

IMES DISCUSSION PAPER SERIES

量子計算の概要： ファイナンスへの応用を例に

いしがきかつあき
石垣克明

Discussion Paper No. 2023-J-10

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES

BANK OF JAPAN

日本銀行金融研究所

〒103-8660 東京都中央区日本橋本石町 2-1-1

日本銀行金融研究所が刊行している論文等はホームページからダウンロードできます。

<https://www.imes.boj.or.jp>

無断での転載・複製はご遠慮下さい。

備考：日本銀行金融研究所ディスカッション・ペーパー・シリーズは、金融研究所スタッフおよび外部研究者による研究成果をとりまとめたもので、学界、研究機関等、関連する方々から幅広くコメントを頂戴することを意図している。ただし、ディスカッション・ペーパーの内容や意見は、執筆者個人に属し、日本銀行あるいは金融研究所の公式見解を示すものではない。

量子計算の概要:ファイナンスへの応用を例に

いしがきかつあき
石垣 克明*

要 旨

近年、テクノロジーの発展に伴い高速計算の需要が高まる中、ミクロな物質の振る舞いや性質（重ね合わせ、干渉、量子もつれ）を演算処理に用いる量子コンピュータに期待が寄せられている。量子コンピュータによる計算（量子計算）の高速化は、従来型のコンピュータでは計算時間の関係で困難とされる問題を解決し得るため、ファイナンスをはじめ様々な分野への応用研究が行われている。こうした背景を踏まえ、本稿では量子計算とファイナンス分野への応用研究をサーベイする。具体的には、量子計算に関心のあるファイナンス研究者や実務家向けに、量子計算の仕組みや量子アルゴリズムについて解説し、金融商品のプライシング、リスク管理、ポートフォリオ最適化といったファイナンスの問題への応用事例を紹介するとともに、量子計算の実務適用の課題についても述べる。

キーワード：量子コンピュータ、量子力学、重ね合わせ、干渉、量子もつれ、量子計算、量子アルゴリズム

JEL classification: C6、G1

* 日本銀行金融研究所（現三菱 UFJ 銀行、E-mail: katsuaki_ishigaki@mufg.jp）

本稿の作成に当たっては、鈴木洋一特任准教授（慶應義塾大学）、田中智樹氏（三菱 UFJ 銀行）、日本銀行のスタッフ等から有益なコメントを頂いた。ここに記して感謝したい。ただし、本稿に示されている意見は、筆者個人に属し、日本銀行の公式見解を示すものではない。また、ありうべき誤りはすべて筆者個人に属する。

1. はじめに

近年、半導体集積密度が1~2年で約2倍になるという半導体加工技術の経験則であるムーアの法則 (Moore [1975]) が限界を迎えつつあると言われている。一方で、機械学習・AI等の大量データを用いた最新テクノロジーは発展を続けており、高速計算に対する需要は益々高まっている。そのため、既存技術とは異なる原理で動作する次世代のコンピュータ開発に対して期待が寄せられている。その候補の1つがマイクロな物質 (量子) の振る舞いや性質 (重ね合わせ、干渉、量子もつれ) を演算処理に用いた量子コンピュータである。量子コンピュータによる計算 (量子計算) は、従来型のコンピュータ (以下、古典コンピュータ) では計算量の観点から困難とされている大規模な因数分解やデータ検索問題を効率的に解くことができると期待されている。既に、量子超越性¹を示す研究報告のほか、D-Wave社やIBM社の商用量子コンピュータの登場を契機として、量子コンピュータ開発に対する注目は高まっており、量子コンピュータの活用可能性がある様々な分野において、量子化学計算や量子機械学習等、量子計算の応用に関する研究が盛んになってきている。

こうした背景を踏まえ、本稿では量子計算とそのファイナンス分野への応用研究をサーベイする。具体的には、量子計算に関心のあるファイナンス研究者や実務家向けに、量子計算の仕組みや高速化が可能となる背景、具体的な量子アルゴリズムについて解説したうえで、金融商品のプライシング、リスク管理、ポートフォリオ最適化といったファイナンスの実務問題への応用事例を紹介する。

本稿の概要をあらかじめ紹介すると以下のとおりである。2節では、量子コンピュータに関する基本事項を説明する。まず、量子コンピュータの歴史的な経緯や現在の取組みを解説し、量子コンピュータの種類 (ゲート方式、アニーリング方式) と課題について整理する。次にショアの因数分解アルゴリズムやグローバーのデータ検索アルゴリズムといった代表的な量子アルゴリズムを解説し、量子計算の仕組み (重ね合わせ、干渉、量子もつれ) や論理回路 (量子回路) についても説明する。

続く3節では、量子計算の基礎についての数学的な説明を行う。情報の基本単位である量子ビットや量子ビットを操作する量子ゲートについて、量子力学と線形代数を用いて具体的な計算例を示しつつ説明する。

次に4節では、量子計算における主要な量子アルゴリズムについて説明する。最初に量子アルゴリズムを概観し、その後、各アルゴリズムについて詳しい説明を行う。本節では、ゲート方式の量子コンピュータについて、誤り耐性量子コンピュータ (FTQC : Fault-Tolerant Quantum Computer) と誤り訂正処理を持たない小規模量子デバイス (NISQ : Noise Intermediate-Scale Quantum) を中心に扱う。FTQCのアルゴリズムについては、まず、量子フーリエ変換、量子位相推定、量子振幅増幅、量子振幅推定といったサブルーチンとして

¹ 量子超越性とは、古典コンピュータでは実用的な時間では解決できない問題を、プログラム可能な量子デバイスによって解決できることである (Preskill [2012])。

用いられる基本アルゴリズムを説明したうえで、量子モンテカルロ積分による数値積分や線形方程式の求解アルゴリズムといった応用アルゴリズムについて説明する。一方、NISQ デバイスのアルゴリズムについては、量子古典ハイブリッド型の計算方式を実装する変分量子回路について概観した後、量子近似最適化アルゴリズムの仕組みを説明する。

5 節では量子計算のファイナンス分野への応用例について紹介する。具体的には、金融商品（例：オプション等）のプライシング、リスク量（例：バリュー・アット・リスク等）の計算、ポートフォリオ最適化といったファイナンス分野の計算問題に対する量子アルゴリズムの応用事例について解説する。量子アルゴリズムによって、これらのファイナンス問題を古典アルゴリズムよりも効率的に計算できる可能性があることを紹介する。

6 節では一連のサーベイ結果を踏まえつつ、量子計算のファイナンス分野への応用研究に関して、実務適用の課題について考察する。最後に 7 節で本稿をまとめる。

本稿の構成は以上に述べたとおりであるが、読者の知識や関心によって読み方を変えることができる。例えば、2 節と 3 節は、4 節以降を読むための量子計算の予備知識の説明であるため、既に量子コンピュータに馴染みのある読者は 4 節から読み進めることが可能である。4 節には主要な量子アルゴリズムをまとめており、5 節を読むためには一読することを推奨する。5 節はファイナンス問題への応用別に特化しており、ある程度独立して読めるようにしてある。研究の詳細ではなく、ファイナンスへの応用研究の全体の流れや課題を知りたい読者は、1 節と 2 節を読んだ後、6 節へ進むこともできる。

2. 量子コンピュータとは²

本節では、量子コンピュータの基本事項と考え方を説明する。まず量子コンピュータの歴史や現状を概説してから、量子コンピュータの実用化に向けた課題を整理し、量子コンピュータによって高速化が期待される問題とアルゴリズムを概説する。次に量子計算による高速化の背景となる量子の性質（重ね合わせや干渉）、量子計算への応用の考え方や量子回路を解説する。

(1) 歴史的経緯

量子コンピュータの歴史を解説する（表 1）。量子コンピュータの誕生は 1980 年代まで遡る。Feynman [1982] が、電子や原子といったミクロな世界の物質（量子）を使ったシミュレーションのアイデアを提案したことに端を発する。その後、Deutsch [1985] により量子コンピュータの理論的な定式化がなされ量子コンピュータの概念ができあがった。

1990 年代になると古典コンピュータの計算アルゴリズムを原理的に超える可能性を持つ量子アルゴリズムの発表が相次いだ。Shor [1994] よって因数分解のアルゴリズムが考案さ

² 量子コンピュータの開発や取組みに関する詳しい解説は、研究開発戦略センター [2018]、藤井ほか [2022]、先端技術ラボ [2020]、藤吉 [2022]、統合イノベーション戦略推進会議 [2022a, b] を参照。

れ、Grover [1996] によってデータ検索のアルゴリズムが考案された。特に因数分解アルゴリズムは、RSA 暗号等の公開鍵暗号の安全性を揺るがす可能性が示唆されたことから注目を浴びた³。これら 2 つのアルゴリズムの登場が量子コンピュータ研究の火付け役となり、以降の量子コンピュータの開発の原動力となった。また Kadowaki and Nishimori [1998] により、ゲート方式とは異なるアニーリング方式の原理が発表された。1990 年代にはアルゴリズム面での大きな進展があったが、量子コンピュータ自体の開発は難しく、大学の研究室で行われる実証実験やサイエンスに近いものであった。

2010 年代になると量子コンピュータの研究開発は転機を迎える。量子コンピュータ開発にかかわる企業が増え、どうやって量子コンピュータを作るかというエンジニアリング的な研究テーマにシフトし始める。まず、D-Wave 社によってアニーリング方式に基づく初の商用量子コンピュータが発表されて話題となり (Zyga [2011])、その後、マルティニスらのグループによって高精度の量子ゲート操作⁴が成功したことで、誤り訂正処理を持つ量子コンピュータ (FTQC) の実現が現実味を帯び始めた (Barends *et al.* [2014])。その後、IBM 社によってゲート方式に基づく量子コンピュータのオンライン公開や商用量子コンピュータの発表により、ユーザーへの門戸を広げることになった (IBM [2016]、IBM [2019])。また、Google 社による量子コンピュータの量子超越性の発表が注目を集めた (Arute *et al.* [2019])⁵。しかし、実務での運用に足る量子コンピュータの開発は依然として難しく、開発目標である FTQC の実現には至っていない。すなわち、現状の量子コンピュータ (NISQ) は、少数量子ビットで誤り訂正処理がないため、十分な精度の計算結果を得ることができない。こうした観点から、Preskill [2018] は、量子コンピュータの開発は、まだ、発展途上にあるので、中長期的な活用方法を検討していくべきと語っている。

³ 量子コンピュータが公開鍵暗号を破る可能性に関する評価については、宇根・菅 [2021]、清藤・四方 [2019] を参照。

⁴ 誤り訂正処理を行うための、量子ゲートのエラー率の上限は 1%とされている。

⁵ Google 社はスーパーコンピュータが約 10,000 年かかるタスクを、53 量子ビットの NISQ を使って 200 秒で実行し量子超越性を主張。これに対して、IBM 社はメモリと HDD を組み合わせてストレージを拡大すれば、古典コンピュータでも同様の実験を 2.5 日で実行できると指摘 (Pednault *et al.* [2019])。またスーパーコンピュータで同様のシミュレーションが可能となっている (Liu *et al.* [2021])。

表 1 量子コンピュータの歴史年表

フェーズ	年	中心人物・企業	事項
黎明期	1982年	ファインマン	量子力学を使ったコンピュータのアイデアを提案
	1985年	ドイチュ	量子コンピュータの計算モデル(量子チューリングマシン)を提唱
第1次発展 サイエンス	1994年	ショア	効率的な因数分解アルゴリズムを考案
	1996年	グローバー	データ検索の効率的なアルゴリズムを考案
	1998年	門脇・西森	量子アニーリング法を提案
	1999年	中村・蔡(NEC)	初の量子ビットの演算に成功
第2次発展 エンジニア リングへ	2011年	D-Wave社	世界初の商用量子コンピュータ(D-Wave One: アニーリング方式)を発表
	2014年	マルチニス	大規模な量子コンピュータに必要な精度の量子ゲートに成功
	2016年	IBM社	量子コンピュータ(ゲート方式)をオンライン公開
	2017年	プレスキル	誤り訂正なし、小規模量子デバイスNISQの活用を問題提起
	2019年	IBM社	商用量子コンピュータ(IBM Quantum System One: ゲート方式)を発表
	2019年	Google社	量子超越性の実証を発表(後にIBM社より反論指摘)

(2) 量子コンピュータの種類と課題

本節では、改めて量子コンピュータの演算処理の方式の種類と、実用化に向けた課題について整理する。本稿では量子コンピュータを、処理方式によってゲート方式とアニーリング方式に大別する。ゲート方式では、情報単位である量子ビットに対してゲート操作を行い、量子ビットの状態を変化させて演算処理を行う。古典コンピュータと同様に論理回路を持つ方式で、幅広い問題を解くことが可能である。加えて、特定の問題に対しては、古典コンピュータ対比での高速化が理論的に保証されている量子アルゴリズムが存在する⁶。一方、アニーリング方式は、量子ビットでイジング・モデル⁷を構成し、外部磁場と量子ビットを相互作用させて、物理エネルギーの最小値を探索する方式である(4節を参照)⁸。物理エネルギーを最適化問題のコスト関数に対応させることで、組合せ最適化問題を解くことが可能である。なお、ゲート方式とは異なり論理回路を持たない。

次に、量子コンピュータ開発に向けた課題について整理する。ゲート方式は基礎研究の段階である。FTQCの開発の難しさはいくつかあるが、量子の性質に起因する理由が2点存在する。1つは、量子状態がデリケートであり、量子的な性質が壊れやすいことである⁹。もう1つは、量子計算のための量子状態の操作には高い精度が求められることである

⁶ ゲート方式の量子コンピュータを実現する物理系には、超電導方式、イオントラップ方式、光方式等の候補があるが、それぞれ長所と短所があり現時点で決定的な手法はない。幅広い問題を解くことが可能であることから、ゲート方式は汎用型とも呼ばれている。

⁷ イジング・モデルとは、2次元格子点の各点に上向きまたは下向きのスピンの配置され、相互作用している設定のモデルである。

⁸ 解ける問題に限られることから、アニーリング方式は特化型とも呼ばれている。

⁹ 量子コンピュータ開発では、真空状態や極低温にして環境との相互作用や外部ノイズ等の影響を極力遮断するような工夫が行われている。

10. したがって、将来的な FTQC の実現には、上記の条件のもとで量子ビット数の拡張や量子ゲートのエラー率の改善等のハードウェア技術の進歩と誤り訂正技術の確立が重要となる。現状の NISQ デバイスは数百量子ビット程度しかなく、量子ゲートのエラー率が高いため、十分な精度での計算が難しい。このため、NISQ デバイスの量子計算を古典コンピュータによって支援する量子古典ハイブリッド型の方式の活用が検討されている。一方、アニーリング方式の研究については、実際の問題に対する応用例が見られ始めており、適用可能な問題を探索しながらノウハウの蓄積・共有が進むことが見込まれる。ただし、アニーリング方式にもハードウェア制限があること、そもそも古典コンピュータ対比での高速性について理論的保証がないことに留意が必要である。

以上、量子コンピュータの種類と課題について述べたが、改めて整理すると表 2 のとおりとなる。ゲート方式は基礎研究の段階であるものの、汎用性の高さや高速性の保証があることから、以降の節ではゲート方式を中心に説明を行っていく。また量子コンピュータと呼ぶ場合、通常はゲート方式を指していることから、本稿でも基本的にそれに倣う¹¹。

表 2 量子コンピュータの種類、課題と展望

	ゲート方式(汎用型)	アニーリング方式(特化型)
種類	<ul style="list-style-type: none"> ・(現状)誤り訂正なし、中小規模デバイス(NISQ) ・(将来)誤り訂正あり、大規模量子コンピュータ(FTQC) 	<ul style="list-style-type: none"> ・量子アニーリングマシン(D-Wave社等)
処理方式	<ul style="list-style-type: none"> ・量子ビットに対して量子ゲート操作を行い演算処理を実行 ・古典コンピュータの仕組みに近いデジタル処理 	<ul style="list-style-type: none"> ・物理エネルギーの最小化問題に帰着 ・磁場をかけて最小解を得るアナログ処理
用途/高速性	<ul style="list-style-type: none"> ・幅広い問題に対応可能(汎用性) ・特定の問題に対する高速性が保証(高速性) 	<ul style="list-style-type: none"> ・組合せ最適化問題のみ(問題特化) ・高速性の理論的な保証はない
研究段階	<ul style="list-style-type: none"> ・基礎研究(FTQCの実用化は10年以上先) →FTQCを目指しNISQ開発が進む 	<ul style="list-style-type: none"> ・応用研究(適用可能な問題を模索) →ノウハウの蓄積・共有、問題の見極めが進む
課題と展望(ハードウェア)	<ul style="list-style-type: none"> ・量子ビット数不足、量子ゲートの高エラー率 →量子ビット数の拡大、誤り訂正技術の確立 	<ul style="list-style-type: none"> ・量子ビット数不足、コヒーレンス時間や結合の制限 →量子ビット数の拡大や上記の制限の改善
課題と展望(ソフトウェア)	<ul style="list-style-type: none"> ・低レイヤ(例:アセンブリ言語)でのプログラミングが必要 →量子力学の知見が不要なソフトウェア環境の整備 	<ul style="list-style-type: none"> ・最適化の定式化の知識がプログラミングに必要 →量子力学の知見が不要なソフトウェア環境の整備

資料：先端技術ラボ [2020]、藤吉 [2022] をもとに筆者作成。

(3) 量子コンピュータの開発状況

2023年3月現在の量子コンピュータ(ゲート方式)の開発状況(図1)を見ると、ビット数が最大の量子コンピュータはIBM社が発表している433量子ビットのマシンであり、

¹⁰ 古典コンピュータの場合、0と1どちらかしか使わないため、閾値を決めて0か1に分けてしまえば演算処理に大きな問題はない。しかし、量子コンピュータの場合、0と1以外の情報、重ね合わせや干渉を利用するため、量子状態を精密に操作する必要がある。さらに量子計算の場合、途中の結果を確認するためには観測が必要であり、観測すると量子状態が壊れてしまう。このため、途中の結果を確認せずに演算を進めると、計算回数の増加によりエラーが蓄積してしまう問題が生じる。

¹¹ アニーリング方式は特定の演算処理を除き、量子アルゴリズムを実行できないため、量子コンピュータとは別の専用マシンとして整理する考え方がある。

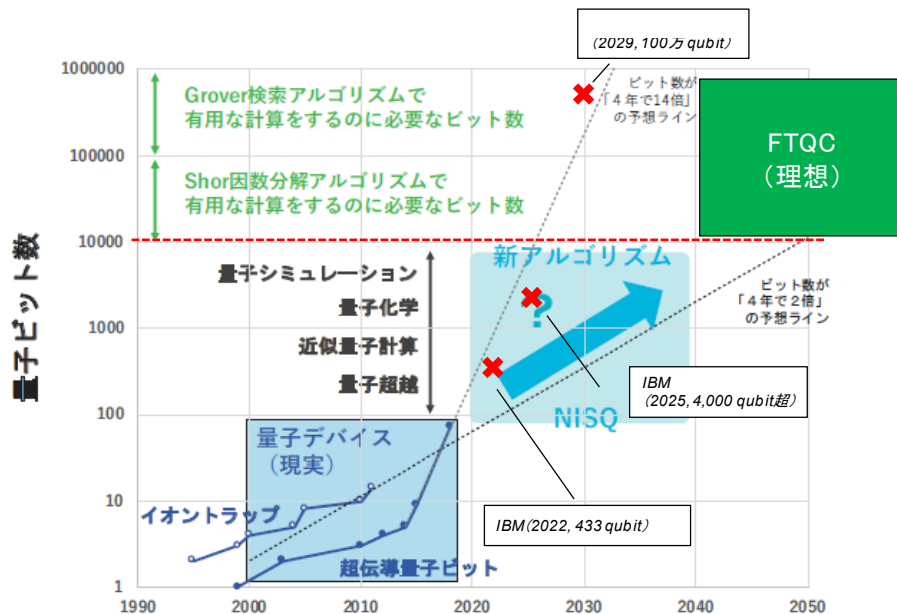
NISQ 領域に位置する。IBM 社や Google 社らのロードマップ¹²⁾のとおりに量子コンピュータの開発が進めば、2030 年代頃には量子アルゴリズムが実行可能な 10,000 量子ビットの水準 (FTQC 領域) に到達する可能性がある。ただし、FTQC 領域には段階があり、すぐに実務利用が可能になるわけではないことに留意が必要である。FTQC 領域への到達地点である 10,000 量子ビット水準は、数個から数十個の量子ビットに対する誤り訂正ができて始める段階 (例: 10 個の (論理) 量子ビットを使った、誤り訂正処理を備えた Shor のアルゴリズムが実行可能になる程度¹³⁾) に過ぎない。また前述のとおり、単なるビット数の拡張以外に量子状態を高精度に制御する技術的なハードルも存在する。

以上を踏まえると、金融業務が量子コンピュータによる計算量の優位性から来る高速化の恩恵を得るためには、技術的に高度なレベルの FTQC が実現している必要があると考えられ、実務適用可能な時期を見積もることは難しい。しかしながら、過去の歴史を踏まえると、近年、量子コンピュータの開発スピードは加速しており、ブレイクスルーによって実務利用可能な FTQC の実現が早まる可能性もある。そのため、金融実務家が量子コンピュータの仕組みや量子アルゴリズムについて、「今」理解することには意味があると考えられる。

¹²⁾ IBM 社は 2025 年までに 4,000 超の量子ビットを搭載する量子コンピュータを実現するとのロードマップを公開している (IBM [2022])。Google 社は 2029 年までに 100 万量子ビットを搭載した誤り訂正処理を持つ量子コンピュータを開発する計画である (Castellanos [2021])。

¹³⁾ 量子ビットは論理ビットと物理ビットに分けられる。論理ビットはアルゴリズムの実装に必要なビットであり、物理ビットは誤り訂正を用いて論理ビットを守りながら実行するのに必要なビットである。量子コンピュータの場合、エラー発生確率にもよるが、100~1,000 物理ビットを用いて 1 つの論理ビットを符号化するため、FTQC による量子アルゴリズムの実行には最低限 10,000 程度の量子ビット数が必要と考えられる (藤井ほか [2022])。誤り訂正処理とは、複数ビットにより冗長性を持たせて 1 つの論理ビットを表すことで、エラーが発生しても元のデータを復元できる手法である。例えば、物理ビット 3 つを用いて 1 つの論理ビットを表す (物理ビットの過半数が 1 であれば論理ビットが 1 になる)。もし、3 つのうち 1 つが 0 になっても、残り 2 つが 1 なので、論理ビットとしては 1 として扱う。いくつか量子誤り訂正方法が提案されており、Nielsen and Chuang [2010] や 嶋田・情報処理学会出版委員会 [2020] を参照されたい。

図 1 量子コンピュータ（ゲート方式）の開発ロードマップ



資料：研究開発戦略センター [2018] をもとに筆者作成。

(4) 代表的な量子アルゴリズム

本節では、代表的な4つの量子アルゴリズムの概要を紹介する¹⁴。まず、各アルゴリズムの問題設定と高速化のポイントについて説明する(表3)。

ショアの解法(Shor [1994])は、因数分解を行うアルゴリズムであり、暗号解読の分野で威力を発揮する。主流の暗号方式として使われているRSA公開鍵暗号を破る可能性があると考えられ、量子コンピュータ研究の火付け役となった。重ね合わせと干渉を用いた量子フーリエ変換によって剰余の周期を高速に発見できることを利用している。グローバーの解法(Grover [1996])は、データ検索アルゴリズムである。データを重ね合わせ状態として表現し、干渉によって対象データを絞り込むことで高速化を図っている。HHL¹⁵の解法(Harrow, Hassidim and Lloyd [2009])は、線形方程式の求解アルゴリズムである。線形方程式の求解は汎用性が高いアルゴリズムであることから、幅広い応用が期待されている。線形方程式の係数行列の固有値・固有ベクトルを高速に計算する位相推定アルゴリズム(Kitaev [1995])を利用している。量子モンテカルロ積分(QMCI: Quantum Monte-Carlo Integration) (Montanaro [2015])は、数値積分を行うアルゴリズムである。数値積分を確率

¹⁴ これまでおよそ60種類の量子アルゴリズムが考案されている。本項では因数分解やデータ検索といった応用可能性が明確になっているアルゴリズムを取り上げている。その他のアルゴリズムについてはAlgorithm Zoo (Jordan [2022])を参照されたい。

¹⁵ 考案者らの名前の頭文字を取ってHHLと呼ばれる。

振幅の推定に置き換えて、振幅推定のアルゴリズム (Brassard *et al.* [2000]) により高速化している。

次に、量子アルゴリズムの高速性について計算量の観点で解説する。計算量とは、アルゴリズムにおけるサブルーチンの呼び出し回数のオーダーであり、計算誤差や問題サイズなどの関数である。上で挙げた4つの解法をはじめとするいくつかの問題と量子アルゴリズムについては、計算量が古典アルゴリズムより少なく済むことが理論的に保証されている。例えば、HHLの解法による線形方程式求解は指数加速となる、すなわち、古典アルゴリズムの計算量が量子アルゴリズムの計算量の指数関数のオーダーで増加することが知られている。同様に、グローバーの解法や量子モンテカルロ法は、2次加速を達成する量子アルゴリズムである。すなわち、古典アルゴリズムの計算量が量子アルゴリズムの計算量の2次関数のオーダーとなることが知られている。

モンテカルロ法による数値積分を例に、より具体的な数値例を挙げる。計算誤差 ϵ 以内で積分値を推定するには、古典モンテカルロ積分の場合には計算誤差の2乗に反比例する計算量 $O(\epsilon^{-2})$ が必要となるが、量子モンテカルロ積分の場合には計算誤差に反比例する計算量 $O(\epsilon^{-1})$ で済む。具体的には、計算誤差を $\epsilon = 0.001$ とすると、古典モンテカルロ積分では1,000,000回程度の乱数生成が必要となるが、量子モンテカルロ積分では乱数生成に対応する量子ゲートを1,000回程度適用すれば良い。

表 3 代表的な量子アルゴリズム¹⁶

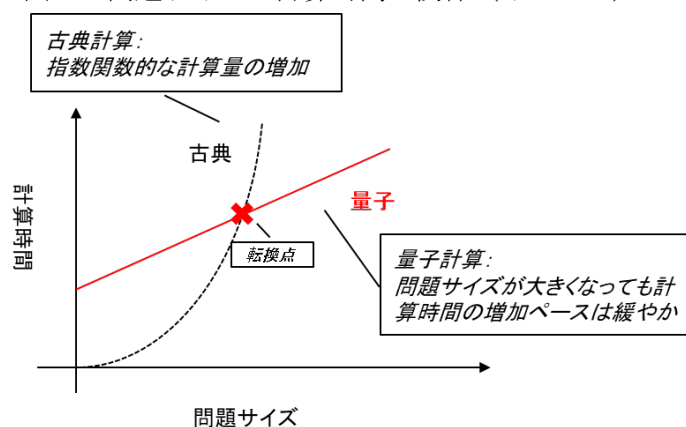
解法 (発表者・発表年)	問題	応用分野	高速化ポイント	計算量(サイズ N 、誤差 ϵ)	
				古典	量子
ショアの解法 (Shor, 1994)	因数分解	暗号解読	重ね合わせと干渉による量子フーリエ変換により剰余の周期を高速に見出す	$e^{O(N^{\frac{1}{3}} \log N^{\frac{2}{3}})}$ (一般数体篩法)	$O(N^3)$
グローバーの解法 (Grover, 1996)	データ検索	データベース検索等	データを重ね合わせて並列化し、干渉により対象を高速に絞り込んで探索	$O(N)$ (線形探索)	$O(\sqrt{N})$
量子モンテカルロ積分 (QMCI) (Brassard, 2000, Montanaro, 2015)	数値積分 (期待値計算)	ファイナンス (プライシング等)	被積分関数を確率振幅に置き換えて、高速に推定することで積分を計算	$O(\epsilon^{-2})$ (古典モンテカルロ)	$O(\epsilon^{-1})$
HHLの解法 (Harrow, Hassidim and Lloyd, 2009)	線形方程式求解	機械学習・最適化等	固有値と固有ベクトルの線形結合で表される解のベクトルを高速に計算	$O(N)$ (共役勾配法)	$O(\log N)$

このように、量子コンピュータの計算時間の増加ペースは、古典コンピュータに対して緩やかである。このため、問題サイズがある一定のサイズより大きくなる（または計算誤差が一定水準より小さくなる）と、量子コンピュータが古典コンピュータに対して、計算速度の優位性（量子加速）を示すと考えられる（図 2）。ただし、量子加速が発生する問題サイズについては留意点がある。通常、計算速度を比較するには、特定の問題に対する

¹⁶ 計算量はサブルーチンの呼び出し回数のオーダー。ショアの解法の場合は、 N 桁の整数に対する計算量（周期探索）。一般数体篩法はおおよその計算量を記載。グローバーの解法の場合は、データの個数 N に対する計算量（クエリ（データベースへのアクセス）回数）。HHLの解法の場合は、線形方程式の次元 N に対する計算量。スパース性が満たされた条件のもとでサイズ N にのみ注目して算出。QMCIの場合は、推定誤差 ϵ 以内に抑えるための計算量（関数の評価回数）。

実行時間（＝サブルーチン 1 回当たりの所要時間×サブルーチン呼び出し回数）をもとに行うべきである。しかし、現状では古典コンピュータと比較可能な量子コンピュータは実現していないため、上述のように計算量を比較して量子コンピュータの高速性を評価することが多い。ただし、量子コンピュータの場合、計算アルゴリズム以外の処理（例：データセットの準備や結果の読み取り）で計算時間が増加する可能性があるため、量子コンピュータが古典コンピュータよりも高速となる問題サイズの転換点は現時点ではよく分かっていない。

図 2 問題サイズと計算時間の関係（イメージ）



(5) 量子コンピュータの仕組み

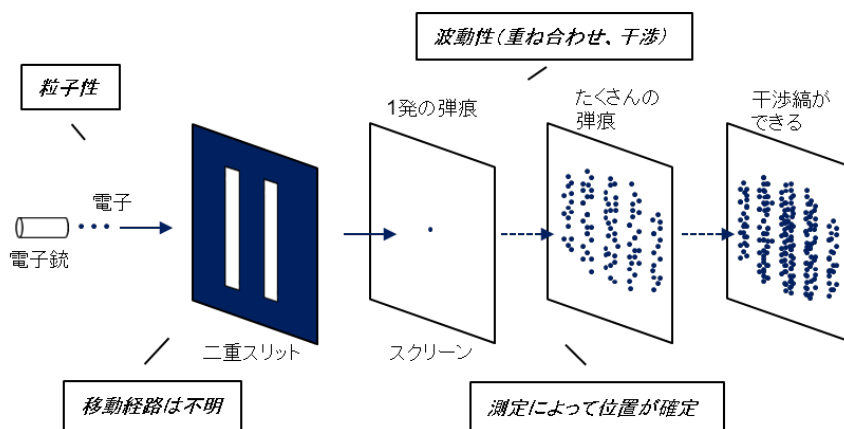
本節では、量子コンピュータの仕組みを理解するために最低限必要な範囲で、量子力学の基本と量子の性質を用いた演算方法について概説する。量子とは電子や原子等のミクロな物質の総称を意味し、量子の振る舞いを記述するための物理学が量子力学と呼ばれる。量子の性質（粒子と波動の二重性）はわかりにくいので、まず電子の二重スリット実験を通して、量子が持つ性質のイメージを説明する。次に、重ね合わせや干渉といった量子の波動性を用いた演算方法について解説する。計算が高速化される仕組みや例についても説明する。さらに量子コンピュータの論理回路（量子回路）を紹介しつつ、古典コンピュータとの共通点や違いを説明する。

イ. 粒子と波動の二重性

電子を電子銃でスクリーンに向かって発射する実験を行う（図 3）。電子銃とスクリーンとの間に、2つのスリットが入った板を置いておく。今、1個の電子をスクリーンに向かって発射すると、電子はどちらかのスリットを通過してスクリーンに衝突して1つの弾痕を残す。これを複数回繰り返すと弾痕が増えていき、最終的に濃淡のある干渉縞の分布ができる。もし電子が純粋な「粒子」なのであれば、このような縞の分布にはならず、ス

リットの2つの穴の延長線上にのみ2本の縞ができるはずである。このように多くの縞模様ができるということは、電子が単純な「粒子」ではなく「波」として2つのスリットをすり抜け、その後、互いの波が干渉し、強め合ったところに縞ができたと考えるのが自然である¹⁷。量子力学では、この現象を理解するために、「電子は確率的な振幅を持つ波として振る舞う粒子である」と考える。発射された電子は、「波」として両方のスリットを通過し、振幅を強め合い・打ち消し合って進み、スクリーンに衝突して弾痕を残す。確率的な振幅が強め合う場所は衝突回数が多く干渉縞が濃くなり、逆に打ち消し合う場所は衝突回数が少なく干渉縞が薄くなる。このように、量子は観測されるまでは波として重ね合い・干渉し合い、観測によって粒子として特定の状態に確定する性質を持つ。

図3 二重スリット実験（電子の波動性）



ロ. 量子力学を用いた計算方法

量子力学を用いた計算の考え方を説明する。説明をわかりやすくするためにスピンの概念を導入する（図4）。電子にはスピンと呼ばれる方向と大きさ（ベクトル）で表される量子の状態が存在しており¹⁸、前述の二重スリット実験の場合と同様に波動の性質がある。スピンの場合、観測するまでは、上向きと下向きのスピンの重ね合わせ状態で存在しているが、観測によってスピンの向きが上向きか下向きかどちらかに確定する。

このスピンを用いた計算の仕組みを考える。まず、情報単位の割り当てと操作が可能な仕組みを導入する。スピンに対して量子ビットと呼ばれる情報単位を割り当てる。ここでは上向きのスピンを $|0\rangle$ 、下向きのスピンを $|1\rangle$ と表記する¹⁹。観測後の量子ビットは $|0\rangle$ か

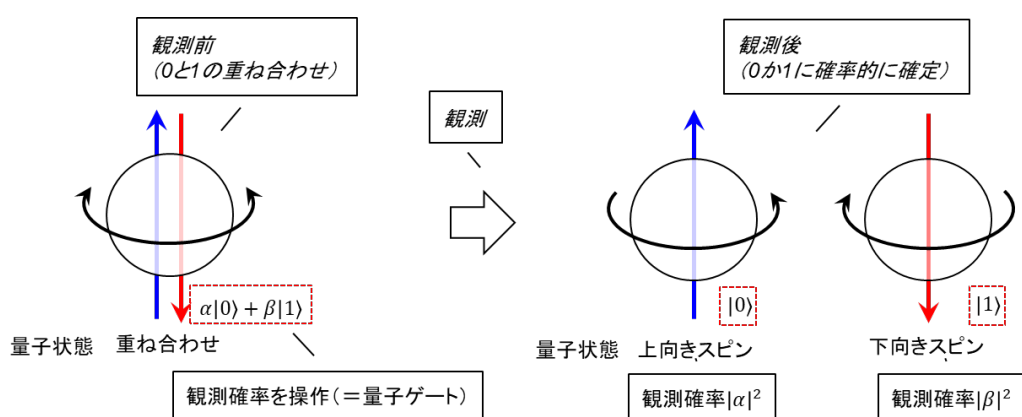
¹⁷ 片方のスリットを閉じて電子を発射すると、電子はもう片方の開いているスリットを通過するしかないが、その場合、干渉縞は出現しなくなる。様々な考察や実験がされているが、結局、発射された電子は波動のように振る舞い、二重スリットを同時に通過してスクリーンに到達したと考えるしかない。

¹⁸ 量子力学におけるスピンは、マクロな世界における物体の自転とは異なるものであるが、イメージとしては同様のものと考えて差し支えない。スピンによって磁気モーメントが発生する。

¹⁹ この記法（ブラケット記法）の定義は3節で改めて説明する。

$|1\rangle$ のどちらかであるが、観測前の量子ビットは重ね合わせ状態であるため $\alpha|0\rangle + \beta|1\rangle$ のように表現できる。ここで α, β は確率振幅と呼ばれ、 $|\alpha|^2, |\beta|^2$ は $|0\rangle, |1\rangle$ 状態の観測確率を意味する。次に量子ビットの状態に対する操作を考える。スピンの場合、外部磁場によって重ね合わせや干渉²⁰を作り出すことができ、量子状態の確率振幅 α, β を操作することで、演算処理の基本的な仕組みができる。例えば、入力状態が $|0\rangle$ の場合には、 β を増幅させて $|1\rangle$ の観測確率 $|\beta|^2$ を上げ、反対に入力状態が $|1\rangle$ の場合には、 α を増幅させて $|0\rangle$ の観測確率 $|\alpha|^2$ を上げれば $|0\rangle$ と $|1\rangle$ の切り替えができる。これは論理回路における NOT 演算に対応し、このような量子ビット操作を量子ゲートと呼ぶ。

図 4 量子（スピン）を用いた計算



ハ. 重ね合わせや干渉の利用

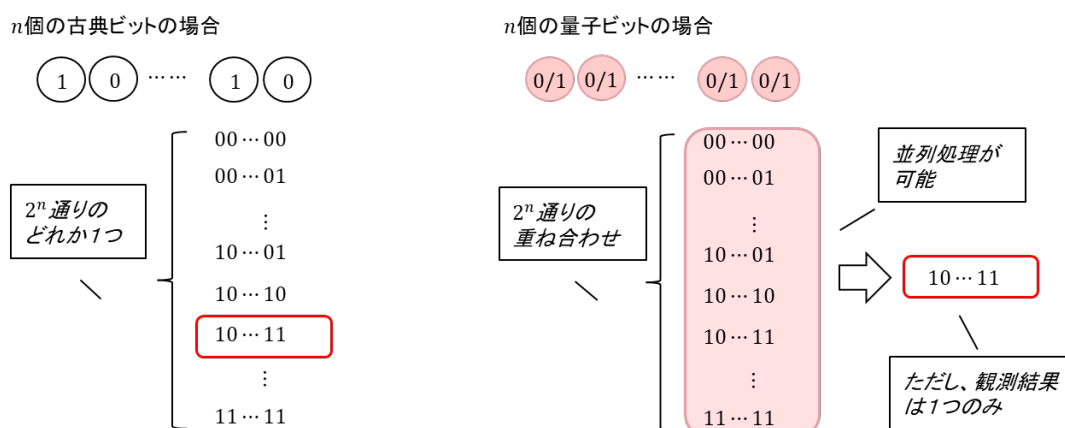
量子コンピュータのデータ構造や演算処理には、重ね合わせや干渉といった古典コンピュータにはない特徴があり、これらが量子コンピュータの高速性を生み出す理由として考えられる。具体的には、重ね合わせによって複数データを持たせて並列的な処理が可能であること、および干渉によって特定データに作用して効率的な演算ができることである。以下、具体例を挙げて説明する。

まず、古典・量子のコンピュータのデータ構造の違いについて説明する（図 5）。 n 個の古典・量子ビットの列を考える。古典コンピュータの場合、 n 個の古典ビットがあれば、 2^n 個のパターンを表現することが可能である。ただし、古典ビットは 0 か 1 のどちらかしか持てないため、1 度に持つことができるデータは 2^n 個のパターンのうち 1 つだけである。そのため、古典ビット 1 つ 1 つに対して処理を加えていく必要があり、直列的な演算方法と言える。一方、量子コンピュータの場合、 n 個の量子ビットがあれば、重ね合わせによって同時に 2^n 個のパターンを表現することが可能である。しかも、重ね合わせは、0 と 1 の両方の状態なので並列的に操作を加えることが可能である。単純に考えると、量子

²⁰ 「干渉」は、前述の二重スリット実験の例であれば、スリットの隙間の大きさやスリットとスクリーンとの位置等を変えて干渉縞の模様を変えることに対応する。

コンピュータは大量のデータに同時にアクセスできるため、計算量が少なくなると期待される。しかし、1度の観測によって得られる結果は、複数パターンのうち1つだけであるため、重ね合わせを活かして演算を行い、欲しい結果を効率的に取り出す工夫が必要となる。

図 5 古典と量子のデータの持ち方の違い



資料：武田 [2020] をもとに筆者作成。

次に、量子状態を干渉させることによる効率的な演算について、2節(4)でも紹介したグローバーのデータ検索アルゴリズムを例に説明する(図6)。まず、 N 個の未整序なデータがあり、その中に特定の「条件」を満たす“当たり”(検索対象)を1つ見つける問題を考える。古典計算のアルゴリズム(例：線形探索)の場合、1つ1つ中身を確認していくと“当たり”を見つけるのに平均 $N/2$ 回程度の試行が必要となる。一方、グローバーの検索アルゴリズムの場合、重ね合わせ状態²¹にあるデータに対して、干渉によって“当たり”のデータの測定確率を増幅させることで、平均 \sqrt{N} 回程度の試行で“当たり”を見つけることができる。

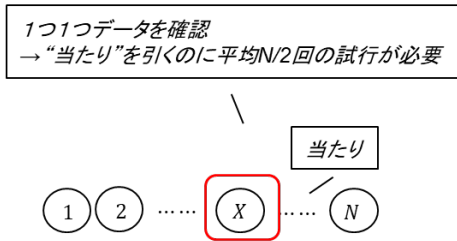
²¹ 重ね合わせ状態はアダマールゲートと呼ばれる量子ゲートを用いて生成される。1個の量子ビットにアダマールゲートを適用すると、観測確率が等しい0と1の2つの状態の重ね合わせが生成される。 n 個の量子ビットにアダマールゲートを適用すると、全体で $N (= 2^n)$ 通りの状態の重ね合わせを作ることができる。このとき、重ね合わせ状態は数学的には N 個の N 次元基底ベクトルの線形結合として表現される。アダマールゲートによる計算手順については3節(4)の計算例を参照されたい。

図 6 データ検索アプローチの違い

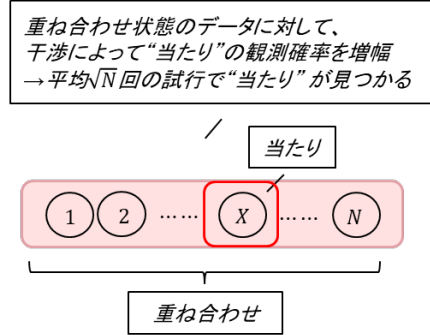
問題設定:

N 個の未整序データから“当たり”を見つけ出す。

古典アルゴリズム(線形探索)の場合:



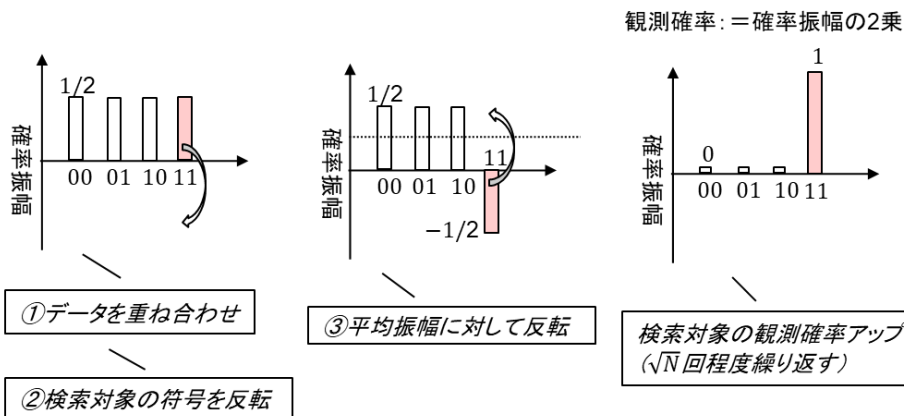
量子アルゴリズム(グローバー)の場合:



以下にグローバーの検索アルゴリズムの手順を示す(図 7)。ここでは簡単のために2量子ビットの場合のデータを考える。

- ① データの重ね合わせ状態を作成し、 $1\sim N$ のデータに対して並列的にアクセスできるようにする。
 - ② 検索対象の“当たり”(図の例では11)が満たすべき条件を持つデータに対して、確率振幅の符号を反転させる。
 - ③ 各データを振幅の平均値(1/4)に対して反転させる。
- ①と②の作業を、データのサイズに応じて \sqrt{N} 回だけ繰り返す。すると、“当たり”以外の観測確率は減り、“当たり”の確率振幅は符号が反転して増幅される(干渉)。最後に結果を観測すると“当たり”状態が高確率で得られる。

図 7 グローバーのデータ検索アルゴリズムの概要



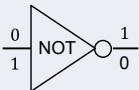

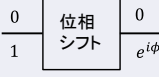
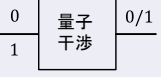
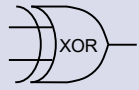
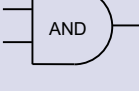
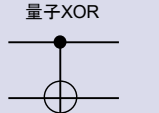
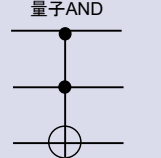
以上がグローバーのデータ検索アルゴリズムである。重ね合わせによるデータへの同時アクセス(並列処理)と、対象状態の確率振幅の増幅・打消し(干渉)によって、データ

検索を高速化していることがわかる。また量子ビットの数が増えるほど、古典アルゴリズムに対する量子アルゴリズムの高速性が顕著になる。

二. 量子回路

量子コンピュータにおける論理回路（量子回路）について説明する（表 4）。論理回路とは論理ゲート同士を線で繋いだものであり、入力出力とゲート操作の関係を視覚化できる。一般的に、古典コンピュータでは基本の論理回路（NOT、AND、OR、XOR（制御 NOT））を組み合わせることで、加減乗除や高度な数値計算等あらゆる計算（万能計算）を行うことができる。同様に、量子コンピュータにおいても基本となる論理演算を組み合わせることで量子回路を作ることによって万能計算が可能である（ソロヴェイ=キタエフの定理）。加えて、量子コンピュータの場合には、量子特有の性質（重ね合わせ・干渉）を利用した論理演算²²により、グローバルのデータ検索のような量子アルゴリズムを実行することができる。また量子回路は古典コンピュータの論理回路とは異なり可逆（例えば、演算処理の結果の数字から演算処理前の入力変数を可逆的に得ることができる）であるため、情報消去の発熱による電力消費や熱損失の問題が少ないとされる（ランダウアの原理）。

表 4 古典と量子のコンポーネントの違い

	古典計算	量子計算
情報単位	古典ビット 0 or 1	量子ビット 0/1 重ね合わせ
1入力の論理演算（ゲート）		<p>1量子ビットゲート</p>  <p>重ね合わせ・干渉</p>  
多入力の論理演算（ゲート）	 	<p>多入力量子ビットゲート 可逆回路(ユニタリ行列)</p>  

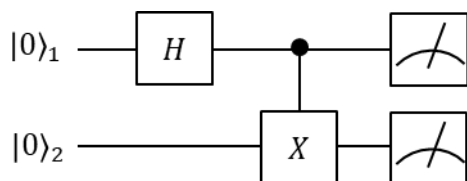
資料：武田 [2020] をもとに筆者作成。

²² 例えば、1ビット入力の論理演算として、位相シフトや重ね合わせ・干渉（例えば、アダマールゲート）による演算が可能である。位相シフトは確率振幅の位相（表 4 中の ϕ ）を変化させる演算であり、アダマールゲートは重ね合わせ状態を作り出す演算である（3 節を参照）。

量子回路の例として、エンタングルメント（量子もつれ）を作成する量子回路を取り上げて、量子回路の読み取り方を説明する。エンタングル状態とは量子ビット同士が特殊な相関関係²³を持つ状態であり、量子計算による計算の高速化と関係があると考えられている。例えば、1番目と2番目の量子ビットが共に0になる状態 $|0\rangle_1|0\rangle_2$ と、両者が共に1になる状態 $|1\rangle_1|1\rangle_2$ の重ね合わせがエンタングル状態である（3節参照）。第1量子ビットが0で測定されるときは、第2量子ビットは0に確定し、第1量子ビットが1で測定されるときは、第2量子ビットは1に確定する。つまり、片方の量子ビットの状態からもう片方の量子ビットを推定することができる。

以下の図8が量子回路である。量子回路では左から処理が始まり右で終わる。量子回路の左端が入力の量子ビット列 $|0\rangle_1, |0\rangle_2$ であり、右端が量子ビットの測定を意味する測定器の回路記号である。入力と出力の間には量子ゲートを配置し、量子ビットと作用させる量子ゲートを線で繋げる。エンタングル状態はアダマールゲートと制御NOTゲート²⁴を組み合わせて作られる。アダマールゲートはHで表している。制御NOTゲートの場合、制御ビット側に●を付し、標的ビット側にNOTゲートの回路記号Xを置いて、制御ビットの線と標的ビットの量子ゲートを線で結ぶ。図8の量子回路の処理順を説明すると、まず1番目の量子ビット $|0\rangle_1$ にアダマールゲートを作用させる。次に1番目の量子ビットを制御ビットとし、2番目の量子ビットにNOTゲートを作用させる。最後に1番目と2番目の量子ビットを測定して結果を得る。

図8 エンタングルメントを生成する量子回路



3. 量子計算の基礎²⁵

本節では、量子計算の基礎について数学的な説明を行う。量子計算では複素数のベクトルや行列による線形代数を用いる。情報の基本単位である量子ビットは、複素数ベクトルで定義され、量子ゲートによる量子ビットの操作はユニタリ行列として定義される。

²³ この性質は量子計算だけでなく、量子テレポーテーションへの応用がある。

²⁴ 制御ゲートは、制御ビットが特定の状態の場合に、標的ビットに処理を行うゲート（3節参照）。

²⁵ 3節、4節については、Nielsen and Chuang [2010]、Schuld and Petruccione [2021]、嶋田・情報処理学会出版委員会 [2020]、湊ほか [2021]、西村 [2022]、渡邊 [2021]、IBM [2017]、QunaSys [2023] を参考に記載。

(1) 量子ビット

量子計算における情報単位を量子ビットと呼び、複素数成分のベクトルを用いて表現する。古典ビットは0と1を用いて2進数で表現されるが、量子計算の場合、対応する量子ビットは、2次元の基底ベクトルを用いて以下で定義される。 $\{|0\rangle, |1\rangle\}$ は計算基底と呼ばれ、正規直交基底となる。なお、括弧 $| \)$ の表記については本節(3)にて説明する。

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (1)$$

古典ビットが0か1のどちらか片方しか一度に表現できないのに対して、量子ビットの場合は、 $|0\rangle$ と $|1\rangle$ の重ね合わせ状態が許容される。重ね合わせ状態 $|\psi\rangle$ は、数学的には以下のように複素係数 $\alpha, \beta \in \mathbb{C}$ による線形結合として表現される。

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad (2)$$

$$|\alpha|^2 + |\beta|^2 = 1. \quad (3)$$

α, β は確率振幅と呼ばれ、 $|\alpha|^2$ と $|\beta|^2$ はおのおの $|0\rangle$ と $|1\rangle$ の測定確率を意味する。そのため確率の合計値が1となるように規格化条件 $|\alpha|^2 + |\beta|^2 = 1$ を加えておく。

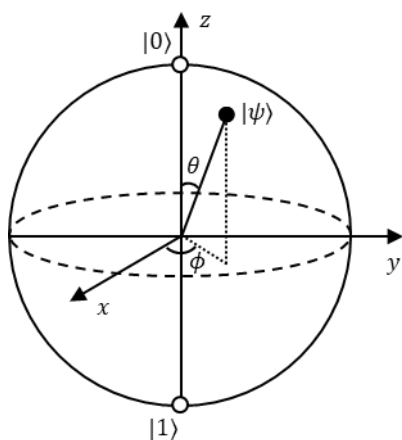
複素数成分のベクトルである量子ビットは幾何学的に視覚化して解釈することが可能である。実数パラメータ θ, ϕ_1, ϕ_2 と虚数単位 i を用いて $\alpha = e^{i\phi_1} \cos \theta, \beta = e^{i\phi_2} \sin \theta$ の変数変換を行うと、以下の数式が得られる。パラメータ ϕ は位相と呼ばれ、 $|0\rangle$ と $|1\rangle$ のずれを表現している。位相は測定確率に影響しないが、確率振幅の増幅・打消し（干渉）に影響し、パラメータ θ は量子ビットの測定確率の大きさ（確率振幅）に影響する。

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right)|1\rangle, \quad (4)$$

$$0 \leq \theta \leq \pi, 0 \leq \phi < 2\pi, \phi = \phi_2 - \phi_1. \quad (5)$$

これにより、量子ビットは実3次元空間内の単位球面上の点 $(\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$ として理解できる。この球はブロッホ球と呼ばれている（図9）。ブロッホ球面上の量子ビットの移動による量子状態の操作は、本節(4)で説明する1量子ビットゲートに対応する。

図 9 ブロッホ球



(2) 量子ゲート

量子ビットの状態操作を量子ゲートと呼び、数学的にはユニタリ行列を用いて表現する。ユニタリ行列 U とは、以下を満たす複素数成分の正方行列である。

$$U^\dagger U = U U^\dagger = I. \quad (6)$$

ここで \dagger (ダガー) は行列の共役転置、 I は単位行列とする。ユニタリ行列の定義より $U^{-1} = U^\dagger$ がわかる。またユニタリ行列による量子ビットへの作用は、 $|\psi'\rangle^2 = |U\psi\rangle^2 = |\psi\rangle^2$ となり、量子ビットのベクトルとしての大きさ (ノルム) を変えない。つまり、測定確率の合計は 1 に保たれる。

(3) ブラケット記法

量子ビットの状態や量子ゲートによる作用を簡潔に記述するためにブラケット記法を導入する。以下にブラケットの定義とブラケットを用いた計算例を示す。

- ブラとケットの定義

列ベクトルで表される量子ビットをケットと呼び、対象を $|\ \rangle$ で囲んで表記する。一方、行ベクトルで表される量子ビットをブラと呼び、対象を $\langle \ |$ で囲んで表記する。ただし、*記号は複素共役とする。

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha|0\rangle + \beta|1\rangle, \quad (7)$$

$$\langle\psi| = (|\psi\rangle)^\dagger = (\alpha^* \ \beta^*) = \alpha^*\langle 0| + \beta^*\langle 1|, \quad (8)$$

$$\alpha, \beta, \alpha^*, \beta^* \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1. \quad (9)$$

- 量子ゲートによる作用

量子ゲート U を量子ビット $|\psi\rangle$ に作用させて共役転置を取ると、量子ゲート U^\dagger を右から $\langle\psi|$ に作用させたものと同じになる。

$$|\psi'\rangle = U|\psi\rangle = \begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = \alpha'|0\rangle + \beta'|1\rangle, \quad (10)$$

$$\langle\psi'| = (U|\psi\rangle)^\dagger = \langle\psi|U^\dagger = ((\alpha')^* \quad (\beta')^*) = (\alpha')^*\langle 0| + (\beta')^*\langle 1|, \quad (11)$$

$$\alpha', \beta', (\alpha')^*, (\beta')^* \in \mathbb{C}, |\alpha'|^2 + |\beta'|^2 = 1. \quad (12)$$

- 内積

ブラとケットの順序の積（ブラケット）は、量子ビット同士の内積となる。計算例により、量子ゲートの作用によって量子ビットのノルム（測定確率の合計）は変わらないことがわかる。

$$\langle\psi|\psi\rangle = (\alpha^* \quad \beta^*) \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = |\alpha|^2 + |\beta|^2 = 1, \quad (13)$$

$$\langle\psi'|\psi'\rangle = \langle\psi|U^\dagger U|\psi\rangle = \langle\psi|\psi\rangle = 1. \quad (14)$$

- 直積（外積）

ケットとブラの順序の積（ケットブラ）は量子ビットの直積（外積）となる。これにより、量子状態の反転や射影行列²⁶を作ることができる。

$$|\psi\rangle\langle\psi| = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} (\alpha^* \quad \beta^*) = \begin{pmatrix} \alpha\alpha^* & \alpha\beta^* \\ \beta\alpha^* & \beta\beta^* \end{pmatrix}. \quad (15)$$

- 例：反転行列

量子ゲート $U_0 = 2|0\rangle\langle 0| - I$ 、 $U_1 = 2|1\rangle\langle 1| - I$ を $|\psi\rangle$ に作用させると、

$$U_0|\psi\rangle = (2|0\rangle\langle 0| - I)(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle, \quad (16)$$

$$U_1|\psi\rangle = (2|1\rangle\langle 1| - I)(\alpha|0\rangle + \beta|1\rangle) = -\alpha|0\rangle + \beta|1\rangle, \quad (17)$$

となって、 U_0, U_1 はおのおの $|1\rangle, |0\rangle$ に対して反転させる量子ゲートであることがわかる。

- 例：射影行列

基底ベクトル $|i\rangle$ の直積 $|i\rangle\langle i|, i = 0, 1$ は射影行列と呼ばれる。両側から量子ビット $|\psi\rangle$ で挟んで計算すると、量子ビットが $|i\rangle$ として測定される確率が得られる。

$$|\langle 0|\psi\rangle|^2 = \langle\psi|0\rangle\langle 0|\psi\rangle = (\alpha^* \quad \beta^*) \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = |\alpha|^2, \quad (18)$$

$$|\langle 1|\psi\rangle|^2 = \langle\psi|1\rangle\langle 1|\psi\rangle = (\alpha^* \quad \beta^*) \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = |\beta|^2. \quad (19)$$

²⁶ 正方行列 P が $P^2 = P, P^\dagger = P$ を満たす時、 P を射影行列と呼ぶ。2つ目の性質はエルミート行列を示す。

(4) 代表的な量子ゲート

量子アルゴリズムの理解に必要な量子ゲートの定義・用途や計算方法を説明する。量子ゲートは作用させる量子ビットの数で分けられる。1量子ビットに作用する場合は1量子ビットゲート、2量子ビットに同時に作用する場合は2量子ビットゲートと呼ぶ（表5, 表6）。2量子ビット以上のゲートには制御ゲートと呼ばれるゲートが存在する。制御ゲートでは、片方を制御ビット、もう片方を標的ビットとし、制御ビットの値が特定の値の場合にのみ、標的ビットに1量子ビットゲートを作用させる。なお、3量子ビット以上の任意のゲートは、1量子ビットゲートと2量子ビットゲートを組み合わせて作ることができる。

- NOTゲート（表記：X）

1量子ビットに作用し、量子ビットの $|0\rangle$ と $|1\rangle$ を切り替える基本的なゲート。ブロッホ球上のX軸180度回転に相当する。その他にY軸、Z軸を回転軸とする180度回転に相当する量子ゲートもあり、これらを合わせてパウリゲートとも呼ぶ。

（ブラケット記法による定義）

$$X|0\rangle := |1\rangle, \quad (20)$$

$$X|1\rangle := |0\rangle. \quad (21)$$

（行列成分による定義）

$$X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \quad (22)$$

（行列成分での計算確認）

$$X \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle, \quad (23)$$

$$X \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle. \quad (24)$$

- アダマールゲート（表記：H）

1量子ビットに作用し、量子ビットの $|0\rangle$ と $|1\rangle$ の重ね合わせ（以下、 $|+\rangle$, $|-\rangle$ と表記する場合もある）を作るゲート。入力量子ビットが $|1\rangle$ の場合、 $|1\rangle$ の符号（位相）が反転した重ね合わせができる。また、アダマールゲートを2回作用させると、重ね合わせが消えて元の量子ビットに戻る。

（ブラケット記法による定義）

$$H|0\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle, \quad (25)$$

$$H|1\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle. \quad (26)$$

(行列成分による定義)

$$H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (27)$$

(行列成分での計算確認)

$$H \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad (28)$$

$$H \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \quad (29)$$

● 位相シフトゲート (表記: $P(\phi)$)

1量子ビットに作用し、位相 ϕ をシフト (Z軸を軸に ϕ 回転) させるゲート (図 10)。量子ビット $|0\rangle$ と $|1\rangle$ の間に位相 ϕ を作ることができる。位相 $\phi = \pi/2, \pi/4$ の場合、それぞれS,Tゲートと呼ぶ。

(ブラケット記法による定義)

$$P(\phi)|0\rangle := |0\rangle, \quad (30)$$

$$P(\phi)|1\rangle := e^{i\phi}|1\rangle. \quad (31)$$

(行列成分による定義)

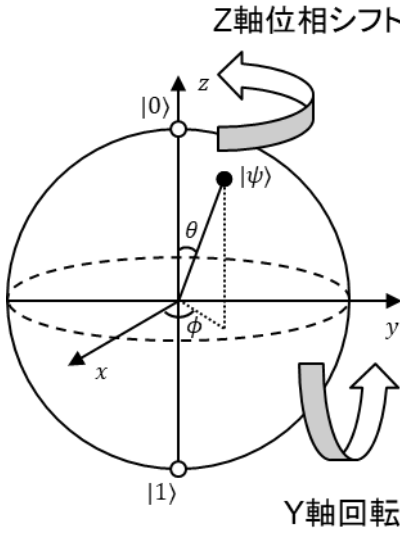
$$P(\phi) := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}. \quad (32)$$

(行列成分での計算確認)

$$P(\phi) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle, \quad (33)$$

$$P(\phi) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = e^{i\phi} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = e^{i\phi}|1\rangle. \quad (34)$$

図 10 位相シフト・回転ゲートのイメージ



● 回転ゲート (表記: $R(\theta)$)

1 量子ビットに作用し、ブロッホ球上の軸の周りに θ 回転させるゲート (図 10)。Y 軸上の回転ゲートの場合、量子ビット $|0\rangle$ と $|1\rangle$ の間の振幅の大きさを変化させることができる。その他には X 軸や Z 軸を回転軸とする回転ゲートが存在する。以下の定義は Y 軸回転ゲートのもの。

(ブラケット記法による定義)

$$R(\theta)|0\rangle := \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)|1\rangle, \quad (35)$$

$$R(\theta)|1\rangle := -\sin\left(\frac{\theta}{2}\right)|0\rangle + \cos\left(\frac{\theta}{2}\right)|1\rangle. \quad (36)$$

(行列成分による定義)

$$R(\theta) := \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \quad (37)$$

(行列成分での計算確認)

$$R(\theta) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) \end{pmatrix} = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)|1\rangle, \quad (38)$$

$$R(\theta) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\sin\left(\frac{\theta}{2}\right) \\ \cos\left(\frac{\theta}{2}\right) \end{pmatrix} = -\sin\left(\frac{\theta}{2}\right)|0\rangle + \cos\left(\frac{\theta}{2}\right)|1\rangle. \quad (39)$$

- 制御 NOT ゲート (表記 : CNOT)

制御ビットが 1 の時にのみ、標的ビットに NOT ゲートを作用させる制御ゲート。1 番目を制御ビット、2 番目を標的ビットとする制御 NOT ゲート $\text{CNOT}_{1,2}$ は以下のとおり。2 量子ビットはテンソル積²⁷を用いて 4 次元の基底ベクトルとして表現する。なお、テンソル積の記号 \otimes は明示する必要がない場合は省略される。

(ブラケット記法による定義)

$$\begin{aligned}\text{CNOT}_{1,2}|0\rangle_1 \otimes |0\rangle_2 &:= |0\rangle_1 \otimes |0\rangle_2, \\ \text{CNOT}_{1,2}|0\rangle_1 \otimes |1\rangle_2 &:= |0\rangle_1 \otimes |1\rangle_2, \\ \text{CNOT}_{1,2}|1\rangle_1 \otimes |0\rangle_2 &:= |1\rangle_1 \otimes |1\rangle_2, \\ \text{CNOT}_{1,2}|1\rangle_1 \otimes |1\rangle_2 &:= |1\rangle_1 \otimes |0\rangle_2\end{aligned}\quad (40)$$

$$|0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.\quad (41)$$

(行列成分による定義)

$$\text{CNOT}_{1,2} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.\quad (42)$$

(行列成分での計算確認)

$$\text{CNOT}_{1,2}|1\rangle_1|1\rangle_2 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |1\rangle_1|0\rangle_2.\quad (43)$$

- 制御位相シフトゲート (表記 : CP)

制御ビットが 1、標的ビットが 1 の時にのみ位相をシフトさせるゲート。1 番目を制御ビット、2 番目を標的ビットとする位相 ϕ のシフトゲート $\text{CP}_{1,2}(\phi)$ は以下のとおり。

(ブラケット記法による定義)

$$\begin{aligned}\text{CP}_{1,2}(\phi)|0\rangle_1|0\rangle_2 &:= |0\rangle_1|0\rangle_2, \\ \text{CP}_{1,2}(\phi)|0\rangle_1|1\rangle_2 &:= |0\rangle_1|1\rangle_2, \\ \text{CP}_{1,2}(\phi)|1\rangle_1|0\rangle_2 &:= |1\rangle_1|0\rangle_2, \\ \text{CP}_{1,2}(\phi)|1\rangle_1|1\rangle_2 &:= e^{i\phi}|1\rangle_1|1\rangle_2.\end{aligned}\quad (44)$$

²⁷ 量子ビット $|\psi\rangle, |\phi\rangle$ のテンソル積 $|\psi\rangle \otimes |\phi\rangle$ は以下で定義される。定義より、ベクトルの成分同士を掛け合わせた値を成分とするベクトルを作る操作であるとわかる。

$$|\psi\rangle = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = a_0|0\rangle + a_1|1\rangle, \quad |\phi\rangle = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = b_0|0\rangle + b_1|1\rangle, \quad |\psi\rangle \otimes |\phi\rangle := \begin{pmatrix} a_0|\phi\rangle \\ a_1|\phi\rangle \end{pmatrix} := \begin{pmatrix} a_0b_0 \\ a_0b_1 \\ a_1b_0 \\ a_1b_1 \end{pmatrix}$$

(行列成分による定義)

$$CP_{1,2}(\phi) := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix}. \quad (45)$$

(行列成分での計算確認)

$$CP_{1,2}(\phi)|1\rangle_1|1\rangle_2 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ e^{i\phi} \end{pmatrix} = e^{i\phi}|1\rangle_1|1\rangle_2. \quad (46)$$

- 例：重ね合わせ

2量子ビットのテンソル積に対して、アダマールゲートのテンソル積²⁸を作用させると、2量子ビットの状態の重ね合わせを作成できる。この例では、2量子ビットの重ね合わせによって、10進数表示で0から3までを同時に表現している。

$$\begin{aligned} H^{\otimes 2}|0\rangle^{\otimes 2} &:= (H \otimes H)(|0\rangle \otimes |0\rangle) = H|0\rangle \otimes H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \frac{1}{2}(|0\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle) \\ &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle). \end{aligned} \quad (47)$$

- 例：エンタングルメント

1番目の量子ビットに対してアダマールゲートを適用し、1番目を制御ビット、2番目を標的ビットとするCNOTゲートを適用することでエンタングル状態を作り出せる。

$$\begin{aligned} &(|0\rangle_1 \otimes |0\rangle_2) \xrightarrow{H \otimes I} H|0\rangle_1 \otimes I|0\rangle_2 \\ &= \frac{1}{\sqrt{2}}(|0\rangle_1 + |1\rangle_1) \otimes |0\rangle_2 = \frac{1}{\sqrt{2}}(|0\rangle_1|0\rangle_2 + |1\rangle_1|0\rangle_2) \\ &\xrightarrow{CNOT_{1,2}} \frac{1}{\sqrt{2}}[CNOT_{1,2}(|0\rangle_1|0\rangle_2) + CNOT_{1,2}(|1\rangle_1|0\rangle_2)] = \frac{1}{\sqrt{2}}(|0\rangle_1|0\rangle_2 + |1\rangle_1|1\rangle_2). \end{aligned} \quad (48)$$

²⁸ 量子ゲート U, V のテンソル積 $U \otimes V$ の量子ビットのテンソル積 $|\psi\rangle \otimes |\phi\rangle$ に対する作用は

$$(U \otimes V)(|\psi\rangle \otimes |\phi\rangle) := U|\psi\rangle \otimes V|\phi\rangle$$

で定義される。量子ゲートのテンソル積とは、個別の量子ゲートを量子ビットに作用させてからテンソル積を取る操作に対応する。量子ゲート U, V のテンソル積 $U \otimes V$ の定義を行列成分で確認すると、元の行列の成分同士を掛け合わせて成分とする行列を作る操作であることがわかる。



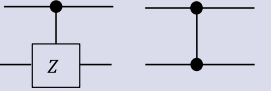
$$U = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}, V = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix}$$

$$U \otimes V := \begin{pmatrix} u_{11}V & u_{12}V \\ u_{21}V & u_{22}V \end{pmatrix} := \begin{pmatrix} u_{11}v_{11} & u_{11}v_{12} & u_{12}v_{11} & u_{12}v_{12} \\ u_{11}v_{21} & u_{11}v_{22} & u_{12}v_{21} & u_{12}v_{22} \\ u_{21}v_{11} & u_{21}v_{12} & u_{22}v_{11} & u_{22}v_{12} \\ u_{21}v_{21} & u_{21}v_{22} & u_{22}v_{21} & u_{22}v_{22} \end{pmatrix}$$

表 5 代表的な 1 量子ビットゲート

ゲート名称/回路記号	意味	ブラケット表示	行列表示	
パウリ	X, σ_x NOT	ビット反転 (X軸 π 回転)	$ 0\rangle \rightarrow 1\rangle$ $ 1\rangle \rightarrow 0\rangle$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
	Y, σ_y	位相・ビット反転 (Y軸 π 回転)	$ 0\rangle \rightarrow i 1\rangle$ $ 1\rangle \rightarrow -i 0\rangle$	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
	Z, σ_z	位相反転 (Z軸 π 回転)	$ 0\rangle \rightarrow 0\rangle$ $ 1\rangle \rightarrow - 1\rangle$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
アダマール	H	重ね合わせ (X-Z軸 $\pi/4$ 回転)	$ 0\rangle \rightarrow +\rangle = (0\rangle + 1\rangle)/\sqrt{2}$ $ 1\rangle \rightarrow -\rangle = (0\rangle - 1\rangle)/\sqrt{2}$	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
位相シフト	S	位相シフト (Z軸 $\pi/2$ 回転)	$ 0\rangle \rightarrow 0\rangle$ $ 1\rangle \rightarrow i 1\rangle$	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
	T	位相シフト (Z軸 $\pi/4$ 回転)	$ 0\rangle \rightarrow 0\rangle$ $ 1\rangle \rightarrow e^{i\pi/4} 1\rangle$	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$
	$P(\phi)$	位相シフト (Z軸 ϕ 回転)	$ 0\rangle \rightarrow 0\rangle$ $ 1\rangle \rightarrow e^{i\phi} 1\rangle$	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$
回転	$R_x(\theta)$	X軸 θ 回転	$ 0\rangle \rightarrow \cos \theta/2 0\rangle - i \sin \theta/2 1\rangle$ $ 1\rangle \rightarrow -i \sin \theta/2 0\rangle + \cos \theta/2 1\rangle$	$\begin{pmatrix} \cos \theta/2 & -i \sin \theta/2 \\ -i \sin \theta/2 & \cos \theta/2 \end{pmatrix}$
	$R_y(\theta)$	Y軸 θ 回転	$ 0\rangle \rightarrow \cos \theta/2 0\rangle + \sin \theta/2 1\rangle$ $ 1\rangle \rightarrow -\sin \theta/2 0\rangle + \cos \theta/2 1\rangle$	$\begin{pmatrix} \cos \theta/2 & -\sin \theta/2 \\ \sin \theta/2 & \cos \theta/2 \end{pmatrix}$
	$R_z(\theta)$	Z軸 θ 回転	$ 0\rangle \rightarrow e^{-i\theta/2} 0\rangle$ $ 1\rangle \rightarrow e^{i\theta/2} 1\rangle$	$\begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$
ユニバーサル	$u(\theta, \phi, \lambda)$	任意操作	$ 0\rangle \rightarrow \cos \theta/2 0\rangle + e^{i\phi} \sin \theta/2 1\rangle$ $ 1\rangle \rightarrow -e^{i\lambda} \sin \theta/2 0\rangle + e^{i(\lambda+\phi)} \cos \theta/2 1\rangle$	$\begin{pmatrix} \cos \theta/2 & -e^{i\lambda} \sin \theta/2 \\ e^{i\phi} \sin \theta/2 & e^{i(\lambda+\phi)} \cos \theta/2 \end{pmatrix}$

表 6 代表的な 2 量子ビットゲート

ゲート名称/回路記号	意味	ブラケット表示 (制御ビット(左)、対象ビット(右))	行列表示	
CNOT		制御量子ビットが1のとき 対象量子ビットを反転	$ 0\rangle \otimes 0\rangle \rightarrow 0\rangle \otimes 0\rangle$ $ 0\rangle \otimes 1\rangle \rightarrow 0\rangle \otimes 1\rangle$ $ 1\rangle \otimes 0\rangle \rightarrow 1\rangle \otimes 1\rangle$ $ 1\rangle \otimes 1\rangle \rightarrow 1\rangle \otimes 0\rangle$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
SWAP		2つの量子ビットを交換	$ 0\rangle \otimes 0\rangle \rightarrow 0\rangle \otimes 0\rangle$ $ 0\rangle \otimes 1\rangle \rightarrow 1\rangle \otimes 0\rangle$ $ 1\rangle \otimes 0\rangle \rightarrow 0\rangle \otimes 1\rangle$ $ 1\rangle \otimes 1\rangle \rightarrow 1\rangle \otimes 1\rangle$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
制御Z (CZ)		制御量子ビットが1のとき 対象量子ビットを位相反転	$ 0\rangle \otimes 0\rangle \rightarrow 0\rangle \otimes 0\rangle$ $ 0\rangle \otimes 1\rangle \rightarrow 0\rangle \otimes 1\rangle$ $ 1\rangle \otimes 0\rangle \rightarrow 1\rangle \otimes 0\rangle$ $ 1\rangle \otimes 1\rangle \rightarrow 1\rangle \otimes - 1\rangle$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$

4. 量子アルゴリズム

本節では、ファイナンス分野の問題に用いられる主要な量子アルゴリズムについて、量子コンピュータの種類 (FTQC、NISQ) に分けて説明していく。量子計算のファイナンスへの応用研究には、FTQC 上でのアルゴリズムの実行を前提としたものが多いため、まずは FTQC 向けの量子アルゴリズムの説明を行う。その後、NISQ デバイス上での量子アルゴリズムである量子古典ハイブリッド計算について説明する。

(1) FTQC アルゴリズム

FTQC 向けのアルゴリズムを、サブルーチンとして用いられる基本アルゴリズムと、具体的な問題を解くための応用アルゴリズムに分けて考える。最初に各アルゴリズムについて概説し、その後で個別アルゴリズムについての詳細な説明を行う。

まず基本アルゴリズムとして、量子フーリエ変換（QFT：Quantum Fourier Transform）、量子位相推定（QPE：Quantum Phase Estimation）、量子振幅増幅（QAA：Quantum Amplitude Amplification）、量子振幅推定（QAE：Quantum Amplitude Estimation）がある（表 7）。これらのアルゴリズムは着目する対象と目的によって分けられる。QFT と QPE では位相に着目する。QFT は量子ビットの値を、位相の異なる確率振幅の重ね合わせに変換する。QPE は、量子ゲートの固有値に含まれる位相を量子ビットの値に変換して推定する。一方、QAA と QAE では振幅に着目する。QAA は特定の量子状態の確率振幅を増幅させて測定確率を高めることを目的とし、QAE は量子状態の確率振幅を推定するのが目的である。これらのアルゴリズムの一般的な実装方法を説明する。

表 7 基本アルゴリズム一覧²⁹

名称 (略称)	概要	応用先 アルゴリズム例	計算量 (サイズ N 、誤差 ϵ)
量子フーリエ変換 (QFT)	QFTは離散フーリエ変換(DFT)の量子版。 入力ビット値を位相と見做して複数の波(確率振幅)に分解。逆変換により、波の重ね合わせから、位相をビット値として抽出できる。	QPE	$O((\log N)^2)$
量子位相推定 (QPE)	量子ゲートの固有値内の位相の推定。 逆QFTによって、固有値(=確率振幅)内の位相を、量子ビットの値として出力する。	ショアの解法、 HHL、 QAE	$O((\log N)^2)$
量子振幅増幅 (QAA)	重ね合わせ状態の中の望みの状態の確率振幅を選択的に大きくして測定確率を高くする。	グローバーの解法、 QAE	$O(\sqrt{N})$
量子振幅推定 (QAE)	量子ビット列の確率振幅の推定。 QAAとQPEを組み合わせて、確率振幅の位相パラメータを量子ビットの値として出力する。	QMCI	$O(\epsilon^{-1})$

次に応用アルゴリズムとして、量子モンテカルロ積分（QMCI：Quantum Monte Carlo Integration）と線形方程式の求解アルゴリズム（HHL：Harrow Hassidim Lloyd）を取り扱う（表 8）。通常のモンテカルロ法による積分は、乱数を発生させて何らかの関数を計算し、それらの期待値を取って数値積分を行うシミュレーション方法のことである。本稿では古典モンテカルロ法等と便宜的に呼んでいる。一方、量子アルゴリズムのQMCIは、被積分関数が符号化された確率振幅を、QAEにより推計して計算するアルゴリズムである。

²⁹ 問題サイズ N と誤差 ϵ として計算量を記載。QPEに必要な計算量はQFTの計算量を記載。ただし、追加で制御ゲートの呼出しの計算量が必要になる。QAAにはグローバーの解法の計算量を記載。QAEには推定対象を生成するゲートの呼出しの計算量を記載。QAEとQMCIの推定部分の計算量は同じ。

乱数生成と重ね合わせ生成の類似より量子モンテカルロ積分と呼ばれる³⁰。プライシングやリスク管理等のファイナンス問題への応用研究としては、古典モンテカルロ法を適用する箇所（例：オプション・プライシングにおける期待値計算）に対して、QAE や QMCI が適用されることが多い。HHL は、線形方程式の係数行列からなる量子ゲートと QPE を用いて固有値問題を解き、固有値と固有ベクトルの線形結合として線形方程式の解を与える。HHL については、線形方程式の求解問題の汎用性が高いため、ファイナンス問題だけでなく機械学習タスクへの応用が検討されている（Duan *et al.* [2020]）。

表 8 応用アルゴリズム

名称 (略称)	概要	応用先	計算量(サイズ N , 誤差 ϵ)	
			古典	量子
量子モンテカルロ積分 (QMCI)	数値積分の計算アルゴリズム。 被積分関数を確率振幅として符号化し、 QAEにより確率振幅を推定して計算。	プライシング、 リスク計測等	$O(\epsilon^{-2})$ (古典モンテカルロ)	$O(\epsilon^{-1})$ (QMCI)
HHL	線形方程式の解法アルゴリズム。解のベクトルが固有値、固有ベクトルの線形結合となるように、QPEと回転ゲートを用いて計算。	ポートフォリオ最適化、 プライシング、機械学習	$O(N)$ (共役勾配法)	$O(\log N)$ (HHL)

イ. 量子フーリエ変換(QFT)

量子フーリエ変換 (QFT) は、離散フーリエ変換 (DFT : Discrete Fourier Transform) の量子版に相当し、量子ビットの値を位相情報に変換させる量子アルゴリズムである。QFT の定義より、QFT を入力量子ビット $|j\rangle$ に作用させると、異なる位相を持つ確率振幅の重ね合わせ状態として出力されることがわかる。数式を展開することで重ね合わせ状態の量子ビット同士の積の形に分解することもできる。これにより、各量子ビットに対して制御位相シフトゲートを適用すれば、QFT が実装できることがわかる。ただし、整数 j を2進数表現 (0/1) に展開している。

$$\text{QFT}|j\rangle := \sum_{k=0}^{2^n-1} \frac{1}{\sqrt{2^n}} \exp\left(\frac{i2\pi kj}{2^n}\right) |k\rangle \quad (49)$$

$$= \frac{1}{\sqrt{2^n}} (|0\rangle + e^{i2\pi 0 \cdot j_n} |1\rangle)(|0\rangle + e^{i2\pi 0 \cdot j_{n-1} j_n} |1\rangle) \dots (|0\rangle + e^{i2\pi 0 \cdot j_1 \dots j_n} |1\rangle) \quad (50)$$

QFT の逆変換 (逆 QFT) を行うことで、確率振幅に含まれる位相情報を量子ビットの値に変換することが可能である。なお、後述する量子位相推定において、逆 QFT は、量子ゲートの固有値を量子ビットの値として取り出すサブルーチンとして用いられる。このため、ここで、逆 QFT の定義を示しておく。 j が2進数で $x = j$ の場合、 $|x\rangle$ を測定すれば確率1で所望の結果が得られる。

³⁰ QAE と QMCI は中核となるアルゴリズムが似ているため同一視されることもある。

$$\text{QFT}^{-1}\left(\sum_{k=0}^{2^n-1} \frac{1}{\sqrt{2^n}} \exp\left(\frac{i2\pi kj}{2^n}\right) |k\rangle\right) := \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \sum_{x=0}^{2^n-1} \exp\left(-\frac{i2\pi k}{2^n}(x-j)\right) |x\rangle \approx |j\rangle. \quad (51)$$

QFT の計算量について説明する。 N 個のサンプルに対して、DFT の計算量は $O(N \log N)$ である。一方、QFT の場合、 n 個の量子ビットに対して計算量は $O(n^2)$ となる。量子の場合、重ね合わせによって $N = 2^n$ とできるため、サンプル数 N で揃えると計算量は $O((\log N)^2)$ となる。 $\log N$ の分だけ計算量が少ないため指数加速になる。ただし、QFT の結果はベクトルであるため、すべての係数を正確に取り出そうとすると、指数関数的に計算量が増えてしまう点には注意が必要である。そのため、確率振幅が最も増幅されるピークの位置のみを知りたい場合に QFT および逆 QFT を用いるのが良い³¹。

ロ. 量子位相推定 (QPE)

量子位相推定 (QPE) とは、量子ゲートの固有値に含まれる位相情報を推定するアルゴリズムである。線形代数の知識より、ユニタリ行列 (量子ゲート) U を固有ベクトル (量子ビット) $|\psi\rangle$ に作用させると、以下のように固有値 $e^{i2\pi\phi}$ が出力される³²。この固有値の位相 ϕ を近似的に求めるのが QPE である。

$$U|\psi\rangle = e^{i2\pi\phi}|\psi\rangle, \quad (52)$$

$$0 \leq \phi < 1, \quad (53)$$

$$\phi = 0.\phi_1\cdots\phi_n. \quad (54)$$

QPE の考え方を説明する。位相 ϕ を量子ビットの値に変換して出力できれば、位相 ϕ を推定できたことになる。その変換手段として、前項で説明した逆 QFT が利用できる。位相 ϕ を含む確率振幅の重ね合わせ状態を、制御ゲートで作り出しておき、逆 QFT を用いて位相 ϕ を量子ビット列に出力して推定するのである。

$$\text{QFT}^{-1}\left(\sum_{k=0}^{2^n-1} \frac{1}{\sqrt{2^n}} \exp(i2\pi k\phi) |k\rangle\right) = |\phi\rangle. \quad (55)$$

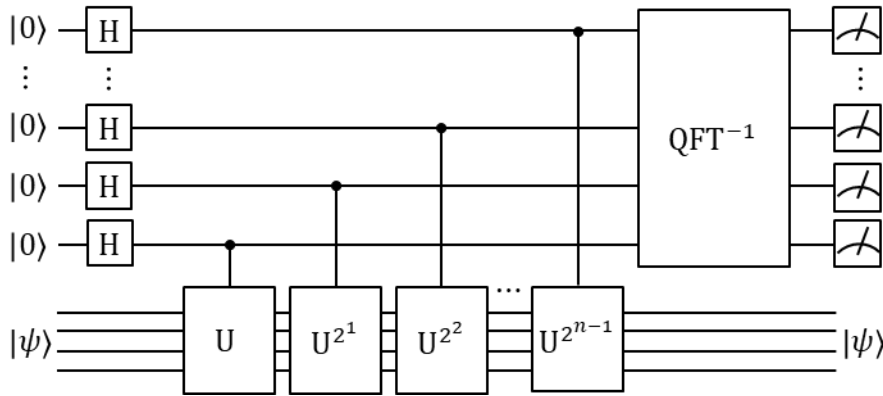
QPE のアルゴリズムを説明する (図 11)。 $|\psi\rangle$ を固有ベクトルの量子ビット列、 $|0\rangle^{\otimes n}$ を補助量子ビット列とする。まず補助量子ビット列 $|0\rangle^{\otimes n}$ に対してアダマールゲート $H^{\otimes n}$ を作用させて重ね合わせ状態を作成する。次に重ね合わせ状態の値 $|k\rangle$ を制御ビットとする制御量子ゲート CU^k を、固有ベクトルの量子ビット列 $|\psi\rangle$ に作用させる。これにより、位相 ϕ を含む確率振幅の重ね合わせ状態を作り出すことができる。最後に逆 QFT を補助量子ビット列に作用させると、位相が量子ビット列 $|\phi\rangle$ として得られる。

³¹ 後述する QPE では、量子ゲートの固有値 (位相) を逆 QFT 結果のピークとして出力している。その他の例として、ショアの解法では、QFT 結果のピーク位置から剰余関数の周期を効率的に求めている。

³² ユニタリ行列の固有値は複素数平面の単位円上の点となるため、必ずこの形 $e^{i2\pi\phi}$ で表される。

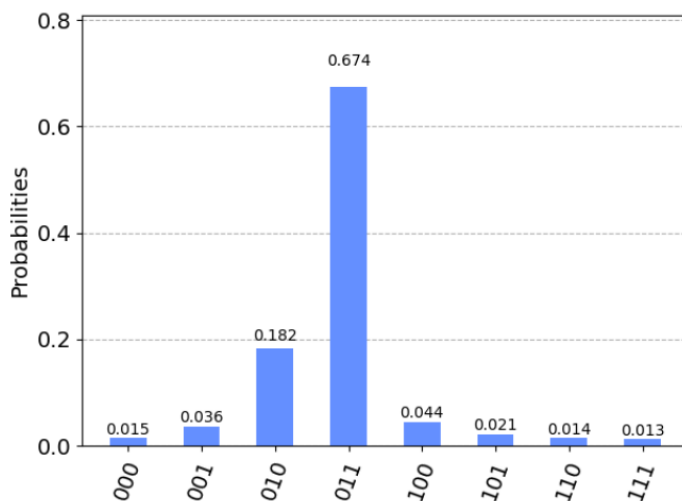
$$|0\rangle^{\otimes n}|\psi\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle|\psi\rangle \xrightarrow{CU^k} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp(i2\pi k\phi) |k\rangle|\psi\rangle \xrightarrow{QFT^{-1}} |\phi\rangle|\psi\rangle \quad (56)$$

図 11 QPE アルゴリズムの量子回路



シミュレータによる QPE アルゴリズムの実行例を示す (図 12)。 $U|\psi\rangle = e^{2i\pi\phi}|\psi\rangle$ を満たす固有値の位相 ϕ を QPE によって推定する。シミュレーション条件として、出力先の量子ビット数を 3、位相 ϕ を $1/3 = 0.333 \dots$ 、固有状態 $|\psi\rangle$ を $|1\rangle$ とする。QPE の実行結果は以下の図 12 のとおり。横軸に量子ビット列の値 (2 進数)、縦軸に測定確率をプロットしている。結果より、011 (2 進数) に測定確率のピークがある。011 は、逆 QFT の式と 10 進数表示により $\hat{\phi} = 0.375 (= 3/2^3)$ であり、少数の量子ビットでも、真の値 $\phi = 0.333 \dots$ をある程度近似できていることがわかる。量子ビット数を増やしていけば真の値に近づいていく。

図 12 QPE アルゴリズムの実行結果 (量子ビット数=3、位相 $\theta = 1/3$)



資料：量子計算のシミュレータ Qiskit (IBM [2017]) を用いて筆者作成。

ハ. 量子振幅増幅(QAA)

量子振幅増幅 (QAA) は、重ね合わせ状態にある特定の状態の確率振幅を、選択的に増幅させるアルゴリズムである。QAA はグローバーのデータ検索の一般化にあたる。

QAA の考え方とアルゴリズムを説明する。まずアルゴリズムの入力対象の量子状態 $|\psi\rangle$ を用意しておく。この量子状態は量子ビット列 $|0\rangle^n$ に何等かの量子ゲート A を作用させて作成し、直交基底 $|\tilde{\psi}_0\rangle, |\tilde{\psi}_1\rangle$ の線形結合によって表現されているものとする。3節にて量子ビットの状態をブロッホ球上の点とみなしたのと同様に、 $|\psi\rangle$ の確率振幅をパラメータ θ, ϕ の関数としてみなすと、以下のように表現できる。

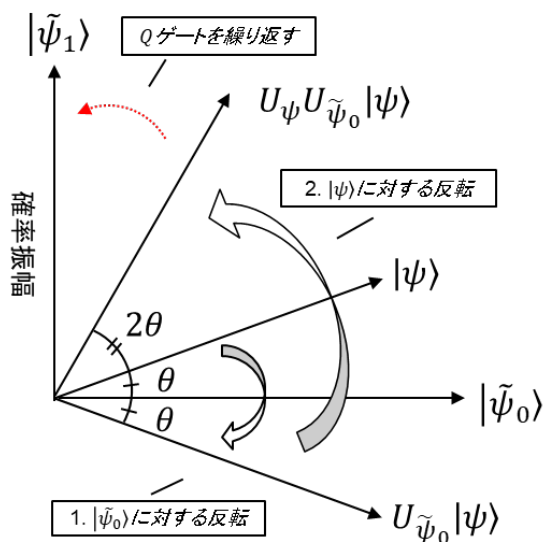
$$|\psi\rangle = A|0\rangle^n = \cos\theta |\tilde{\psi}_0\rangle + e^{i\phi} \sin\theta |\tilde{\psi}_1\rangle. \quad (57)$$

QAA の目的は $|\tilde{\psi}_1\rangle$ の確率振幅 $\sin\theta$ を変化させて、 $|\tilde{\psi}_1\rangle$ の測定確率を高めることである。そのため、QAA では次の2つのステップを繰り返して量子状態の回転を行う。

- ① $|\psi\rangle$ を $|\tilde{\psi}_0\rangle$ 軸に対して反転させる。(= $U_{\tilde{\psi}_0}|\psi\rangle$)
- ② $U_{\tilde{\psi}_0}|\psi\rangle$ を $|\psi\rangle$ 軸に対して反転させる。(= $U_\psi U_{\tilde{\psi}_0}|\psi\rangle$)

ステップ①②により、量子状態 $|\psi\rangle$ が $|\tilde{\psi}_1\rangle$ 方向に 2θ 回転することがわかる。このステップを適切な回数だけ繰り返して、 $|\tilde{\psi}_1\rangle$ の確率振幅を大きくするアルゴリズムが QAA である (図 13)。

図 13 QAA アルゴリズムの幾何学的イメージ



次にステップ①②で作用させた量子ゲート $U_{\tilde{\psi}_0}, U_\psi$ の具体的な形を以下に示す。定義より $U_{\tilde{\psi}_0}$ と U_ψ は、それぞれ、 $|\tilde{\psi}_0\rangle$ 軸と $|\psi\rangle$ 軸に対する反転ゲートであることがわかる。なお、 U_ψ と $U_{\tilde{\psi}_0}$ の積 $Q = U_\psi U_{\tilde{\psi}_0}$ をまとめたものをグローバー演算子と呼ぶ。

$$U_{\tilde{\psi}_0} := 2|\tilde{\psi}_0\rangle\langle\tilde{\psi}_0| - I, \quad (58)$$

$$U_\psi := 2|\psi\rangle\langle\psi| - I = -A(I - 2|0\rangle\langle 0|)A^\dagger. \quad (59)$$

U_ψ と $U_{\tilde{\psi}_0}$ の定義から $Q = U_\psi U_{\tilde{\psi}_0}$ を計算すると、量子ゲート Q は以下の回転行列であることがわかる。 $|\psi_1\rangle$ の確率振幅が最大となるのは、 $\sin(2k+1)\theta \approx 1$ となる時であるため、 Q の適切な適用回数は、 $k \approx [(\pi/2\theta - 1)]/2$ 回程度であることがわかる³³。

$$Q = \begin{pmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{pmatrix}, \quad (60)$$

$$Q^k |\psi\rangle = \cos(2k+1)\theta |\psi_0\rangle + e^{i\phi} \sin(2k+1)\theta |\psi_1\rangle. \quad (61)$$

また量子状態 $|\psi\rangle$ は、量子ゲート Q の固有ベクトル $|\psi_\pm\rangle$ によって展開できて、量子ゲート Q を $|\psi_\pm\rangle$ に作用させると、固有値 $e^{\pm 2i\theta}$ を確率振幅の位相に反映させることができる。

$$|\psi\rangle = \frac{1}{\sqrt{2}} (e^{i\theta} |\psi_+\rangle + e^{-i\theta} |\psi_-\rangle), \quad (62)$$

$$|\psi_\pm\rangle = \frac{1}{\sqrt{2}} (|\psi_0\rangle \mp i e^{i\phi} |\psi_1\rangle), \quad (63)$$

$$Q^k |\psi\rangle = \frac{1}{\sqrt{2}} (e^{i(2k+1)\theta} |\psi_+\rangle + e^{-i(2k+1)\theta} |\psi_-\rangle). \quad (64)$$

このことは前項で説明した QPE を用いて量子ゲート Q の位相 θ が推定できることを示唆する。つまり、位相 θ で表現される量子状態 $|\psi\rangle$ の確率振幅 $\sin \theta$ を推定できることになる。このアルゴリズムは量子振幅推定 (QAE) と呼ばれており、本節 (1) ニにて説明を行う。

二. 量子振幅推定 (QAE)

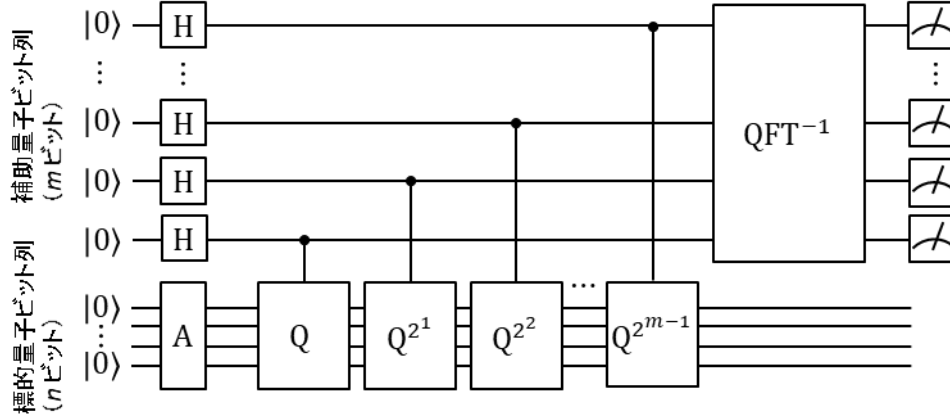
量子振幅推定 (QAE) とは量子状態の確率振幅を推定するアルゴリズムである。ここでは Brassard *et al.* [2000] が提案しているアルゴリズムに沿って QAE の考え方を説明する。QAA の説明と同じ量子状態 $|\psi\rangle$ と量子ゲート Q を用いる。QAE では量子ゲート A によって作成した量子状態 $|\psi\rangle$ の確率振幅 $\sin \theta$ を推定することが目的である。QAA では確率振幅 $\sin \theta$ を増幅させるために量子ゲート Q を用いたが、QAE では量子ゲート Q の固有値 $e^{\pm 2i\theta}$ に含まれる位相 θ を求めることで確率振幅 $\sin \theta$ を推定する。つまり、QAE とは量子ゲート Q の固有値 (位相) の推定と言い換えることができ、QPE のアルゴリズムが適用可能である。以下に QAE のアルゴリズムの流れと量子回路を示す (図 14)。

$$\begin{aligned} |0\rangle^{\otimes m} |\psi\rangle &\xrightarrow{H^{\otimes m}} \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} |k\rangle |\psi\rangle = \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} |k\rangle \left[\frac{1}{\sqrt{2}} (e^{i\theta} |\psi_+\rangle + e^{-i\theta} |\psi_-\rangle) \right] \\ &\xrightarrow{cQ^k} \frac{e^{i\theta}}{\sqrt{2}} \sum_{k=0}^{2^m-1} \frac{1}{\sqrt{2^m}} e^{i2k\theta} |k\rangle |\psi_+\rangle + \frac{e^{-i\theta}}{\sqrt{2}} \sum_{k=0}^{2^m-1} \frac{1}{\sqrt{2^m}} e^{-i2k\theta} |k\rangle |\psi_-\rangle \end{aligned}$$

³³ グローバーのデータ検索では、初期状態として n 量子ビットすべての重ね合わせ状態を用意するため、確率振幅の初期値は $\sin \theta = 1/\sqrt{2^n}$ とわかる。これより計算量を $O(\sqrt{N})$ と見積ることができる。ただし、適切な回数を超えると、対象の測定確率が減少してしまうことに留意されたい。

$$\xrightarrow{\text{QFT}^{-1}} \frac{e^{i\theta}}{\sqrt{2}} \left| \frac{2^m \theta}{\pi} \right\rangle |\psi_+\rangle + \frac{e^{-i\theta}}{\sqrt{2}} \left| 2^m \left(1 - \frac{\theta}{\pi} \right) \right\rangle |\psi_-\rangle. \quad (65)$$

図 14 QAE アルゴリズムの量子回路



QAE のアルゴリズムを説明する。まず補助量子ビット列 $|0\rangle^{\otimes m}$ と標的量子ビット列 $|\psi\rangle$ を用意する。次に補助量子ビット列 $|0\rangle^{\otimes m}$ にアダマールゲートを作用させて重ね合わせ状態を作成する。その後、補助ビット $|k\rangle$ を制御ビットとする制御ゲート CQ^k を標的量子ビット列に作用させると、固有ベクトル $|\psi_{\pm}\rangle$ ごとに確率振幅の異なる位相の重ね合わせ状態が作成される。最後に逆 QFT を補助量子ビット列に作用させれば、位相 θ を含む量子ビットが高い確率で測定される。測定結果 x が得られた場合、以下の変換によって位相 θ と確率振幅 $\sin \theta$ を求めることができる。なお、 $\sin \theta = \sin(\pi - \theta)$ であるから、どちらの結果が得られても推定結果に影響はない。

$$x \approx 2^m \frac{\theta}{\pi}, 2^m \left(1 - \frac{\theta}{\pi} \right) \leftrightarrow \theta \approx \frac{\pi}{2^m} x, \pi \left(\frac{x}{2^m} - 1 \right), \quad (66)$$

$$\sin^2 \theta \approx \sin^2 \left(\frac{\pi}{2^m} x \right) = \sin^2 \left[\pi \left(\frac{x}{2^m} - 1 \right) \right]. \quad (67)$$

QAE の誤差と計算量について説明する。 θ (または $\pi - \theta$) が誤差 $\epsilon = \pm \pi k / 2^m$ で得られる確率の下限は $8/\pi^2 (\approx 81\%, k = 1)$ で与えられる。ここで 2^m はゲート A の呼出し回数³⁴に対応しており、 $N = 2^m$ とすると $N \sim O(1/\epsilon)$ の関係がある。そして、 θ (または $\pi - \theta$) が誤差 ϵ で得られたとき、 $\sin^2 \theta (= a)$ の推計値である $\sin^2 \hat{\theta} (= \hat{a})$ の推計誤差は、誤差の伝播により次式で与えられる。

$$|\hat{a} - a| \leq 2\epsilon \sqrt{a(1-a)} + \epsilon^2 = 2\pi k \frac{\sqrt{a(1-a)}}{2^m} + \left(\frac{\pi k}{2^m} \right)^2 \quad (k > 1). \quad (68)$$

³⁴ ゲート A は最初の量子状態の作成で 1 回、制御ゲート Q で 2 回呼び出される。量子回路の中でゲート Q のべき乗が補助量子ビット列のビット数分呼び出されるため最終的には 2^m 回呼び出されることになる。

θ の推計誤差 ϵ と $\sin^2 \theta$ の推計誤差は同程度であるため、QAE の計算量は $N \sim O(1/\epsilon)$ となる。また任意の $\delta > 0$ に対して、QAE を $O(\log 1/\delta)$ 回繰り返して中央値を取ることで誤差を δ 以内に抑えることができる（冪乗の補題（Power Lemma））。ここで説明した QAE のアルゴリズムはゲート Q や QFT の呼出し回数が多く回路が複雑になる。そのためアルゴリズムの改良が盛んに行われている³⁵。

ホ. 量子モンテカルロ積分(QMCI)³⁶

量子モンテカルロ積分 (QMCI) とは、量子振幅推定 (QAE) を数値積分の計算に応用するアルゴリズムである。QMCI では数値積分の近似式を確率振幅に符号化し、QAE によって確率振幅を推計することで数値積分の値を求める。

QMCI の考え方を説明する。まず d 次元実数空間 $[0,1]^d$ 上の関数 $g(x) = \rho(x)h(x)$ の積分を考える。ただし、QAE を適用するために $g(x)$ は以下のように規格化されているとする。

$$0 \leq \int_{[0,1]^d} \rho(x)h(x)dx \leq 1, \quad (69)$$

$$\int_{[0,1]^d} \rho(x)dx = 1. \quad (70)$$

次に上記の積分の近似 $S(f)$ を考える。 $\rho(x), h(x)$ に対応する関数 $p(x), f(x)$ を用いて近似式を以下で与える。 $p(x), f(x)$ の定義と近似式により $0 \leq S(f) \leq 1$ である。ここで実数値空間の次元は $d = 1$ とし定義域は 2^n 個に分割している。

$$\int_{[0,1]} \rho(x)h(x)dx \approx S(f) := \sum_{x=0}^{2^n-1} p(x)f(x), \quad (71)$$

$$\sum_{x=0}^{2^n-1} p(x) = 1, f(x) = h\left(\frac{x}{2^n}\right). \quad (72)$$

QMCI では、この積分近似式 $S(f)$ を確率振幅に符号化し、QAE により推定することが目的となる。次に上記の近似式を確率振幅として符号化するための量子ゲートを考える。以下のような量子ゲートAを実装すれば、最後尾の補助量子ビットが1として測定される確率が積分近似式 $\sum_{x=0}^{2^n-1} p(x)f(x)$ に一致する。

$$\begin{aligned} |\psi\rangle &= A|0\rangle^{\otimes n}|0\rangle_{\text{anc}} = \sum_{x=0}^{2^n-1} \sqrt{p(x)}|x\rangle \left(\sqrt{1-f(x)}|0\rangle_{\text{anc}} + \sqrt{f(x)}|1\rangle_{\text{anc}} \right) \\ &= \sum_{x=0}^{2^n-1} \sqrt{p(x)}\sqrt{1-f(x)}|x\rangle|0\rangle_{\text{anc}} + \sum_{x=0}^{2^n-1} \sqrt{p(x)}\sqrt{f(x)}|x\rangle|1\rangle_{\text{anc}} \end{aligned} \quad (73)$$

³⁵ 例えば、Suzuki *et al.* [2020]、Grinko *et al.* [2021]や Giurgica-Tiron *et al.* [2022]がある。

³⁶ 宇野 [2019] を参考に記載。

この量子ゲートAは以下で定義される量子ゲートP,Rのテンソル積 $A = R(P \otimes I)$ に分解される。量子ゲートPは確率分布を生成するための量子ゲートであり、量子ゲートRは定義域の値 x に対応する関数値 $f(x)$ を与える制御回転ゲートである。ただし、確率振幅であるため平方根を取っている。

$$P|0\rangle^{\otimes n} := \sum_{x=0}^{2^n-1} \sqrt{p(x)}|x\rangle, \quad (74)$$

$$R|x\rangle|0\rangle_{\text{anc}} := |x\rangle\left(\sqrt{1-f(x)}|0\rangle_{\text{anc}} + \sqrt{f(x)}|1\rangle_{\text{anc}}\right). \quad (75)$$

次の基底変換を行うと、積分近似式が確率振幅として符号化されていることが明らかになる。さらに $\sqrt{S(f)} = \sin \theta$ とおけば、前項で説明したようにQAEが適用できる形となり、QAEを用いて確率振幅 $\sin \theta$ を推計することで数値積分 $S(f) = \sin^2 \theta$ が得られる。

$$\begin{aligned} |\psi\rangle &= \sqrt{1-S(f)}|\psi_0\rangle + \sqrt{S(f)}|\psi_1\rangle \\ &= \cos \theta |\psi_0\rangle + \sin \theta |\psi_1\rangle, \end{aligned} \quad (76)$$

$$|\psi_0\rangle := \sum_{x=0}^{2^n-1} \sqrt{p(x)} \sqrt{\frac{1-f(x)}{1-S(f)}} |x\rangle|0\rangle_{\text{anc}}, |\psi_1\rangle := \sum_{x=0}^{2^n-1} \sqrt{p(x)} \sqrt{\frac{f(x)}{S(f)}} |x\rangle|1\rangle_{\text{anc}}. \quad (77)$$

QMCIの計算量について説明する。QMCIの場合、推定誤差を ϵ に抑えるために必要な関数の評価回数（ゲートAの呼出し回数）は $O(\epsilon^{-1})$ となる。一方、古典モンテカルロ法の場合、必要な関数の評価回数（サンプル・パスの発生回数）は $O(\epsilon^{-2})$ となる。QMCIの計算量は ϵ 倍少ないため2次加速と呼ばれる。ただし、QAE以外の部分の計算量に留意する必要がある。例えば、確率分布や被積分関数の生成アルゴリズム（Grover and Rudolph [2002]）に対して、量子加速が得られないとの研究（Herbert [2021]）があり課題である。

へ. 線形方程式求解(HHL)

HHLとは線形方程式の求解アルゴリズムの1つである。HHLでは係数行列に対するQPEを用いて、係数行列の固有値・固有ベクトルで表現される線形方程式の解を求める。ここではHarrow, Hassidim and Lloyd [2009]らによるHHLの考え方を説明する。まず問題設定として、以下の線形方程式の解を求めることを考える。ただし、係数行列Aはエルミート行列³⁷とし、 $|\mathbf{x}\rangle, |\mathbf{b}\rangle$ は量子ビット列からなるベクトルである³⁸。

$$A|\mathbf{x}\rangle = |\mathbf{b}\rangle \rightarrow |\mathbf{x}\rangle = A^{-1}|\mathbf{b}\rangle, \quad (78)$$

³⁷ 正方行列Aがエルミート行列でない場合は、 $\tilde{A} = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}$ とすれば、 \tilde{A} はエルミート行列となり、 $|\tilde{\mathbf{b}}\rangle = \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix}$ とおいて $\tilde{A}|\mathbf{x}\rangle = |\tilde{\mathbf{b}}\rangle$ とできる。

³⁸ 量子ビット列の確率振幅にベクトル成分を符号化することでベクトルを表現している。これをアナログ（振幅）符号化と呼ぶ。なお、これまでの符号化方法はデジタル（基底）符号化と呼ばれる。

$$|\mathbf{x}\rangle = \sum_k x_k |k\rangle, |\mathbf{b}\rangle = \sum_k b_k |k\rangle. \quad (79)$$

線形代数の知識より、係数行列 A の固有値 λ_k ・固有ベクトル $|u_k\rangle$ を用いると、係数行列 A はスペクトル分解として表現され、ベクトル $|\mathbf{b}\rangle$ は線形結合として表現される。したがって、線形方程式の解 $|\mathbf{x}\rangle$ は、固有値 λ_k ・固有ベクトル $|u_k\rangle$ の線形結合として表現される³⁹。また係数行列 A の量子ゲート e^{iA} はユニタリ行列となりスペクトル分解として表現できる。

$$A = \sum_k \lambda_k |u_k\rangle\langle u_k|, \quad (80)$$

$$|\mathbf{b}\rangle = \sum_k \beta_k |u_k\rangle, \quad (81)$$

$$|\mathbf{x}\rangle = A^{-1}|\mathbf{b}\rangle = \sum_k \frac{\beta_k}{\lambda_k} |u_k\rangle, \quad (82)$$

$$e^{iA} = \sum_k e^{i\lambda_k} |u_k\rangle\langle u_k|. \quad (83)$$

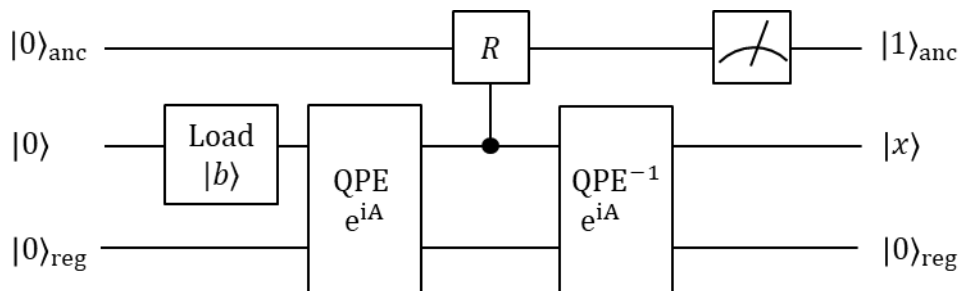
注目すべきは、係数行列 A の固有値 λ_k が、ユニタリ行列 e^{iA} の固有値 $e^{i\lambda_k}$ の位相 λ_k として含まれていることである。ユニタリ行列 e^{iA} に対して QPE を用いて位相 λ_k を取り出し、適当な制御回転ゲートを適用すれば、線形方程式の解 $|\mathbf{x}\rangle$ が得られることを意味する。

以下にアルゴリズムの流れと量子回路を示す (図 15)。

$$\begin{aligned} & |0\rangle_{\text{reg}} |\mathbf{b}\rangle |0\rangle_{\text{anc}} = |0\rangle_{\text{reg}} \sum_k (\beta_k |u_k\rangle) |0\rangle_{\text{anc}} \\ & \xrightarrow{\text{QPE of } e^{iA}} \sum_k \beta_k |\lambda_k\rangle_{\text{reg}} |u_k\rangle |0\rangle_{\text{anc}} \\ & \xrightarrow{\text{CR}} \sum_k \beta_k |\lambda_k\rangle_{\text{reg}} |u_k\rangle \left(\sqrt{1 - \frac{c^2}{\lambda_k^2}} |0\rangle + \frac{c}{\lambda_k} |1\rangle \right)_{\text{anc}} \\ & \xrightarrow{\text{QPE}^{-1} \text{ of } e^{iA}} \sum_k \beta_k |0\rangle_{\text{reg}} |u_k\rangle \left(\sqrt{1 - \frac{c^2}{\lambda_k^2}} |0\rangle + \frac{c}{\lambda_k} |1\rangle \right)_{\text{anc}} \\ & \xrightarrow{\text{measured } |1\rangle_{\text{anc}}} c |0\rangle_{\text{reg}} \left(\sum_k \frac{\beta_k}{\lambda_k} |u_k\rangle \right) |1\rangle_{\text{anc}} = c |0\rangle_{\text{reg}} |A^{-1}\mathbf{b}\rangle |1\rangle_{\text{anc}}. \end{aligned} \quad (84)$$

³⁹ $A^{-1}|\mathbf{b}\rangle = (\sum_k \lambda_k^{-1} |u_k\rangle\langle u_k|)(\sum_l \beta_l |u_l\rangle) = \sum_{k,l} \lambda_k^{-1} \beta_l |u_k\rangle\langle u_k|u_l\rangle = \sum_k \lambda_k^{-1} \beta_k |u_k\rangle$ によって導出している。ただし、 $A^{-1} = \sum_k \lambda_k^{-1} |u_k\rangle\langle u_k|$ と $\langle u_k|u_l\rangle = \delta_{kl}$ を用いた。

図 15 HHL アルゴリズムの量子回路



HHL のアルゴリズムの説明を行う。初期状態 $|0\rangle_{\text{reg}}|b\rangle|0\rangle_{\text{anc}}$ を用意しておく。ここでは $|0\rangle_{\text{reg}}$ をレジスタ量子ビット列、 $|b\rangle$ を入力量子ビット列、 $|0\rangle_{\text{anc}}$ を補助量子ビットと呼ぶ。まず量子ゲート e^{iA} の QPE を実行し固有値 $e^{i\lambda_k}$ の位相 λ_k をレジスタ量子ビット列に出力する。次にレジスタ量子ビット列を制御ビットとし、補助量子ビットを標的ビットとする制御回転ゲートを作用させて、補助量子ビットの確率振幅に位相 λ_k を反映させる。次に QPE の逆演算 (uncomputation) を行い、レジスタ量子ビット列の値を 0 に戻す。補助量子ビットの測定結果が 1 の時、量子状態が線形方程式の解 $|A^{-1}b\rangle$ となる。

HHL の計算量と前提条件について触れる。線形方程式の行列サイズ N に対して、HHL の計算量は $O(\log N)$ である。古典の共役勾配アルゴリズムの計算量が $O(N)$ であるため HHL によって指数的な量子加速が得られると考えられる。しかし、いくつかの注意点がある。まず、入力データの量子状態の用意や結果の読み出し部分の計算量を考慮していない。十分な考慮なく入出力を行うと計算量が $O(N)$ となり指数加速が相殺されてしまう可能性がある。したがって、前提条件として、効率的に入力データを準備できて⁴⁰、何らかの統計値に加工して結果を出力する必要がある。また線形方程式の行列がスパースでない場合、推計結果の精度が落ちてしまうため、問題設定にも留意が必要である。

(2) NISQ アルゴリズム⁴¹

開発途上の量子コンピュータである NISQ デバイスは、量子ビット数が少なく、ノイズ耐性がないため、計算の誤りが多く計算能力が制限される。そのため、古典コンピュータを用いて NISQ デバイスの量子計算をサポートする量子古典ハイブリッド計算方式が NISQ アルゴリズムとして考えられている。具体的には、パラメータ付き量子回路 PQC (Parameterized Quantum Circuit) で量子計算を行い、その計算結果をもとに古典コンピュータ側でパラメータ調整を行い、量子回路を最適化することで所望の結果を得る方式である。この計算方式は、ノイズの影響が無視できない環境や正確な量子回路が分からない状況で有効となる可能性がある手法であり、比較的少ない量子ゲート数 (浅い回路、逆は深い回路) でも高い表現力を持つ量子回路ができる可能性があるとされている。誤りが多い

⁴⁰ 例えば、量子メモリ (qRAM) が提案されている (Giovannetti, Lloyd and Maccone [2008])。

⁴¹ より包括的な NISQ アルゴリズムの説明についてはサーベイ (Cerezo *et al.* [2021]) を参照されたい。

NISQ デバイスであっても比較的执行しやすいため、量子化学計算、組合せ最適化問題や機械学習等への利用方法が研究されている。量子古典ハイブリッド計算方式を実装する量子回路のことを、変分量子回路（VQC : Variational Quantum Circuit）と呼ぶ。

本節では、まず NISQ アルゴリズムの種類と概要を説明する。次に NISQ アルゴリズムのうち、離散ポートフォリオ最適化等の組合せ最適化問題に用いられる量子近似最適化アルゴリズム（QAOA）を取り上げて仕組みを詳しく解説する。QAOA 以外の NISQ アルゴリズムについては 5 節にてファイナンスへの応用例として触れる。

イ. ハミルトニアンとシュレディンガー方程式

NISQ アルゴリズムの理解のために、量子計算で用いられる量子力学の知識を補足する。量子力学ではエネルギー等の物理量は行列で記述され、物理量の観測結果は期待値（平均）として得られる。例えば、エネルギーはハミルトニアン H ⁴² と呼ばれるエルミート行列を用いる。エルミート行列とは $H = H^\dagger$ を満たす正方行列である。特定の量子状態 $|0\rangle, |1\rangle$ のエネルギーを ϵ_1, ϵ_2 とし、ハミルトニアンを $H = \text{diag}(\epsilon_1, \epsilon_2)$ とすると、ハミルトニアンを量子状態で囲むことでエネルギーの期待値が計算できる。

$$\langle H \rangle := \langle \psi | H | \psi \rangle = (\alpha^* \quad \beta^*) \begin{pmatrix} \epsilon_1 & 0 \\ 0 & \epsilon_2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = |\alpha|^2 \epsilon_1 + |\beta|^2 \epsilon_2. \quad (85)$$

また量子状態の時間変化はハミルトニアンからなるシュレディンガー方程式と呼ばれる微分方程式に従っている。量子状態を初期状態 $|\psi(0)\rangle$ から終状態 $|\psi(t)\rangle$ に変化させる行列 e^{-iHt} は時間発展演算子と呼ばれる⁴³。時間発展演算子は量子ゲートに必要なユニタリ性を満たしており、特殊なケースが 3 節 (4) で説明した量子ゲートである。

$$i \frac{d}{dt} |\psi(t)\rangle = H |\psi(t)\rangle \Leftrightarrow |\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle. \quad (86)$$

ロ. NISQ アルゴリズムの種類

NISQ アルゴリズムの種類と概要について説明する（表 9）。NISQ アルゴリズムは、問題設定や応用先に応じていくつかの種類に分けられるが、基本的に古典コンピュータによるパラメータ調整等と NISQ デバイスによる計算の実行を繰り返して所望の結果を得るアルゴリズムである点が変わらない。

⁴² 表記は同じであるがアダマールゲートではないことに注意。文脈によってどちらか判断できる。

⁴³ ここでは簡単のため時間に依存しないハミルトニアンのもとで方程式を解いている。

表 9 NISQ アルゴリズム

名称 (略称)	概要・応用先
変分量子固有値ソルバ (VQE)	量子化学や物性物理学などの量子多体系の基底状態(エネルギー最低状態)を計算するアルゴリズム。
量子近似最適化アルゴリズム (QAOA)	変数0,1だけからなる組合せ最適化問題(0-1整数計画問題)の計算アルゴリズム。イジングモデルのハミルトニアン H のエネルギー期待値の最小化問題に帰着。
量子回路学習 (QCL)	量子回路を学習モデルと見立てて機械学習タスクを行うアルゴリズム。パラメータ付きの量子回路からの出力(測定値)と訓練出力の差(損失関数)を最小化。
変分量子シミュレータ (VQS)	量子系の時間発展をシミュレーションする。

変分量子固有値ソルバ (VQE : Variational Quantum Eigensolver) は、量子化学計算における物質の最小 (基底) エネルギーを求める量子アルゴリズムであり、Peruzzo *et al.* [2014] によって提案されている。分子や物質の性質を決める電子の動きはシュレディンガー方程式によって記述される。この方程式を解くことはハミルトニアン H の固有値問題を解いて固有値 E_i (エネルギー) と固有ベクトル (固有状態) $|\phi_i\rangle$ を求めることと同値である。

$$H|\psi\rangle = E|\psi\rangle \quad (\text{時間に依存しないハミルトニアン}). \quad (87)$$

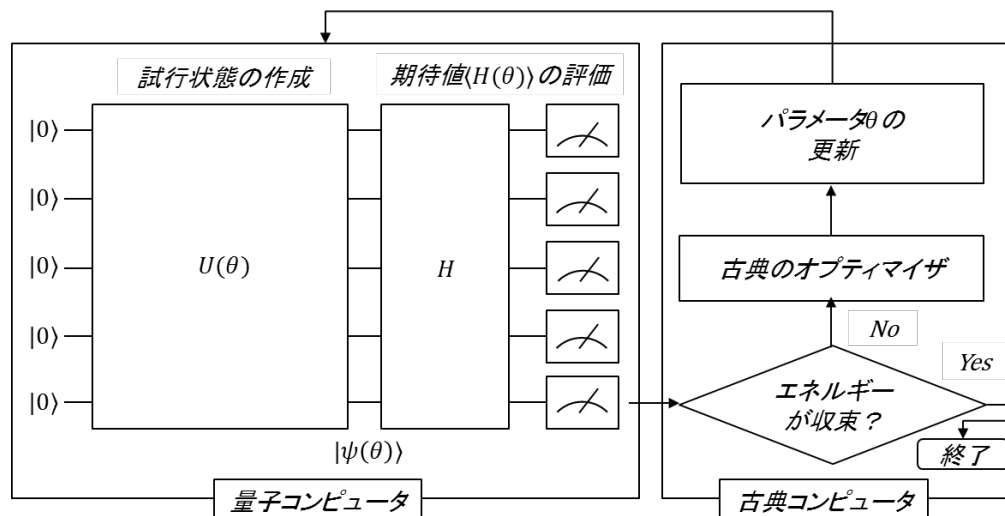
一般に電子の状態は一番エネルギーの低い状態 (基底状態) にあるが、物質が大きくなるほどハミルトニアンの次元は指数関数的に巨大になり方程式を解くことが困難になる。巨大次元のハミルトニアンの最小エネルギー状態を求める有力な手法として変分法が用いられる。任意状態 $|\psi\rangle$ のエネルギー期待値 $\langle\psi|H|\psi\rangle$ は最小エネルギー E_0 よりも高くなるため (変分原理)、変分法ではエネルギー期待値が最も低くなる状態 $|\psi\rangle$ を探索する。

$$\langle\psi|H|\psi\rangle \geq E_0. \quad (88)$$

VQE では変分法によるアプローチを最適化アルゴリズムとして採用する (図 16)。NISQ デバイスの量子回路にパラメータ θ を持たせて量子状態 $|\psi(\theta)\rangle$ を生成し、古典コンピュータ側でエネルギー期待値を最小化するパラメータの探索を行う。

$$\arg \min_{\theta} \langle\psi(\theta)|H|\psi(\theta)\rangle. \quad (89)$$

図 16 VQE アルゴリズムの概要



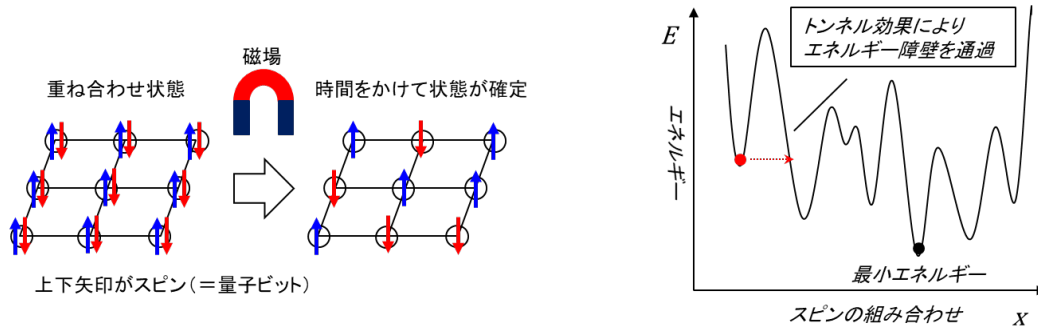
資料：嶋田・情報処理学会出版委員会 [2020] をもとに筆者作成。

量子近似最適化アルゴリズム (QAOA : Quantum Approximate Optimization Algorithm) は、組合せ最適化問題を解くためのアルゴリズムであり、Farhi and Goldstone [2014] によって提案されている。離散変数による組合せ最適化問題には、連続変数の最適化問題とは異なり、厳密解を得ることが難しい問題 (例：巡回セールスマン問題やナップサック問題) が存在する。組合せ最適化問題に特化したアニーリング方式の計算プロセスを変分量子回路によって近似するのが QAOA のアプローチである。

アニーリング方式とは、組合せ最適化問題をイジング・モデルのハミルトニアン H のエネルギー期待値 $\langle \psi | H | \psi \rangle$ の最小化問題に帰着させて解く方式であり、Farhi *et al.* [2000] によって具体的な計算方法が示されている。アニーリング方式では、各格子点のスピンを量子ビットとして扱い、最適化問題をイジング・モデルのハミルトニアンの値に設定する (

図 17 左)。まず強い磁場をかけて各量子ビットを横向き（上向きと下向きスピンの重ね合わせ状態）に初期化する。次に横磁場を少しずつ弱くしていき 0 にする。すると各スピンはスピン間の相互作用の強さに応じた基底状態に確定する。途中で局所的な最小値に落ち着きそうになっても、トンネル効果⁴⁴によって最終的に基底状態に到達すると期待される（図 17 右）。

図 17 イジング・モデルの概要（左）と量子アニーリングによる探索イメージ（右）



上記のプロセスは断熱量子計算⁴⁵と呼ばれ、以下のシュレディンガー方程式を解くことに相当する。自明な基底状態のハミルトニアン H_{ref} を初期状態 ($t = 0$) として用意し、外部磁場を長時間 ($t \rightarrow T$) 与えることで、目的の基底状態のハミルトニアン H_{cost} ⁴⁶が導かれる（図 18）。

$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle = \left[\left(1 - \frac{t}{T}\right) H_{\text{ref}} + \frac{t}{T} H_{\text{cost}} \right] |\psi(t)\rangle, \quad (90)$$

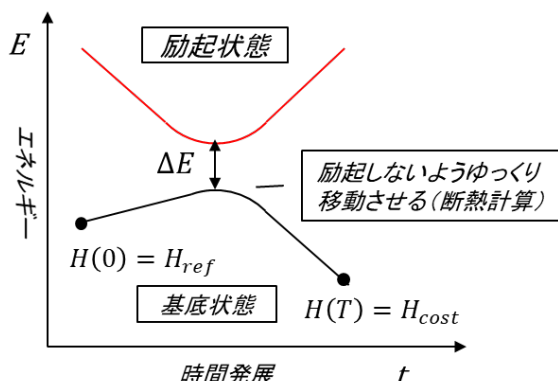
$$H_{\text{ref}} = \sum_i X_i, H_{\text{cost}} = \sum_{i < j} J_{ij} Z_i Z_j + \sum_i h_i Z_i. \quad (91)$$

⁴⁴ トンネル効果とは量子が波の性質によってエネルギー障壁をある確率で通り抜ける現象。

⁴⁵ 断熱量子計算の理論的な説明は、西森ほか [2018] や Albash and Lidar [2018] を参照されたい。

⁴⁶ H_{ref} は横磁場の効果であり、 H_{cost} の右辺第1項はスピン間の相互作用、第2項は局所的な縦磁場の効果である。 X_i, Z_i は*i*番目の格子点の*x*軸、*z*軸方向のスピンで、ビット反転ゲート*X*、位相反転ゲート*Z*に対応する。 J_{ij} はスピン Z_i, Z_j 間の相互作用の強さ、 h_i は*z*軸方向の磁場の強さを表す。

図 18 ハミルトニアンの時間発展

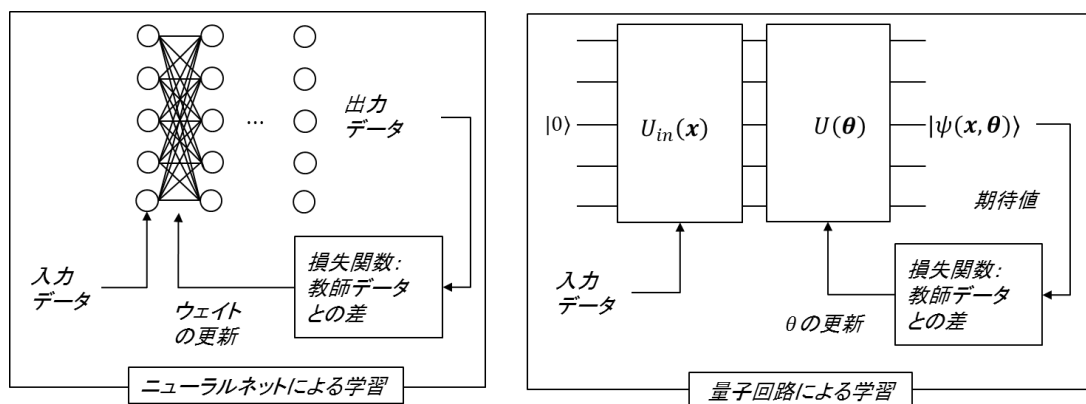


QAOA では、断熱量子計算をゲート方式の量子回路によって離散近似し、NISQ デバイスによってシミュレーションし、古典コンピュータ側でハミルトニアンの期待値を最小化するようにパラメータ探索を行う。QAOA のアルゴリズムの詳細については後で説明する。

$$\arg \min_{\beta, \gamma} \langle \psi(\beta, \gamma) | H | \psi(\beta, \gamma) \rangle. \quad (92)$$

量子回路学習 (QCL : Quantum Circuit Learning) は、量子回路を学習モデルに見立てて量子回路の出力と訓練出力の差 (損失関数等) を最小化するアルゴリズム (図 19) であり、Mitarai *et al.* [2018] によって提案されている。近年、機械学習の分野において、深いニューラル・ネットワークを用いて複雑な関数を近似するディープ・ラーニングが脚光を浴びている。QCL はこのニューラル・ネットワークを量子回路に置き換えた学習手法である。量子力学の重ね合わせの性質を活かして指数関数的に多数の基底関数を用いて学習できるため、モデルの表現力が向上するとされている。また量子回路のユニタリ性 (パラメータのノルムが 1 に制限) が、ある種の正則化項として働き、オーバー・フィッティングを防げると考えられている。

図 19 QCL アルゴリズムの概要



資料 : 嶋田・情報処理学会出版委員会 [2020] をもとに筆者作成

変分量子シミュレータ (VQS : Variational Quantum Simulator) は、量子状態の時間発展のシミュレーションを行うアルゴリズムであり、Li and Benjamin [2017] によって提案されている。通常、量子状態の時間発展をシミュレートするにはシュレディンガー方程式を数値的に解く必要があるが、古典コンピュータでシミュレートする場合、ハミルトニアン サイズ (粒子やスピンの数) が大きくなると、計算時間が指数関数的に増大してしまう。そのため量子コンピュータを用いて量子状態を直接シミュレーションするのが VQS の発想である。VQS では NISQ デバイス側と古典コンピュータ側で入力パラメータや出力結果をやり取りして、各時点における量子状態をシミュレーションする。

このように NISQ アルゴリズムの多くは、何らかのコスト関数を最小化するように古典コンピュータ側で量子回路のパラメータを調整し、量子コンピュータ側で計算して結果を確認するという、量子古典ハイブリッド型の最適化アルゴリズムといえる。

NISQ アルゴリズムの留意点について述べる。NISQ アルゴリズムは幅広い分野での応用が可能であるが、ヒューリスティックな近似アルゴリズムであるため、古典コンピュータに対する高速性が保証されていない。またパラメータ数が増えるほど古典コンピュータ側の計算負荷が増えるため最適化計算のボトルネックになる可能性もある。

ハ. 量子近似最適化アルゴリズム(QAOA)

QAOA の仕組みとアルゴリズムを説明する。QAOA では n 桁ビット列 $\mathbf{z} = z_1 z_2 \dots z_n, z_i \in \{0,1\}$ に関して、コスト関数 $C(\mathbf{z}) = \sum_{\alpha} C_{\alpha}(\mathbf{z})$ が最小となるビット列 \mathbf{z} の探索問題を考える。ここで $C_{\alpha}(\mathbf{z})$ はビット列 \mathbf{z} を引数とする何らかの関数で、例えば、イジング・モデルと同様の形式 $C_{\alpha}(\mathbf{z}) = z_i \cdot z_j$ で表される項とする。次にパラメータ $\beta := \{\beta_1, \dots, \beta_p\}$ 、 $\gamma := \{\gamma_1, \dots, \gamma_p\}$ を導入し、次のような量子状態 $|s\rangle$ 、 $|\beta, \gamma\rangle$ を考える。

$$|s\rangle := |+\rangle^{\otimes n} = \left[\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \right]^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} |z\rangle \quad (93)$$

$$|\beta, \gamma\rangle := U_X(\beta_p) U_C(\gamma_p) \dots U_X(\beta_1) U_C(\gamma_1) |s\rangle. \quad (94)$$

量子状態 $|s\rangle$ は n 量子ビットの重ね合わせ状態であり、量子状態 $|\beta, \gamma\rangle$ は、量子状態 $|s\rangle$ に対して、以下の量子ゲート $U_C(\gamma_{(i)})$ 、 $U_X(\beta_{(i)})$ を作用させて得られる量子状態である。

$$U_C(\gamma_{(i)}) := e^{-i\gamma_{(i)} C(\mathbf{z})} = \prod_{\alpha} e^{-i\gamma_{(i)} C_{\alpha}(\mathbf{z})}, \quad ((i) = 1, \dots, p) \quad (95)$$

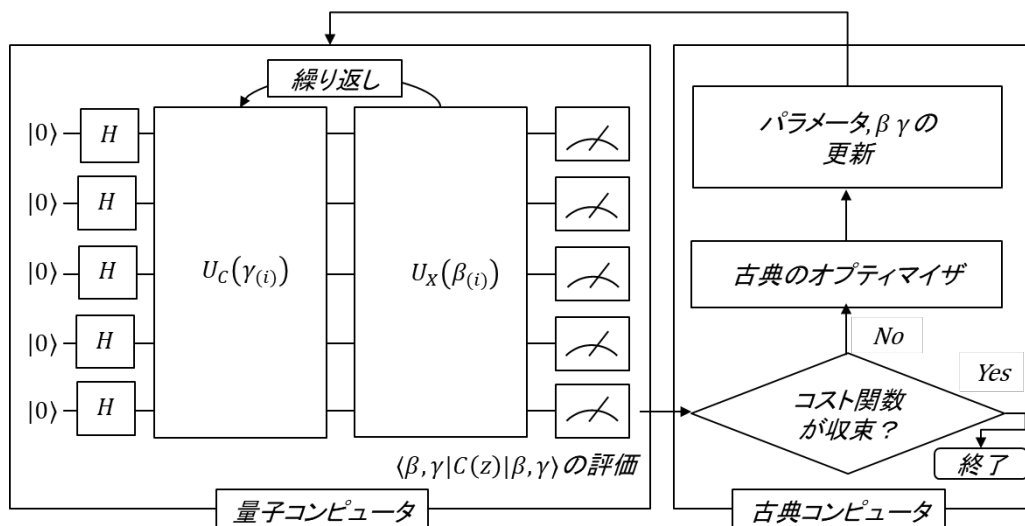
$$U_X(\beta_{(i)}) := e^{-i\beta_{(i)} B} = e^{-i\beta_{(i)} \sum_{j=1}^n X_j} = \prod_j e^{-i\beta_{(i)} X_j}, \quad ((i) = 1, \dots, p) \quad (96)$$

\mathbf{z} は引数のビット列 $\mathbf{z} = z_1 z_2 \dots z_n$ を位相反転ゲート $Z_1 \dots Z_n$ で置き換えたものである。 $B = \sum_{j=1}^n X_j$ はミキシング演算子と呼ばれ、 X_j はビット反転ゲート (NOT ゲート) である。また

$U_C(\gamma), U_X(\beta)$ は断熱量子計算の離散近似に対応している⁴⁷。コスト関数 $C(Z)$ の期待値 $\langle \gamma, \beta | C(Z) | \gamma, \beta \rangle$ を最小にする γ, β を探索することで、元々の最適化問題の答えを探し出すとするのが QAOA の狙いである。以下に QAOA のアルゴリズムの手順を示す (図 20)。

- ① 量子コンピュータ上で重ね合わせ状態 $|s\rangle = |+\rangle^{\otimes n}$ を作る。
- ② パラメータ β, γ に対応する $U_C(\gamma_{(i)})$ 、 $U_X(\beta_{(i)})$ を作用させて状態 $|\beta, \gamma\rangle$ を得る。
- ③ 量子コンピュータでコスト関数 $\langle \beta, \gamma | C(Z) | \beta, \gamma \rangle$ を評価 (測定) する。
- ④ 古典コンピュータで $\langle \beta, \gamma | C(Z) | \beta, \gamma \rangle$ を小さくするようパラメータ β, γ を更新する。
- ⑤ ①～④を繰り返し、最適解 β^*, γ^* を得る。
- ⑥ 状態 $|\beta^*, \gamma^*\rangle$ に対して、z方向の射影測定を複数回実行する。
- ⑦ 最も良さそうな測定結果 $z_1 z_2 \dots z_n$ を元々の最適化問題の解として採用する。

図 20 QAOA アルゴリズムの概要



資料：嶋田・情報処理学会出版委員会 [2020] をもとに筆者作成

5. 量子計算のファイナンス分野への応用

本節では、量子計算のファイナンス問題への応用例を紹介する (表 10)。具体的には、金融商品のプライシング、バリュー・アット・リスク (VaR : Value at Risk) やグリークス等のリスク計測手法、そしてポートフォリオ等の最適化問題に対する量子計算の応用研究の解説を行う。なお、本稿では主要なファイナンス応用例の紹介に絞っており、機械学習

⁴⁷ シュレディンガー方程式 $i\hbar \frac{d}{dt} |\psi\rangle = H(t) |\psi\rangle$ の時間発展は、ユニタリ行列 $U(t) = \exp(-i \int_0^t H(t) dt)$ で与えられる。イジング・モデルのハミルトニアン $H = H_{\text{ref}} + H_{\text{cost}} = \sum X_{(i)} + C(Z)$ を考えて Trotter 分解で離散近似すると、ユニタリ行列 $U_C(\gamma_{(i)}) = \prod_{\alpha} \exp(-i\gamma_{(i)} C_{\alpha}(Z))$ と $U_X(\beta_{(i)}) = \prod_j \exp(-i\beta_{(i)} X_j)$ の積の形で表せる。パラメータ $\gamma_{(i)}$ 、 $\beta_{(i)}$ は $\gamma_{(i)} = (1-t)\Delta t$ 、 $\beta_{(i)} = t\Delta t$ でありパラメータ p は時間分割数に対応。

関連タスクへの応用例は含んでいない。そのため、より広範な応用例について興味のある読者は既存サーベイ論文 (Herman *et al.* [2022])を参照されたい⁴⁸。

表 10 ファイナンス分野への量子アルゴリズムの応用例

ファイナンス分野	数値計算例	量子アルゴリズムの例
デリバティブのプライシング	・オプション、CDO等	・量子モンテカルロ積分 (QMCI) ・微分方程式求解
リスク計測	・VaR、グリークス、CVA、ストレステスト	・量子モンテカルロ積分、量子勾配法 ・変分量子回路 (CVAIに対して)
ポートフォリオ最適化	・連続最適化 ・組合せ最適化	・線形方程式求解 (HHL等) ・変分量子回路、アニーリング
その他最適化 機械学習	・信用スコアリング ・予兆管理等 ・機械学習	・変分量子回路、アニーリング ・線形方程式求解 (HHL等) →線形回帰等の機械学習手法

(1) プライシングへの応用

金融商品のプライシングへの量子計算の応用研究を紹介する。金融機関では大量の金融商品を扱っており、金融商品の価格評価等のための計算が大量に発生している。ブラック＝ショールズ・モデルによるヨーロッパン・オプションであれば、オプション価格の解析解が存在するため計算時間が問題になることはない。しかし、複雑なプライシング・モデルによるエキゾチックなデリバティブ商品等の場合には解析解が存在しないため、モンテカルロ法による計算負荷の高いシミュレーションを行う必要が生じる。そのため、量子計算によるプライシングの高速化は金融機関のビジネスにおけるインパクトになり得る。

金融商品の評価手法にはいくつかあるが、そのうちの1つがモンテカルロ法による数値積分である (以下、古典モンテカルロ法)。誤差を ϵ とする時の古典モンテカルロ法の計算量が $O(\epsilon^{-2})$ であるのに対して、量子計算の量子モンテカルロ積分 (QMCI) の計算量は $O(\epsilon^{-1})$ で抑えられるため2次の量子加速が得られる (4節参照)。この理由により、古典モンテカルロ法による計算部分を、QMCI (あるいはQAE) で置き換えて高速化を図るアプローチが取られることが多い。

現状、FTQC は実現していないため、プライシングへの応用研究の中心は、簡単な問題設定 (例：ブラック＝ショールズ・モデルによるプレーンな商品) に対する量子アルゴリズムの提案と演算リソースの見積もりや、古典コンピュータやNISQデバイスでの数値実験であった。もっとも、近年の動向として、複雑な商品や評価モデル (例：エキゾチック・デリバティブ、ボラティリティ・モデル) を扱う、より実用的な問題設定に対する量子アルゴリズムが提案されるようになってきている。また偏微分方程式の有限差分法に対

⁴⁸ その他のサーベイとしては Egger *et al.* [2020]、Boulard *et al.* [2020]、Gómez *et al.* [2022]、Orús, Mugel and Lizaso [2019 b]、Pistoia *et al.* [2021] を参照。

する量子アルゴリズムや変分量子回路による数値シミュレーション等、QMCI 以外の新たなプライシング手法も提案されている。

量子計算のデリバティブのプライシングへの応用例を紹介していく。まず、オプションのプライシングへの量子計算の最初の応用研究として、Rebentrost, Gupt and Bromley [2018] は、QMCI によるオプション・プライシングを提案し、ブラック＝ショールズ・モデルによるヨーロピアン型やアジア型のオプションに対して2次の量子加速が得られることを示している。ただし、原資産価格の確率分布が所与であること、および効率的にペイオフ関数が計算可能であることを仮定している。次に、Stamatopoulos *et al.* [2020] は、QMCI によるオプション・ポートフォリオや経路依存型オプション（バリア・オプション）の評価方法を提案している。彼らは、量子コンピュータの実機を用いて、プライシング用の量子回路のパフォーマンス調査や簡単な誤り補償処理も行っており、実用的なプライシングには、多数の量子ゲートを実行するためのハードウェアが必要であると述べている。

QMCI を用いたオプション価格推定アルゴリズムの要点と結果について、Rebentrost, Gupt and Bromley [2018] に沿って解説する。まずブラック＝ショールズ・モデルのヨーロピアン・オプションの価格について説明しておく。ブラック＝ショールズ・モデルは、原資産価格 S_t が幾何ブラウン運動に従うモデルであり、以下の確率微分方程式で表現される。ブラック＝ショールズ・モデルの場合、満期 T における原資産価格 S_T の密度関数 $p(S_T)$ は対数正規分布に従う。またペイオフ関数 $f(S_T)$ を以下で決める。ここではストライク K のコール・オプションとする。

$$dS_t = rS_t dt + \sigma S_t dW_t \leftrightarrow S_T = S_0 e^{\sigma W_T + \left(r - \frac{\sigma^2}{2}\right)T}, \quad (97)$$

$$p(S_T) = \frac{1}{S_T \sigma \sqrt{2\pi T}} e^{-\frac{(\ln S_T - \mu)^2}{2T}}, \quad (98)$$

$$f(S_T) = \max\{S_T - K, 0\}. \quad (99)$$

ヨーロピアン・オプションの価格 Π は、満期 T 時点のペイオフ関数 $f(S_T)$ の現在価値の期待値であるから、以下の数値積分として計算される。なお、 e^{-rT} は無リスク金利 r による割引率である。またペイオフ関数は離散化した原資産価格 i を用いて $f(i)$ とし、 $0 \leq f(i) \leq 1$ となるようにスケールしておく。

$$\Pi = e^{-rT} E_Q[f(x)] \approx e^{-rT} \sum_i p(i) f(i), \quad (100)$$

オプション・プライシングの量子アルゴリズムを説明する。

- ① 量子ビット列 $|0\rangle^{\otimes n}$ に満期時点の原資産価格の確率分布 $\sum_i \sqrt{p(i)} |i\rangle$ を生成する。
- ② 量子ビット列の値 $|i\rangle$ を制御ビット、補助ビット $|0\rangle_{\text{anc}}$ を標的ビットとする制御回転ゲートを作用させて、ペイオフ関数の状態を生成する。

$$\sum_i \sqrt{p(i)} \sqrt{1 - f(i)} |i\rangle |0\rangle_{\text{anc}} + \sum_i \sqrt{p(i)} \sqrt{f(i)} |i\rangle |1\rangle_{\text{anc}}$$

- ③ QAE により補助ビットが $|1\rangle_{anc}$ となる確率 $\sum_i p(i)f(i)|i\rangle$ を得る。
- ④ 割引率の乗算とスケーリングによってオプション価格 Π を得る。

$$\begin{aligned}
|0\rangle^{\otimes n}|0\rangle_{anc} &\rightarrow \sum_{i=0}^{2^n-1} \sqrt{p(i)}|i\rangle|0\rangle_{anc} \\
&\rightarrow \sum_{i=0}^{2^n-1} \sqrt{p(i)}\sqrt{1-f(i)}|i\rangle|0\rangle_{anc} + \sum_{i=0}^{2^n-1} \sqrt{p(i)}\sqrt{f(i)}|i\rangle|1\rangle_{anc} \\
&\xrightarrow{\text{QAE}} \sum_{i=0}^{2^n-1} p(i)f(i) \xrightarrow{\text{scaling}} \Pi = e^{-rT} \sum_i p(i)f(i) \tag{101}
\end{aligned}$$

Rebentrost, Gupt and Bromley [2018] は、古典モンテカルロ法と QMCI の計算量を数値計算によって検証した。その結果、理論的結果と整合的な形で、古典モンテカルロ法では計算誤差と計算量の関係が $\epsilon_C \sim O(1/\sqrt{N})$ 、QMCI では $\epsilon_Q \sim O(1/N)$ となることを示した。このことは、QMCI によって実際に計算高速化（2 次加速）が起きることを示している。

複雑な商品性のデリバティブのプライシングに対する量子アルゴリズムの研究も行われている。Tang *et al.* [2021] は、複雑な構造を持つ金融商品である債務担保証券 CDO (Collateralized Debt Obligation) に対して、QMCI によるプライシング方法を提案している。すなわち、正規 - 逆ガウス法の量子回路を実装し、古典モンテカルロ法の代わりに QAE を適用して CDO トランシェごとの損失を推定している。

Chakrabarti *et al.* [2021] は、事前に定めた条件を満たすと自動的に早期償還が発生するエキゾチックな金融商品である、オート・コーラブル、TARF (Target Accrual Redemption Forward)⁴⁹ のプライシングに対する量子アルゴリズムを提案し、プライシングに必要なリソース見積りを行っている。事前学習させた変量子回路と FTQC の組合せによる再パラメータ化によるプライシング方法を導入し、同手法が確率ボラティリティや局所ボラティリティに拡張できると述べている。もっとも、オート・コーラブルのプライシングを量子コンピュータで行うには、8 千個の論理量子ビットと深さ 54 百万の位相シフトゲート (Tゲート) が必要であるとし、現在の技術水準では実現は難しいとしている。

アメリカン・オプションやバミューダン・オプション等、早期権利行使が可能なオプション取引のプライシングに対する量子アルゴリズムの応用研究も進められている。

Doriguello *et al.* [2021] は、アメリカン・オプションの継続価値を最小二乗法によって求める最小二乗モンテカルロ法の量子アルゴリズムを提案して 2 次に近い量子加速が得られるとしている。同様に Miyamoto [2021] は、QMCI とチェビシェフ補間を組み合わせ、バミューダン・オプションの継続価値を近似するアルゴリズムを提案している。

⁴⁹ これらは金融機関で頻繁に取引されているが、経路依存型であるため計算負荷が高い。

複雑なプライシング・モデルに対する量子アルゴリズムの応用研究も行われている。*Kaneko et al.* [2020] は、ボラティリティ・スマイルを考慮する局所ボラティリティ・モデルの量子回路の実装方法を提案している。量子振幅推定 (QAE) タイプと擬似乱数 (PRN : pseudo-random number) タイプ⁵⁰の2つの量子アルゴリズムを考え、FTQC 上での具体的な量子回路の実装を示し、量子回路の幅 (量子ビット数) や深さ (T ゲート数) でのリソース見積を行っている。なお、実務上の観点としては、局所ボラティリティ関数を所与とし、キャリブレーション⁵¹していないこと等に留意が必要である。

QMCI とは異なるアプローチのプライシング手法も検討されている。1つはオプション価格が従う偏微分方程式に対して、量子アルゴリズムによる有限差分法を用いて解を求める手法である。*Miyamoto and Kubo* [2021] は、原資産が複数あるデリバティブの価格が従う偏微分方程式に対して、線形微分方程式の求解アルゴリズムに基づく微分方程式のソルバを用いる方法を提案している。通常、確率振幅に埋め込まれているオプション価格をそのまま結果として取り出そうとすると測定回数が指数関数的に増加してしまう。そのため、現時点のオプション価格が将来時点のオプション価格の期待値で形成されることから、期待値を計算して結果を取り出す工夫を行うことで指数加速になるとしている。本手法に用いる微分方程式のソルバや QAE には FTQC が前提となることから、*Kubo et al.* [2022] は、変分量子シミュレータ (VQS) や内積計算を行う SWAP テストといった NISQ デバイスで可能なアルゴリズムを組み合わせるプライシングを行っている。また *Kubo et al.* [2021] は、確率微分方程式を3項ツリー・モデルで近似し、変分量子回路によってシミュレーションする方法を提案している。

もう1つはオプション価格が従う拡散方程式を、量子力学の基礎方程式であるシュレディンガー方程式 (波動方程式) に変換してプライシングするアプローチである。*Fontanela, Jacquier and Oumgari* [2021] と *Radha* [2021] は、変分量子回路を用いてオプション価格が従うシュレディンガー方程式を時間発展させてプライシングしている。一方、*Gonzalez-Conde et al.* [2022] は、変分量子回路による最適化を行わない量子回路によるプライシング方法を提案している⁵²。

その他に興味深いプライシングのアプローチがある。*Ramos-Calderer et al.* [2021] は、標準的な2進数表現ではなく、1進数表現⁵³による資産価格 $|\psi\rangle_{\text{unary}}$ を用いた QMCI によるオ

⁵⁰ 擬似乱数 (PRN) は、通常のモンテカルロ法において資産価格の動きをシミュレートするために用いられる。PRN タイプは通常のモンテカルロ法に類似した手法であり、アルゴリズムの説明については *Miyamoto and Shiohara* [2020] や *Kaneko et al.* [2021] を参照されたい。これら文献では、PRN タイプのアルゴリズムを、多次元のクレジット・ポートフォリオのリスク計測に適用している。

⁵¹ 実務では、ヨーロッパ・オプションの市場価格とモデル価格が等しくなるようにモデルのパラメータを調整する必要があり、このパラメータ調整はキャリブレーションと呼ばれる。

⁵² 非エルミートなハミルトニアンによる時間発展をユニタリ行列として埋め込む工夫を行っている。

⁵³ 例えば、3ビットに対して、2進数で符号化すると、 $0 = (000)_2, 1 = (001)_2, 2 = (010)_2, 3 = (011)_2, \dots, 7 = (111)$ となるが、1進数では、 $0 = (000)_1, 1 = (001)_1, 2 = (010)_1, 3 = (100)_1$ となる。この符号化方法は一般的な1進数表示とは異なるが、W状態と呼ばれるエンタングル状態の1つで、量子ビットの抜け落ちに対してロバストであるとされている。

オプションのプライシング方法を提案している。1進数表現によって、量子回路の構造や深さを簡素化できるため、複雑な計算ができないNISQデバイスに向いているとしている。

$$|\psi\rangle_{\text{binary}} = \sum_{i=0}^{2^n-1} \psi_i |i\rangle$$

$$= \psi_0 |00 \dots 01\rangle + \psi_1 |00 \dots 10\rangle + \dots + \psi_{2^{n-2}} |11 \dots 10\rangle + \psi_{2^{n-1}} |11 \dots 11\rangle, \quad (102)$$

$$|\psi\rangle_{\text{unary}} = \sum_{i=0}^{n-1} \psi_i |i\rangle = \sum_{i=0}^{n-1} \psi_i (\otimes_{j=0}^{n-1} |\delta_{ij}\rangle)$$

$$= \psi_0 |00 \dots 01\rangle + \psi_1 |00 \dots 10\rangle + \dots + \psi_{n-2} |01 \dots 00\rangle + \psi_{n-1} |10 \dots 00\rangle. \quad (103)$$

An *et al.* [2021] は、量子計算によるマルチレベル・モンテカルロ法 (MLMC : Multi-Level Monte Carlo) を提案している。マルチレベル・モンテカルロ法とは、異なる時間幅のサンプルを生成するモンテカルロ法であり、時間幅のレベルごとに適切なサンプリング回数を設定することで効率的な計算が可能になるとされる。

$$\mathbb{E}[P_L] = \sum_{l=0}^L \mathbb{E}[P_l - P_{l-1}] = \sum_{l=0}^L \left(\frac{1}{N_l} \sum_{i=0}^{N_l} (P_l^{l,i} - P_{l-1}^{l,i}) \right). \quad (104)$$

ここで、 P_l はレベル l でのペイオフ、 N_l はレベル l でのサンプル数である。量子MLMCによって計算量が $O(\epsilon^{-1-1/r})$ (r は確率微分方程式の強近似のオーダー) となることを示し、量子MLMCはオプションのプライシングやグリークス計算に応用できると述べている。

(2) リスク管理への応用

リスク管理手法に対する量子計算の応用研究を紹介する。金融機関では金融規制対応やポジション管理のために、バリュー・アット・リスク (VaR)、期待ショートフォール (CVaR : Conditional VaR)、グリークス、CVA (Credit Valuation Adjustment) 等のリスク指標の計測やストレステストを行っており、量子計算による高速化はビジネス上意味があると考えられる。これらリスク指標の計測において、古典モンテカルロ法で計算可能な部分を、QMCI (QAE) で置き換えるアプローチが多い。プライシングへの応用の場合と同様に、QMCI (QAE) の量子アルゴリズムの実行にはFTQCが前提となるため、古典コンピュータ上で可能なビット数程度の問題設定での応用研究が中心である。

リスク計測およびリスク管理手法への量子計算の応用例について紹介する。Woerner and Egger [2019] は、量子計算によるVaRやCVaRのリスク評価アルゴリズムを提案している。2分探索法とQAEを組み合わせて、損益分布からVaRやCVaRを推定している。Egger *et al.* [2021] は、Woerner and Egger [2019] の手法を適用して所要自己資本評価式におけるVaR推計部分の計算を行っている。

QMCIを用いたVaRとCVaRの推定アルゴリズムの要点を解説する。 X を損益とし、 $100(1-\alpha)\%$ を信頼水準とする時のVaRとCVaRはそれぞれ以下で定義される。ただし、 $f(x)$ は X の密度関数とし、 $\text{VaR}_\alpha(X)$ と $\text{CVaR}_\alpha(X)$ は X の正負の符号と合わせている。

$$\text{VaR}_\alpha(X)(= l_\alpha) := \inf\{x | \mathbb{P}[X \leq x] > \alpha\} \approx \inf\{x | \sum_{i=0}^{l_\alpha} p(i) > \alpha\}, \quad (105)$$

$$\text{CVaR}_\alpha(X) := \mathbb{E}[X | X \leq \text{VaR}_\alpha(X)] = \frac{1}{\alpha} \int_{-\infty}^{\text{VaR}_\alpha} x f(x) dx \approx \frac{l_\alpha}{\mathbb{P}[X \leq l_\alpha]} \sum_{i=0}^{l_\alpha} \frac{i}{l_\alpha} p(i). \quad (106)$$

先に VaR 推定の量子アルゴリズムから説明する。

- ① 量子ビット列 $|0\rangle^n$ に対して、損益分布 $p(x)$ の状態 $\sum_i \sqrt{p(i)} |i\rangle$ を生成する。
- ② 量子ビット $|i\rangle$ が適当な閾値 l 以下の場合に、補助量子ビット（最後尾）を1に反転させて、 $\sum_{i=l+1}^{2^n-1} \sqrt{p(i)} |i\rangle |0\rangle_{\text{anc}} + \sum_{i=0}^l \sqrt{p(i)} |i\rangle |1\rangle_{\text{anc}}$ を生成する。
- ③ 補助量子ビット $|1\rangle_{\text{anc}}$ に対して、QAEによる振幅推定と2分探索を繰り返して、分位点 α を満たす測定確率 $l_\alpha (= \text{VaR}_\alpha(X))$ を得る。

$$\begin{aligned} |0\rangle^n |0\rangle_{\text{anc}} &\rightarrow \sum_i \sqrt{p(i)} |i\rangle |0\rangle_{\text{anc}} \\ &\rightarrow \sum_{i=l+1}^{2^n-1} \sqrt{p(i)} |i\rangle |0\rangle_{\text{anc}} + \sum_{i=0}^l \sqrt{p(i)} |i\rangle |1\rangle_{\text{anc}} \\ &\xrightarrow{\text{QAE binary search}} l_\alpha (= \text{VaR}_\alpha(X)) \text{ s.t. } \sum_{i=0}^{l_\alpha} p(i) |i\rangle = \alpha \end{aligned} \quad (107)$$

今度は CVaR 推定の量子アルゴリズムを説明する。

- ① 補助量子ビットが $|1\rangle_{\text{anc}}$ かつ量子ビット列 $|i\rangle$ が l_α 以下の場合に、 $\sqrt{i/l_\alpha}$ 分の回転ゲートにより $\sum_{i=0}^{l_\alpha} \sqrt{1-i/l_\alpha} \sqrt{p(i)} |i\rangle |1\rangle_{\text{anc}} + \sum_{i=0}^{l_\alpha} \sqrt{i/l_\alpha} \sqrt{p(i)} |i\rangle |1\rangle_{\text{anc}}$ を生成する。
- ② QAEで振幅を推定して補助量子ビットが1となる確率 $\sum_{i=0}^{l_\alpha} (i/l_\alpha) p(i)$ を得る。
- ③ 最後にスケールリングして、 $\text{CVaR}(= l_\alpha / \mathbb{P}[X \leq l_\alpha]) \sum_{i=0}^{l_\alpha} (i/l_\alpha) p(i)$ を得る。

$$\begin{aligned} &\sum_{i=l_\alpha+1}^{2^n-1} \sqrt{p(i)} |i\rangle |0\rangle_{\text{anc}} + \sum_{i=0}^{l_\alpha} \sqrt{p(i)} |i\rangle |1\rangle_{\text{anc}} \\ &\xrightarrow{\text{rotation}} \sum_{i=l_\alpha+1}^{2^n-1} \sqrt{p(i)} |i\rangle |0\rangle_{\text{anc}} + \left(\sum_{i=0}^{l_\alpha} \sqrt{1-\frac{i}{l_\alpha}} \sqrt{p(i)} |i\rangle |1\rangle_{\text{anc}} + \sum_{i=0}^{l_\alpha} \sqrt{\frac{i}{l_\alpha}} \sqrt{p(i)} |i\rangle |1\rangle_{\text{anc}} \right) \\ &= \left(\sum_{i=l_\alpha+1}^{2^n-1} \sqrt{p(i)} |i\rangle + \sum_{i=0}^{l_\alpha} \sqrt{1-\frac{i}{l_\alpha}} \sqrt{p(i)} |i\rangle \right) |0\rangle_{\text{anc}} + \sum_{i=0}^{l_\alpha} \sqrt{\frac{i}{l_\alpha}} \sqrt{p(i)} |i\rangle |1\rangle_{\text{anc}} \\ &\xrightarrow{\text{QAE}} \sum_{i=0}^{l_\alpha} \frac{i}{l_\alpha} p(i) \xrightarrow{\text{scaling}} \frac{l_\alpha}{\mathbb{P}[X \leq l_\alpha]} \sum_{i=0}^{l_\alpha} \frac{i}{l_\alpha} p(i) \end{aligned} \quad (108)$$

Stamatopoulos *et al.* [2022] は、オプション取引のグリークス計算に対して Gilyén, Arunachalam and Wiebe [2019] による量子勾配推定アルゴリズムを応用してシミュレーションを行っている。Alcazar *et al.* [2022] は、NISQ デバイスによる量子古典ハイブリッド型の計算により、量子回路生成と振幅推定を繰り返して CVA の値を計算する方法を提案している。Skavysh *et al.* [2022] は、金融機関が被る期待損失を QMCI によって推計することでストレステストへの応用を提案している。

(3) ポートフォリオ最適化への応用

ファイナンス分野の最適化問題への量子計算の応用研究として、ポートフォリオ最適化を紹介する。ポートフォリオ最適化とはリスクやリターン等の基準を考慮して、最適な資産の組合せ（ポートフォリオ）を見つけることである。金融機関では、ポートフォリオ最適化の数理モデルを作成して、株式ポートフォリオ選択やアセットアロケーション等に用いている。投資対象の資産数や最適化の頻度が少なければ、計算量は問題とはならないが、膨大な投資対象から資産を選択する組合せ最適化問題を考える場合には、計算量が爆発的に増大する可能性がある。したがって、量子計算によるポートフォリオ最適化の高速化には意義があると考えられる。

最適化問題は、離散的な対象を扱う離散最適化（組合せ最適化）と連続的な対象を扱う連続最適化に分類され、それぞれ最適化のアプローチが異なる。量子計算の場合も同様で、最適化問題の分類に応じて、異なるアプローチが取られる。離散ポートフォリオの最適化に対しては、量子アニーリングや NISQ デバイスによる変分量子回路（QAOA、VQE 等）が用いられ、実用性を意識した問題設定のもとで各手法のパフォーマンス比較が行われている。一方、連続ポートフォリオの最適化に対しては、HHL を線形方程式のソルバとする最適化アルゴリズムが提案されている。HHL の実行には量子メモリ（qRAM）を持つ FTQC の実現が必要であるため、現状は、古典コンピュータ上でのシミュレーションや NISQ デバイスによる実証研究の段階にある。

離散ポートフォリオ最適化への応用例を紹介する。まず量子アニーリングを用いたポートフォリオ最適化の先駆的な研究である Rosenberg *et al.* [2016 b] は、離散的なポートフォリオに対して、ポートフォリオのリスクを最小化して期待リターンを最大化する最適化（平均分散モデル）を行い、高い確率で最適解が得られることを数値的に示している。Rosenberg and Rounds [2018] は、ショートポジションが可能となる条件のもとでのポートフォリオ最適化を行い、リスク・リターンが改善することを確認している。

離散ポートフォリオ最適化については、変分量子回路等を用いた実用性評価の研究も進められている。Hodson *et al.* [2019] は、ポートフォリオの多期間リバランス問題を解くために QAOA とその改良版である量子交代演算子アンザツ（QAOAz : Quantum Alternating Operator Ansatz）（Hadfield *et al.* [2019]）を用いてパフォーマンスを実験的に評価してい

る⁵⁴。少数銘柄の株式ポートフォリオの最適問題に対して、QAOA と QAOAz によって最適解に近い結果が得られると述べている。また、Slate *et al.* [2021] は、ランダム・ウォークの量子版の概念である量子ウォークによる最適化アルゴリズム (QWOA : Quantum Walk Optimization Algorithm) を提案し、QAOA よりも有意に良いパフォーマンスが達成できることを数値的に示している。高いパフォーマンスの背景として、量子ウォークのパスに対して、資産の組合せを符号化することで、QAOA に比べて探索空間を絞り込んで効率的にパラメータ探索を行うことができると主張している。Mugel *et al.* [2022] は、実データによるポートフォリオの動的最適化に対して、変分量子固有値ソルバ (VQE)、量子アニーリングやテンソル・ネットワーク⁵⁵等の計算方式を適用してデータサイズごとにパフォーマンスを比較している。結果、現在の量子デバイスの実装段階では、量子アニーリングやテンソル・ネットワークが大規模なデータを扱えると述べている。

QAOA による離散ポートフォリオ最適化へのアプローチについて要点を説明する。対象ポートフォリオのリターンを最大化し、リスクを最小化するアセットアロケーション問題を考える。資産 i を保有するかどうかを表す変数を $x_i \in \{0,1\}$ とすると、コスト関数 $C(\mathbf{x})$ は以下で表される。

$$C(\mathbf{x}) = -\sum_i x_i \mu_i + \gamma \sum_{i,j} x_i x_j \sigma_{ij} + \left(\sum_i x_i - N \right)^2. \quad (109)$$

ただし、 μ_i は資産 i の平均リターン、 σ_{ij} は資産 i, j のリターンの共分散、 γ はリスク選好度のパラメータとする。第3項は資産数 N を一定とするための制約条件である。このようにコスト関数を設定してQAOAによるパラメータ x_i の探索を行えば、平均分散アプローチを扱うことができる。なお、組合せ最適化問題の多くは $x_i \in \{0,1\}$ を変数とする2次制約なし2値最適化 (QUBO : Quadratic Unconstrained Binary Optimization) の形式で表現されるが、変数変換 $x = \frac{1}{2}(1+z)$, $z_i \in \{-1,1\}$ によってイジング・モデルに変換して用いる。

次に量子アルゴリズムの連続ポートフォリオ最適化への応用例を紹介する。Rebentrost and Lloyd [2018] は、連続ポートフォリオの最適化問題を、線形方程式で表現される非制約2次計画問題として定式化し、HHLを用いて線形方程式の解を求めるアルゴリズムを提案している。入力データの準備や結果の読み出し等、HHLの扱いには注意が必要であるが、ベストな古典アルゴリズム (共役勾配法) に対して指数加速になると主張している⁵⁶。Kerenidis, Prakash and Szilágyi [2019] は、ポジション制約や予算制約等の制約を設けた

⁵⁴ QAOA の場合、投資制約 (ネット・ロング数) $(\sum_i z_i - D)^2$ を含んだコスト関数を最小化する必要がある。一方、QAOAz の場合には、エンタングルを初期状態に取り入れ、ロング・ショート内でのスワップ操作によるパラメータ探索を行うことで、投資制約がないコスト関数の最小化を可能にしている。

⁵⁵ テンソル・ネットワークとは、テンソルを用いて表した情報を、小さなテンソルの縮約 (積) のネットワークとして表現し、グラフの形に図形化した概念である。量子回路をテンソル・ネットワークと見なすことで、高速に計算できる場合がある。テンソル・ネットワークは、量子技術に着想を得て既存技術の性能を高める古典アルゴリズムであり、量子インスパイア型アルゴリズムと呼ばれる。

⁵⁶ HHL とは、線形方程式の係数行列の固有値と固有ベクトルを効率的に計算して線形方程式の解を求める量子アルゴリズムである (4 節参照)。

連続ポートフォリオ最適化に対するアルゴリズムを提案している。ポートフォリオの最適化問題を、実行可能領域が2次錘となる凸計画問題として定式化し、HHLを線形方程式のソルバとして組み込んだ内点法 (Kerenidis and Prakash [2018]) を用いて解を求めるアルゴリズムである。ベストな古典アルゴリズムに対して多項式加速になるとされる。

NISQデバイス上でのHHLの実装に関する研究も進められている。Yalovetzky *et al.* [2022] は、NISQデバイス上でのHHLアルゴリズム (NISQ-HHL) を提案し、小規模ポートフォリオの最適化の実験に取り組んでいる。NISQ-HHLにおける量子位相推定 (QCL-QPE) は少ない量子ゲート数で実行できる利点があるとしている。

HHLを用いた連続ポートフォリオ最適化のアプローチについて要点を解説する。以下のポートフォリオのリスク最小化問題は、等式制約付きの関数極値問題として定式化されており、ラグランジュ未定乗数法によって解くことができる。関数 $L(\mathbf{w}, \eta, \theta)$ は線形方程式に変換できるため、HHLを適用することで、ポートフォリオのリスクを最小化する最適資産ウェイトを得ることができる。

$$\min_{\mathbf{w}} \sigma_p := \sum_{i,j=1}^N w_i w_j \sigma_{ij} \quad \text{s.t.} \quad \sum_{i=1}^N w_i \mu_i = \mu_p^*, \quad \sum_{i=1}^N w_i s_i = \xi, \quad (110)$$

$$\begin{aligned} L(\mathbf{w}, \eta, \theta) &= \sum_{i,j=1}^N w_i w_j \sigma_{ij} + \eta \sum_{i=1}^N (w_i \mu_i - \mu_p) + \theta \sum_{i=1}^N (w_i s_i - \xi) \\ &\rightarrow \begin{pmatrix} 0 & 0 & \mathbf{R}^T \\ 0 & 0 & \mathbf{S}^T \\ \mathbf{R} & \mathbf{S} & \Sigma \end{pmatrix} \begin{pmatrix} \eta \\ \theta \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mu_p \\ \xi \\ 0 \end{pmatrix}. \end{aligned} \quad (111)$$

ここで、関数 $L(\mathbf{w}, \eta, \theta)$ の第1項は最適化対象のポートフォリオのリスク、第2項はポートフォリオのリターンの制約条件、第3項は予算の制約条件である。また \mathbf{R}, \mathbf{S} は各資産のリターンベクトルと価格ベクトルであり、 Σ は各資産のリターンによる共分散行列である。

(4) その他最適化問題への応用

ポートフォリオ最適化以外の応用例を紹介する。先行研究のほとんどは、組合せ最適化問題として定式化されたファイナンスの問題に対して、量子アニーリング方式を適用するものである。Milne, Rounds and Goddard [2017] は、クレジット・スコアの計算モデルの特徴量 (変数) 選択アルゴリズムに対して量子アニーリングを用いている。相互に独立かつ影響力が大きい特徴量が選択されるように、2次制約なし2値最適化 (QUBO) 問題として定式化している。結果、同手法は、再帰的特徴量除去⁵⁷に対して、正確性を失わずに少

⁵⁷ すべての特徴量から始めてモデルを作成し、そのモデルにおける最も重要度が低い特徴量を削減していき、事前に定めた数の特徴量になるまで、モデル作成と特徴量削減を繰り返していく方法。

ない数の特徴量を選択できるとしている。Bouayoun [2019] は、XVA ポートフォリオのリバース・ストレステストに対して、量子アニーリングを適用している。XVA (X-Valuation Adjustment)⁵⁸とは、デリバティブ取引における価格調整の総称である。XVA の評価には複数のリスクファクターのモンテカルロ・シミュレーションが必要となるため一般に計算負荷が大きい。XVA による損失を最大化させるショック・シナリオの組合せを探索するために、量子アニーリング方式での計算を行っている。Rosenberg *et al.* [2016 a] はスワップ取引のネッティング問題や裁定機会の探索問題への量子アニーリングの適用を提案している。清算機関の立場におけるスワップ取引のネッティング対象の選択を、エクスポージャーの最小化問題として定式化している。Rosenberg [2016] では、取引資産と交換レートを、グラフ上のノードとエッジに見立てて、取引サイクルにおける交換レートの積を最大化する問題として定式化している。そのほか、Orús, Mugel and Lizaso [2019 a] は、金融危機の予測問題を金融ネットワークの均衡状態の崩壊を予測する問題と見なして量子アニーリングの適用を提案している。

6. 実務適用に向けた課題

本節では、量子計算の実務適用に向けた課題について論じる。プライシング、リスク管理、ポートフォリオ最適化の応用分野ごとに、量子計算の有効性と課題を整理する。

第1に、プライシングへの応用研究の動向と課題について述べる。これまで、簡単な問題設定（例：ブラック＝ショールズ・モデルを用いたヨーロピアン・オプション）でのプライシングが量子計算の応用研究の中心であったが、最近は実務的な問題設定（例：局所ボラティリティ・モデルや早期権利行使付きオプション）でのプライシングへの応用に研究テーマがシフトしてきている。プライシングに対する量子計算によるアプローチとしては、量子振幅推定 (QAE) や量子モンテカルロ積分 (QMCI) がプライシング計算のサブルーチンとして組み込まれることが多い。しかし、これら以外にも偏微分方程式を解いてオプション価格を得る方法や変量子回路で確率微分方程式をシミュレーションする方法等、量子計算による様々なアプローチが試みられている。

実務適用への課題として、古典モンテカルロ法の代替手法である QMCI の課題について述べる。古典モンテカルロ法は、擬似乱数を発生させて確率分布を生成し平均値を取る手法であり、一般に確率分布が陽に分からない場合に用いられる。しかし、現在の QMCI のアルゴリズムは、確率分布を所与と仮定している。例えば、オプションであれば原資産価格の確率分布は対数正規分布などが仮定される。そのため、古典モンテカルロ法を用いるのと同じ理由で、現状の QMCI をそのまま利用することはできない。したがって、確率分布を効率的に生成する量子アルゴリズム（例：qGAN (Zoufal, Lucchi and Woerner [2019])）の開発や QMCI を用いないプライシングのアプローチの検討が量子計算の実務適用におい

⁵⁸ XVA にはカウンターパー・ティリスク評価調整 (CVA)、自身のデフォルト・リスク評価調整 (DVA)、ファンディング評価調整 (FVA) 等があり、一部は実務適用されている。

て重要になると考えられる。また、既存のプライシング手法には古典モンテカルロ法より高いパフォーマンスを示している代替手法が存在するため、それとの差別化を考える必要がある。例えば、準モンテカルロ法は、QMCIと同程度の計算量で済むことが分かっている。さらに、実務の場面ではモンテカルロ法を用いずに偏微分方程式やツリー・モデルによってプライシングすれば十分な場合もある。そのため、量子アルゴリズムの改良・開発はもとより、量子アルゴリズムの有効な使い方を見つけ出して、古典アルゴリズムに対する優位性を示す必要があると考えられる。

第2に、リスク管理への応用研究の動向と課題について述べる。VaR、CVaR、グリークス、CVA やストレステストといった実務で用いられる主要なリスク計測・リスク管理手法に対して、既にいくつかの量子アルゴリズムが提案されている。多くの場合、QAE や QMCI がサブルーチンとして用いられ、小規模な問題設定のシミュレーションによって2次の量子加速が得られることが示されている。

プライシングの場合と同様に、QMCI アルゴリズムの課題は、確率分布を所与と仮定している点にある。例えば、VaR やストレステストでは事前に損益分布を仮定している。そのため、確率分布を効率的に生成する量子アルゴリズムなしに、現状の QMCI をそのまま利用することは難しい。また既存のリスク計測手法には古典モンテカルロ法以外の代替手法が存在する場合もある。例えば、過去データが十分にあればヒストリカル・シミュレーション法を用いることで少ない計算量で VaR が計算可能である。プライシングの場合と同様に、量子アルゴリズムの改良・開発と共に有効な使い方の模索が必要と考えられる。

第3に、ポートフォリオ最適化への応用研究の動向と課題について述べる。離散ポートフォリオの最適化に対しては、量子アニーリングや NISQ デバイスによる変分量子回路等、組合せ最適化に適用できる可能性があると考えられる手法が用いられており、比較的実用性のある問題設定のもとで、パフォーマンスや特性の評価が行われている。またテンソル・ネットワークや量子ウォーク最適化アルゴリズム等、新しい最適化アルゴリズムの開発も行われている。今後、より実用的な問題設定のもとでの検証が行われ、各アルゴリズムについての理解と改良が進むと考えられる。しかしながら、量子アニーリングや NISQ アルゴリズムにも注意が必要である。まず、これらは高速化の理論保証がないヒューリスティックなアルゴリズムであることが挙げられる。また、大規模な最適化問題を扱う場合に、調整すべきパラメータ数が増えることで、古典コンピュータ側の負担が増大することによる影響に十分に留意する必要がある。

連続ポートフォリオの最適化に対しては、HHL アルゴリズムを線形方程式のソルバとする最適化アルゴリズムが提案されている。HHL の活用によって計算の指数加速が期待されるが、HHL を実行するためには、初期量子状態の準備や計算結果の読み取りのための計算量がネックであり、実現には高度な FTQC が必要となる。また、HHL アルゴリズムの特性として、計算精度を維持するために、線形方程式の行列にスパース性が求められる。これ

ら HHL の欠点を解消するためのアルゴリズムの改良・開発、あるいは HHL が適している問題設定を考える必要がある。

現状では、どのファイナンス分野の応用先についても、実用化を阻むアルゴリズム上の課題が存在する。しかしながら、量子計算のファイナンス分野への応用研究は始まったばかりである。今後、ファイナンスへの応用研究が進めば、ブレイクスルーとなる新たな量子アルゴリズムや量子アルゴリズムの有効な使い方が提案される可能性がある。

7. まとめ／結論

本稿では、次世代の計算マシンとしての期待が高まっている量子コンピュータに関して、ファイナンス研究者や実務家向けに、量子計算の基礎や量子アルゴリズムを紹介し、金融商品のプライシング、リスク管理、ポートフォリオ最適化といったファイナンス分野への応用研究のサーベイを行った。

近年、将来の誤り耐性量子コンピュータ (FTQC) の実現を見越して、ファイナンス分野の問題を高速に解くための量子アルゴリズムが数多く提案されている。現時点では、ハードウェア・ソフトウェア (アルゴリズム含む) 両面での課題が多く残されており、量子計算による計算の高速化が実現するかどうかは未知数である。もっとも、量子アルゴリズムのメカニズムを理解することによって、将来的に、実務上も有用性の高いアルゴリズムの開発や実装に繋がる可能性も考えられる。本サーベイが理解深耕の一助になれば幸いである。

参考文献

- IBM, Qiskit, IBM, 2017年 (<https://qiskit.org/>, 2023年2月28日)
- QunaSys, Quantum Native Dojo!, QunaSys, 2019年
(<https://dojo.qulacs.org/ja/latest/index.html>, 2023年2月28日)
- 宇根正志・菅和 聖、「量子コンピュータ開発の進展と次世代暗号」、『金融研究』 第40巻第4号、日本銀行金融研究所、2021年、55～96頁
- 宇野隼平、「量子コンピュータを用いた高速数値積分」、2019年
研究開発戦略センター、「みんなの量子コンピューター ～情報・数理・電子工学と拓く新しい量子アプリ～」、CRDS-FY2018-SP-04、科学技術振興機構、2018年
- 清藤武暢・四方順司、「量子コンピュータが共通鍵暗号の安全性に与える影響」、『金融研究』、第38巻第1号、日本銀行金融研究所、2019年、45～72頁
- 西森秀稔（著）・大関真之（著）・須藤彰三（監修）・岡 真（監修）、『量子アニーリングの基礎』、共立出版、2018年
- 西村治道、『基礎から学ぶ量子計算: アルゴリズムと計算量理論』、オーム社、2022年
先端技術ラボ、「量子コンピュータの概説と動向 ～量子コンピューティング時代を見据えて～」、日本総合研究所、2020年
- 渡邊靖志、『入門講義 量子コンピュータ』、講談社、2021年
- 嶋田義皓（著）・情報処理学会出版委員会（監修）、『量子コンピューティング: 基本アルゴリズムから量子機械学習まで』、オーム社、2020年
- 統合イノベーション戦略推進会議、「量子技術イノベーション戦略 ロードマップ改訂」、内閣府、2022年 a
- 、「量子未来社会ビジョン ～量子技術により目指すべき未来社会ビジョンとその実現に向けた戦略～」、内閣府、2022年 b
- 藤井啓祐・市川 翼・山下 眞・山本 俊・根来 誠、「研究開発の動向」、『量子情報技術 科学技術に関する調査プロジェクト報告書』、国立国会図書館、2022年
- 藤吉栄二、「量子コンピュータ ～2030年に向けたロードマップ～」、第330回 NRI メディアフォーラム、野村総合研究所、2022年
- 武田俊太郎、『量子コンピュータが本当にわかる! — 第一線開発者がやさしく明かすしくみと可能性』、技術評論社、2020年
- 湊雄一郎・加藤拓己・比嘉恵一郎・永井隆太郎、『IBM Quantum で学ぶ量子コンピュータ』、秀和システム、2021年
- Albasha, Tameem, and Daniel A. Lidar, “Adiabatic Quantum Computation,” *Reviews of Modern Physics*, 90(1), 2018.
- Alcazar, Javier, Andrea Cadarso, Amara Katarbarwa, Marta Mauri, Borja Peropadre, Guoming Wang, and Yudong Cao, “Quantum Algorithm for Credit Valuation Adjustments,” *New Journal of Physics*, 24(2), IOP Publishing, 2022, 023036.

- An, Dong, Noah Linden, Jin-Peng Liu, Ashley Montanaro, Changpeng Shao, and Jiasu Wang, “Quantum-Accelerated Multilevel Monte Carlo Methods for Stochastic Differential Equations in Mathematical Finance,” arXiv:2012.06283, 2021.
- Arute, Frank, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis, “Quantum Supremacy Using a Programmable Superconducting Processor,” *Nature*, 574, 2019, pp. 505-510.
- Barends, Rami, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O’Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and John M. Martinis, “Superconducting Quantum Circuits at the Surface Code Threshold for Fault Tolerance,” *Nature*, 508, 2014, pp. 500-503.
- Bouayoun, Assad, “Quantitative Multi-period Reverse Stress Testing using Quantum and Simulated Annealing”, HSBC, 2019.
- Bouland, Adam, Wim van Dam, Hamed Joorati, Jordanis Kerenidis, and Anupam Prakash, “Prospects and Challenges of Quantum Finance,” arXiv:2011.06492, 2020.
- Brassard, Gilles, Peter Høyer, Michele Mosca, and Alain Tapp, “Quantum Amplitude Amplification and Estimation,” *Quantum Computation and Quantum Information*, American Mathematical Society, 2002, pp. 53-74.
- Castellanos, Sara, “Google Aims for Commercial-Grade Quantum Computer by 2029,” *The Wall Street Journal*, 2021 (available at <https://www.wsj.com/articles/google-aims-for-commercial-grade-quantum-computer-by-2029-11621359156>、2023 年 2 月 28 日).
- Cerezo, Marco, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio and Patrick J. Coles, “Variational Quantum Algorithms,” *Nature Reviews Physics*, 3(9), 2021, pp. 625-644.

- Chakrabarti, Shouvanik, Rajiv Krishnakumar, Guglielmo Mazzola, Nikitas Stamatopoulos, Stefan Woerner, and William J. Zeng, “A Threshold for Quantum Advantage in Derivative Pricing,” *Quantum*, 5, 2021, pp. 463.
- Deutsch, David, “Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer,” *Proceedings of the Royal Society of London A*, 400(1818), 1985, pp. 97-117.
- Doriguello, João F., Alessandro Luongo, Jinge Bao, Patrick Reberstrost, and Miklos Santha, “Quantum Algorithm for Stochastic Optimal Stopping Problems with Applications in Finance,” *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, 232, 2022.
- Duan, Bojia, Jiabin Yuan, Chao-Hua Yu, Jianbang Huang, and Chang-Yu Hsieh, “A Survey on HHL Algorithm: From Theory to Application in Quantum Machine Learning,” *Physics Letters A*, 384(24), 2020.
- Egger, Daniel J., Claudio Gambella, Jakub Marecek, Scott Mcfaddin, Martin Mevissen, Rudy Raymond, Andrea Simonetto, Stefan Woerner, and Elena Yndurain, “Quantum Computing for Finance: State-of-the-Art and Future Prospects,” *IEEE Transactions on Quantum Engineering*, 1, 2020.
- , Ricardo García Gutiérrez, Jordi Cahué Mestre, and Stefan Woerner, “Credit Risk Analysis using Quantum Computers,” *IEEE Transactions on Computers*, 70(12), 2021, pp. 2136-2145.
- Farhi, Edward, and Jeffrey Goldstone, “A Quantum Approximate Optimization Algorithm,” arXiv:1411.4028, 2014.
- , Jeffrey Goldstone, Sam Gutmann, and Michael Sipser, “Quantum Computation by Adiabatic Evolution,” arXiv: 0001106, 2000.
- Feynman, Richard P, “Simulating Physics with Computers,” *International Journal of Theoretical Physics*, 21(6-7), 1982, pp. 467-488.
- Fontanela, Fillipe, Antonine Jacquier, and Mugad Oumgari, “A Quantum Algorithm for Linear PDEs Arising in Finance,” *SIAM Journal on Financial Mathematics*, 12(4), 2021.
- Gilyén, András, Srinivasan Arunachalam, and Nathan Wiebe, “Optimizing Quantum Optimization Algorithms via Faster Quantum Gradient Computation,” *Proceedings of the Thirties Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '19*, 2019, pp.1425-1444.
- Giovannetti, Vittorio, Seth Lloyd, and Lorenzo Maccone, “Quantum Random Access Memory,” *Physical Review Letters*, 100, 2008, pp. 160501.
- Giurgica-Tiron, Tudor, Iordanis Kerenidis, Farrokh Labib, Anupam Prakash, and William Zeng, “Low Depth Algorithms for Quantum Amplitude Estimation,” *Quantum*, 6, 2022, pp. 745.
- Gómez, Andrés, Álvaro Leitao, Alberto Manzano, Daniele Musso, María R. Nogueiras, Gustavo Ordóñez, and Carlos Vázquez, “A Survey on Quantum Computational Finance for Derivatives Pricing and VaR,” *Archives of Computational Methods in Engineering*, Springer, 2022, pp.1-27.

- Gonzalez-Conde, Javier, Angel Rodriguez-Rozas, Enrique Solano, and Mikel Sanz, “Simulating Option Price Dynamics with Exponential Quantum Speedup,” arXiv:2101.04023, 2022.
- Grinko, Dmitry, Julien Gacon, Christa Zoufal, and Stefan Woerner, “Iterative Quantum Amplitude Estimation,” *npj Quantum Information*, 7(52), 2021.
- Grover, Lov K., “A Fast Quantum Mechanical Algorithm for Database Search,” *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*, 1996, pp. 212-219.
- , and Terry Rudolph, “Creating Superpositions that Correspond to Efficiently Integrable Probability Distributions,” arXiv:0208112, 2002.
- Hadfield, Stuart, Zihui Wang, Bryan O’Gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas, “From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz,” *Algorithms*, 12(2), 2019, pp. 34.
- Harrow, Aram W., Avinatan Hassidim, and Seth Lloyd, “Quantum Algorithm for Linear Systems of Equations,” *Physical Review Letters*, 103, 2009, pp. 150502.
- Herbert, Steven, “No Quantum Speedup with Grover-Rudolph State Preparation for Quantum Monte Carlo Integration,” *Physical Review E*, 103, 2021, pp. 063302.
- Herman, Dylan A., Cody Googin, Xiaoyuan Liu, Alexey Galda, Ilya Safro, Yue Sun, Marco Pistoia, and Yuri Alexeev, “A Survey of Quantum Computing for Finance,” arXiv:2201.02773, 2022.
- Hodson, Mark, Brendan Ruck, Hugh Ong, David Garvin, and Stefan Dulman, “Portfolio Rebalancing Experiments using the Quantum Alternating Operator Ansatz,” arXiv:1911.05296, 2019.
- IBM, “IBM Makes Quantum Computing Available on IBM Cloud to Accelerate Innovation,” *Cision*, 2016 (available at <https://www.prnewswire.com/news-releases/ibm-makes-quantum-computing-available-on-ibm-cloud-to-accelerate-innovation-300262512.html>、2023 年 2 月 28 日).
- , “IBM Unveils World’s First Integrated Quantum Computing System for Commercial Use,” *IBM Newsroom*, 2019 (available at https://newsroom.ibm.com/2019-01-08-IBM-Unveils-Worlds-First-Integrated-Quantum-Computing-System-for-Commercial-Use#assets_all、2023 年 2 月 28 日).
- , “Expanding the IBM Quantum roadmap to anticipate the future of quantum-centric supercomputing,” *IBM Blog*, 2022 (available at <https://research.ibm.com/blog/ibm-quantum-roadmap-2025>、2023 年 2 月 28 日).
- Jordan, Stephen. *Quantum Algorithm Zoo*, 2022 (available at <https://quantumalgorithmzoo.org/>、2023 年 2 月 28 日).
- Kadowaki, Tadashi, and Hidetoshi Nishimori, “Quantum Annealing in the Transverse Ising Model,” *Physical Review E*, 58(5), 1998, pp. 5355-5363.

- Kaneko, Kazuya, Koichi Miyamoto, Naoyuki Takeda, and Kazuyoshi Yoshino, "Quantum Speedup of Monte Carlo Integration with respect to the Number of Dimensions and its Application to Finance," *Quantum Information Processing*, 20(185), 2021.
- , Koichi Miyamoto, Naoyuki Takeda, and Kazuyoshi Yoshino, "Quantum Pricing with a Smile: Implementation of Local Volatility Model on Quantum Computer," *EPJ Quantum Technology*, 9(7), 2022.
- Kerenidis, Iordanis, Anupam Prakash, and Dániel Szilágyi, "Quantum Algorithms for Portfolio Optimization," *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, AFT'19, 2019, pp. 147-155.
- , and Anupam Prakash, "A Quantum Interior Point Method for LPs and SDPs," *ACM Transactions on Quantum Computing*, 1(1), 2020, pp. 1-32.
- Kitaev, Alexei Y., "Quantum Measurements and the Abelian Stabilizer Problem," *Electronic Colloquium on Computational Complexity*, TR96-003, 1996.
- Kubo, Kenji, Yuya O. Nakagawa, Suguru Endo, and Shota Nagayama, "Variational Quantum Simulations of Stochastic Differential Equations," *Physical Review A*, 103, 2021, pp. 052425.
- , Koichi Miyamoto, Kosuke Mitarai, and Keisuke Fujii, "Pricing Multi-Asset Derivatives by Variational Quantum Algorithms," arXiv:2207.01277, 2022.
- Li, Ying, and Simon C. Benjamin, "Efficient Variational Quantum Simulator Incorporating Active Error Minimization," *Physical Review X*, 7, 2017, pp. 021050.
- Liu, Yong, Fang Li, Xin Liu, Haohuan Fu, Yuling Yang, Jiawei Song, Pengpeng Zhao, Zhen Wang, Dajia Peng, Huarong Chen, Chu Guo, Heliang Huang, Wenzhao Wu, Dexun Chen, "Closing the "Quantum Supremacy" Gap: Achieving Real-Time Simulation of a Random Quantum Circuit Using a New Sunway Supercomputer," *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 3, 2021, pp. 1-12.
- Milne, Andrew, Maxwell Rounds, and Phil Goddard, "Optimal Feature Selection in Credit Scoring and Classification using a Quantum Annealer," White Paper, 1QBit, 2017.
- Mitarai, Kosuke, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii, "Quantum Circuit Learning," *Physical Review A*, 98(3), 2018, pp. 032309-1-032309-6.
- Miyamoto, Koichi, and Kenji Kubo, "Pricing Multi-Asset Derivatives by Finite Difference Method on a Quantum Computer," *IEEE Transactions on Quantum Engineering*, 3, 2021.
- , and Kenji Shiohara, "Reduction of Qubits in Quantum Algorithm for Monte Carlo Simulation by Pseudo-random Number Generator," *Physical Review A.*, 102, 2020, pp. 022424.
- , "Bermudan Option Pricing by Quantum Amplitude Estimation and Chebyshev Interpolation," *EPJ Quantum Technology*, 9(3), 2022.
- Montanaro, Ashley, "Quantum Speedup of Monte Carlo Methods," *Proceedings of the Royal Society A*, 8, 2015.

- Moore, Gordon E., “Progress in Digital Integrated Electronics,” *Electron Devices Meeting*, 21, . 1975, pp. 11-13.
- Mugel, Samuel, Carlos Kuchkovsky, Escolástico Sánchez, Samuel Fernández-Lorenzo, Jorge Luis-Hita, Enrique Lizaso, and Román Orús, “Dynamic Portfolio Optimization with Real Datasets Using Quantum Processors and Quantum-Inspired Tensor Networks,” *Physical Review Research*, 4, 2022, pp. 013006.
- Nielsen, Michael A., and Isaac L. Chuang, *Quantum Computation and Quantum Information 10th Anniversary Edition*, Cambridge University Press, 2010 (木村達也訳『量子コンピュータと量子通信』、オーム社、2004年) .
- Orús, Román, Samuel Mugel, and Enrique Lizaso, “Forecasting Financial Crashes with Quantum Computing,” *Physical Review A*, 99, 2019 a, pp. 060301.
- , Samuel Mugel, and Enrique Lizaso, “Quantum Computing for Finance: Overview and Prospects,” *Reviews in Physics*, 4, 2019 b, pp. 100028.
- Pednault, Edwin , John Gunnels, Dmitri Maslov, and Jay Gambetta, “On “Quantum Supremacy”,” *IBM Website*, 2019 (available at <https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/>、2023年2月28日).
- Peruzzo, Alberto, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Aspuru-Guzik, and Jeremy L. O'Brien, “A Variational Eigenvalue Solver on a Quantum Processor,” *Nature Communications*, 5(4213), . 2014. pp. 1-7.
- Pistoia, Marco, Syed Farhan Ahmad, Akshay Ajagekar, Alexander Buts, Shouvanik Chakrabarti, Dylan Herman, Shaohan Hu, et al, “Quantum Machine Learning for Finance,” *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, ICCAD Special Session Paper, IEEE, 2021.
- Preskill, John, “Quantum Computing Computing and the Entanglement Frontier,” arXiv:1203.5813, 2012.
- , “Quantum Computing in the NISQ Era and Beyond,” *Quantum*, 2, 2018, pp. 79.
- Radha, Santosh Kumar, “Quantum Option Pricing using Wick Rotated Imaginary Time Evolution,” arXiv:2101.04280, 2021.
- Ramos-Calderer, Sergi, Adrián Pérez-Salinas, Diego García-Martín, Carlos Bravo-Prieto, Jorge Cortada, Jordi Planagumà, and José I. Latorre, “Quantum Unary Approach to Option Pricing,” *Physical Review A*, 103, 2021, pp. 032414.
- Rebentrost, Patrick, and Seth Lloyd, “Quantum Computational Finance: Quantum Algorithm for Portfolio Optimization,” arXiv:1811.03975, 2018.
- , Brajesh Gupta, and Thomas R. Bromley, “Quantum Computational Finance: Monte Carlo Pricing of Financial Derivatives,” *Physical Review A*, 98(2), 2018.

- Rosenberg, Gili, “Finding Optimal Arbitrage Opportunities using a Quantum Annealer,” White Paper, IQBit, 2016.
- , Clemens Adolphs, Andrew Milne, and Andrew Lee, “Swap Netting using a Quantum Annealer,” White Paper, IQBit, 2016 a.
- , Poya Haghnegahdar, Phil Goddard, Peter Carr, Kesheng Wu, and Marcos López de Prado, “Solving the Optimal Trading Trajectory Problem Using a Quantum Annealer,” *IEEE Journal of Selected Topics in Signal Processing*, 10(5), 2016 b, pp. 1053-1060.
- , and Maxwell Rounds, “Long-Short Minimum Risk Parity Optimization Using a Quantum or Digital Annealer,” White Paper, IQBit, 2018.
- Schuld, Maria, and Francesco Petruccione, *Machine Learning with Quantum Computers*. Springer, . 2021 (大関真之・荒井俊太・篠島匠人・高橋茶子・御手洗光祐・山城 悠訳『量子コンピュータによる機械学習』、共立出版、2020年) .
- Shor, Peter W., “Algorithms for Quantum Computation: Discrete Logarithms and Factoring,” *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994.
- Skavysh, Vladimir, Sofia Priazhkina, Diego Guala, and Thomas R. Bromley, “Quantum Monte Carlo for Economics: Stress Testing and Macroeconomic Deep Learning,” Staff Working Paper, Bank of Canada, 2022.
- Slate, N., E. Matwiejew, S. Marsh, and J. B. Wang, “Quantum Walk-based Portfolio Optimisation,” *Quantum*, 5, 2021, pp. 513.
- Stamatopoulos, Nikitas, Daniel J. Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen, and Stefan Woerner, “Option Pricing using Quantum Computers,” *Quantum*, 4, 2020, pp. 291.
- , Guglielmo Mazzola, Stefan Woerner, and William J. Zeng, “Towards Quantum Advantage in Financial Market Risk using Quantum Gradient Algorithms,” *Quantum*, 6, 2022, pp. 770.
- Suzuki, Yohichi, Shumpei Uno, Rudy Raymond, Tomoki Tanaka, Tamiya Onodera, and Naoki Yamamoto, “Amplitude Estimation without Phase Estimation,” *Quantum Information Processing*, 19(75), Springer, 2020.
- Tang, Hao, Anurag Pal, Tian-Yu Wang, Lu-Feng Qiao, Jun Gao, and Xian-Min Jin, “Quantum Computation for Pricing the Collateralized Debt Obligations,” *Quantum Engineering*, 3(4), 2021, pp. e84
- Woerner, Stefan, and Daniel J. Egger, “Quantum Risk Analysis,” *npj Quantum Information*, 5, 2019. pp. 1-8.
- Yalovetzky, Romina, Pierre Minssen, Dylan Herman, and Marco Pistoia, “NISQ-HHL: Portfolio Optimization for Near-Term Quantum Hardware,” arXiv:2110.15958, 2022.
- Zoufal, Christa, Aurélien Lucchi, and Stefan Woerner, “Quantum Generative Adversarial Networks for Learning and Loading Random Distributions,” *npj Quantum Information*, 5(103), 2019.

Zyga, Lisa, “D-Wave Sells First Commercial Quantum Computer,” *Phys.org website*, 2011
(available at <https://phys.org/news/2011-06-d-wave-commercial-quantum.html>、2023 年 2 月 28
日).