10-2021

Quantum computing: Computational excellence for Society 5.0

Paul R. GRIFFIN
Singapore Management University, paulgriffin@smu.edu.sg

Michael BOGUSLAVSKY
Tradeteq, London

Junye HUANG
IBM Quantum, Singapore

Robert J. KAUFFMAN
Singapore Management University, rkauffman@smu.edu.sg

Brian R. TAN
B. R. I. T. Management Consulting, Singapore

Citation

GRIFFIN, Paul R.; BOGUSLAVSKY, Michael; HUANG, Junye; KAUFFMAN, Robert J.; and TAN, Brian R.. Quantum computing: Computational excellence for Society 5.0. (2021). *Data Science and Innovations for Intelligent Systems: Computational Excellence and Society 5.0*. 1-32.
Available at: https://ink.library.smu.edu.sg/sis_research/6851

Quantum Computing: Computational Excellence for Society 5.0

Paul R. Griffin[1], *Michael Boguslavsky*[2], *Junye Huang*[3], *Robert J. Kauffman*[4], *and Brian R. Tan*[5]
1 School of Information Systems Singapore Management University, Singapore
2 Tradeteq, London, United Kingdom
3 IBM Quantum, Singapore
4 Copenhagen Business School Denmark, and School of Information Systems, Singapore Management University, Singapore
5 B.R.I.T. Management Consulting, Singapore

Abstract: In this chapter, we consider which general business problems may be suitable for exploring the utilization of quantum computing and provide a framework for applying quantum computing. The characteristics of quantum computing systems are mapped into business problems to show the potential advantages of quantum computing. The framework shows how quantum computing can be applied in general, and a use case is offered for quantum machine learning (QML) related to the credit ratings of small and medium-size enterprises (SMEs).

CONTENTS

Published in Data science and innovations for intelligent systems : Computational excellence and society 5.0 / edited by Kavita Taneja, et al. Boca Raton, FL: CRC Press, 2022.
DOI: 10.1201/9781003132080-1

1.1 INTRODUCTION

Quantum computing is a key part of building an intelligent systems infrastructure for Society 5.0 and can be used in the future across the main pillars of fintech, healthcare, logistics, and artificial intelligence (AI). Intelligent systems based on data science are machines that are sufficiently advanced to be able to perceive and react to external events. Quantum computers offer various avenues to go beyond systems using classical computers and extend computational excellence beyond its current state. Digital innovation underpins the concept of Society 5.0 for a better future with an inclusive, sustainable, and knowledge-intensive society that uses information computing. A key to realizing this society is to utilize gargantuan volumes of data in real-time in intelligent systems. The sharing of information in Society 4.0 has been insufficient and integration of data problematic, whereas Society 5.0 integrates cyberspace and physical space. For example, in Society 5.0, the huge amount of data from physical Internet of Things (IoT) devices are required to be analyzed, processed, and fed back to robotic devices interacting with people in various forms. In Japan alone, the next 15 years is expected to see a growth in IoT and robotics of US$20 bn and US$70 bn, respectively (JapanGov News, 2019). However, the aim of Society 5.0 is to balance economic development and solutions for social issues to bring about a human-centered society. This chapter shows how quantum computing can be applied to many current challenges and open up new opportunities with innovative ways that align better with human thinking.

Classical computing has brought society great benefits over many years from the abacus 3,300 years ago through to modern computing from Alan Turing in the 20th century – to the latest smartphones we are now familiar with. Computers have enabled products and services that humans cannot provide alone such as increasing productivity, enhancing communication, storing vast amounts of data, sorting, organizing, and searching through information amongst many more. However, many problems still exist such as data security, scalability, manageability, and interoperability. Furthermore, Moore's Law increases in computing power is now beginning to fail (Loeffler, 2018). Stefan Filipp, a quantum scientist at IBM Research, has stated that to "*continue the pace of progress, we need to augment the classical approach with a new platform, one that follows its own set of rules. That is quantum computing*" (Singh, 2019). Using the advantages of quantum over classical computing, it is possible to increase computing capacity beyond anything that classical computers can achieve.

Quantum computing was suggested in the 1980s by Manin (1980) and Feynman (1982). In the past few years, it has become a reality and accessible to everyone, with IBM putting the first quantum computer on the cloud in 2016. Now, in 2021, there are dozens of quantum computers online with processing capabilities much better than the first one. While there is little doubt that quantum computers can outperform classical computers for some processes, such as unstructured search problems (Grover, 1996), it is not clear whether and how quantum computing will be advantageous for a particular business need or, indeed, worth the effort to investigate further.

This chapter is aimed at providing a framework to assess the likelihood that quantum computing will be an area that is worthwhile to get involved in for particular business opportunities and challenges. The main differences for quantum computing are *superpositio*n and *entanglemen*t. Traditional computers use bits of either 0 or 1. In contrast, quantum computers use qubits existing in a state that is best described as the probability of being either 0 or 1. This is called the *superposition of states* (Nielsen & Chuang, 2010).[1] Qubits also exhibit entanglement, whereby they may be spatially nearby or far apart, may interact with one another at certain times, and yet are not able to be characterized as being independent of one another. The result is that two qubits may work together as if they were one larger qubit. This is fundamentally different from bits that are always kept separate in classical computing. However, current quantum computers are noisy and have an insufficient number of qubits to be able to show provable advantages. And even the widely publicized Google experiment (Arute et al., 2019) is still held to be contentious (Pednault, 2020). Even more contentious are *annealer-type quantum computers* (Rønnow et al., 2014). These will not be covered in this chapter as a result, and we will focus on *gate-type quantum computers* instead.

Considering the Society 5.0 issues of data volumes, real-time processing and linking data, we present a framework to assess what business needs may potentially be addressed by quantum computing and how quantum computing is different to classical systems. There are four areas of concern: the data, the processing, the infrastructure, and the environment. (See Figure 1.1.) First, data may be complex,

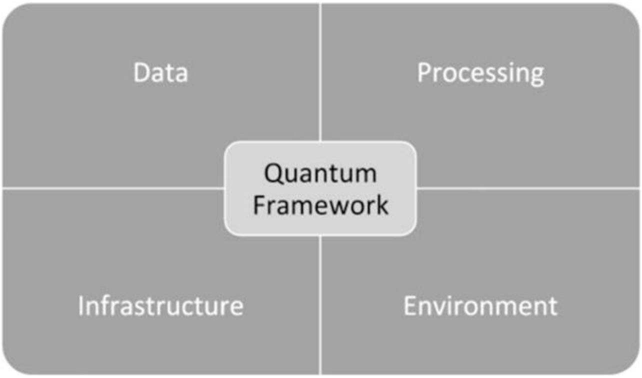


FIGURE 1.1 The main areas of concern in the quantum application framework.

natively quantum or probabilistic, for example, chemical reactions involving quantum particles and financial derivatives pricing including many predictor variables. Second, processing may involve a range of different data analytics approaches, such as simulation, machine learning (ML), optimization, and AI. Third, current quantum computing infrastructure solutions vary in their types and specifications, with the overall marketplace in the throes of rapid technological innovation. Last, the environment may have legal regulations in terms of what can be done, how data can be used, and how physical and human resources can be used while protecting private information.

The remainder of this chapter is laid out as follows. Section 1.2 reviews the main differences between quantum computing and classical computing. It introduces the concepts of qubits, quantum states, and quantum operations. It also gives overviews of different approaches to quantum hardware, as well as the end-to-end process of running a quantum algorithm. Available software and debugging challenges are also discussed, along with some common quantum algorithms. Section 1.3 looks at current managerial concerns and related business problems that quantum computing can potentially address. Section 1.4 offers a framework showing the areas that need to be addressed when considering whether to build a quantum computing solution. Using this framework, Section 1.5 provides an example of its application for a *quantum neural network* (QNN) solution for the credit rating of *small and medium-size enterprises* (SMEs). Section 1.6 looks at the future of quantum computing, covering improvement areas for theoretical development, hardware, and integration with analytics software.

While there is no expectation that quantum computers will outperform current classical models in the near term, exploring the requirements and limitations of current quantum computers can be useful for thinking through how to develop a future system to meet business objectives when quantum computing reaches a suitable level of maturity.

1.2 QUANTUM COMPUTING FUNDAMENTALS

This section reviews the fundamental properties of quantum computers, and the end-to-end workflow from data preparation to analyzing the quantum circuit output. We further discuss the available software, with a focus on Qiskit as an example. Hardware and available algorithms are also assessed.

1.2.1 KEY CONCEPTS

The fundamental concepts of superposition, entanglement, interference, qubits, quantum gates, the concept of Bloch's sphere, and adiabatic annealers are key to understanding quantum computers. We will explain these concepts here.

Superposition. The fundamental processing and storage unit of quantum computers is the quantum bit, also called a *qubit.* A qubit can represent 0 or 1 like a classical bit. However, it can also represent 0 and 1 in a superposition state. A classical bit is either in the state 0 or 1. In contrast, a qubit is in a state which can be characterized by a general formula, $|\psi\rangle = a|0\rangle + b|1\rangle$, where a and b are probability amplitudes represented by complex numbers. Although a qubit can be in a superposition, when it is measured or read out in a computer language, the result can only be 0 or 1 and not both at the same time. However, by measuring a qubit multiple times, the probability of obtaining 0 or 1 is obtained. The probability of 0 is $|\ \mathrm{a}\ |^2$ and 1 is $|\ \mathrm{b}\ |^2$, and due to the conservation of the total probability being equal to 1, $|\ a\ |^2 + |\ b\ |^2 = 1$.

Entanglement. Another strange phenomenon related to qubits is entanglement. It refers to the correlation of different qubits in a system. If we extend the general formula for a qubit state to two qubits, we have: $|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$. Qubits are entangled if the state of one qubit is correlated with by the other. For example, when $b = c = 0$, the state becomes $a|00\rangle + d|11\rangle$. In this state, the two qubits' states are strongly correlated so that when the first qubit is $|0\rangle$ the second qubit is always also $|0\rangle$, and when the first qubit is $|1\rangle$ the second qubit is always $|1\rangle$.

Interference. Quantum states can interfere with each other due to the *phase* difference between the probability amplitudes. *Quantum interference* is similar to wave interference and can be understood in the same way. For example, when two waves act together in phase, their probability amplitudes sum to create a higher probability, and when the probability amplitudes are out of phase, their amplitudes cancel out. (See Figure 1.2.) This phenomenon can be utilized by quantum algorithms to speed up their calculations. For example, the Grover (1996) search algorithm uses interference to increase the probability of finding the right answer and reduce the probability of the wrong ones.

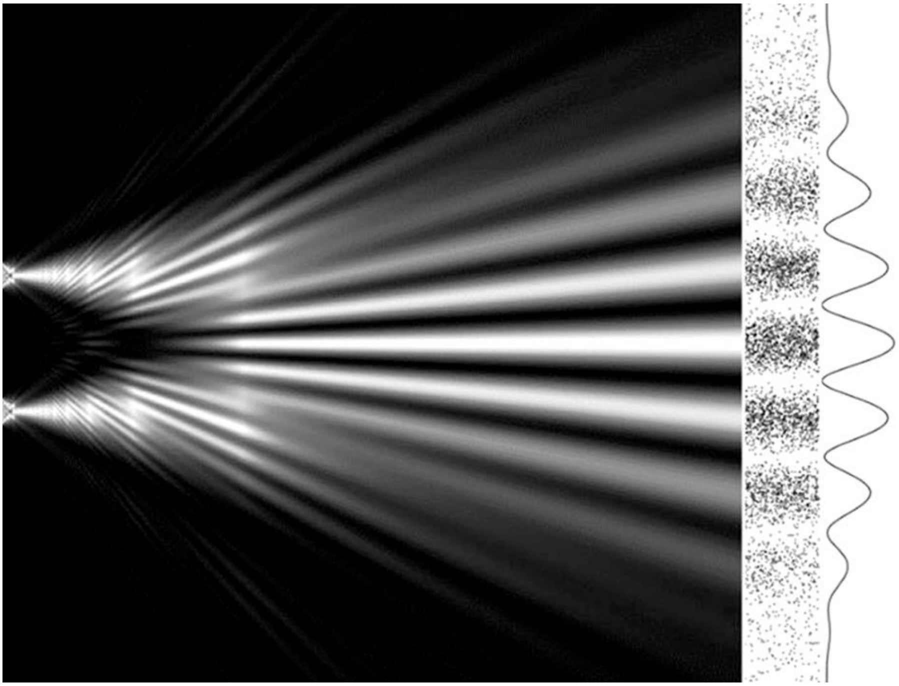


FIGURE 1.2 Interference pattern from a simulated double-slit experiment with electrons. *Source:* Alexandre Gondran, distributed under a CC-BY-SA-4.0 license.

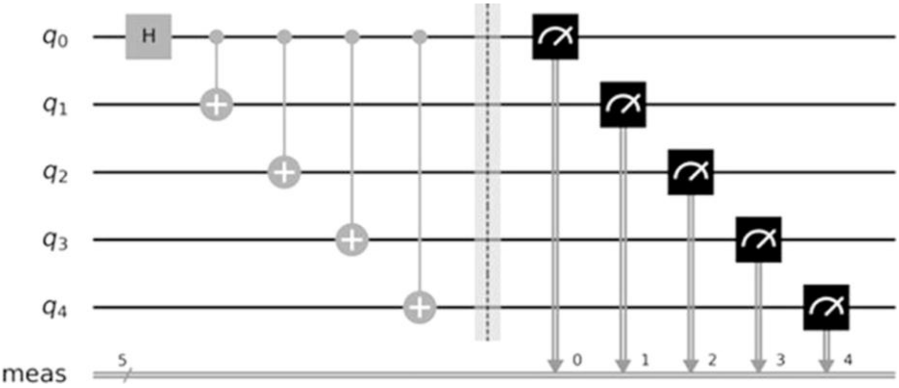


FIGURE 1.3 An example of a quantum circuit.

Quantum gates. Quantum algorithms can be represented by quantum circuits. Quantum circuits consist of quantum gates like the AND gate and the XOR gate from classical logic circuits. A common set of universal gates which can perform any quantum operation includes the Clifford gates and the T gates. This set of universal gates is composed of CNOT, H, S, and T gates.

Hadamard (H) gates map $|0\rangle$ to $|0\rangle + |1\rangle$ and $|1\rangle$ to $|0\rangle - |1\rangle$. H gates are ubiquitous in quantum algorithms for generating superpositions. S and T gates are part of the phase shift gates family. They leave $|0\rangle$ unchanged and map $|1\rangle$ to $e^{i\phi}|1\rangle$ (with $\phi = \pi/2$ and $\pi/4$ for the S and T gates, respectively). Phase gates are important for manipulating phases of quantum states to achieve intended interference.

Controlled NOT (CNOT) gates act on two qubits, unlike the H, S, and T gates, which act on a single qubit only. CNOT gates perform a NOT operation on the second qubit whenever the first qubit is in the state $|1\rangle$ and can be described as mapping the state $|a, b\rangle$ to the state $|a, a \oplus b\rangle$. CNOT is important for generating entanglement between qubits.

An example of a quantum circuit is shown next. (See Figure 1.3.) This circuit has an H gate on qubit-0 and four CNOT gates between qubit-0 (the control) and each of the remaining qubits (the target). Finally, the qubits are measured, and the outcomes are stored in classical registers.

1.2.2 Hardware, Software, Alorithms, and Workflow

We introduce a range of available software and hardware, followed by a general workflow for a quantum program's execution. The IBM Quantum Experience and Qiskit are used together as an example to illustrate a typical workflow.

Available hardware. As of January 2021, there were a handful of quantum computing hardware providers. They include: IBM Quantum Experience, Rigetti Quantum Cloud Services, AWS Braket, Microsoft Azure Quantum, Xanadu

Quantum Cloud, D-Wave Leap, Quantum Inspire (from QuTech), and Origin Quantum Cloud. All except AWS Braket and Azure Quantum have their own hardware systems. AWS Braket has three different external hardware providers: D-Wave (annealer), IonQ (trapped ions), and Rigetti (superconducting qubits), while Azure Quantum has IonQ (trapped ions), Honeywell (trapped ions), and Quantum Circuits (superconducting qubits). IBM Quantum Experience, Quantum Inspire, and Origin Quantum Cloud have systems that are open for the public free of charge, while the others charge for access.

IBM Quantum Experience (IQX) is the leading provider of quantum computing services. In 2016, IQX put the first quantum computer on the cloud. Since then, the platform has grown tremendously and now has more than 10 quantum systems (of up to 15 qubits) with free access and an additional more than 10 premium access quantum systems (with up to 65 qubits) for the IBM Quantum Network's partners. IQX has more than 250,000 users who collectively run more than 1 billion quantum circuits each day, and have published more than 250 related research papers (IBM.com, 2020). The IBM Quantum Network has more than 200 partners in industry and academia. IBM also released its quantum hardware roadmap in the annual Quantum Summit in September 2020, and laid out the steps toward building quantum systems with more than a million qubits. IBM aims to release processors with 127 qubits in 2021, 433 qubits in 2022, and 1,121 qubits in 2023 over the next three years (Gambetta, 2020). The quantum systems will be large enough to investigate the implementation of *quantum error correction* (QEC), to open the door to the practical implementation of many quantum algorithms.

Available software. There is a range of software development kits (SDKs) available for writing and running quantum programs, including Qiskit from IBM, Cirq from Google, QDK from Microsoft, Forest from Rigetti, and ProjectQ from ETH Zurich (LaRose, 2019). All of these SDKs, except QDK, are based on Python, which allows easy integration of the Python ecosystem's capabilities for scientific computing and ML. Many companies also work on quantum software packages for specific domain applications to interface with the SDKs mentioned previously. Notable examples include: Qiskit Aqua for chemistry, ML and optimization, from IBM; PennyLane for ML from Xanadu; OpenFermion for chemistry, from Google; and TensorFlow Quantum for ML, from Google.

Available algorithms. The most famous quantum algorithm is attributable to Shor (1997), who showed an exponential quantum speed-up related to the best- known classical algorithms for factorization. It is likely to threaten the existing cryptography infrastructure, however, it also requires a large number of qubits and gate operations, which also require QEC capabilities. The best estimate is that it will require on the order of millions of physical qubits (Gidney & Eker, 2019). Other famous textbook algorithms such as Grover's algorithm also require QEC capabilities, which will be difficult to implement in NISQ devices within the next three to five years.

There is a class of quantum-classical hybrid algorithms called *variational quantum eigensolver algorithms* (VQE, a kind of the more general variational quantum algorithms, VGAs). These are suitable for implementation on NISQ devices (Peruzzo et al., 2014). Such algorithms contain both quantum circuits and classical procedures that are invoked iteratively. The quantum circuits in each iteration have a small number (< 100) of qubits and a small number (also < 100) of quantum gates, and these can be run with current NISQ devices. The results of the quantum circuits are then fed to classical procedures for calculation and optimization to determine the parameters of a quantum circuit in the next iteration of the algorithm. This hybrid approach enables harnessing the power of quantum computers in the NISQ era before QEC becomes widely available.

Quantum computing workflow stages and examples. A typical quantum program workflow includes three stages: data loading, data processing execution, and data extraction.

- In the *data loading stage*, data are loaded into the memory of classical computer and converted to the states represented by qubits.
- In the *data processing execution stage*, a qubit's state is changed by the application of quantum gates. A compiler converts a logical circuit to a circuit that can be executed on a physical quantum processor, considering physical qubit connections and native gate sets.
- In the *data extraction stage*, all qubit states are read out by measurements. Due to the probabilistic nature of quantum states, multiple measurements are usually needed to sample the probability distribution in order to obtain a meaningful understanding of the solutions.

We next will look at Qiskit. It consists of four different modules: Terra for writing and running quantum circuits; Aer for high-performance simulation; Ignis for analyzing and mitigating noise; and Aqua for quantum algorithms and applications. Using Qiskit and the IQX, we illustrate an example of how a quantum program is executed on a quantum computer via the cloud. (See Figure 1.4.)

For this circuit, we don't need to load any data as the qubits are initialize to $|0\rangle$ by default. In the circuit execution stage, the program written in Python first needs to go through a compiler to convert it to a circuit that can run on a five-qubit quantum processor, such as the *ibmq_vigo* quantum processor. The H gate in Figure 1.3 is converted to a U2 gate that is native to IBM quantum processors. Such processors are superconducting devices that are usually not connected to all of the other qubits. However, the circuit written in Qiskit does not take this into account. So, it's the compiler's job to convert the logical circuit in Figure 1.3 to a circuit that can be run on the actual hardware. (See Figure 1.4 again.)

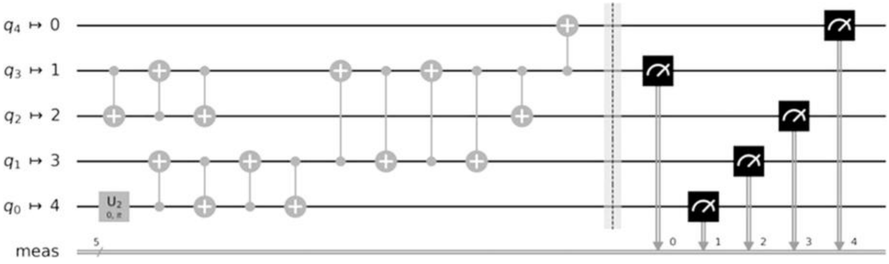


FIGURE 1.4 An example of a compiled quantum circuit.

If a CNOT gate is applied to qubits that are not directly connected, the compiler will need to apply a SWAP gate to move the quantum states of the qubits to two qubits that are connected. SWAP gates are implemented by using three CNOT gates, each of which generally has a ten times higher error rate than single qubit gates have. In the NISQ era, quantum processors do not have QEC and any errors may make a quantum program fail completely.

In Qiskit, this optimization procedure is implemented by a *transpile*r (i.e., a source-to-source compiler or transcompiler) consisting of a series of passes that optimize circuits based on different algorithms. The transpiled circuit is converted to a *quantum object* (Qobj) *file* and then sent to the IQX system (McKay et al., 2018). The Qobj file's contents control the electronics necessary to convert the program into microwave pulses transmitted to the quantum processor inside a cryogenic dilution refrigerator, which ensures that a low temperature is maintained. At the end of a quantum program, all the qubits will be read out, and the circuits and measurements will have run multiple times to sample the measurement outcome instead of only capturing a single-point measure.

1.3 QUANTUM COMPUTING NEEDS AND SERVICE INDUSTRY APPLICATIONS

We next consider a variety of business needs, and a new perspective on operational, managerial, and strategic problem-solving and decision-making that will benefit from the application of the quantum computing approach in the services industries. In addition to sharing our perspective on the reasons for the upwelling of interest in the new computational methods, we also provide some characteristics of appropriate business applications in operations, and the travel and hospitality services. We further offer a deeper reading of several prominent problem areas in contemporary financial services.

1.3.1 Business Needs and Concerns

Organizations operate in complex and dynamic economies driven by increasing social connectivity, the free flow of decentralized information, and technological advances that allow tracking of individuals using a network perspective. To deal with this increase in complexity and making good decisions (Nijs, 2015), managers in different kinds of organizations should be encouraged to move from a linear transaction logic and toward a networked logic that emphasizes value.

From the perspective of a *networked logic of value* (i.e., business value across business units, processes, product families, business partners, and smart networks in global value chains), and to make good decisions effectively, there is a need for businesses and organizations to capture and understand information from different sources. Information from the *internal environment*, such as employee-based or patent-related data, can help an organization to orchestrate its resources so it can develop higher-level capabilities and competencies. Likewise, information from the *external environment*, such as customer preference data, competitor information, or macro-environment trends, can aid an organization in decisions related to pricing, market segmentation, customer targeting, and product positioning. Understanding how to utilize both internal and external information types is critical for organizations to achieve sustainable advantage, driven by a networked logic of value.

To obtain relevant and meaningful information for decisions, organizations must consider four areas of managerial concern and related business problems. (See Table 1.1.)

Relevant information can only be obtained if the data used is accurate and captures what is being measured in a meaningful way. Data pertaining to real-life situations, such as customer preferences, competitor actions or employee data are inherently unstable, complex and dynamic. They often occur as range estimates (rather than just as point estimates); they also exhibit stochastic variation or are not easily measured or estimated. They interact with each other in complex probabilistic, interdependent and constrained ways (Ménard, Ostolic, Patel, & Volz, 2020).

1. *Data and information.* It is necessary to use well-specified and reliable data inputs to obtain good information for decision-making, lest such a decision process will devolve to GIGO thinking – or “garbage in, garbage out.” This acronym implies that bad inputs will result in bad output, and this principle applies more generally to all analysis and logic that cannot support meaningful evidence-based conclusions.
2. *Explanatory power and processing.* Considerations about respondent bias (Furnham, 1986) and the explanatory power of the solution methodology used should also be considered. In addition, the nature of processing is often linked with the types of data to be analyzed, as well as the computational activities that are required to achieve a solution for a complex problem.

TABLE 1.1 Concerns and Related Business Problems to Consider

Managerial Concerns	Examples of Related Business Problems
Data and information	Conversion of data with regard to numerous forms to information/knowledge; text, sentiments, natural language, images/videos, map/spatial representations; and firm/competition/market sources
Explanatory power and processing	Power of varied solutions/approaches; robustness/vulnerability to data problems; effectiveness of handling different kinds of complex problems; trying to match human-related conditions
Infrastructure and capabilities	Tech infrastructure; computational/staff support; human limitations in working on NP-hard vs. soft organizational problems; and limit to value in firm-level data gathering/problem models
Environment and regulation	Business sustainability, fair allocation of resources; appropriate use of physical/human resources; ensure compliance; and protect personal info (customers/partners/employees)

3. *Infrastructure and capabilities.* Organizations also must make judgment calls on the type of resources to utilize to develop capabilities and competencies in information gathering and interpretation. Resource types under consideration include the organization's choice of IT systems, its processes, systems, and staff capabilities.
4. *Environmental and regulatory constraints.* Since organizations operate within a larger societal context, they must develop systems within environmental sustainability and other constraints. Thus, governmental regulations on personal information protection and privacy laws, financial constraints, and security considerations are all important.

1.3.2 DECISION PROBLEM FRAMING AND COMPUTATION

Quantum computing is relevant in such situations, not just because of its potential computing supremacy or its advantage for some kinds of problems, but because of the way in which it achieves solutions via optimization, simulation, and unique distillation approaches (Arute et al., 2019). To understand how quantum computing can achieve supremacy, we should understand its source of power. The power from quantum computing comes from the qubit, exhibiting the characteristics of a range estimate and with multiple qubits, it is possible to express additional levels of correlation and stochasticity, to characterize problems of higher complexity (Arora & Barak, 2009). To create computational power, computer scientists have developed processes that leverage the nature of the qubit, through the integration of its stochastic range and point estimates, in a manner that makes it possible (with additional hardware developments) to obtain problem solutions with faster parallelism and speed. In contrast, classical computers operate in a linear manner, which slows them down. Both quantum computing and classical computing are competitive on problems of small to modest size.

There are three implications based on the source of power and differences in computing methods:

1. Quantum computing, like traditional computing, supports parallelism. A key contrast is that classical computers use more linear approaches to processes of computation though.
2. Suitably large *quantum machine learning* (QML) systems may lead to radical advances in the creation of thinking machines and AI that approaches the capabilities of humans.
3. Quantum computers are suitable for problems that are complex, combinatorial, and stochastic, where judgment in decision making is important. A wisdom-oriented approach towards thinking, by considering range-estimate inputs and requiring judgment becomes more relevant when compared to an intelligence-based approach (Jeste et al., 2010).

With these characteristics in mind, we consider some business problems in different service industries to which quantum computing seems uniquely suited.

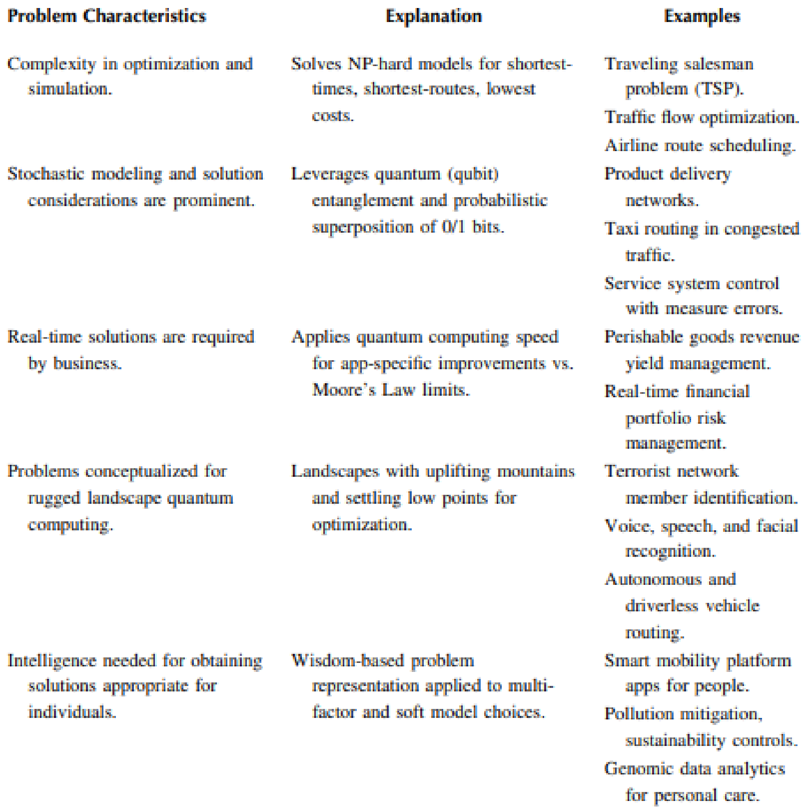
1.3.3 THE RANGE OF BUSINESS PROBLEM AREAS THAT CAN BE ADDRESSED

The business issues to which quantum computing methods apply possess characteristics that often are observed. (See Table 1.2.) We caution the reader to recognize, however, that quantum computing in late 2020 is at the apex of its hype cycle, but there are not yet “killer applications” or completed commercial tools that will permit applications of extreme complexity.

First, most problems to which the methods are appropriate involve considerable computational complexity for traditional computers to handle – especially problems that are non-deterministic and of *non-polynomial time-complexity* (NP-hard). In

---

TABLE 1.2 Problems that Quantum Computing Addresses: Optimization of Complexity, Stochastic Drivers, Real-Time Computation, Intelligent Simulation, and Rugged Landscape Analytics



Problem Characteristics	Explanation	Examples
Complexity in optimization and simulation.	Solves NP-hard models for shortest-times, shortest-routes, lowest costs.	Traveling salesman problem (TSP). Traffic flow optimization. Airline route scheduling.
Stochastic modeling and solution considerations are prominent.	Leverages quantum (qubit) entanglement and probabilistic superposition of 0/1 bits.	Product delivery networks. Taxi routing in congested traffic. Service system control with measure errors.
Real-time solutions are required by business.	Applies quantum computing speed for app-specific improvements vs. Moore’s Law limits.	Perishable goods revenue yield management. Real-time financial portfolio risk management.
Problems conceptualized for rugged landscape quantum computing.	Landscapes with uplifting mountains and settling low points for optimization.	Terrorist network member identification. Voice, speech, and facial recognition. Autonomous and driverless vehicle routing.
Intelligence needed for obtaining solutions appropriate for individuals.	Wisdom-based problem representation applied to multi-factor and soft model choices.	Smart mobility platform apps for people. Pollution mitigation, sustainability controls. Genomic data analytics for personal care.

addition, stochastic modeling is required in many problem settings, such as taxi routing in congested city streets, and the control of complex systems when volatility and measurement errors are present. Third, an increasingly important problem characteristic is business solutions that must produce a computational result in real-time or near real-time in order to support a business process. Intraday revenue yield pricing for perishable goods (e.g., airline seats, hotel rooms and rental cars) is an exemplar. Fourth, other problems require consideration of "rugged landscapes," which make it hard for ML solutions that do hill-climbing optimization to succeed. Finally, a fifth class requires machine intelligence to find solutions that are tailored to individuals, such as healthcare genomics-based treatment decisions. Ideally, what is required is a more wisdom-based problem representation, so it is possible to find good solutions in the presence of hard and soft constraints, and objective and subjective goals.

In the routing optimization domain of NP-hard problems, the well-known *traveling salesman problem* (TSP) stands out as one to which heuristic methods have been successful, for example, to address the 1.9 million-city World TSP Tour (http://www.math.uwaterloo.ca/tsp/world/index.html). More value-laden quantum computing use cases with strong industrial relevance are related to the airline industry. Othmani, Ettl, and Guonaris (2020) have identified areas in the progression of quantum computing since the 1900s development of quantum science, then quantum-ready proofs-of-concept, and commercial advantage for solving real-world problem in the industry. The applications they point to include: contextually enhanced service personalization; untangling operational disruptions (irregular ops management) in air travel, that requires dynamic updating and optimization of crew, take-off slots, and equipment, all while trying to address customer satisfaction and cost over-run concerns. Quantum computing offers promise for solving the vexing matter of overcoming the fragmentation of the overall problem. According to the authors, quantum computing is able to address data and variable range-uncertainty, while re-aggregating the fragmented optimization sub-problems that will produce a more efficient and competitive advantage.

In the past 10 years, applied research and corporate practices with data analytics and problems that can be studied from the large-scale availability of the digital traces of people as consumers, social media users, users of mobile phones, and the tracking of people who exercise has given rise to new directions in predictive modeling and "living analytics" (as with our work at the Living Analytics Research Centre (LARC, https://larc.smu.edu.sg/) of Singapore Management University). It stands to reason that new directions in data analytics are now able to support innovative work in other areas that are open to innovations with quantum computing methods. Many business and social problems are naturally modeled in ways that play to the strengths of the quantum paradigm. Examples are the modeling and recognition of complex voice, facial expressions and emotions, and other image recognition problems. Even fast computation by traditional computers has not been sufficient due to limitations in how such problems have been approached in computational terms.

Another direction lies in creating the computational basis for doing things such as terrorist network member identification and driverless, autonomous vehicle routing

(Burkacky, Pautasso & Mohr, 2020). These involve the conceptualization of related problems with *rugged landscape optimization* (i.e., discontinuous modeling surfaces), so that the typical methods of *hill climbing* in optimization and ML are reduced in their power, especially with dynamic changes in their content and environment. We note the innovations based on the "changing landscape" analogy for problem identification and quantum computational methods. Finally, there are many such problems that require a smart approach, for "wicked" operational, social, and healthcare problems, like smart mobility platform design for changing urban traffic and transportation opportunities (Akrout, 2020), and policy analytics for pollution and sustainability, as well as genomic data analytics.

1.3.4 The Unique Role of Quantum Computing in Financial Services Applications

Among the various settings noted above, it's beneficial to consider problem areas that can potentially be treated in the financial services space by quantum computing approaches. They include offering enhanced power, flexibility, and representational authenticity to achieve smart, nearly real-time solutions for complex optimization, stochastic modeling, and intelligent choice problems. Financial markets are essentially complex systems that exhibit a high degree of *randomness*, for example, in the movement of equities' and derivatives' market prices, as well as foreign exchange rate pairs and interest rates. Their *volatilities* have been observed, and numerous applications and problem contexts have been analyzed in university and industry research. They have used portfolio construction models and optimization, neural networks (NNs), and Monte Carlo simulation, and ML and AI approaches. These further leverage configural patterns that are hard for human analysts to identify in big data, due to the presence of stochastic variation and the distributional mechanics of asset returns. In all these cases, the large size of the data sets used to calibrate changing risk levels in day- to-day operations typically require a huge amount of computational power for understanding the necessary relationships in the data so meaningful forecasts can be made (Lee, 2020).

Quantum computing is attractive for financial applications, like many-asset, multi-market portfolio construction and risk management controls. It also is relevant for establishing intraday trading paths to find ways to successfully exploit cross-asset or cross-market arbitrage opportunities. Other domains include near real-time, irrevocable settlement of securities trades, and the voluminous and volatile payments in continuous matching systems that maintain adequate settlement system participants' liquidity. These exhibit knowledge about the degree of the cross-state independence and correlations of asset price realizations from distributions that are the basis for the evolving lattice of assets prices in a market, which could utilize the property of *quantum entanglement*, such that it is present.

Finance practitioners know that a measurement of the state of such systems at any time is inherently random, though there is underlying cross-correlations, which result in the cointegrated evolution of asset prices and value-at-risk, a popular loss estimate at some level of statistical certainty over time. These states can be described in mathematical finance as *wave functions*, which quantum computing hardware has the potential to speed up the simulations in comparison to more speed- limited classical computer hardware and algorithms. Another consideration going forward is that the extent to which computational speed and power are likely to become available with lower electricity costs, and beneficial implications for more sustainable electricity consumption by financial institutions.

An application to *securities trading settlement transactions,* involving the exchange of a delivered financial instrument for an irrevocable cash payment, has been reported by Barclays Bank and IBM Research (Braine, Egger, Glick, & Woerner, 2019). The authors utilize quantum algorithms to make a mixed-binary math programming model's optimization faster. This is done for value maximization of continuously submitted batches of securities trades, with settlement processing that considers counter-party credit, collateral facilities, and regulatory compliance rules as objective function constraints. Traditional computation uses problem representations that support simulated annealing, which mimics a physical process of heating some material and lowering its temperature gradually. This reduces material defects, and the overall energy of the system. The analogy for finance is related to how computational algorithms approximate globally optimal solutions in a complex and constrained system, based on the weighted sum of settled transactions. The basic math programming formulation is transformed into an unconstrained model with a lambda-penalty function (like Kuhn-Tucker quadratic optimization) to support a solution with one qubit for each of the securities transactions to be settled, and a wave function for each possible settlement transaction that together form a batch for actual settlement. Quantum computation leverages unique features of the stochastic optimization, while reducing the liquidity risk for a participant that cannot settle its net funds position, and credit risk for system participants, should there be a fall in the value of traded securities.[2]

Another finance problem that deserves comment was presented by Woerner & Egger (2019). It is on *quantum risk analytics*, in which the quantum *amplitude estimation* (AE) algorithm is estimated for an unknown parameter. The problem the authors applied their approach to is the estimation of the uncertain price evolution for a U.S. Treasury bill, with a daily trading value of about US$500 bn and aggregate government debt as of 2016 of approximately US$14.8 tr. T-bill prices are subject to yield curve risk over time (i.e., as a fixed income security, from changing interest rates across the bills' maturities). For their analysis, the authors used constant maturity treasury (CMT) rates, to calculate the daily risk of a one-bill portfolio toy problem.[3] They did this for one-day changes in CMT based on the distributions of three correlated, underlying explanatory principal components called *shift, twist,* and *butterfly*. They are known to account for 96% of CMT's daily variations. Among them, only the first two were retained due to the low correlation (quantum entanglement) of the third component with them (i.e., due to its independence and thus lack of suitability for quantum analysis).

Three qubits were used to represent shift uncertainty, and two were used to represent twist uncertainty. The authors' approach was intended to deliver a quadratic speed-boost in comparison to traditional Monte Carlo simulation for T-bill diffusion in two-asset portfolios, which adds to forecasting problem complexity. They admitted that their sample simulation was not able to be to scaled up to more qubits for estimation, since they lacked the requisite quantum computing hardware to demonstrate the power-gain beyond massively-parallel, traditional computer hardware.[4]

We next turn to a discussion of a new framework for the implementation of quantum computing solutions, with special attention given to more technical aspects than have been discussed so far.

1.4 APPLICATION FRAMEWORK

In this section, we provide a framework describing the areas to consider when applying quantum computing to a business need. This is based on the assumption that advantage may be gained from using quantum computing, and a deeper look into what that entails is required. Taking into account what has been discussed in Section 1.2 on the basics of quantum computing and in Section 1.3 on current business needs, we now describe a framework for applying quantum computing to particular business problems. We will cover the four areas of concern presented in Figure 1.1 (data, processing, technical, and governance issues) by looking at the data preparation, algorithm, quantum circuit design, and some issues around integration. (See Figure 1.5.) These implementation areas are likely to be iterative as development continues towards the final solution and the state-of-the-art continues to evolve in all areas of quantum computing.

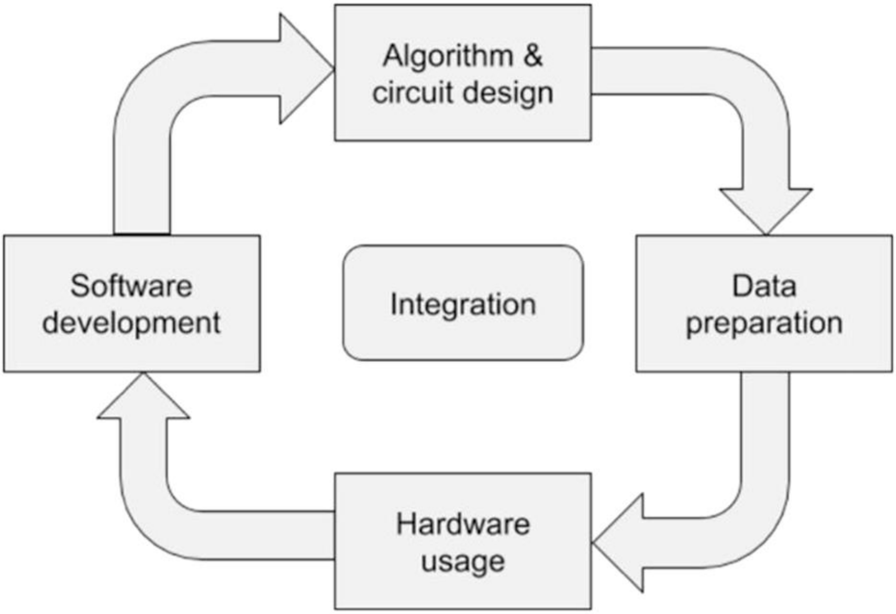


FIGURE 1.5 The iterative development cycle.

1.4.1 ALGORITHM DESIGN

Once the business needs have been defined, we must consider designing a suitable algorithm for the problem(s) involved in the business need. This can be one of the hardest steps as it involves both a knowledge of the business and a grasp of quantum algorithms. However, there are a number of algorithms already available that have been well established and used on a range of problems. Some of the algorithms can be found in online tutorials with sample code, and there are also consultancies available to advise potential users further. If there is nothing available that is suitable, then a new algorithm can be designed. It is most likely that even well-described algorithms will require modification for the business need. There are also companies working on embedding quantum algorithms into common data analytics packages, such as MATLAB, to utilize the power of quantum computing while hiding the complexity of quantum algorithms (http://horizonquantum.com/).

We next list a number of common problems that could exploit quantum properties and provide some algorithms examples that are suitable to explore. (See Table 1.3.)

TABLE 1.3 Problem Type, Quantum Properties, and Sample Algorithms

Problem Type	Quantum Property	Sources of Sample Algos
Black-box and oracle problems, discrete logarithm problems, integer factorization, boson sampling problems.	Interference between qubits provides an efficient quantum Fourier transform operation.	Deutsch and Jozsa (1992). Bernstein and Vazirani (1997). Quantum phase estimation (Simon, 1997). Shor's algorithm (Shor, 1997).
Searching an unstructured data set, BQP-complete problems, decision making, modelling and simulation, linear systems.	Amplitude amplification using conversion between probability amplitude and phase.	Grover (1996).
Element distinctness, triangle- finding, rugged landscape.	Quantum walks display exponential speed-up.	Ambainis (2007).
Graph theory problems, minimize the energy expectation to find the ground state energy.	Hybrid quantum/classical algorithms.	Quantum approximation optimization algo (QAOA) (Farhi, Goldstone, & Gutmann, 2014). Variational quantum eigensolver VQE) (Peruzzo et al., 2014).
ML and optimzation including: least squares fitting, semidefinite programming (SDP), NNs, and combinatorial optimization.	Uses quantum superposition.	QAOA (Farhi et al., 2014). VQE (Peruzzo et al., 2014).

The entire scope of the software that needs to be developed can be broken into various components: preparing and loading the data with classical code; creating a quantum circuit using qubits and quantum gates; executing the quantum circuit with classical code using quantum libraries; and processing the output of the quantum computer with classical code.

1.4.2 SOFTWARE DEVELOPMENT

With an algorithm decided, we now must develop the classical application that runs the algorithm with the quantum circuit. The classical code may just run the quantum circuit once or it may run it many times, and then rebuild the quantum circuit depending on the output. An example is when a hybrid neural network (NN) uses classical gradient descent to optimize the quantum weights in a hidden layer, or via a variational design where the quantum gates are rebuilt after each run.

Data preparation. The data to be input to the algorithm will need to be prepared. This will depend on the type of data and the type of quantum circuit. Depending on the quantum circuit, the data will have to be loaded in different ways. Data can be loaded either as binary data, as qubit rotations, or as quantum states.

The first way, *binary data loading,* is similar to a classical bit, where each one is encoded directly onto the pure state of the qubit. A classical 0 is encoded as a $|0\rangle$ state and 1 as a $|1\rangle$ state. This is simple to understand and, assuming that the default state of a qubit is $|0\rangle$, then the circuit will have either a NOT gate immediately applied to the input qubit for a $|1\rangle$ state or nothing, so the qubit stays in a $|0\rangle$ state. However, this method has the lowest data density of information encoding and is not utilizing the potential range of probability amplitude and phase amplitude superposition of the qubit.

Encoding classical values directly onto qubit probability rotation angles is possible using rotation gates, either on the probability amplitude or the phase amplitude after scaling the values between 0 and π radians. Going beyond π actually decreases the amplitude, for example $0 = 2\pi$. Further note also that the probabilities have an angular dependence on the rotation angle. To apply a probability of P to the qubit will need an angle θ, where θ (radians) = $\arccos(2P - 1)$, using an Ry gate for amplitude probability and an Rx gate for phase probability. This method is simple to apply and is easy to use with simulators and real backends. However, it still does not utilize the possibility of applying values to all quantum states.

Classical values can also be applied onto the qubit quantum states. In IQX, this is implemented using the *statevector* object and the *initialize* function of the quantum circuit object. This can utilize the full availability of states and map complex data directly onto the qubits. However, there is a need to normalize the *statevector* data for real computers before initialization. Some suggestions for how to handle the different data types include: *binary* – use binary; *integers* – scale use rotation or state; *floats* – scale and use rotation or state; *complex* – use state; text – convert to integer, scale and use rotation or state; *image* – convert to integer, scale and use rotation or state; and *objects* – serialize and convert to integer, and also scale and use rotation or state.

Creating the quantum circuit. The code to create a quantum circuit defines the registers, adds single and multiple qubits gates and measurement instructions. Registers are of two types: quantum and classical. Quantum registers contain qubits and classical registers contain classical bits. Classical registers will be used to collect the results of the quantum circuit execution and may be used to interact with the quantum circuit during the execution on simulators, such as the IF statement in Qiskit (Foy, 2019).

Qubits are initialized in the ground $|0\rangle$ state. The first operation is the a rotation gate to encode a classical value or a Hadamard gate to put the qubit into a 50-50 probability superposition. Single- and multi-qubit gates are then used to operate on qubits affecting the whole state of the quantum register. So, measurements are performed on the qubits in the quantum registers. The measured qubits may be the same as the input qubits or they may be ancilla used to collect the final circuit output.

Executing the quantum circuit. Two architectures are now considered: (1) a static quantum circuit that is executed once and has all the logic in it, and (2) a variational approach where the quantum circuit is modified after each run most likely to converge upon a result.

- *Static quantum circuit*: The quantum circuit is designed and executed on a quantum computer. Data are prepared in the classical code and initialized onto the input qubits of the quantum circuit. The code then defines a quantum backend which executes the quantum circuit, and then waits until a result is returned. (See Figure 1.6.) Before attempting to run the circuit, the circuit depth should be checked to ensure that it does not exceed the *coherence time* of the quantum computer, the amount of time a quantum state can continue in

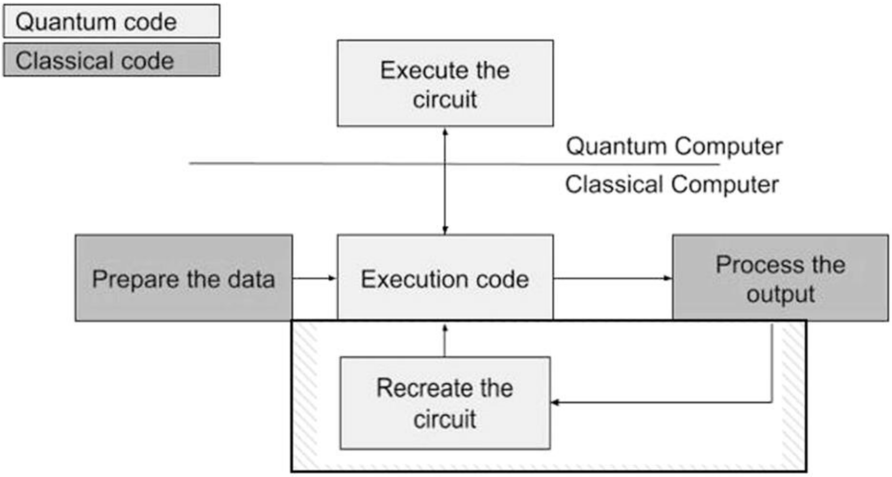


FIGURE 1.6 The combination of classical and quantum code to execute a quantum circuit. The shaded area is the extra step for variational quantum circuits.

the same stable form. (See the glossary in Appendix Table A1 for additional details.) It is likely that the high-level quantum circuit gate operations will need to be decomposed into the native gates of the quantum computer to understand the actual circuit depth that will be executed on the backend. Also, it is necessary to check the scaling of the algorithm's quantum depth with the quantum width, since algorithms can scale significantly.

- *Variational quantum circuits:* The entire system couples the quantum and classical code. The classical code recreates the quantum circuit after each time it is executed, depending on the results from the quantum circuit. (See Figure 1.6.)

The execution of a quantum circuit takes milliseconds, but the whole time for the call to the quantum computer can take time to come back, especially if the circuit jobs are queued and there is a significant queue depth. In this case, the calling code needs to have a wait or callback function to continue the processing after the call is complete.

Processing the output of the quantum computer. When a quantum computer provides an output, it needs to be retrieved by a classical computer and processed to be used by downstream systems. Measurement destroys the quantum state on the qubit and is usually performed at the end of a quantum circuit's execution. Measurements are only performed for the probability amplitude on one axis (the basis) of the Bloch sphere, which on IBM machines is the z-axis. The results are acquired from executing the quantum circuit a number of times, called shots, which default to 1,024. Each time a quantum circuit is executed it only returns a bit (0/1), depending on the probability amplitude for each state in the quantum register. For example, in Table 1.4 there are two qubits that have four states in total. The counts are how many times the circuit gave a 1 in that state. These counts can be calculated to obtain the probability of each state occurring in the output. (See Table 1.4.)

To find the probability of each qubit being in the |1⟩ state sum the probabilities for that qubit across all the states. In the above example, with the probability of q_0 being |1⟩, and 0.33 + 0.36 = 0.69, whereas q_1 is 0.31. The probability outputs can then be analyzed by classical code.

TABLE 1.4 The Count and Probability Example for a 2-Qubit Circuit

State	Count of 1	Probability
00	160	16%
01	336	33%
10	158	15%
11	370	36%

1.4.3 HARDWARE FOR QUANTUM COMPUTING

The quantum hardware used is also important to the success of the solution. As well as the quantity of the input data that needs to be processed, the fidelity, qubit coupling and coherence time of the qubits are also important. Noise in the qubits and gates will also need to be considered however, noise may be a useful for some business cases such as decision making.

Circuit width and number of physical qubits. Inspecting an algorithm will give a good idea about the quantity of logical qubits required. From the number of logical qubits plus the number of qubits needed for correcting errors, the total number of physical qubits can be determined. For example, if the problem approach is to do optimization, and there are 10 features to optimize on and another 3 qubits are needed for error correction, then at least 13 physical qubits are needed.

Circuit depth and coupling. The algorithm that has been designed will give an idea of the circuit depth. But the qubit coupling of the quantum device will also affect the circuit depth by potentially adding more gates to move information around. For example, on the IQX, with the *ibmq_vigo* processor device, there is no direct coupling between qubit 0 and 4. So, if qubit 0 is entangled with qubit 4, there has to be more gates (SWAP gates) used to move the quantum information between the qubits. (See Figure 1.7.) Each gate adds to the circuit depth and time spent by the circuit that could be used elsewhere. Matching the circuit design to the coupling can increase performance. As the circuit's mismatch to the coupling increases, the number of gates increases and the coherence times can run over what is desired.

Device adjustments. Finally, on some backends such as IQX, it is possible to adjust the qubit interactions at the lowest level of the physical interactions. In IQX, for example, it is possible to adjust the microwave pulse that controls the qubit rotation. This can also help with reducing errors. Finally, it also is worth noting that

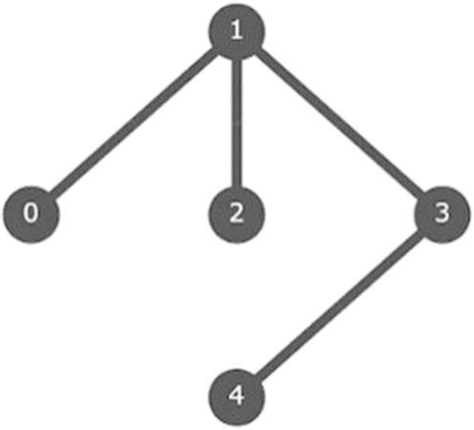


FIGURE 1.7 Coupling map for ibmq_vigo v1.2.1.

qubits can have more than two states. While the fidelity of even the third state is not sufficient for computation, this property would increase the computing power further than now with m^n states, where m is the number of states in a qubit and n is the number of qubits, if a quantum register were possible.

1.4.4 INTEGRATION WITH OTHER IT SYSTEMS IN THE FIRM

Currently, as quantum computing is still being used for research purposes, there is no integration with companies' internal IT systems. However, when a significant quantum advantage has been shown for business processes, then the quantum computer must be integrated into business flows so its value can be realized. It will then be important to consider data privacy and proprietary data processing issues. In light of this, current cloud-based quantum computers need to be assessed on how they transmit, store and delete data and quantum circuits that are used. Simulations can likely be performed locally for developing and testing quantum circuits, but to gain quantum advantage, real quantum computers will need to be used. Though it is possible to buy a real quantum computer now, the initial and maintenance costs are known to be quite high.

1.5 CASE: IMPLEMENTING A QUANTUM NEURAL NETWORK FOR CREDIT RISK

We next return to the quantum computing application framework described earlier, to illustrate a current QML project that the first two coauthors are currently working on.

1.5.1 CREDIT RISK ASSESSMENT

The business problem of quantitative credit risk assessment is not new. The first systematic data-driven approach to this problem dates back at least to 1968 when Altman (1968) published his pioneering *Z*-score formula for predicting bankruptcy. That work used linear discriminant analysis to predict the likelihood of bankruptcy from four observed financial ratios of a company. Since then, this formula has been refined many times with thousands of academic papers and hundreds of business applications introducing new features more sophisticated prediction models using larger company data sets. The vast majority of these modeling improvements are still focused on accounting data. This means that the model inputs are changing very slowly (e.g., once a year for a typical non-listed company). In addition, the input feature set is quite narrow – with no more than a dozen accounting data fields that are available for most corporates, and the input features are observed with a significant delay. These constraints limit the complexity of sensible models and the use of powerful modern ML approaches would result in heavily over-engineered solutions.

Our interest in credit scoring comes from the need for risk assessment in trade finance. Trade financing often exposes lenders to the risks of SMEs, as many of them operate in emerging markets with little timely and reliable information available. This means that classical accounting-based scoring approaches have limited utility in many cases. However, rapid digitization of trade finance means that a wealth of new information is available to lenders. The credit decisions can be made based on the information available at the level of individual transactions. Many trade finance users are engaged in high-intensity flows of relatively homogenous transactions. Many lenders have access to transaction data of dozens or even hundreds of thousands of borrowers. Moreover, these observed transactions are often inter-linked, either geographically or along industry verticals. Transaction data credit scoring makes the credit assessment problem with too few features available into a problem with a thousand or even millions of feature observations per day. Tradeteq uses sophisticated graph ML algorithms running on cloud GPU farms to calibrate and optimize these models.

The ongoing introduction of blockchains and IoT into the supply chains means that potentially relevant data flows are about to intensify by another three or four orders of magnitude, with real-time shipment location and conditions tracking available for more and more goods in transit. Thus, transaction credit scoring will soon become more taxing for existing computing systems.

Of course, classical computing and ML, in particular, are developing quite quickly, with rapid progress at all levels of the technological stack – from hardware to algorithms. However, to be ready for the increasing intensity of the data flows, we need to explore all available options for future ML systems, including the new QML systems.

Credit risk assessment problems are similar to certain problems that were already explored on quantum systems, for example, classification and regression ML and optimization problems. However, one still needs to ascertain to which extent peculiarities of credit risk problems are well suited for quantum or hybrid quantum and classical ML architectures. Credit risk ML problems often exhibit strongly imbalanced data sets with asymmetric noise, where one class is much less frequent and noisier than others. Supply chain graph topology matters a lot for the risk and needs to be reflected in a transaction risk model. It remains to be seen how well quantum systems will be able to handle these problem features though. The current state of quantum systems and QML algorithms means that they are quite far behind the capabilities of modern classical ML architectures. Even for relatively simple company credit scoring, our current models use over 300 features per company. On current quantum systems with limited number of qubits, it is not possible to process this number of features. and reaching these levels will take years. Theoretically, quantum systems may offer large advantages as they can process a very large number of possibilities simultaneously. In the future it is expected that quantum systems will reach a scale to be useful for these problems. To be ready for that, we need to build expertise and community knowledge about QML techniques and their capabilities.

We hope that quantum computing may become a good fit for these problems because, as explained earlier, the problems are dealing with high and growing volumes of complex data. So, model recalibration needs to be done quickly. Quantum algorithms have already been developed and tested for a number of similar problems, including classification ML, optimization, and graph ML. Our data and models are already residing in computing clouds, and we have already solved the accompanying security and access problems. Thus, connectivity to quantum systems is not going to add massive architectural complexity. Overall, *quantum neural networks* (QNNs) alongside classical processing seem to be a good match for the business needs.

1.5.2 Algorithm Design for a Quantum Neural Network (QNN)

The literature for QNNs contains two basic designs, one for a simple 2-qubit perceptron similar to what is offered by Entropica Labs (2020) and another involving a hybrid quantum/classical model that uses PyTorch and Qiskit (Jupyter Book Community, 2020).[5]

2-qubit perceptron. Here, q_0 and q_1 are the input qubits and q_2 is the output qubit from a Toffoli gate. R_{y1} and R_{y2} are the input values as rotations, and R_{y3} and R_{y4} are the neural network weights to be trained. (See Figure 1.8.) The circuit is recreated with new input and weight rotations and re-run for each data row. The output of the circuit is compared to the expected 0/1 output (binary in this case for “defaulted or not”). The best set of weights is the optimum accuracy from this routine. The main steps of the algorithm are: Data is prepared and loaded; a quantum circuit is created; the quantum circuit is executed; the output of the quantum circuit is processed.

The hybrid approach for this routine involved these steps. (See Figure 1.9.) Data is prepared and loaded; the data is input to the classical first layer; a quantum circuit is created (see Figure 1.10) based on the output of the first layer; the quantum circuit is executed; the output from the quantum circuit is used as input to the classical last layer.

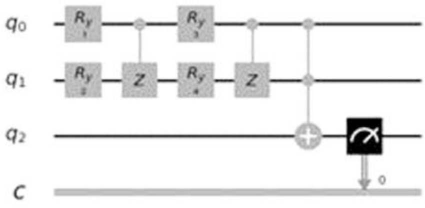


FIGURE 1.8 The 2-qubit perceptron quantum circuit.

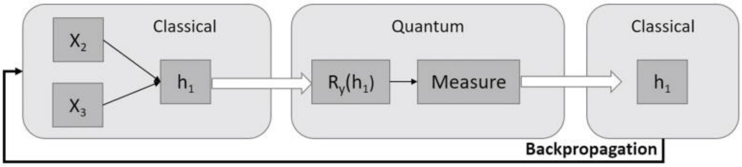


FIGURE 1.9 Hybrid algorithm using a quantum circuit as a hidden layer in a back-propagation NN.

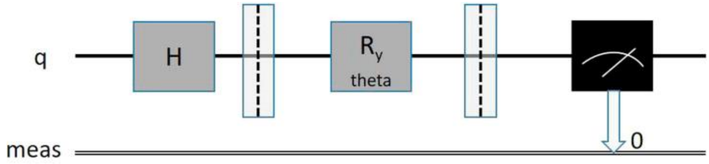


FIGURE 1.10 The quantum circuit used in the hidden layer.

Finally, the output of the last layer is fed back to the input layer adjusted for back-propagation.

Our approach was to first assess two features of credit data using these simple (1- or 2-qubit) designs and then to expand them to more features. There are other algorithms written, including variational approaches (e.g., Tacchino et al., 2019) in which the states hold weight information, as well as approaches that can also capture the training accuracy in the phase (e.g., Liao et al., 2019). As the implementation of these models is not easy, it is important to start small and increase complexity after getting meaningful results on smaller problems.

1.5.3 SOFTWARE DESIGN FOR QNN

With the algorithm decided, the application needed to be coded for data preparation and loading, executing the quantum circuit, and performing output processing. Qiskit was chosen as the we already had some experience and access to the IBM Q machines. First, the sample code was executed to ensure that all worked as expected, and then the code was modified to use the credit rating data.

Data preparation. Before the data is encoded onto the qubits, it is important to cap outliers. The cut-off points are determined empirically, and the step is performed classically. This step enables consistent scaling of feature values between 0 and π radians. Additionally, this step ensures that when the model is deployed for prediction, new data points will not take on values out of the usual range.

The data are encoded as amplitude rotations on the input qubits. These are coded into the circuit that is re-built at each data row. Rotation was used so far for the perceptron and hybrid models as it is straightforward and maps relatively easily to data features. State was not used due to the normalization issue with the scaling needing to be done across the data set not at each data row. With other models, such as Liao et al. (2019) with the whole data set applied in one shot, then this would be the best approach allowing for enormous data inputs and processing. The data set is going to be huge in the end state, but for now a smaller set of 2,000 rows of data with 40 features can be prepared.

- *For the 2-qubit perceptron* (see Figure 1.8), classical data preparation is used at first. The values for each row of data are put onto a z-rotation of the input qubits, and two more rotations are made after an entanglement for the neuron weights. A range of weight values are tested for each row of data, the result is stored

outside the quantum circuit, and the best set of weights chosen as the final best solution.
- *For the hybrid approach*, there is 1 qubit in a hidden layer between two classical layers (Jupyter Book Community, 2020). The output from the first classical layer is applied as a rotation to the qubit in the hidden layer. The output of the hidden layer then is used as the input to the second classical layer.

The quantum circuit's design can be further modified:

- *For the simple perceptron*, different single and multiple gates can be selected, and then the ROC (receiver operating characteristic) and AUC (area under the curve) metrics can be tested for each design.
- *For the hybrid approach*, the circuit starts in a simple way with 1 qubit and is then expanded with multiple qubits and entanglement operations.

Executing the quantum circuit. We next consider how the quantum circuit will be run.

- *For the simple perceptron*, the same quantum circuit is executed for each combination of weight values for each data row and the accuracy of the output is stored. The most accurate set of weights is then chosen as the best NN model and the test data set is measured for the ROC and AUC metrics.
- *For the hybrid approach*, the quantum circuit is embedded with the hidden layer of a classical NN model. The quantum-related code is coded into a QuantumClass with Qiskit and a QuantumClassicalClass with PyTorch.

Processing the output. The output of the both models is a single probability of classification, and a threshold is applied to give a final binary result for the prediction of whether a company will default or not. For the simple perceptron model the output qubit probability of 1 is used. For the hybrid model the output of the output classical layer provides the classification output. In both instances, the outputs are analyzed and visualized using standard statistics packages, such as Sklearn (sclkit-learn.org).

1.5.4 HARDWARE FOR QUANTUM CREDIT SCORING

For the credit-scoring use case, we expect to need a large number of qubits to be available to enable good use of a range of features. IQX was chosen given the availability of machines with a relatively large number of qubits for the quantum circuits, effective software development with Qiskit, and good support. One issue with these models is if there is the use of queuing on the backends compared to having a dedicated time slot. Having a queue means that executing the quantum circuit multiple times can significantly increase the overall time if there is a long queue. This is not such a big problem for a proof-of-concept, but would not be feasible for production.

Coupling, meanwhile, is not a problem for small circuits. But, as the size of the NN grows in terms of features, weights and entanglement, the circuit will need to align more with the coupling of the device and the design will need to be optimized for the topology.

Further, coherence time has not been an issue so far with small circuits but moving to more sophisticated designs using variational circuits and storing accuracy in a state will require more qubits and much deeper circuits. The whole quantum circuit must then execute and optimize within the coherence time. Qubit and gate noise has also not been such a big issue with the smaller circuits, but again, with more sophisticated circuits errors are expected and will need to be dealt with. A potential trade-off could be made between circuit complexity and quantum speed- up, as the hybrid model can alternate between classical and quantum layers to avoid noise problems due to circuit depth, at the expense of pure-quantum speed-ups.

1.5.5 Issues for Moving from a Stand-Alone to an Integrated System

Our work on credit scoring is currently a stand-alone project, and it is not yet integrated into a production process. Thus, there is no integration downstream as yet, but a data pipeline was developed for upstream use, before the data are sent to the models. This involved cleaning, scaling, and linearizing the data using standard packages. The output of the models has been compared to a benchmark using classical ML, and ranking coefficients have been used to compare how the classical and quantum solutions differed. The data on this project are publicly available, with some proprietary processing required so that using cloud computing for this research is fine. Also, with the hybrid model, the quantum circuit model is less likely to be required to be protected.

1.6 CONCLUSION

Quantum computing is positioned to be key for the future and for Society 5.0. There are many business needs that could be greatly helped by the use of quantum algorithms. However, the quantum advantage has not yet been demonstrated beyond any doubt and exploring quantum computing has a steep learning curve. In this chapter, we have given some background for the key features of quantum computing, mapped them to current business needs, and shown how to begin to apply them to actual business problems. In the future, we are looking forward to improved quantum hardware, better software development environments, a greater understanding of the physics of quantum phenomena, and greater networked thinking to better utilize quantum computing. Quantum hardware is consistently improving for qubit fidelity and coherence times. In addition to the rapid improvements in the qubits of quantum computers in the coming years (Hackett, 2020), software development environments also will improve in usability and visualization as well as integration with current data analytics packages, such as the recent IBM Circuit composer upgrade and Horizon Quantum's integration with MATLAB®(http://horizonquantum.com/). Though advances are being made in understanding quantum phenomena, it is not an easy path and there still is no guarantee of success. But as Johann Wolfgang von Goethe is credited with saying:

> *Whatever you can do, or dream you can, begin it. Boldness has genius, power and magic in it.*

ACKNOWLEDGMENTS

We benefited from the book editors' guidance and suggestions related to our revision of this chapter. Paul Griffin thanks Singapore Management University, the Monetary Authority of Singapore, Tradeteq, and OneConnect for their generous support. Michael Boguslavsky also acknowledges the Monetary Authority of Singapore. Junye Huang thanks IBM Corporation for its support. Robert J. Kauffman acknowledges Danske Bank, Copenhagen Airport A/S, Think Tank DEA, and the Endowed Chair in Digitalization at Copenhagen Business School for financial assistance. Finally, Brian R. Tan would like to thank Kreative Kommunication, Vietnam, a marketing agency, for its support. All errors and omissions are the sole responsibility of the authors.

NOTES

1 For a glossary with terms and definitions, see Appendix Table A1 at the end of this chapter.

2 In fact, Barclays has been testing the operation of quantum computing with quantum computing hardware that only uses 16 qubits at a maximum (Crosman, 2019). In other words, inadequate representational and computational power for a fully scaled version of its securities trade settlement process, which was reported as having been run on a 7-qubit quantum processor and a depth of 3 gates, with 6 participants and 7 security trades to be settled.

3 This used the T-bill CMT historical distribution of 1-day prices for maturity at time *t* as a random variable.

4 For more background, see Egger et al. (2020) and Orús, Enrique and Lizaso (2019) for finance domain quantum computing application areas, with consideration of future research, hardware, and applications.

5 See Wittek (2014) for QML fundamentals and Biamonte et al. (2017) for a broader review of QML methods.

REFERENCES

Akrout, M. (March 2020). *Will quantum computing be the next generation of automotive technology?* Retrieved from https://www.prescouter.com/2020/03/will-quantum-computing-be-the-next-generation-of-automotive-technology/

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance*, *23*(4), 189–209.

Ambainis, A. (2007). Quantum walk algorithm for element distinctness. *SIAM Journal of Computing*, *37*(1), 210–239.

Arora, S., & Barak, B. (2009). *Computational complexity: A modern approach*. Cambridge: Cambridge University Press.

Arute, F., Arya, K., Babush, R., Bacon, D., Bardin, J. C., Barends, R., … Martinis, J. M. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, *574*, 505–510.

Bernstein, E., Vazirani, U. (1997). Quantum complexity theory. *SIAM Journal of Computing*, *26*(5), 1411–1473.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, *549*(7671), 195–202.

Braine, L., Egger, D. J., Glick, J., & Woerner, S. (2019). *Quantum algorithms for mixed binary optimization applied to transaction settlement*. Retrieved from https://arxiv.org/abs/1910.05788

Burkacky, O., Pautasso, L., & Mohr, N. (2020, September 2). *Will quantum computing drive the automotive future?* Retrieved from https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/will-quantum-computing-drive-the-automotive-future#

Crosman, P. (2019, July 16). *Why banks like Barclays are testing quantum computing*. Retrieved from https://www.americanbanker.com/news/why-banks-like-barclays-are-testing-quantum-computing

Deutsch, D., & Jozsa, R. (1992). Rapid solutions of problems by quantum computation. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *439*(1907), 553–558.

Dolev, S. (2018, January 6). *The quantum apocalypse is imminent*. Retrieved from https://techcrunch.com/2018/01/05/the-quantum-computing-apocalypse-is-imminent/

Egger, D. J., Gambella, C., Marecek, J., McFaddin, S., Mevissen, M., Raymond, R., … Yndurain, Y. (2020). *Quantum computing for finance: State of the art and future prospects*. Retrieved from https://arxiv.org/abs/2006.14510

Entropica Labs. (2020). *A quantum perceptron*. Retrieved from https://qml.entropicalabs.io/

Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2000, January 28). *Quantum computation by adiabatic evolution*. Retrieved from https://arxiv.org/abs/quant-ph/ 0001106v1

Farhi, E., Goldstone, J., & Gutmann, S. (2014, November 14). *A quantum approximate optimization algorithm*. Retrieved from https://arxiv.org/abs/1411.4028

Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics, 21*, 467–488.

Foy, P. (2019, December 25). *Introduction to quantum programming with Qiskit*. Retrieved from https://www.mlq.ai/quantum-programming-with-qiskit/.

Furnham, A. (1986). Response bias, social desirability and dissimulation. *Personality and Individual Differences*, *7*(3), 385–400.

Gambetta, J. (2020, September 15). *IBM's roadmap for scaling quantum technology*. Retrieved from https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap.

Gidney, C., & Eker, M. (2019, December 5). *How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits*. Retrieved from https://arxiv.org/abs/1905.09749 Grover L. K. (1996). A fast quantum mechanical algorithm for database search. *Proceedings, 28th Annual ACM Symposium on the Theory of Computing*, 212–219.

Hackett, R. (2020, November 15). *IBM plans a huge leap in superfast quantum computing by 2023*. Retrieved from https://fortune.com/2020/09/15/ibm-quantum-computer-1-million-qubits-by-2030/

IBM.com (2020, August 20). *IBM delivers its highest quantum volume to date, expanding the computational power of its IBM cloud-accessible quantum computers*. Retrieved from https://newsroom.ibm.com/2020-08-20-IBM-Delivers-Its-Highest-Quantum-Volume-to-Date-Expanding-the-Computational-Power-of-its-IBM-Cloud-Accessible-Quantum-Computers.

Japan Gov News (2019, October 3). *Evolving innovation*. Retrieved from https://www.japan.go.jp/investment/evolving _innovation.html.

Jeste, D. V., Ardelt, M., Blazer, D., Kraemer, H. C., Vaillant, G., & Meeks, T. W. (2010). Expert consensus on characteristics of wisdom: A Delphi method study. *Gerontologist, 50*(5), 668–680.

Jupyter Book Community. (2020). *Hybrid quantum-classical neural networks with PyTorch and Qiskit*. Retrieved from https://qiskit.org/textbook/ch-machine-learning/machine-learning-qiskit-pytorch.html.

LaRose, R. (2019, March 22). *Overview and comparison of gate level quantum computing software platforms*. Retrieved from https://arxiv.org/abs/1807.02500

Lee, R. S. T. (2020). *Quantum finance: Intelligent forecast and trading systems*. Singapore: Springer.

Liao, Y., Ebler, D., Liu, F., & Dahlsten, O. (2019, November 20). *Quantum advantage in training binary neural networks*. Retrieved from https://arxiv.org/pdf/1810.12948.pdf

Loeffler, J. (2018, November 29). *No more transistors: The end of Moore's Law. InterestingEngineeering.com.* Retrieved from https://interestingengineering.com/no-more-transistors-the-end-of-moores-law.

Manin, Y. I. (1980). *Computable and Noncomputable (Vychisimloe y Nevchislimoe)*. Moscow: Sov. Radio.

McKay, D. C., Alexander, T., Bello, L., Biercuk, M. J., Bishop, L., Chen, J., … Gambetta, J. M. (2018, September 10). *Qiskit backend specifications for Open QASM and OpenPulse experiments*. Retrieved from https://arxiv.org/abs/1809.03452

Ménard, A., Ostolic, I., Patel, M., & Volz, D. (2020, February 6). *A game plan for quantum computing*. Retrieved from https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/a-game-plan-for-quantum-computing

Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information* (10th anniversary edition). Cambridge: Cambridge University Press.

Nijs, D. E. L. W. (2015). Introduction: Coping with growing complexity in society. *World Futures: The Journal of New Paradigm Research*, *71*(1-2), 1–7.

Orús, R., Enrique, M., & Lizaso, E. (2019). Quantum computing for finance: Overview and prospects. *Reviews in Physics, 4*, 100028.

Othmani, I., Ettl, M., & Guonaris, Y. (2020). *Exploring quantum computing use cases for airlines: A new technology is cleared for take-off*. Retrieved from https://www.ibm.com/thought-leadership/institute-business-value/report/quantum-airlines

Pednault, E. (2020, October 10). *On "quantum supremacy"*. Retrieved from https://www.ibm.com/blogs/research/2019/10/o"n-quantum-supremacy/

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M. Y., Zhou, X. Q., Love, P. J., … O'Brien, J. L. (2014). A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, *5*(4213). DOI: 10.1038/ncomms5213

Roffe, J. (2019, July 25). *Quantum error correction: An introductory guide*. Retrieved from https://arxiv.org/abs/1907.11157

Rønnow, T. F., Wang, Z., Job, J., Boixo, S., Isakov, S. V., Wecker, D., … Troyer, M., (2014). Defining and detecting quantum speedup. *Science*, *345*(6195), 420–424.

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

Simon, D. R. (1997). On the power of quantum computation. *SIAM Journal of Computing*, *26*(5), 1474–1483.

Singh, S. (2019, December 28). *Quantum computing: Solving problems beyond the power of classical computing*. Retrieved from https://economictimes.indiatimes.com/tech/software/quantum-computing-solving-problems-beyond-the-power-of-classical-computers/articleshow/73011174.cms

Tacchino, F., Macchiavello, C., Gerace, D., & Bajoni, D. (2019). An artificial neuron implemented on an actual quantum processor. *npj Quantum Information, 5*(26). DOI: 10.1038/s41534-019-0140-4

Wittek, P. (2014). *Quantum machine learning: What quantum computing means to data mining*. New York: Academic Press.

Woerner, S., & Egger, D. J. (2019). Quantum risk analysis. *npj Quantum Information*, *5*(15). DOI: 10.1038/s41534-019-0130-6

Zhao, R., Tanttu, T., Tan, K. Y., Hensen, B., Chan, K. W., Hwang, J. C. C., … Dzurak, A. S. (2019). Single-spin qubits in isotopically enriched silicon at low magnetic field. *Nature Communication*, *10*, 5500.

APPENDIX A: GLOSSARY OF TERMS

TABLE A1 Quantum Computing Terms and Definitions

Term	**Definition**
Adiabatic quantum eigenvector computing (AQC)	Emphasizes the resting state of a complex system minimization for Hamiltonian problems (Farhi et al., 2000).
Backend	Quantum hardware to run a quantum circuit and measure the final states of the qubits.
Coherence time	The amount of time a quantum state is stable form, such that it will be possible to conduct a solution procedure or optimization experiment before it is extinguished.
Fidelity	A metric for the similarity or closeness of two quantum states for a qubit.
Gate	A single or multi-qubit operation for the circuit model of quantum computing
Intelligence	Ability to obtain knowledge and skills and apply them to solve problems.
Noisy intermediate-scale quantum (NISQ)	An acronym for the current state of quantum computers that have 10-100 qubits (intermediate scale) without error correction (noisy).
NP Hard	Problems with a non polynomial time complexity
Qiskit	An open-source software developer kit (SDK) for writing and running gate-based quantum computing program on hardware such as IBM’s Q Experience backends.
Qubit	The analog of a 0/1 bit in standard computing, but for which a quantum computer has different realizations using quantum particles.
Quantum approx. optim. Algo. (QAOA)	An algorithm developed for quantum computing to estimate the solutions for hard combinatorial optimization problems using a gate-based approach.
Quantum error correction (QEC)	Approach used in quantum computing to correct data that has noise introduced from the quantum computer (Roffe, 2019).
Quantum machine learning (QML)	The analysis of data from complex settings for problems that can be simulated or optimized using the computation methods of a quantum computer.
Variational quantum eigensolver (VQE)	A quantum algorithm that enables the estimation of an eigenvalue for a Hamiltonian (Peruzzo et al., 2014).
Wicked problems	Problems with no stopping rule or a non-exhaustive set of potential solutions.
Wisdom	Exhibiting knowledge and experience to arrive at a good judgment.