SEMINARARBEIT

Mögliche Einsatzgebiete von Quantencomputer im Finanzwesen

ausgeführt am

Institut für
Finanz- und Versicherungsmathematik
TU Wien

unter der Anleitung von

Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Gerhold

durch

Christoph Holub
Matrikelnummer: 11729957

Wien, am 01.07.2021

Inhaltsverzeichnis

1 Einleitung

Durch den Einsatz von Computern haben sich viele Bereiche unseres Lebens grundlegend verändert. Viele schwierige und aufwendige Aufgaben können, durch die Rechenleistung von modernen Computern nun schnell und einfach gelöst werden. Eine höhere Rechenleistung spielt daher in vielen Branchen bereits einen essentiellen, wenn nicht sogar marktentscheidenden Faktor. Darunter auch im Finanz- und Versicherungswesen. Viele Tätigkeiten basieren mittlerweile darauf, einen Vorteil gegenüber anderen Marktteilnehmer zu bekommen, indem gewisse Aufgaben schneller und effizienter bewältigt werden können. Ein sehr deutliches Beispiel wäre hierfür der High-Frequency-Handel.

Quantencomputer, welche eine schnellere Rechengeschwindigkeit als herkömmliche Computer versprechen, könnten in diesem Hinblick einen erheblichen Einfluss auf die technischen Voraussetzungen haben, welche ein Finanzunternehmen haben muss, um konkurrenzfähig zu bleiben. Eine schnellere Rechenleistung ist allerdings nicht der einzige Grund warum Quantencomputer in der Zukunft eine immer wichtigere Rolle einnehmen werden. Durch immer kleiner werdende Computerteile werden quantenmechanische Effekte zu einem immer weniger vernachlässigbaren Problem.

Es ist damit nachvollziehbar, dass ein immer größer werdendes Interesse an der Konstruktion und Umsetzung eines effektiv funktionierenden Quantencomputer entsteht. So experimentieren zum Beispiel Google, oder IBM mit ersten Prototypen eines Quantencomputer und können in sehr zugeschnittenen Aufgabestellungen bereits Verbesserungen in der Rechengeschwindigkeit des Quantencomputers gegenüber eines klassischen Computers aufzeigen (Arute et al., 2019). Viele Hürden und Hindernisse sind jedoch noch zu bewältigen, bevor ein kommerzieller Einsatz von Quantencomputern vorstellbar ist.

Ziel dieser Seminararbeit ist es den Lesenden, einen Überblick darüber zu geben, welches Potenzial Quantencomputer für die Finanzindustrie haben können. Da es für das allgemeine Verständis der Thematik sinnvoll ist, wird auch ein stückweit auf die physikalischen Hintergründe des Quantencomputers eingegangen, obgleich der Fokus nicht darauf liegen soll.

Die Arbeit orientiert sich an der Arbeit ‘Prospects and challenges of quantum finance’ (Bouland, van Dam, Joorati, Kerenidis & Prakash, 2020) von Adam Bouland, Wim van Dam, Hamed Joorati, Iordanis Kerenidis, Anupam Prakash, welche am 12 November 2020 in der Fachzeitschrift ‘Nature’ veröffentlicht wurde.

2 Überblick Quantencomputer

2.1 Geschichtlicher Hintergrund

Als die klassische Mechanik, Anfang-mitte des 20. Jahrhunderts bei der Beschreibung der Vorgänge innerhalb eines Atoms versagte, mussten die physikalischen Grundsätze bei mikroskopischen Messgrößenordnungen überarbeitet werden. In Anlehnung an die klassische Mechanik spricht man von der Quantenmechanik. Viele Gesetzesmäßigkeiten die im alltäglichen Leben selbsverständlich scheinen, verlieren ihre intuitiven Eigenschaften je 'näher' man sie betrachtet. In dem Kontext, der immer kleiner werdenden Computerbestandteile, ist es daher unumgänglich sich mit quantenmechanischen Eigenschaften auseinanderzusetzen. Wie wir sehen werden ist es auch möglich diese neuen Erkenntnisse in Form eines Quantencomputer für eine bessere Rechenleistung zu verwenden. Besonders relevant wird für uns der Wellen-Teilchen-Dualismus, das Superpositionsprinzip, welches für die grundlegenden Unterscheidung von Bits zu Qubits herangezogen wird und schließlich die Quantenverschränkung.

Die erste Überlegung zu einem Quantencomputer entstanden Mitte der 1970er Jahre als es zunehmend ersichtlich wurde, dass der Umgang mit Quantenphänomenen auf klassischen Computer nicht vollständig möglich ist. Die ersten Konzepte wurden schließlich 1980 von dem amerikanischen Physiker Paul Benioff, als auch parallel dazu, von dem russisch-deutschen Mathematiker Juri Manin entwickelt. Bekannt wurde die Idee des Quantencomputers schließlich durch den Vortrag von Richard Feynman 1981, indem er die Vorteile des Quantencomputers für das Lösen von modernen Aufgaben in der Physik und Chemie beschrieb. (Feynman, 2018)

2.2 Quantenmechanische Grundlagen

2.2.1 Wellen-Teilchen-Dualismus

Ein guter Ausgangspunkt für den Wellen-Teilchen-Dualismus ist die Frage ob Licht sich als Wellen oder als Teilchen fortbewegt. So hat Newton damals mit seiner Korpuskulartheorie, das Bild von Licht als Teilchen oder Korpuskeln stark geprägt. Später durchgeführte Experimente deuten darauf hin, dass Licht durchaus auch Eigenschaften einer Welle besitzt. Ausschlaggebend dafür war das Doppelspaltexperiment von Jung, welches sich mit diversen Weiterführung zu einem der wichtigsten Demonstrationen des Wellen-Teilchen-Dualismus entwickelte. Der grobe Aufbau des Experiments ist wie folgt:

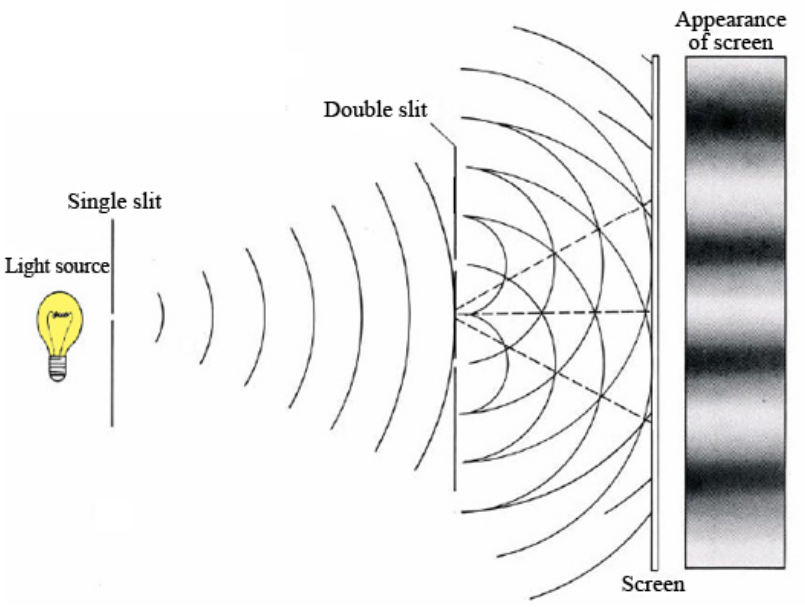

Abbildung 2.1: Youngs Doppelspaltexperiment
(*The World of Physics*, 2012)

Ein Quantenobjekt, in unserem Beispiel ein Photon, nimmt, falls es in den Spalten nicht gemessen wird, Welleneigenschaften an. Diese manifestieren sich anhand der Interferenz, welche nach dem Doppelspalt auf einem Bildschirm sichtbar gemacht wird. Falls bei dem den Spalten aber gemessen wird, durch welche der beiden Spalten das Photon passiert, verschwindet das Interferenzmuster und das Photon bekommt Teilcheneigenschaften. Dieses Phänomen wurde schließlich in der "Kopenhagener Deutung"1927 insofern beschrieben, dass ein Quantenobjekt sowohl Wellen- und Teilcheneigenschaften annehmen kann, sich aber bei Beobachtung für einen Zustand entscheiden muss. Für uns ist es wichtig hier anzumerken, dass bereits leichte Verunreinigungen oder Themperaturveränderungen ein Quantenobjekt zur Messung führt und damit der Wellenzustand verloren geht. Es ist daher technisch sehr schwierig bei Quantencomputer eine höhere Anzahl an Qubits gleichzeitig und konstant in einen Quantenzustand zu versetzen.

2.2.2 Superposition

Das Superpositionsprinzip besagt, dass sich physikalische Größen überlagern können, ohne sich dabei zu behindern. Quantenobjekte können, wie bei der Kopenhagener Deutung erwähnt, zu gewissen Zeitpunkten sowohl als Teilchen als auch als Welle gesehen werden. Das Quantenobjekt befindet sich damit in 'Superpostionen'. Solche Zustände lassen sich durch Wellenfunktionen beschreiben in unserem Fall durch die Schrödingergleichung. Durch die Linearität und Homogenität der Schrödingergleichung sind auch alle Linearkombinationen der Lösung eine Lösung. Eine Superpostion wird daher als Linearkombination der n verschieden möglichen Zustände $|\varphi_i\rangle$ *mit* $i = 0, ..., n$ des Quantenobjekts beschrie-

ben. Dabei verwendet man die Dirac-Notation, auch Bra-Ket-Notation genannt. Für ein Element v aus einem Vektorraum V verwendet man $|v\rangle$ um den Vektor v darzustellen. Man spricht von einem "Ket" $|v\rangle$. Zu jedem "Ket"$|\cdot\rangle$ existiert ein "Bra" $\langle\cdot|$ aus dem Dualraum von V. Ein "Bra" $\langle\cdot|$ angewendet auf ein "Ket" $|\cdot\rangle$ ergibt somit wieder eine konventionelle Darstellung des Skalarprodukts: $\langle\cdot|\cdot\rangle$. Ein Gesamtzustand $|\psi\rangle$ lässt sich somit durch eine Überlagerung der Einzelzustände $|\varphi_i\rangle$ darstellen:

$$|\psi\rangle = \sum_{i=1}^{n} c_i|\varphi_i\rangle$$

2.2.3 Quantenverschränkung

Sei ein physikalisches System aus mehreren Teilsystemen zusammengesetzt. Falls der Zustand eines oder mehrerer dieser Teilsysteme nicht unabhänging von dem gesamten System betrachtet werden kann, so spricht man von Verschränkung. In Bezug auf ein System von Quantenobjekten, spricht man von Quantenverschränkung, wenn der Zustand einzelner Elemente von den anderen, beziehungsweise von dem gesamten System abhängt. Ein einfaches Beispiel werden wir in 2.3 betrachten.

2.3 Theoretische Funktionsweise eines Quantencomputers

Ein klassischer Computer verwendet für seine Rechnungen "bits"(kurz für "binary digit"). Dabei handelt es sich, um die kleinste Information zur Verarbeitung von Computerdaten. Ein Bit ist ein System, welches zwei Zustände annehmen kann: 0 und 1. Diese zwei Möglichkeiten können physikalisch mit einem Stromkreis emuliert werden. In der Regel weißt man einem System die Zahl 1 zu , falls Strom fließt und 0 falls kein Strom fließt. Sehr vereinfacht dargestellt: Bekommt ein Computer einen Input, wird dieser ihm eine Kombination aus Bits zuweisen. Mithilfe verschiedener Schaltungen und Manipulationen des Stromkreise wird eine neue Kombination erzeugt, die schließlich als Output wiedergegeben wird. Das bietet die Möglichkeit zeitaufwendige Aufgaben schnell, als eine Anzahl vieler kleiner Aufgaben abzuarbeiten. Das Ein- und Auslesen sowie das Interpretieren der Bit-Kombinationen erfolgt über eine Software beziehungsweise über das Betriebssystems. In weiterer Folge werden wir sehen, dass das Übersetzen von klassischen Daten in eine Quantensoftware nicht immer einfach zu bewerkstelligen ist. Entscheidend für uns ist hierbei, dass Bits nur zwei Zustände annehmen können.

Quantencomputer verwenden statt Bits, sogenannte Qubits als kleinste verarbeitbare Information. Qubits können nun die Zustände 0,1 und deren Superposition annehmen da deren Position durch die Lösung der Schrödingergleichung gegeben ist. Dabei gelten die Zustände 0 und 1 als Basiszustände: $|0\rangle$ und $|1\rangle$. Durch die Linearität der Schrödingergleichung lässt sich der Zustand eines Qubits auch als Linearkombinationen von $|0\rangle$ und $|1\rangle$ darstellen, also:

$$\alpha|0\rangle \ + \beta|1\rangle, \ |\alpha|^2 + |\beta|^2 = 1$$

Dabei darf man sich den Zustand des Qubits nicht als Zahl zwischen 0 und 1 vorstellen. Bei jeder Messung muss sich schließlich für einen der Zustände entscheiden: α und β sind hierbei lediglich die Wahrscheinlichkeiten, welcher der beiden Zustande bei der Messung angenommen wird, weshalb diese auch normiert sind. Physikalisch kann dieser Zustand mit einem polarisierten Photon, oder ionisierten Atom umgesetzt werden. Dieser Quantenzustand ist sehr fragil und bei jeglicher Verunreinigung oder Störung verliert er seine Form und nimmt einen der beiden Basiszustände an. Dies ist auch der Grund warum Quantencomputer nur sehr schwierig auf allen ihren verfügbaren Qubits laufen können.

Ein Zustand $|\Psi\rangle$ eines 2-Qubits-System wird ebenfalls in Superposition der Basiszustände der Qubits dargestellt:

$$|\Psi\rangle = \alpha|0\rangle|0\rangle + \beta|0\rangle|1\rangle + \gamma|1\rangle|0\rangle + \delta|1\rangle|1\rangle$$

$$\text{mit} \quad |\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$$

Formal betrachtet, spannen die zwei orthonormalen Basiszustände $|0\rangle$ und $|1\rangle$ eines Qubits einen zwei-dimensionalen Hilbertraum auf. Somit lassen sich n Qubits in Superposition in einem 2^n dimensionalen Hilbertraum darstellen.

Das Superpositionsprinzip lässt vermuten, dass in einem N-Qubit System gleichzeitig 2^N Information gespeichert werden können. Das funktioniert nicht, da jegliche Messung, in diesem Fall das Abspeichern des Zustands eines Qubits, diesen zwingt eine Form anzunehmen. Das Holevo-Theorem zeigt, dass der übertragbare Informationsgehalt eines N-Qubit System sowie der eines klassischen N-Bit System genau N Bit beträgt. Ein anderer vorkommender Effekt bei Systemen, die auf mehreren Qubits arbeiten ist die Quantenverschränkung. Gegebenenfalls lassen sich nach diversen Schaltungen Zustände von Systemen vereinfachen und als Produkt darstellen (In diesem Fall mit $\gamma = \delta = 0$, ohne Normierung):

$$|\Psi\rangle = (\frac{1}{\sqrt{2}})(|0\rangle|0\rangle + |0\rangle|1\rangle) = (\frac{1}{\sqrt{2}})(|0\rangle(|0\rangle + |1\rangle))$$

Damit wüssten wir in diesem Fall den Zustand $|0\rangle$ des ersten Qubits und dass sich der zweite in der Superposition $(|0\rangle + |1\rangle)$ befindet. Die beiden Qubits bilden somit voneinander unabhängige Teilsysteme. Falls diese Produktdarstellung nicht möglich ist(In diesem Fall mit $\beta = \gamma = 0$, ohne Normierung):

$$|\Psi\rangle = (\frac{1}{\sqrt{2}})(|0\rangle|0\rangle + |1\rangle|1\rangle)$$

erkennt man die Verschränkung. Vor allem diese Quantenverschränkung ist eine vielver-

sprechende Eigenschaft welche ein klassischer Computer nicht verwenden kann.

2.4 Aussichten

Die genannten quantenmechanischen Grundlagen des Quantencomputers würden einen Quantenprozessor potentiell ermöglichen, gewisse Rechnungen wesentlich schneller zu lösen, als ein klassischer Prozessor. Auch wenn die Technologie um den Quantencomputer noch sehr unentwickelt ist, gelang es dem Google-Quantencomputer bereits 2019 das immense Potenzial eines Quantencomputers zu demonstrieren. Der auf 53-Qubits laufende Quantenprozessor Sycamore löste eine, ihm sehr zugeschnittene, statistische Berechnung in 200 Sekunden: Ein aktueller Supercomputer hätte für diese Aufgabe theoretisch 10000 Jahre (Arute et al., 2019) gebraucht. Andere Rechnungen darunter jene von IBM, schätzen die Rechenzeit eines klassischen Supercomputer für eine äquivalente Aufgabe allerdings auf 2,5 Tage (Edwin Pednault & Gambetta, 2019). Die neuen Rechengeschwindigkeiten würden für viele Aufgabenbereiche neu Möglichkeiten schaffen. Für uns werden in weiterer Folgen die Auswirkung von Quantentechnologien auf Monte-Carlo-Simulationen, Portfoliooptimierungen und Deep-Learning-Methoden von Interesse sein.

Auch wenn Quantencomputer ein immenses Potential haben, sind diese noch lange nicht in der Lage ihre Möglickeiten vollkommen auszuschöpfen. Zuallererst kann man sich Quantencomputer nicht als klassischen Computer vorstellen, der schlicht über eine bessere Rechenleistung verfügt. Die Algorithmen die ein Quantencomputer verarbeiten kann sind aufgrund seiner, auf der Quantenmechanik basierenden Natur, von Grund auf unterschiedlich zu jenen die auf klassischen Computer laufen. Das Ein- und Auslesen von Daten, welche für einen Quantencomputer verarbeitbar sind, beziehungsweise das Übersetzen solcher Daten auf einen klassischen Computer ist aktuell noch schwierig umzusetzen und erfordert oft noch zusätzlichen Rechenaufwand. Auch wenn solche Hindernisse in Zukunft eventuell besser gelöst werden können, erschweren sie aktuell dennoch die Arbeit mit Quantencomputern. Ein weiteres großes Problem ist der Erhalt der Superpostion. Wie bereits angedeutet, genügt eine kleine Verunreinung um den Quantenzustand zu verlieren und damit laufende Rechnungen zunichte zu machen. Ein damit verbundenes Problem ist die hohe noise-Rate der aktuellen Quantencomputer-Prototypen. Während der CPU-Rechenfehler bei modernen klassischen Computer mit der error storage rate des RAMs vernachlässigt werden kann, verliert sich die Rechenfähigkeit eines aktuellen Quantencomputers nach 10^2 bis 10^3 Rechenoperationen bereits im noise. Das bedeutet, dass die Resultate durch Rechenfehler des Quantencomputers von den wirklichen Daten abweichen würden und somit in keiner sinnvollen Relation mehr stehen würden. Auf lange Sicht wird es Korrektur-Algorithmen geben, welche die Fehlerrate, wie bei einem klassischen Computer, reduzieren werden. Dafür wird statt dem RAM voraussichtlich ein QRAM entwickelt, der wiederum verfügbare Qubits in Anspruch nehmen muss und damit auch etwas Geschwindigkeit verloren gehen könnte.

All dies lässt darauf schließen, dass wir erst am Anfang der Quantencomputer stehen und von gewerblich verwendeten, vollständig entwickelten Quantencomputer noch weit entfernt

sind. Es werden deshalb Technologien um den Quantencomputer in zwei Phasen gegliedert. Zum einen werden in den nächsten Jahren Quantencomputer-Prototypen noch weit davon entfernt sein eine verlässliche Fehlerkorrektur zu implementieren: man spricht von dem "Noisy Intermediate-Scale Quantum" (NISQ) Zeitraum. Zum anderen wird dann jene Zeit kommen in der die Fehlerkorrektur vollständig implementiert sein wird. Viele theoretischen Überlegungen beziehen sich bereits auf diese Zeit. Infolgedessen wird auch oft eine Art Mittelweg gewählt, in dem Quantencomputer und klassische Computer gemeinsam arbeiten, wobei der klassische Computer häufig eine fehlerüberprüfende Funktion einnimmt.

3 Monte Carlo Simulationen

3.1 Theoretische Grundlagen

3.1.1 Grundlagen von Monte-Carlo-Simulationen

Bei Monte-Carlo-Simulationen handelt es sich um ein stochastisches Verfahren, indem anhand einer vielfachen Realisierung eines Zufallsexperiments dessen Mittelwert numerisch approximiert werden soll. Ausschlaggebend ist hierfür das Gesetz der Großen Zahlen. Dieses Verfahren wird beispielsweise bei der Bewertung und Preisfindung von exotischen Optionen verwenden, wenn der tatsächliche Preis analytisch nicht oder schwer zu berechnen ist. Für eine Verteilung mit Mittel μ und Varianz σ^2, kann mit k Realisierungen der Verteilung ein approximierter Mittelwert $\tilde{\mu}$ mithilfe der Chebychev Ungleichung abgeschätzt werden:

$$\mathbf{P}(|\tilde{\mu} - \mu| \geqslant \varepsilon) \leqslant \frac{\sigma^2}{k\varepsilon^2}$$

Für einen Schätzfehler ε braucht es $k = O(\frac{\sigma^2}{\varepsilon^2})$ Proben, um μ adequat approximiert zu haben. Falls wir nun $\tilde{\mu}$ genauer darstellen wollen, oder den Fehler so klein wie möglich werden lassen wollen, wird eine großere Anzahl an Monte-Carlo-Simulationen notwendig. Für komplexe Verteilungen wie zum Beispiel mittelfristige bis langfristige Verlustverteilungen für verschiedene Portfolios, kann es deshalb zu einen großen Rechenaufwand, mit langen Rechenzeit führen.

3.1.2 Potenzial von Quantencomputer

Für Monte-Carlo-Simulation können Quantencomputer eine deutliche Verbesserung in der Berechnungsgeschwindigkeit herbeiführen. Sie könnten theoretisch die Abhängigkeit von k bezüglich σ und ε fast quadratisch verbessern. Damit würde man für einen Schätzfehler ε nur $k = \tilde{O}(\frac{\sigma}{\varepsilon})$ Simulation statt $O(\frac{\sigma^2}{\varepsilon^2})$ benötigen (Montanaro, 2015). Eine solche Beschleunigung könnte dementsprechend einen deutlichen Einfluss in das Bewerten von Finanzinstrument haben.

Eine vereinfachte Darstellung eines Quantenalgorithmuses für Monte-Carlo-Simulation hat einen folgenden Aufbau: Sei A ein Quantenalgorithmus welcher eine Zufallsvariable $v(A)$ erzeugt. Es wir nun versucht den Erwartungswert $\mathbf{E}(v(A)) =: \mu$ abzuschätzen. Die Varianz der erzeugten Zufallsvariablen sei σ^2. Für die einfachste Form der Approximierungen, nehmen wir an, dass A mit 0 und 1 beschränkt ist.

Des Weiteren gilt:
- A läuft auf n-Qubits, die mit $|0\rangle$ initialisiert sind.
- A bekommt $k < n$ Inputs.
- Für $x \in \{0,1\}^k$ erzeugt A nun den Output $\phi(x)$ mit $\phi : \{0,1\}^k \to [0,1]$ ($bzw.\mathbf{R}$).
- Weiters sei W ein auf $k+1$ Qubits laufender Operator sodass:

$$W : |x\rangle|0\rangle \mapsto |x\rangle(\sqrt{1-\phi(x)}|0\rangle + \sqrt{\phi(x)}|1\rangle)$$

Wird schließlich nachdem die k Inputs gegeben wurden, der Operator W auf den $k+1$ Qubits ausgeführt, so ist die Wahrscheinlichkeit als Output 1 zu erhalten $v(A)$. Für eine Schätzung dieser Wahrscheinlichkeit verwendet man einen sogenannten 'Quantum amplitute estimator'. Da eine genau Beschreibung des Quantum-amplitute-estimator für unsere Zwecke nicht notwendig ist, wird hier darauf verzichtet. Für eine genau Beschreibung siehe (Brassard, Høyer, Mosca & Tapp, 2002).

Nach t-facher Iteration des Quantum Amplitude Estimation erfüllt der geschätze Mittelwert $\tilde{\mu}$:

$$|\tilde{\mu} - \mu| = O(\tfrac{\sqrt{\mu}}{t} + \tfrac{1}{t^2}).$$

Somit genügen $O(\frac{1}{\varepsilon})$ Iterationen um $\tilde{\mu}$ mit dem Fehler ε i.e $|\tilde{\mu} - \mu| \leqslant \varepsilon$ abzuschätzen.
Falls wir nun vorraussetzen, dass $\mathbf{Var}(v(A)) \leqslant \sigma^2$ so bekommen wir, die bereits angedeutet fast quadratische Beschleunigung. Es benötigt dafür $\tilde{O}(\frac{\sigma}{\varepsilon})$. Die Idee hierbei ist den Output von A in kleinere disjunte Intervalle zu unterteilen und schließlich, ähnlich wie bei der bereits erläuterten Methode zu bearbeiten. Auf eine genauere Beschreibung soll hier nicht eingegangen werden.

All dies bezieht sich aber eher theoretisch auf die Zukunft in der fehlertolerante Quantencomputer bereits vorhanden sind, welche noch nicht in absehbarer Zukunft verfügbar sein werden. Aktuell existieren noch zwei relevante Probleme. Einerseits ist die hohe noise-Rate von Quantencomputer sehr hinderlich für eine sinngerechte Durchführung vieler verschiedener Algorithmen auf Quantencomputer. Ein anderes Problem tritt auf wenn eine größere Anzahl an Simulationen durgeführt werden muss. Vor allem bei Monte-Carlo-Simulationen ein häufig auftretendes Anliegen. Ein Quantencomputer muss alle Simulation am Stück, in Serie durchführen, da sofern er die Qubits in Superposition versetzt hat, diese nicht in Superposition abspeichern kann. Erst danach, kann er mit der Nachbearbeitung der gesammelten Daten beginnen. Das Postprozessing, wie auf einem klassischen Computer, kann somit also nicht am Ende einer Simulation stattfinden, sondern erst nach allen durch-

geführen Simulation. Kombiniert mit der hohen Fehleranfälligkeit von aktuellen Quantenprozessoren würden somit die Ergebnisse in den meisten Fällen unbrauchbar werden. Zum heutigen Standpunkt wären daher in einer solchen Form, auf Quantencomputern gerechnete, Monte-Carlo-Simulation in einem kommerziellen Umfeld nicht sehr realistisch. Um das Ergebnis solcher Simulation brauchbar zu machen, müssten die verwendeten Algorithmen sehr einfach gestaltet werden, beziehungsweise eine sehr geringe Anzahl an Simulationen durchgeführt werden.

3.2 Aussichten im NISQ Zeitraum

In weiterer Folge sei Q ein Quantenalgorithmus mit Tiefe D, Aufwand $O(\frac{1}{\varepsilon})$, welcher p-mal in Serie ausgeführt werden muss. Die sich ergebende Frage ist nun, wie sehr sich D reduzieren lässt, um mit dementsprechend wachsenden p weiterhin schneller zu arbeiten als parallel ausführbare klassische Monte-Carlo-Algorithmus mit Aufwand $O(\frac{1}{\varepsilon^2})$. Auf die Quanten-Fourier-Transformation, die am Ende jedes Quantenalgorithmus ausgeführt wird, um eventuelle Fehler zu überprüfen, soll hier nicht weiter eingegangen werden, da diese ohnehin den Aufwand, nur um eine Konstante verringert.

3.2.1 Erste Überlegung

Einen Ansatz die Relation zwischen Schaltungstiefe und notwendigen Ausführungen zu bestimmen geht auf (Burchard, 2019) zurück. Diese Diese besagt, dass für Quantenalgorithmen, welche eine Monte-Carlo-Simulation p-mal parallel und maximal D-mal in Serie aufruft, folgendes gelten muss:

$$pD^2 \geqslant \Omega(1/\varepsilon^2),$$

wobei ε die Länge des Konfidenzintervalls des Algorithmus ist. Im Vergleich: Bei klassischen Algorithmen gilt $pD \geqslant \Omega(1/\varepsilon^2)$. Um den Quantenalgorithmus Q auszuführen, benögt man allerdings ungefähr pD Operationen. Demnach würde mit geringer werdender Tiefe, die Quantenbeschleunigung allmählich verschwinden und sich dem klassischen Algorithmus mit $D = 1$ und $p = 1/\varepsilon^2$ annähern. Unter diesen Voraussetzungen ist daher bei p-facher Parallelisierung von Q nur eine Beschleunigung um den Faktor $O(\sqrt{p})$ möglich.

Auch wenn diese Ansätzer eher pessimistisch erscheint, wäre eine solche Beschleunigung ein bedeutsam genug, Quantencomputer in die Praxis einzuführen. Zwar ist es noch ein langer Weg das volle Potenzial von Quantencomputern für Monte-Carlo-Simulation auszuschöpfen, mit diesen Überlegungen wäre allerdings eine zeitnahe, erste Beschleunigung durchaus möglich. Anhand dieser Perspektive, entsteht bereits ein immer größer werdendes Interesse, Quantencomputer für Monte-Carlo-Simulationen zu implementieren.

Eine weitere Überlegung besteht darin, die aktuellen Schwäche eines Quantenalgorithmus mit der Hilfe von klassischen Algorithmen auszubessern. So wurden bereits Konzepte erarbeitet, Monte Carlo-Simulationen in p kleine, gleichgroße Subprobleme zu unterteilen, diese schnell mit einem Quantencomputer auszuführen und die erhaltenen Ergebnisse

mit einem klassischen Computer zu verarbeiten. Allerdings stößt auch dies in der praktischen Umsetzung auf Hindernisse. Zum einen wird p in den meisten Fällen sehr groß und die Wahrscheinlichkeit, dass p in dem gewollten Konfidenzintervall liegt, muss mit einem zusätzlichen Aufwand $O(log(p))$ berechnet werden. Zum Anderen wird die Unterteilung von Monte-Carlo-Simulation in viele kleinere Rechnungen nicht immer möglich sein.

3.2.2 Alternativer Lösungsansatz

Ein anderer, sehr vielversprechender Ansatz entstammt aus der Arbeit von Kerenidis and Prakash, welcher die Wechselwirkung klassischer und Quanten-Monte-Carlo-Simulationen besser umsetzt. Die Grundidee besteht darin Monte-Carlo-Simulationen mit unterschiedlicher Tiefe zu berechnen. Mit einer Maximum-Likelihood-Schätzung, welche auf einen klassischen Computer laufen soll, wird anschließend der Mittelwert bestimmt. Erste Konzepte arbeiten mit einer exponentieller steigender Tiefe bis zu einer maximalen Tiefe $O(\frac{1}{\varepsilon})$. Die Effizienz dieser Methode ist allerdings weiterhin umstritten (Aaronson & Rall, 2020). Jener Algorithmus von Kerenidis und Prakash funktioniert auf ähnliche Weise, verwendet aber polynomial wachsende Tiefen bis zu einer maximalen Tiefe $D \ll O(\frac{1}{\varepsilon})$ (Kerenidis & Prakash, 2020). Es wird daher eine geringere Tiefe benögt, aber mehr Durläufe p. Des Weiteren wurde gezeigt, dass die benötigten Tiefen allerdings problemspezifisch ausgewählt werden können und sogar in speziellen Fällen die von Buchard aufgestellt Grenze ignorieren können (Tanaka et al., 2020).

Wie schaut es letztendlich mit der möglichen Beschleunigung aus? Der Standard - Quantenalgorithmus Q in 3.2.1 erreicht seine $O(\frac{1}{\varepsilon})$ Beschleunigung, falls die fixen Schaltungstiefen größer als $\frac{1}{\varepsilon}$ werden können. Ansonsten ist keine Beschleunigung vorhanden. Für die alternative Methode von Kerenidis und Prakash ist die Beschleunigung gleich wie der Standard-Quantenalgorithmus für die Schaltungstiefen größer als $\frac{1}{\varepsilon}$, bricht aber die klare Grenze bei $\frac{1}{\varepsilon}$ und nimmt kontinuierlich bei kleiner werdender Schaltungstiefe ab.

4 Portfoliotheorie

Ein weiteres mögliches Aufgabengebiet sind Portfolio-Optimierungen. Es geht hierbei darum aus einer großen Auswahl von Finanzprodukten, mit unterschiedlichen erwarteten Renditen und Risiken, ein Portfolio zu entwickeln, welches definierte Anforderungen erfüllt. Da Einschränkungen einen erheblichen Einfluss darauf haben, wie leicht sich ein solches Portfolio finden lässt, betrachten wir nur die unbeschränckte Portfolio-Optimierung und die Portfolio-Optimierung ohne die Möglichkeit auf Leerverkäufe. Diese Unterschiede werden sich dann bei der Implementierung in Quantenalgorithmen zeigen, weshalb wir diese getrennt betrachten werden.

4.1 Grundlagen

Das Ziel der Portfolio-Optimierung ist ein Portfolio zu finden, welches für das kleinste Risiko, den höchsten erwarteten Profit erzielt. Dies basiert auf der grundlegenden Behauptung, dass der zu erwartende Gewinn in Relation zu dem damit verbundenen Risiko steht. Wenn Portfolio A den gleichen erwarteten Profit wie ein Portfolio B aufweist, der potentielle Gewinn von B allerdings eine größere Varianz besitzt als A, so gilt das Portfolio B als ineffektiv und daher nicht erstrebenswert. Wie die Erwartungswerte errechnet werden, spielt hierbei keine Rolle, man könnte zum Beispiel Monte-Carlo-Simulationen hernehmen. Um dies nun zu formalisieren: Für einen Finanzmarkt mit n Vermögenswerte sei eine Renditenfunktion $R(t) \in \mathbf{R}^n$ zum Zeitpunkt $t \in T$ und $R^i(t)$ die Rendite des $i-ten$ Vermögenswert mit $i \leqslant n$. Des Weiteren ist μ die erwartete Rendite und Σ die Kovarianzmatrix von μ, sodass:

$$\mu = \frac{1}{T} \sum_{t \in T} R(t)$$
$$\Sigma = \frac{1}{T-1} \sum_{t \in T} (R(t) - \mu)(R(t) - \mu)^T$$

Sei nun der Portfoliovektor $w = (w_0, ..., w_n)^T \in \mathbf{R}^n$ wobei w_j die Anzahl an Investitionen in den $j-ten$, Vermögenswert darstellt mit $j \leqslant n$. Damit ist $R_w(t)$ die Rendite des Portfolios w zum Zeitpunkt t für welche gilt:

$$R_w(t) = \sum_{i=0}^{n} w_i R^i(t)$$

und folglich auch der Erwartungswert $\mu_w \in \mathbf{R}$ des Portfolios w:

$$\mu_w = \mathbf{E}(R_w) = w^T \mu$$

sowie die Varianz $\sigma_w^2 \in \mathbf{R}$ des Portfolios w:

$$\sigma_w^2 = \mathbf{E}(R_w - \mu_w))^2) = w^T \Sigma w$$

Damit lässt sich ein unbeschränktes Portfolio-Optimierungsproblem wie folgt darstellen:

$$\begin{gathered} min\ w^T \Sigma w \\ \text{mit } R^T w = \mu \wedge A^T w = b \text{ als NB} \end{gathered}$$

wobei die letztere Bedingung definieren soll, wie viel Prozent des verfügbaren Kapitals investiert werden soll. Im Fall $b = 1$ wäre es beispielsweise das gesamte Kapital.

Ein Lösungsansatz eines solchen Portfolio-Optimierungsproblem ist die Verwendung von Lagrange-Multiplikatoren η und Θ (Appendix):

$$\mathcal{L}(w, \eta, \Theta) = \tfrac{1}{2} w^T \Sigma w + \eta(R^T w - \mu) + \Theta(A^T w - b)$$

Damit wär das Lösen eines $n + 2$ dimensionalen linearen Gleichungssystem notwendig:

$$\begin{pmatrix} 0 & 0 & R^T \\ 0 & 0 & A^T \\ R & A & \Sigma \end{pmatrix} \begin{pmatrix} \eta \\ \Theta \\ w \end{pmatrix} = \begin{pmatrix} \mu \\ b \\ 0 \end{pmatrix}$$

4.2 Unbeschränkte Portfolio Optimierung

Bei der Umsetzung solcher Rechnungen in einen Quantencomputer ergeben sich nun einige Unterschiede bezüglich der Restriktionen, die man für seine Portfolio-Optimierungen umsetzen will. Beginnen wir daher mit der unbeschränkte Portfolio-Optimierungen.

Wie in der oberen Gleichung zu sehen, genügt es für unbeschränkte Portfolio - Optimierungen das obere $n+2$-dimensionales Gleichungssystem zu lösen. Fortan werden wir dieses Gleichungssystem der Gleichung: $Mx = b$ gleichsetzen. Die Rechenzeiten hängen damit zusammen wie schnell solche linearen Gleichungen gelöst werden können. Die Laufzeit eines klassischen LGS-Löser skalieren bestenfalls wie $O(n^\omega)$, wobei ω von der behandelten Matrix abhängt. Es gilt $\omega \leqslant 2,373)$. (Buchberger, 1990), (Le Gall, 2014). Sehr oft wird allerdings die Cholesky Zerlegung verwendet welche die Rechenzeit auf $O(n^3)$ erhöht. Für eine unbeschränkte Portfolio Optimierung im Zeitraum T und mit N Anlagemöglichkeiten benötigt

ein klassischer Algorithmus $O(TN^2)$ Operationen für die Erstellung der Kovarianzmatrix und schließlich $O(n^3)$ Operationen für das Implementieren der Pseudo-Inversen. Für den 'quantum linear system solver' ist die Laufzeit so nicht klar abschätzbar. Sie hängt wie bei dem klassischen Algorithmus stark mit den Eingenschaften von M zusammen. Zudem ist es auch wichtig zu gewissen Momenten einen sogenannten Quantenzugang (engl: quantum access), zu den Elementen M und b herstellen zu können. Vereinfacht bedeutet dies soviel, dass die verwendeten Qubits durch die Wahrscheinlichkeiten ihrer Superpositionen die Matrix M, und den Vektor b beschreiben. Des Weiteren ist auch die Frobenius Norm, die Konditionszahl und die Sparcity der Matrix M notwendig um eine ε-Approximation der Lösung zu bekommen (Appendix). Eine vereinfachte Zusammenfassung der Vorgänge wäre folgende:

- $M \in \mathbf{R}^{N \times N}$ mit Eigenwerten $\lambda_i \in [-1, -\frac{1}{\kappa}] \cup [\frac{1}{\kappa}, 1]$
- Zum Zeitpunkt T_M existiert ein Quantenzugang zu M
- Es wird ein Zustand $|b\rangle$ für den Zeitpunkt T_b vorbereitet

$\Rightarrow$ ein Zustand $|M^{-1}b\rangle$ mit Abweichung ε wird in $O((T_M \kappa\zeta + T_b\kappa) log(\kappa\zeta/\varepsilon))$ mit $\zeta(M) = min(\frac{\|M\|_F}{\|M\|_2},\ sparcity(M))$ erzeugt.

Hier ist es auch wichtig hervorzuheben, dass es sich bei der Lösung $M^{-1}b$ um einen Quantenzustand handelt. Eine klassische Lösung würde eine weitere Rechnung benötigen, mit folgenden Aufwänden:

- $O\left(\frac{N}{\varepsilon^2}\right)$ für eine $\varepsilon-$ Approximation mit l_2-Norm
- $O\left(\frac{1}{\varepsilon^2}\right)$ für eine $\varepsilon-$ Approximation mit $l_\infty-$Norm

Ein aktueller Quantencomputer benötigt somit eine insgesamte Laufzeit von:

$$O\left(\frac{N}{\varepsilon^2}(T_M \kappa\zeta + T_b\kappa) log(\kappa\zeta/\varepsilon)\right)$$

für das eine $\varepsilon-$Approximation mit l_2-Norm der Lösung eines unbedingten Portfolio-Optimierung-Problem [RL18]. Eine genaue Gegenüberstellung zu der Laufzeit eines klassischen Algorithmus ist daher nur bedingt möglich.

Es ist hier ebenfalls angebracht hervorzuheben, dass die Vorteile eines Quantencomputer explizit für die Portfolio-Optimierung noch nicht so breit untersucht worden sind wie für

Monte-Carlo-Simulationen. So ist bei dem "quantum linear solver"von einer potenziellen exponentiellen Beschleunigung die Rede (Gilyén, Su, Low & Wiebe, 2019), während eine solch genau Auskunft für das Lösen von unbeschränkten Portfolio-Optimierung-Problemen noch nicht möglich ist.

4.3 Portfolio-Optimierung ohne Leerverkäufen

Das obere Gleichungssystem spiegelt nur einfachste Form der Portfolio-Optimierung wieder. Oft werden noch weitere Bedingungen hinzugefügt, welche in der Realität angewandt werden müssen. So wird beispielsweise die Anzahl der Wertpapiere aus gewissen Anlageklassen beschränkt. Dies kann unternehmensintern oder aufsichtsrechtlich verpflichtend sein (siehe damals noch Solvency I für Versicherungsunternehmen). Es kann auch schlich der Fall sein, dass nicht genug Geld zur Verfügungen steht größere Mengen an gewissen Wertpapieren zu kaufen. Wir werden uns hier allerdings nur mit der "positivity constraint"befassen, welche uns verwehrt eine negative Stückzahl der Anlage zu erwerben. Dies wäre theoretisch mit dem "Leerverkaufen"möglich. Eine solche Problemstellung wurde sich schließlich so darstellen lassen:

$$\begin{gathered} min\ w^T \Sigma w \\ \text{mit } R^T w = \mu \wedge A^T w = b \\ \text{und } w_i > 0,\ \forall i \in \{1, ..., n\} \text{ als NB} \end{gathered}$$

Eine Herangehensweise dieses Problem mit einem Quantencomputer zu lösen, entstammt der Arbeit von Kerenidis, Prakash und Szilágyi's (Kerenidis, Prakash & Szilágyi, 2019), welche sich mit einem Bedingte Portfolio-Optimierungs-Problem mit r "positivity constraints". Die Aufgabe wird auf ein second order cone program (SOCP) heruntergebrochen und schließlich mit einem Innere-Punkte-Verfahren gelöst. Dabei wird bei jeder Iteration ein ähnliches Gleichungssystem gelöst wie oben. Die Ergebnisse werden mit einem sogenannten "quantum tomography procedure"bearbeitet und für die nächsten Gleichungen vorbereitet (Kerenidis & Prakash, 2020). Der Quantenalgorithmus für ein Innere-Punkte-Verfahren skaliert wie $O(\sqrt{r} log \frac{1}{\varepsilon}))$. Die dabei ebenfalls verwendete "quantum tomography procedure"hat den Nachteil aktuell noch sehr ungenau zu sein, weshalb das Ergebnis auch hier wieder nur eine Abschätzung sein kann. Insgesamt hat der Quantenalgorithmus eine Laufzeit von:

$$O(N \frac{\zeta \kappa}{\delta^2} \sqrt{r} \frac{1}{\varepsilon})$$

Im Vergleich dazu der bestmögliche klassische Algorithmus:

$$O(N^3 \sqrt{r} \frac{1}{\varepsilon})$$

wobei:

• r die Anzahl der positivity constraints, welche der Rank der SOCP entspricht. Für das unbeschränkte Portfolio, gelte der Fall $r = 1$

• ε die Lagrange-Dualität der Lösungen der SOCP

• $\zeta,\ \delta$ quantifizierende Parameter für die Distanz der Zwischenlösungen des SOCP

• κ maximale Konditionszahl der Matrizen aus dem Innere-Punkte-Verfahren für die SOCP

Wie man sieht, ist eine Beschleunigung möglich, falls $N^2 > \frac{\zeta\kappa}{\delta^2}$

Es wurde auch bereits eine Überprüfung durchgefürht, infolgedessen die täglichen Aktienpreise der SP500 Unternehmen in einem solchen Portfolio-Optimierungs-Problem bewertet wurden. Es wird geschätzt, dass ein fehlertoleranter Quantenalgorithmus eine solche Aufgabe in einer Zeit von $O(N^{2,37})$ (Kerenidis et al., 2019) lösen könnte, was eine polynomisch skalierende Verbesserung zu dem klassischem Algorithmus wäre.

Portfolio-Optimierungen können sich daher durchaus als Aufgabenstellung anbieten, für die Quantencomputer einen relevanten Unterschied machen können. Ohne der fehlertoleranten Quantencomputer ist allerdings ein großteils der Aufgaben noch stark von den spezifischen Parametern abhängig und somit die eventuelle Beschleunigung nicht immer genau absehbar. Aber auch hier zeigen sich einige Experimente als wegweisend für den Vorteil der Quantencomputer.

5 Deep Learning

Die letzte Aufgabenstellung, welche wir uns anschauen werden, ist das maschinelle Lernen. Dabei handelt sich um ein künstliches System, welches aus einer großen Anzahl an Beispielen Muster erkennen soll und diese schließlich umsetzen kann. Dadurch, dass immer großere Datensätze zur Verfügung stehen und zu verarbeiten sind, gewinnt diese Methode in jüngerer Vergangenheit immer mehr an Wichtigkeit. Sie wird unter anderem im automatisierten Portfoliomanagement, dem automatisierten Handel von Wertpapieren als auch bei Erkennung von Finanzbetrug eingesetzt. Wir werden uns hier grob auf die drei Kategorien von maschinellen Lernen beschränken nämlich dem überwachten, unüberwachten und dem bestärkenden Lernen. Viele Quantentechnologien, welche für diese Aufgabenstellungen zur Verfügung stehen, sind oft für andere Rechnungen entwickelt worden, weshalb wir auf mögliche Beschleunigungen nur oberflächlich eingehen werden.

5.1 Unüberwachtes Lernen

Beginnen wir mit der allgemeinsten Form des maschinenellen Lernens, dem unüberwachten Lernen. Ein solcher Algorithmus bekommt eine Menge an ungeordneten Daten, versucht daraus in ein statistisches Modell zu erzeugen und Zusammenhänge zu erkennen.

5.1.1 Clustering

Die meist verbreitete Methode hierfür ist das Clustering. Dabei werden Daten in Kategorien, sogenannte Cluster eingeteilt, die sich durch charakteristische Merkmale voneinander unterscheiden. Es wurden bereits Quanten-Alternativen zu einem klassischen K-Means Algorithmus angedacht für welchen die Laufzeit lediglich poly-logarithmisch mit der Größe des Datensatzes skaliert. Dieser Quantenalgorithmus basiert stark auf größere Matrizenmultiplikation und Distanzabschäzung und ist womöglich bereits in näherer Zukunft einsetzbar, was unter anderem für die Abschätzung des Portfoliorisikos von Relevanz sein könnte.

5.2 Überwachtes Lernen

Beim überwachten Lernen bekommt der Algorithmus ein geordnetes Paar an Daten und es wird versucht diese Daten in einen adequaten Zusammenhang zu bringen, um schließlich Hypothesen darüber auszustellen. Ein Beispiel an geordneten Datensätzen könnte $(V(t), P(t))$ sein mit $P(t)$ der Preis einer Aktie zum Zeitpunkt t und $V(t)$ die Volatilität zum Zeitpunkt t.

5.2.1 Lineare Regression

Bei der linearen Regression wird ein lineares Modell angenommen, um Zusammenhänge der analysierten Zufallsvariablen zu erstellen, welche als Linearkombination der, von der Dimension des System abhängigen, Regressionskoeffizienten dargestellt werden. Für unser Beispiel von oben bräuchten wir hier eine einfache Regression mit zwei Parametern und die Regression würde mit einer Geraden durch die Punktewolke $(V(t), P(t))$ beschrieben werden. Hier ist auch die kleinste-Quadrate-Schätzung relevant um ein solches Problem zu lösen. Ein Quantenalgorithmus für eine solche Aufgabe würde sich im Kern nicht stark unterscheiden. Es wurden bereits Algorithmen zu dem ”least square”, also der kleinste-Quadrate-Schätzung entwickelt. Diese haben allerdings eine hohe Schaltungstiefe, was den Einsatz in näherer Zukunft problematisch macht. Andere Quantenalgorithmen welche effiziente Distanzabschätzer, efficient quantum distance estimators, versprechen könnten in diesem Bereich eine quadratischen Beschleunigung ermöglichen.

5.2.2 Klassifikationsverfahren

Bei Klassifikationsverfahren werden Daten in bereits vorher definierte Klassen eingeteilt. Oft wird dabei ein ”Nearest Centroid Classifier”verwendet. Dabei werden die zu verarbeitenden Daten mit den definierten ”Centroid”verglichen und ”näherste”herausgefiltert. Sehr bekannt ist hierbei der Rocchio Algorithmus welcher einen solchen Algorithmus für Texte durchführt und so zum Beispiel große Mengen an Feedback, oder geschriebene Informationen verarbeiten kann. Bei einer quantencomputerunterstützten Umsetzung eines Klassifikationsverfahren ist auch hier eine zeitnahe Umsetzung möglich da bereits viele Quantenalgorithmen entwickelt wurden, welche effizient Distanzen berechnen können, beziehungsweise den ”nähersten Centroid”durch Clustering erkennen können.

5.3 Bestärkendes Lernen

Bei bestärkendem Lernen handelt es um eine Reihe an Methoden, die versucht einen eigenen sogenannten Software-Agenten, aus verschiedene Annahmen, die best-passende auszusuchen. Dies geschieht meistens mithilfe einer Nutzenfunktion. Der Software-Agend muss eine gewisse Handlung aufführen und berechnet anhand der Nutzenfunktion welcher der ihm zur Verfügung stehenden Möglichkeit den größten Nutzen in der Zukunft haben werden. Anhand davon kreiert er ein damit entstehendes System. Ein mögliches Anwendungsbeispiel wäre hierführ der algorithmische Handel von Wertpapieren. Überlegungen diese Rechnungen mit einem Quantencomputer durchzuführen sind allerdings, zu aktuellem Stand, noch relativ unentwickelt.

6 Zusammenfassung

Quantencomputer sind ein vielversprechendes Werkzeug um viele maschinelle Probleme mit einer revolutionären Geschwindigkeit lösen zu können. Wie wir gesehen haben können viele computerunterstützte Rechnungen, die aktuell in der Finanzbranche umgesetzt werden, mit vollständig entwickelten Quantencomputer deutlich schneller vollzogen werden. Diese Beschleunigungen würden nicht nur in vielen Bereichen zu einem Industriestandard werden, sondern ermöglichen im manchen Fällen sogar Ergebnisse zu erhalten, die zuvor durch ihren hohen Rechenaufwand gar nicht erst in Echtzeit möglich waren. Diese Erwartungshaltung muss allerdings aktuell noch stark gedämpft werden. Ein fehlertoleranter Quantencomputer mit einer funktionierenden QRAM ist noch fern des technisch Möglichen. Oft wurde gezeigt, dass mit den aktuellen Hindernissen und Einschränkung die ein aktueller Quantencomputer mit sich bringt, die extreme Beschleunigung ausbleibt. Dennoch ist in vielen Gebieten das Interesse groß, so früh wie möglich eine schnellere Computerleitung herbeizubringen und durch meistens kreative Zusammenspiele zwischen klassischen und Quantencomputer wird eine zukunftsnahe Integration von Quantencomputer in gewisse Bereiche nicht mehr unvorstellbar. Ob diese zu jenen Zeitpunkt bereits die fundamentalen Änderungen herbringen die sie versprechen, bleibt allerdings abzuwarten.

7 Appendix

7.1 Holevos-Theorem

Auch wenn n Qubits eine größere Menge an Informationen durch das Superpositionsprinzip darstellen können, ist die daraus erhaltbare Anzahl an klassisch lesbaren Informationen, gleich jener von n klassischen bits.

7.2 Chebychev Ungleichung

Sei X eine Zufallsvariable mit $E(X) = \mu$ und $V(X) = \sigma^2$. Für k eine strikt positiv, reelle Zahl gilt:

$$P(|X - \mu| \geqslant k\sigma) \leqslant \frac{1}{k^2}$$

7.3 Lagrange Multiplikation

Die Methode der Lagrange Multiplikation wird verwendet um die Extrema einer Funktion mit mehreren Variablen unter Nebenbedingungen zu finden. Dafür betrachten wir eine Funktion $f(x, y, z)$. Für Funktionen mit einer anderen Anzahl an Variablen funktioniert dies analog. Zudem seien n Nebenbedingungen. Diese können auf folgende Form gebracht werden:

$NB_1 : g_1(x, y, z) = 0$
$\vdots$
$NB_n : g_n(x, y, z) = 0$

Nun kann man folgende Hilfsfunktion mit den Parametern $(\lambda_1, ..., \lambda_n)$:

$$L(x, y, z, \lambda_1, ..., \lambda_n) = f(x, y, z) + \lambda_1 g_1(x, y, z) + ... + \lambda_n g_n(x, y, z)$$

Wir bestimmen nun die Extremstellen die Funktion L indem wir die partiellen Ableitung Null setzen:

$$\frac{\partial L(x, y, z, \lambda_1, ..., \lambda_n)}{\partial x} = 0$$
$$\frac{\partial L(x, y, z, \lambda_1, ..., \lambda_n)}{\partial y} = 0$$

$$\frac{\partial L(x, y, z, \lambda_1, ..., \lambda_n)}{\partial z} = 0$$
$$\frac{\partial L(x, y, z, \lambda_1, ..., \lambda_n)}{\partial \lambda_1} = 0$$
$$\vdots$$
$$\frac{\partial L(x, y, z, \lambda_1, ..., \lambda_n)}{\partial \lambda_n} = 0$$

Die Lösungen sind schließlich die Extrema.

7.4 Frobenius Norm

Sei eine $m \times n$ Matrix A mit den Einträgen $a_{i,j}$. Die Frobenium Norm $\| \bullet \|_F$ ist eine eine Matrizen-Norm für die gilt:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{i,j}|^2}$$

7.5 Sparcity

Die Sparctiy ist eine Kennzahl dafür wie viele Einträge einer "dünnbesetzten Matrix"(im engl. Sparce Matrix) 0 sind. Die Anzahl der Null-Einträge einer Matrix werden dabei durch die Anzahl aller Einträge der Matrix dividiert.

Beispiel:

$$A := \begin{pmatrix} 1 & 3 & 0 \\ 0 & 0 & 4 \\ 2 & 0 & 0 \end{pmatrix} \Rightarrow Sparcity(A) = \frac{5}{9}$$

Literatur

Aaronson, S. & Rall, P. (2020, Jan). Quantum approximate counting, simplified. *Symposium on Simplicity in Algorithms*, 24–32. Zugriff auf http://dx.doi.org/10.1137/1.9781611976014.5 doi: 10.1137/1.9781611976014.5

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., ... others (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, *574* (7779), 505–510.

Bouland, A., van Dam, W., Joorati, H., Kerenidis, I. & Prakash, A. (2020). Prospects and challenges of quantum finance. *arXiv preprint arXiv:2011.06492*.

Brassard, G., Høyer, P., Mosca, M. & Tapp, A. (2002). Quantum amplitude amplification and estimation. *Quantum Computation and Information*, 53–74. Zugriff auf http://dx.doi.org/10.1090/conm/305/05215 doi: 10.1090/conm/305/05215

Buchberger, B. (Hrsg.). (1990). *J. Symb. Comput.*, *9* (3).

Burchard, P. (2019). *Lower bounds for parallel quantum counting.*

Edwin Pednault, D. M., John Gunnels & Gambetta, J. (2019). *On "quantum supremacy".* Zugriff auf https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/

Feynman, R. P. (2018). Simulating physics with computers. In *Feynman and computation* (S. 133–153). CRC Press.

Gilyén, A., Su, Y., Low, G. H. & Wiebe, N. (2019, Jun). Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. Zugriff auf http://dx.doi.org/10.1145/3313276.3316366 doi: 10.1145/3313276.3316366

Kerenidis, I. & Prakash, A. (2020, Oktober). A quantum interior point method for lps and sdps. *ACM Transactions on Quantum Computing*, *1* (1). Zugriff auf https://doi.org/10.1145/3406306 doi: 10.1145/3406306

Kerenidis, I., Prakash, A. & Szilágyi, D. (2019). Quantum algorithms for portfolio optimization. *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*.

Le Gall, F. (2014). Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation* (S. 296–303). New York, NY, USA: Association for Computing Machinery. Zugriff auf https://doi.org/10.1145/2608628.2608664 doi: 10.1145/2608628.2608664

Montanaro, A. (2015, Sep). Quantum speedup of monte carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *471* (2181), 20150301. Zugriff auf http://dx.doi.org/10.1098/rspa.2015.0301 doi: 10.1098/rspa.2015.0301

Tanaka, T., Suzuki, Y., Uno, S., Raymond, R., Onodera, T. & Yamamoto, N. (2020). *Amplitude estimation via maximum likelihood on noisy quantum computer.*

The world of physics. (2012). Zugriff auf https://sites.psu.edu/passionmalencia/2012/