit will tell us which state the qubit sequence will be: 0 or 1.

$$\mathcal{A}|0\rangle_{n+1} = \sqrt{1-a}|\Psi_0\rangle_n|0\rangle + \sqrt{a}|\Psi_1\rangle_n|1\rangle$$

Amplitude estimation is the algorithm that tries to find out the amplitude *a* of the interested ending state 1. According to Brassard, et al. (2002), *a* can be estimated using additional sampling qubits and quantum phase estimation, as explained in section 2.6.

In this example, Qiskit is used to show the quantum computing process. Qiskit is a quantum programming toolkit that was developed by IBM Research and the Qiskit development community. Qiskit allows us to define quantum circuits, perform operations and measurements, and visualize the results. Many visualizations of examples in this report are produced using Qiskit. Qiskit can be installed in a Python environment. Programs developed using Qiskit can be run on prototype IBM quantum computers or quantum simulators, which can be your personal computer. Many other choices are available for developing and testing programs that are targeted for quantum computing, such as the Ocean Software development kit designed by D-Wave, a pioneer in the quantum computing industry. Other popular choices include Google's Cirq and Microsoft's Q#.

Most quantum programming development kits include prebuilt functions and routines that are helpful for quantum computing. Therefore, developing a quantum program is not very different from writing a program in Python or other programming language, except that the problems need to be solved using quantum computing bases and algorithms. In addition, some desired functionalities may not be readily available and may not to be built from the basic components of quantum computing.

Qiskit is hosted at https://qiskit.org/ with detailed documentation, examples, and instruction of installation. The following example uses version 0.43.2 and is built based on an existing Qiskit example (https://qiskit.org/ecosystem/finance/tutorials/03_european_call_option_pricing.html). However, the example is revised in the following aspects:

- More documentation to explain the codes leveraging on the concepts introduced in this report
- Revised to accommodate latest updates of Qiskit
- Complementing with error analysis to show the impact of estimation error (quantum amplitude estimation) and sampling error separately.

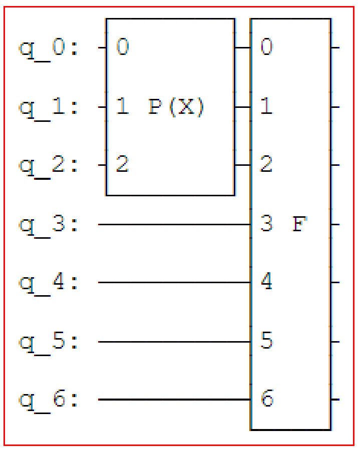
Qiskit Program


```
# Import packages and relevant function
import numpy as np
from scipy.stats import norm
from qiskit import QuantumCircuit
from qiskit.algorithms import IterativeAmplitudeEstimation, EstimationProblem #Aplittude Estimation
from qiskit.circuit.library import LinearAmplitudeFunction #define piecewise linear function
from qiskit_aer.primitives import Sampler #sampling
from qiskit_finance.circuit.library import LogNormalDistribution #Lognormal distribution by quantum circuits

# Setting parameters
S = 100.0 # stock spot price
K = 100.0 #at-the-money call option
eq_vol = 0.25 # annualized equity volatility
```

```
rfr = 0.03 # risk free interest rate
T = 42 / 252 # approximately 2 months until maturity, assuming 252 business days

# Resulting parameters for log-normal distribution
mu = (rfr - 0.5 * eq_vol**2) * T + np.log(S)
sigma = eq_vol * np.sqrt(T)
mean = np.exp(mu + sigma**2 / 2)
variance = (np.exp(sigma**2) - 1) * np.exp(2 * mu + sigma**2)

# Set up the lower and upper bound of the equity price to be considered in the option pricing.
S_lower = np.maximum(0, mean - 3 * np.sqrt(variance))
S_upper = mean + 3 * np.sqrt(variance)

# Create the equity price distribution following the lognormal model.
n_qubits = 3 # number of qubits to represent the distribution of equity price
eq_price_model = LogNormalDistribution(n_qubits, mu=mu, sigma=sigma**2, bounds=(S_lower, S_upper))

# Create the payoff function max(0,S(T)-K) as a piecewise linear function
# LinearAmplitudeFunction creates a circuit implementing a piecewise linear function on qubit amplitudes.
# Starting from the lower bound of the considered range, with a value of 0 (f_min) and a slope of 0 until the stock price reaches the strike price K, the slope becomes 1, with a maximum value of S_upper subtracted by K.
breakpoints = [S_lower, K]
slopes = [0, 1]
offsets = [0, 0]
f_min = 0
f_max = S_upper - K
c_approx = 0.25 # the scaling factor used in approximation of the amplitude function F using Taylor expansion

eu_call_payoff = LinearAmplitudeFunction(
    n_qubits,
    slopes,
    offsets,
    domain=(S_lower, S_upper),
    image=(f_min, f_max),
    breakpoints=breakpoints,
    rescaling_factor=c_approx,
)
```

```
# Create a quantum circuit with both the distribution function of the ending stock price S(T) and the payoff function for quantum amplitude estimation
num_qubits = eu_call_payoff.num_qubits # number of qubits needed for the payoff function in terms of a piecewise function
eu_call = QuantumCircuit(num_qubits)
eu_call.append(eq_price_model, range(n_qubits))
eu_call.append(eu_call_payoff, range(num_qubits))

# Visualize the circuit for the European equity index call pricing model
eu_call.draw()
```

Output




```
# Here it contains 7 qubits with q_3 the target qubit to be measured for amplitude estimation.

# Perform amplitude estimation
# Set target precision and confidence level
epsilon = 0.01
significance_level = 0.05 #confidence level = 1 - 0.05 = 0.95

# Define the estimation problem by supplying the circuit and objective/target qubit to measure
problem = EstimationProblem(
    state_preparation=eu_call,
    objective_qubits=[3], # q_3 is the target qubit to measure which is the amplitude function F.
    post_processing=eu_call_payoff.post_processing,
)

# Create amplitude estimation
ae_obj = IterativeAmplitudeEstimation(
```

```
    epsilon_target=epsilon, alpha=significance_level, sampler=Sampler(run_options={"shots": 1000, 'seed':
123})
) #number of simulations; random number generation seed: 123

result = ae_obj.estimate(problem)
conf_int = np.array(result.confidence_interval_processed)
print("Estimated value: %.4f" % (result.estimation_processed))
print("Confidence interval: [%.4f, %.4f]" % tuple(conf_int))
```

Output
Estimated value: 4.5458
Confidence interval: [4.2918, 4.7998]

```
# Error analysis
print("Estimated value using amplitude estimation algorithm: %.4f" % (result.estimation_processed))
x = eq_price_model.values
y = np.maximum(0, x - K)
exact_value = np.dot(eq_price_model.probabilities, y)
print('Exact expected value based on sampling:%.4f' % exact_value) #amplitude estimation error

# Calculate the theoretical price using Black Scholes formula
def B_S_EU_Call_Price(S, K, T, r, sigma):
    d1 = (np.log(S/K) + (r + sigma**2/2)*T) / (sigma*np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    return S * norm.cdf(d1) - K * np.exp(-r*T)* norm.cdf(d2)
print("Theoretical call option price: %.4f" % (B_S_EU_Call_Price(S, K, T, rfr, eq_vol))) #sampling error
```

Output
Estimated value using amplitude estimation algorithm: 4.5458
Exact expected value based on sampling: 4.4218
Theoretical call option price: 4.3140

Here the estimation error is 0.124, the difference between the estimated value using the amplitude estimation algorithm and exact expected value based on sampling. The sampling error is 0.1078, the difference between the exact expected value based on sampling and the theoretical call option price using Black Sholes formula.

Through this simple example, it is clear that, as long as basic functionalities are available in quantum computing development tools, actuaries can still write programs to be used in quantum computers like writing programs using any other Python packages. However, actuaries need to get familiar with these existing quantum computing functionalities and potentially contribute by building new functionalities for insurance related applications. Actuaries can deal with the mathematical models in quantum computing, while leveraging on advances of quantum computers and general quantum algorithms.

Endnotes

---

[1] A **superposition** state is represented mathematically as

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

where

$|\psi\rangle$: a superposition state.

$|0\rangle$ and $|1\rangle$: basis states of the qubit. $|0\rangle$ stands for vector [1, 0] and $|1\rangle$ stands for vector [0,1]

α and β: the amplitudes of the qubit being in the 0 or 1 state respectively. They can be real numbers or complex numbers with both real and imaginary components.

In quantum mechanics, the probability of measuring a particular outcome is given by the square of the absolute value of the probability amplitude corresponding to that outcome. The probability of measuring the qubit in the state $|0\rangle$ is given by $|\alpha|^2$, and the probability of measuring the qubit in the state $|1\rangle$ is given by $|\beta|^2$. It is worth noting that the sum of the probabilities of all possible outcomes for a given observable must always add up to 1, meaning that the probabilities of measuring all possible outcomes must be normalized. This normalization condition is often written mathematically as $|\alpha|^2 + |\beta|^2 = 1$

[2] **Pauli – X gate:** In quantum computing, the X gate is a one-qubit quantum gate that is used to perform a NOT operation. The X gate transforms the basis state $|0\rangle$ into the basis state $|1\rangle$, and vice versa. Mathematically, the X gate is represented by the following matrix:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Given a state $|0\rangle$, the X gate will change the state to $|1\rangle$

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

Similar, X-gate will transform $|1\rangle$ to $|0\rangle$**.**

$$X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$

[3] The **Hadamard gate** changes a basic state to a superposition state, as specified below.

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

[4] Another important gate in quantum computing is Controlled NOT gate, also known as **CNOT gate**. It involves two qubits, with the first qubit as a target qubit whose state will be changed based on the second qubit, which is a control qubit.

A set of two qubits $|a\rangle$ can be defined as a vector of four elements, representing the four possible basic states $|00\rangle, |01\rangle, |10\rangle$, and $|11\rangle$.

$$|a\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$$

The CNOT gate can be defined as the following matrix:

$$CNOT = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$$

$$CNOT|a\rangle = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}a_{00}\\a_{01}\\a_{10}\\a_{11}\end{bmatrix} = \begin{bmatrix}a_{00}\\a_{01}\\a_{11}\\a_{10}\end{bmatrix}$$

[5] The details on the operation of the circuit that generates the **Bell state** in the example is given as follows:

- Apply Hadamard gate the first qubit: $H|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- Adding the second qubit, which is $|0\rangle$, the superposition of the two qubits becomes $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$.
- With the CNOT operation, when the control circuit is in state $|1\rangle$, then the target qubit will be changed to state $|1\rangle$ as well. The superposition then becomes $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

[6] The detail of the calculation in the example of **Grover's algorithm** is provided in table E.1.

Table E.1

GROVER'S ALGORITHM ILLUSTRATION

Item	Unsorted Starting Point	Iteration 1		Iteration 2[8]	
		Flip the Target	Flip around Average	Flip the Target	Flip around Average
A_1	0.354[3]	0.354	0.177[6]	0.177	-0.088
A_2	0.354	0.354	0.177	0.177	-0.088
A_3	0.354	0.354	0.177	0.177	-0.088
A_4	0.354	0.354	0.177	0.177	-0.088
A_5	**0.354**	**-0.354**[4]	**0.884**[7]	**-0.884**	**0.972**
A_6	0.354	0.354	0.177	0.177	-0.088
A_7	0.354	0.354	0.177	0.177	-0.088

A_8	0.354	0.354	0.177	0.177	-0.088
Average		0.265[5]		0.044	

Notes:

1. Total number of items to be searched (N): 8.
2. Item A_5 is assumed to be the desired item.
3. The initial amplitude of each item is the same and calculated as $\frac{1}{\sqrt{N}}$, which is $\frac{1}{\sqrt{8}}$ in this example.
4. The sign of the desired item's amplitude is changed.
5. After flipping the sign of the desired item's amplitude, the average of the amplitudes of all *N* items is calculated. In this example, it is $\frac{3\sqrt{2}}{16}$.
6. The reflection of the amplitude against the average amplitude calculated in the previous step is used as the new reflection. In this example, for undesired items, it is $2 \times \textit{average amplitude} - \textit{current amplitude} = 2 \times \frac{3\sqrt{2}}{16} - \frac{1}{\sqrt{8}} = \frac{\sqrt{2}}{8}$.
7. For the desired item, the new amplitude becomes $2 \times \textit{average amplitude} - \textit{current amplitude} = 2 \times \frac{3\sqrt{2}}{16} + \frac{1}{\sqrt{8}} = \frac{5\sqrt{2}}{8}$.
8. The second iteration follows the same process of the first iteration.

[7] Current pre-quantum encryption technologies, such as RSA, rely on the difficulty of factorization of a large number, because usually the trial-and-error approach will be used, which requires much time and computing resources to succeed. Factorization is a process to find all the prime number factors. For example, 210 can be factorized as 2 X 3 X 5 X 7. **Shor's algorithm** can be used to factorize a large number much faster on quantum computers, which renders the current encryption useless. The algorithm finds the factors using the following process:

1. Assuming N is the number to be factorized, we first give a guess p, which may potentially be a factor of N, or shares a factor with N. For example, if *N* equals 51 and *g* equals 9, we can check if our guess is useful.

2. The Euclidean algorithm can be used to check if N and our guess g have any common factors. Here, we want to derive the greatest common factor (gcf) of N and g. Gcf is also known as the greatest common divisor (gcd). The Euclidean algorithm determines the larger of the two numbers and replaces the larger number with the differences between the larger number and the integer multiple of the smaller number. This step is repeated until the difference becomes zero.

   - gcf(51, 9) = gcf(51 – 9 X 5, 9) = gcf(6, 9)
   - gcf(6, 9) = gcf(6, 9 – 6) = gcf (6, 3)
   - gcf(6, 3) = gcf(6 – 2 X 3, 3) = gcf(0, 3) = 3

   Three is then found as a factor of N in this example, with the remaining factor 17 (51/3) identified as well.

3. However, we are not always lucky with our guess of g, especially when N is large. The guess g needs to be adjusted when gcf(N, g) equals 1, also known as coprimes. For example, gcf(15, 7) equals 1. With N and g, we make guesses of p and construct the following equation, with both p and m as integers:

$$g^p = m \times N + 1$$

$$g^p - 1 = m \times N$$

$$\left(g^{\frac{p}{2}} - 1\right)\left(g^{\frac{p}{2}} + 1\right) = m \times N$$

The guess of g is then changed to $g^{\frac{p}{2}} - 1$ and $g^{\frac{p}{2}} + 1$.

In the example with N equal to 15 and g equal to 7, by trial and error, we can find p equals 4, which satisfies the desired equation.

$$7^4 = 160 \times 15 + 1$$

$$\left(7^{\frac{4}{2}} - 1\right)\left(7^{\frac{4}{2}} + 1\right) = 160 \times 15$$

$$(48)(50) = 160 \times 15$$

Our new guess of g becomes 48 and 50. Using the Euclidean algorithm, gcf(48, 15) equals 3 and gcf(50, 15) equals 5. It is also notable that, once we find a valid p, we can find multiple values of p that meet the requirements. As shown in the following equation, if $(g^p - 1)$ satisfies the requirement, so is $(g^{np} - 1)$ if n is a positive integer.

$$g^{np} - 1 = (g^p - 1)\left(g^{(n-1)p} + g^{(n-2)p} + \cdots + g^p + 1\right), n \in \{1,2,3, \dots\}$$

The key in Shor's algorithm is how the value of p can be found in an efficient way. When N is large, finding the p that satisfies the equation is not a trivial task. However, with quantum computers and algorithms such as quantum Fourier transform, the periodicity of p allows it to be found quickly, which makes current encryption powerless.

[8] The selected algorithms are available at https://csrc.nist.gov/projects/post-quantum-cryptography/selected-algorithms-2022 with the announcement at https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms.

References

Aaronson, Scott, 2015. “Read the Fine Print.” Nature Physics, 11(4):291–293.

AVIVA, 2021. “Quantum leap: A new computing revolution brings security risks and investment opportunities.” https://www.avivainvestors.com/en-ca/views/aiq-investment-thinking/2021/07/quantum-computing/

Basheer, Afrad, A Afham, and Sandeep K Goyal, 2020. “Quantum k-nearest neighbors algorithm.” https://arxiv.org/abs/2003.09187

Berry, Andrew C. 1941. "The Accuracy of the Gaussian Approximation to the Sum of Independent Variates". Transactions of the American Mathematical Society. 49 (1): 122–136.

Borujeni, S. E., S. Nannapaneni, N. H. Nguyen, E. C. Behrman, and J. E. Steck, 2021. “Quantum circuit representation of Bayesian networks.” Expert Systems with Applications.

Chakrabarti, Shouvanik, Rajiv Krishnakumar, Guglielmo Mazzola, Nikitas Stamatopoulos, Stefan Woerner, and William J. Zeng, 2021. “A Threshold for Quantum Advantage in Derivative Pricing.” https://arxiv.org/abs/2012.03819

Cherrat, El Amine, Snehal Raj, Iordanis Kerenidis, Abhishek Shekhar, Ben Wood, Jon Dee, Shouvanik Chakrabarti, Richard Chen, Dylan Herman, Shaohan Hu, Pierre Minssen, Ruslan Shaydulin, Yue Sun, Romina Yalovetzky, and Marco Pistoia, 2023. “Quantum Deep Hedging.” https://arxiv.org/abs/2303.16585

Date, Prasanna and Thomas Potok, 2021. “Adiabatic quantum linear regression.” Scientific reports, 11(1):21905.

Giurgica-Tiron, Tudor, Sonika Johri, Iordanis Kerenidis, Jason Nguyen, Neal Pisenti, Anupam Prakash, Ksenia Sosnova, Ken Wright, and William Zeng, 2021. “Low depth amplitude estimation on a trapped ion quantum computer” https://arxiv.org/abs/2109.09685

Grover, Lov K., 1996. “A fast quantum mechanical algorithm for database search.” https://arxiv.org/abs/quant-ph/9605043

Havlíček, Vojtěch, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta, 2019. “Supervised learning with quantum-enhanced feature spaces.” *Nature*, 567(7747):209–212.

Herman, Dylan, Cody Googin, Xiaoyuan Liu, Alexey Galda, Ilya Safro, Yue Sun, Marco Pistoia, and Yuri Alexeev, 2022. “A Survey of Quantum Computing for Finance.” https://arxiv.org/abs/2201.02773

Herr, Daniel, Benjamin Obert, and Matthias Rosenkranz, 2021. “Anomaly detection with variational quantum generative adversarial networks.” *Quantum Science and Technology*, 6(4).

Hodson, Mark, Brendan Ruck, Hugh Ong, David Garvin, and Stefan Dulman. 2019. “Portfolio rebalancing experiments using the quantum alternating operator ansatz.” https://arxiv.org/pdf/1911.05296.pdf

Kerenidis, Iordanis, Anupam Prakash, and Dániel Szilágyi, 2021. “Quantum algorithms for second-order cone programming and support vector machines.” *Quantum*, 5:427.

Kitaev, A.Yu. 1995. “Quantum measurements and the Abelian Stabilizer Problem.” https://arxiv.org/abs/quant-ph/9511026

Lloyd, Seth, Masoud Mohseni, and Patrick Rebentrost, 2014. “Quantum principal component analysis.” *Nature Physics*, 10(9):631—633.

Kalra, Angad, Faisal Qureshi, and Michael Tisi. 2018. “Portfolio asset identification using graph algorithms on a quantum annealer.”

Milne, Andrew, Maxwell Rounds, and Phil Goddard, 2017. “Optimal feature selection in credit scoring and classification using a quantum annealer.” *White Paper 1Qbit*. http://1qbit.com/files/white-papers/1QBit-White-Paper-%E2%80%93-Optimal-Feature-Selection-in-Credit-Scoring-and-Classification-Using-a-Quantum-Annealer_-_2017.04.13.pdf

Mitarai, K., M. Negoro, M. Kitagawa, and K. Fujii, 2018. “Quantum circuit learning.”

Miyamoto, Koichi and Kenji Kubo, 2021. “Pricing multi-asset derivatives by finite difference method on a quantum computer.”

Mosca, Michele, and Marco Piani, 2022. “Quantum Threat Timeline Report 2022.” Global Risk Institute. https://globalriskinstitute.org/mp-files/2022-quantum-threat-timeline-report-dec.pdf/

Munich Re, 2021. “Quantum Technology and Application Consortium (QUTAC): Munich Re participates in the establishment of the new consortium.” https://www.munichre.com/en/company/media-relations/media-information-and-corporate-news/corporate-news/2021/2021-06-10-qutac.html

Rosenberg, Gili and Maxwell Rounds, 2018. “Long-short minimum risk parity optimization using a quantum or digital annealer.” *White Paper 1Qbit*. http://1qbit.com/files/white-papers/1QBit-White-Paper-%E2%80%93-Long-Short-Minimum-Risk-Parity-Optimization-Quantum-or-Digital-Annealer.pdf

Sakuma, T. 2020. “Application of deep quantum neural networks to finance.” https://arxiv.org/abs/2011.07319

Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost, 2013. “Quantum algorithms for supervised and unsupervised machine learning.”

Shang, Kailan, and Zakir Hossen. 2019. Liability-Driven Investment: Benchmark Model. https://www.soa.org/498f7f/globalassets/assets/files/resources/research-report/2019/liability-driven-investment.pdf

Shang, Kailan. 2021. “Deep Learning for Liability-Driven Investment.” https://www.soa.org/4964e1/globalassets/assets/files/resources/research-report/2021/liability-driven-investment.pdf

Shor, P. W. 1994. "Algorithms for quantum computation: discrete logarithms and factoring" *Proceedings 35th Annual Symposium on Foundations of Computer Science*:124-134.

Slate, N., E. Matwiejew, S. Marsh, and J. B. Wang, 2021. “Quantum walk-based portfolio optimisation.” *Quantum*, 5:513.

Stamatopoulos, Nikitas, Daniel J. Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen, and Stefan Woerner, 2020. “Option pricing using quantum computers.” *Quantum*, 4:291.

Tang, Hao, Anurag Pal, Tian-Yu Wang, Lu-Feng Qiao, Jun Gao, and Xian-Min Jin, 2021. "Quantum computation for pricing the collateralized debt obligations." *Quantum Engineering*, 3(4)

Warren, G. Geoff, 2019. Choosing and using utility functions in forming portfolios. *Financial Analysts Journal*, 75:39–69.

Zhao, Yunpeng Zhao, Haiyan Wang, Kuai Xu, Yue Wang, Ji Zhu, and Feng Wang, 2022. "Adaptive Algorithm for Quantum Amplitude Estimation." https://arxiv.org/abs/2206.08449

About The Society of Actuaries Research Institute

Serving as the research arm of the Society of Actuaries (SOA), the SOA Research Institute provides objective, data-driven research bringing together tried and true practices and future-focused approaches to address societal challenges and your business needs. The Institute provides trusted knowledge, extensive experience and new technologies to help effectively identify, predict and manage risks.

Representing the thousands of actuaries who help conduct critical research, the SOA Research Institute provides clarity and solutions on risks and societal challenges. The Institute connects actuaries, academics, employers, the insurance industry, regulators, research partners, foundations and research institutions, sponsors and non-governmental organizations, building an effective network which provides support, knowledge and expertise regarding the management of risk to benefit the industry and the public.

Managed by experienced actuaries and research experts from a broad range of industries, the SOA Research Institute creates, funds, develops and distributes research to elevate actuaries as leaders in measuring and managing risk. These efforts include studies, essay collections, webcasts, research papers, survey reports, and original research on topics impacting society.

Harnessing its peer-reviewed research, leading-edge technologies, new data tools and innovative practices, the Institute seeks to understand the underlying causes of risk and the possible outcomes. The Institute develops objective research spanning a variety of topics with its strategic research programs: aging and retirement; actuarial innovation and technology; mortality and longevity; diversity, equity and inclusion; health care cost trends; and catastrophe and climate risk. The Institute has a large volume of topical research available, including an expanding collection of international and market-specific research, experience studies, models and timely research.

Society of Actuaries Research Institute
475 N. Martingale Road, Suite 600
Schaumburg, Illinois 60173
www.SOA.org