量子コンピュータを用いた高速数値積分

宇野 隼平

Fast Numerical Integration Algorithms using Quantum Computer

Shumpei UNO

近年,高速計算が可能な次世代のコンピュータの候補として,量子コンピュータに大きな注目が集まっている.本稿では,量子コンピュータの今後の重要な応用先のひとつとして期待される,数値積分を実行する量子アルゴリズムについて紹介する.量子コンピュータを使用することで,これまでの代表的な数値積分手法(モンテカルロ積分)に較べて,ほぼ2乗の高速化が果たされる.本稿では,近年の著者らの研究を含めて,3種類の量子積分アルゴリズムを紹介する.

(キーワード): 量子コンピュータ, 量子アルゴリズム, 数値積分, 量子振幅推定, モンテカルロ積分

1 はじめに

近年、これまでのコンピュータとは異なる原理により動作する量子コンピュータに大きな注目が集まっている。これまで、IBM、Intel、Google等の大手ITメーカーの他、多くのベンチャー企業も量子コンピュータ開発への参入を表明しており、開発競争が本格化し始めて来ている[1]. 既に、小規模な量子コンピュータをクラウドを通じて利用する環境が提供されており[2,3]、今後の急速な発展に備えて、試験的に量子コンピュータを利用することも可能である。

現在利用可能な量子コンピュータは, まだ小規模 でノイズの多いものであるが、今後、もし大規模な 量子コンピュータが実現すれば、これまで計算時間 の関係により、解くことが難しかった問題を解決す ることが期待されており、現在、科学技術の様々な分 野で、 量子コンピュータの活用に向けた検討が盛ん に行われている. 例えば、慶應義塾大学には、量子コ ンピュータを用いた実用的なアプリケーションの研 究開発を、産学連携により行うことを目的に掲げた IBM Q ネットワークハブが設立され [4], 金融業界 からみずほフィナンシャルグループ, 三菱 UFJ フィ ナンシャル・グループ, 化学業界から JSR, 三菱ケミ カルの4社が参画している. 著者もみずほフィナン シャルグループの一員として本活動に参画し、様々な バックグラウンドを持つ研究者と議論することで,量 子コンピュータの将来的な応用方法について検討を

i サイエンスソリューション部 チーフコンサルタント 博士 (理学) 行っている.

量子コンピュータの重要な応用先の一つとして,多 次元数値積分が期待されている.数値積分は、例えば、 金融工学 [5, 6], 物理シミュレーション [7, 8, 9], 安全 工学 [10, 11], 量子化学計算 [12, 13] を始めとして, 社 会的・学術的に幅広い分野で必要不可欠な計算であ る. 数値積分は、これまでのコンピュータ(以下、「従 来型のコンピュータ」という)では、多くの場合、モン テカルロ法を用いて計算されるが、量子コンピュータ を用いることで、モンテカルロ法と較べて2乗の速度 向上が見込まれるアルゴリズムが知られている[14]. 近年, これらの数値積分の量子アルゴリズムを, 金融 工学を中心として, 具体的な問題へ適用するための 提案が行われている [15, 16, 17, 18]. このアルゴリ ズムの一部は、IBM がオープンソースとしている公 開している量子コンピュータ向けの SDK(Software Development Kit) の Qiskit [19] に, ライブラリとし て組み込まれており、データを与えるだけで、手軽に、 量子コンピュータの金融工学への応用を試験するこ とが出来る状況である.

これまで提案されている多くの量子数値積分アルゴリズムでは Brassard らの量子振幅推定法 [20] が用いられている. Brassard らの量子振幅推定法は、比較的多くの演算操作が必要なアルゴリズムであり、非常に高い量子コンピュータの性能が要求される.一方で、現在の量子コンピュータでは、ビットの量子状態を保つことの困難さ等に起因して、可能な演算操作の回数に制限がある. このため、なるべく早い時期に量子コンピュータを活用するためには、Brassard

らの量子振幅推定法 [20] に較べて演算操作の回数の 少ないアルゴリズムを開発することが重要な課題で ある.この課題に対する筆者らの最近の取り組み [21] についても本稿で簡単に紹介する.

本稿の構成は以下のとおりである。まず,2節では,従来型のコンピュータにおいて代表的な数値積分手法であるモンテカルロ法について紹介する。その後,3節において、量子コンピュータを用いた3種類の数値積分手法を紹介する。

2 従来型のコンピュータによる積分計算

本節では、数値積分問題の定式化及び従来型のコンピュータによる計算手法の概要について述べる. なお、ここでの数値積分問題の定式化は主に [22, 23] に従う.

d次元の数値積分問題は、有界な実関数 $g: \mathbb{R}^d \to \mathbb{R}$ の積分

$$I(g) = \int_{I_d} g(x)dx \tag{1}$$

が解析的に評価出来ない場合に、数値的に近似値を求める問題である.ここで I_d は d 次元空間内の有界な部分空間 $I_d \subset \mathbb{R}^d$ である.

式 (1) は一般性を失わずに, 上界, 下界から決まる 既知の定数によりスケールすることで, 以下の形に書 き直すことが出来る.

$$I(h) = \int_{\tilde{I}_d} \rho(x)h(x)dx \tag{2}$$

ここで, $\tilde{I}_d = [0,1]^d$ は d 次元超立方体, h は 0 から 1 の間に値域 ii をとる実関数 $h: \tilde{I}_d \rightarrow [0,1]$ であり, また, ρ は以下を満たす実関数である:

$$\int_{\tilde{I}_d} \rho(x) dx = 1 \tag{3}$$

式 (2) の近似値を求めるための最も単純なアルゴリズムは、積分の各次元を M 個に分割し、矩形近似することである. この時、積分 I(h) の近似は

$$I(h) \sim S(f) = \sum_{x \in J_d} p(x)f(x) \tag{4}$$

と和の形で計算することが出来る. ここで, J_d は M^d 個の整数格子点 $J_d = \{0,1,2,\cdots,M-1\}^d$ である.

また, p(x) は $\rho(x)$ を, \tilde{I}_d を M^d 個に等分割した各超立方体上で積分した関数であり, $\sum_{x\in J_d} p(x)=1$ を満たす. f(x) は, h(x) の格子点上での値である:

$$f(x_1, x_2, \cdots, x_d) = h\left(\frac{x_1}{M}, \frac{x_2}{M}, \cdots, \frac{x_d}{M}\right)$$
 (5)

1次元の積分区間を M 等分に分割した場合に、誤 差が $\varepsilon \sim O(M^{-1})$ で表せることから、式(4)を用 いて誤差 ε を達成するためには、被積分関数 f を $O(\varepsilon^{-d})$ 回程度評価する必要がある. 典型的な金融の 問題では, 積分の次元 d(金融資産の種類) は数百~数 千次元であるため、例えば、d=100として、誤差を $\varepsilon \sim O(10^{-3})$ 程度まで評価するとすると、積分の近似 値 S(f) を計算するために、被積分関数 f を $O(10^{300})$ 回程度評価する必要がある. これは、f が 1 回の浮 動小数点演算で計算できたとしても、1PFlops(1秒 間に 10^{15} 回の浮動小数点演算が可能) のスパコンを 使って, $O(10^{285})$ 秒 ~ $O(10^{278})$ 年かかる計算であ る. 現在の宇宙年齢が $O(10^{10})$ 年であることを鑑み ると, 矩形近似 (4) を使って, 積分計算を実行するこ とは実質上不可能である. この困難は主に、演算回数 が次元 d の増加に対して、指数関数的に増加する事に より生じている.

以上の事情により、高次元の積分を評価する場合には、式 (4) を直接評価することは無く、モンテカルロ法により評価を行うのが一般的である。モンテカルロ法では、式 (2) の p(x) を確率密度関数とみなすことで、式 (2) を統計的期待値と見なして計算を行う。例えば、p(x) に従う確率変数 x の N 個のサンプル x_1,x_2,\cdots,x_N が得られたとき、(2) の近似値は以下の式により与えられる。

$$I(h) \sim \frac{1}{N} \sum_{i=1}^{N} h(x_i) \tag{6}$$

Chebyshev の不等式により、サンプルの数 N(被積分関数 h の評価回数) と誤差 ε の関係は、 $N \sim O(\epsilon^{-2})$ となることが知られている [6]. モンテカルロ法は式(4) のような直接的な和の評価に較べて、遥かに効率的な手法であるが、一方で、実用上は、 $N \sim O(\epsilon^{-2})$ の値は大きくなることがあり (例えば、誤差を $\varepsilon \sim O(10^{-5})$ にしようと思えば、 $N \sim O(10^{10})$ 程度のサンプル数が必要になる)、モンテカルロ法による数値積分評価は様々な分野で多くの時間を占める計算の一つである.

以下で説明するように、量子コンピュータを用いたアルゴリズムでは、被積分関数 f の評価回数 N と

 $^{^{} ext{ii}}$ この段階では h の値域を [0,1] の範囲に制限する必要は無いが,後の量子アルゴリズムの際の便利のため,値域を設定した

誤差 ε の関係は $N \sim O(\epsilon^{-1})$ となり, 従来型のコンピュータのモンテカルロ計算と較べて, 効率的に数値積分を行うことが可能である.

3 量子アルゴリズム

本節では量子コンピュータを用いて数値積分を行う3種類のアルゴリズムについて紹介する.

まず、3.1 節において量子計算計算モデルの基本事項をごく簡単に述べる.次に、3.2 節において、現在広く用いられている Brassard らの手法 [20] を用いた数値積分手法について述べる. その後、3.3 節において、Brassard らの手法に較べて演算数が少ないと考えられる Abrams の手法 [22] について述べる. 最後に、3.4 節において、本稿の著者を含めた共同研究である、鈴木らの手法 [21] について述べる.

3.1 量子計算モデルの基本事項

ここでは、以下で用いる (ゲート型) 量子コンピュータの計算モデルについて、ごく簡単に述べる. より詳細な内容については、教科書 [24, 25] 等を参照されたい.

従来型のコンピュータでは、ビットは 0 又は 1 の確定した状態を取るが、量子コンピュータのビット (以下、「量子ビット」という) は重ね合わせ状態を取ることが出来る. より具体的には、量子ビットの状態 $|\phi\rangle$ は、2 次元ヒルベルト空間 $\mathcal{H}_2=\mathbb{C}^2$ 内の単位ベクトル

$$|\phi\rangle = \alpha \,|0\rangle + \beta \,|1\rangle \tag{7}$$

で表される. ここで, α , β は $|\alpha|^2 + |\beta|^2 = 1$ を満たす任意の複素数であり, \mathcal{H}_2 の正規直交基底 $|0\rangle$, $|1\rangle$ は計算基底と呼ばれる.

量子コンピュータの状態は、複数の量子ビットを合わせた状態として表される.具体的には、n 個の量子ビットを組み合わせた状態 $|\psi\rangle_n$ として、n 個の \mathcal{H}_2 のテンソル積の状態空間 $\mathcal{H}_2^{\otimes n}=\mathbb{C}^{2^n}$ 内の単位ベクトルとして表される.

$$|\psi\rangle_n = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle_n \tag{8}$$

ここで、 α_x は $\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$ を満たす 2^n 個の複素数の組であり、確率振幅と呼ばれる.また、 $|x\rangle$ は、計算基底 $|0\rangle$, $|1\rangle$ の n 個の積状態である.以下、 $x\in\{0,1\}^n$ を、非負の整数 $\{0,1,2,\cdots,2^{n-1}\}$ の 2 進数表記と見なすこととする.

量子コンピュータの演算は、量子コンピュータの状態 $|\psi\rangle_n$ を別の状態 $|\psi'\rangle_n$ に遷移させるユニタリ演算子 U として表される.

$$|\psi'\rangle_n = U |\psi\rangle_n \tag{9}$$

本稿では、量子コンピュータの計算は、初期状態 $|0\rangle_n$ に対して、様々なユニタリ演算子 U_1, U_2, \cdots を作用させることで進んでいくこととする. なお、本稿で用いるユニタリ演算子は付録 A にまとめる.

最後に、量子コンピュータの計算結果の読み出し (測定) について述べる. 本稿では、計算終了時の量子状態が $|\psi\rangle_n = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle_n$ で表されるとしたとき、計算結果の読み出しを行うことで、n ビット列 $x\in\{0,1\}^n$ が確率 $|\alpha_x|^2$ で得られることとする.

以下では、この計算モデルを用いた数値積分計算の 概要を述べる.

3.2 Brassard らの量子振幅推定法を用いた数値積分

本小節では、数値積分が量子振幅推定の問題に書き 直せることを利用して、数値積分を行う量子アルゴ リズムについて述べる.本小節の記載は主に [15] に 従う.

数値積分の量子アルゴリズムでは、式 (4) の数値積分の矩形近似値を算出することを目的とする. なお、以下では、表記の簡潔さのため、数値積分の次元 d を 1 次元として取り扱い、式 (4) の積分区間の分割数 $M=2^n$ 個とした近似値

$$S(f) = \sum_{x=0}^{2^{n}-1} p(x)f(x)$$
 (10)

を求めることとするが、任意の有限次元 d の積分への拡張は容易である.

付録 A の式 (58) 及び (59) から構成される演算子

$$\mathcal{A} = \mathcal{R}(\mathcal{P} \otimes \mathbf{I}_1) \tag{11}$$

を計算の初期状態 $\ket{0}_{n+1}$ に作用させると

$$|\Psi\rangle_{n+1} = \mathcal{A} |0\rangle_{n+1}$$

$$= \sum_{x=0}^{2^{n}-1} \sqrt{p(x)} |x\rangle_{n} \left(\sqrt{f(x)} |0\rangle + \sqrt{1 - f(x)} |1\rangle\right)$$
(12)

が得られる. 正規直交基底 $|\tilde{\Psi}_0\rangle_{n+1}$ 及び $|\tilde{\Psi}_1\rangle_{n+1}$ を

$$|\tilde{\Psi}_{0}\rangle_{n+1} = \sum_{x=0}^{2^{n}-1} \sqrt{p(x)} \sqrt{\frac{f(x)}{S(f)}} |x\rangle_{n} |0\rangle$$

$$|\tilde{\Psi}_{1}\rangle_{n+1} = \sum_{x=0}^{2^{n}-1} \sqrt{p(x)} \sqrt{\frac{1-f(x)}{1-S(f)}} |x\rangle_{n} |1\rangle$$
(13)

と導入すると、式 (12) の $|\Psi\rangle_{n+1}$ は

$$|\Psi\rangle_{n+1} = \sqrt{S(f)} |\tilde{\Psi}_0\rangle_{n+1} + \sqrt{1 - S(f)} |\tilde{\Psi}_1\rangle_{n+1}$$
(14)

と書き直すことが出来る. 式 (14) では, 数値積分の近似値 S(f) の平方根が $|\tilde{\Psi}_0\rangle_{n+1}$ の確率振幅として現れており, もし確率振幅を十分な精度で求めることが出来れば, 数値積分の近似値を求められることになる.

今,仮に,式 (14) の $|\Psi\rangle_{n+1}$ を繰り返し測定することで確率振幅を推定することを考える (以下,2 種類の測定結果を取る独立な試行を「ベルヌーイ試行」という). この時,ベルヌーイ分布の標準偏差の式により,サンプル数 N と誤差 ε の関係は, $N\sim O\left(\varepsilon^{-2}\right)$ であり,従来型のコンピュータによるモンテカルロ法と同程度のサンプル数が必要となる.このため,量子コンピュータを用いることによる加速を得るためには,確率振幅を推定するための何らかの工夫が必要となる.

単純なサンプリングに比べて効率的に確率振幅を推定する方法としては、Brassard らによって提案された手法が知られている [20]. 以下、モンテカルロ法における被積分関数 f の評価回数 N に対応して、量子計算における関数の呼び出し回数 N を式 (11) で定義される演算子 A の演算回数と定義する. この時、Brassard らの手法を用いた確率振幅の推定では、誤差 ε を達成するための演算子 A の呼び出し回数 N は、 $N\sim O(\varepsilon^{-1})^{\text{iii}}$ であり、モンテカルロ法のほぼ平方根程度の関数の呼び出し回数で、同程度の誤差を達成することが出来る. これは、誤差を $\varepsilon\sim O(10^{-5})$ にしようと思えば、 $N\sim O(10^{5})$ 程度の関数の呼び出し回数で良いこととなり、モンテカルロ法と較べて大幅な速度向上が見込まれる.

以下に、Brassard らの振幅推定法を用いた S(f) の推定方法の概要を述べる.

Brassard らの振幅推定法は,量子振幅増幅法 [20, 26] と位相推定法 [24] から構成される.以下,便宜の

ため, $\sqrt{S(f)}=\sin\theta, (0\leq\theta\leq\frac{\pi}{2})$ と置く. 式 (14) の $|\Psi\rangle_{n+1}$ に作用する, 量子振幅増幅法の演算子 ${\bf Q}$ は, 以下のように定義される.

$$\mathbf{Q} = \mathbf{U}_{\Psi} \mathbf{U}_{\tilde{\Psi}_{0}}$$

$$\mathbf{U}_{\tilde{\Psi}_{0}} = \mathbf{I}_{n} \otimes Z$$

$$\mathbf{U}_{\Psi} = \mathcal{A} \left(\mathbf{I}_{n+1} - 2 \left| 0 \right\rangle_{n+1} \left\langle 0 \right|_{n+1} \right) \mathcal{A}^{\dagger}$$
(15)

振幅増幅演算子 ${f Q}$ を式 (14) の $|\Psi\rangle_{n+1}$ に j 回作用させると、

$$\mathbf{Q}^{j} |\Psi\rangle_{n+1} = \sin((2j+1)\theta) |\tilde{\Psi}_{0}\rangle_{n+1} + \cos((2j+1)\theta) |\tilde{\Psi}_{1}\rangle_{n+1}$$
(16)

となり、 \mathbf{Q} は式 (12) で張られる部分空間 \mathcal{H}_{Ψ} 内の回転演算子として作用することがわかる. 部分空間 \mathcal{H}_{Ψ} 内における \mathbf{Q} の固有状態 $|\Psi_{+}\rangle$ は、

$$|\Psi_{\pm}\rangle_{n+1} = \frac{1}{\sqrt{2}} \left(|\tilde{\Psi}_1\rangle_{n+1} \mp i \, |\tilde{\Psi}_0\rangle_{n+1} \right)$$
 (17)

と構成することが可能であり、それぞれ固有値 $e^{\pm 2i\theta}$ に対応する. もし、振幅増幅演算子 ${\bf Q}$ の固有値 $e^{\pm 2i\theta}$ のどちらかを推定することが出来れば、積分の近似値 $S(f)=\sin^2\theta$ を計算出来ることとなる.

量子コンピュータでユニタリ演算子の固有値を求める方法として、量子位相推定法が知られている [24, 27]. 量子位相推定法は、ユニタリ演算子とその固有ベクトルが与えられた時に、効率的に固有値を算出する手法である. 量子位相推定法を適用するため、式 (14) の $|\Psi\rangle_{n+1}$ を、 \mathbf{Q} の固有ベクトル $|\Psi\pm\rangle_{n+1}$ で展開すると、

$$|\Psi\rangle_{n+1} = \frac{1}{\sqrt{2}} \left(e^{i\theta} |\Psi_+\rangle_{n+1} + e^{-i\theta} |\Psi_-\rangle_{n+1} \right) (18)$$

と書き直せる.

次に, $|\Psi_{n+1}\rangle$ に使用する n+1 ビットに加えて,新たに m 個の補助ビットを追加した量子状態 $|0\rangle_m |\Psi\rangle_{n+1}$ を考える.量子位相推定法では,この追加した m 個の補助ビットに位相の推定結果が出力される.まず,状態 $|0\rangle_m |\Psi\rangle_{n+1}$ の m 個の補助ビットに対して,Hadamard 演算子を作用させると,量子状態は

$$\begin{split} &H^{\otimes m}\left|0\right\rangle_{m}\left|\Psi\right\rangle_{n+1}\\ &=\frac{1}{\sqrt{2^{m}}}\sum_{u=0}^{2^{m}-1}\left|y\right\rangle_{m}\left(e^{i\theta}\left|\Psi_{+}\right\rangle_{n+1}+e^{-i\theta}\left|\Psi_{-}\right\rangle_{n+1}\right) \end{split}$$

iii 本稿の計算量オーダーは、対数の因子を除いたものである

(19)

となる. 式 (19) の量子状態に, m 量子ビット $|y\rangle_m$ を制御ビットとする制御ユニタリ演算子 $^{\mathbf{c}}\mathbf{Q}$ を作用させると,

$${}^{\mathbf{c}}\mathbf{Q}\frac{1}{\sqrt{2^{m}}}\sum_{y=0}^{2^{m}-1}|y\rangle_{m}\left(e^{i\theta}|\Psi_{+}\rangle_{n+1}+e^{-i\theta}|\Psi_{-}\rangle_{n+1}\right)$$

$$=\frac{e^{i\theta}}{\sqrt{2}}\sum_{y=0}^{2^{m}-1}e^{2iy\theta}|y\rangle_{m}|\Psi_{+}\rangle_{n+1}$$

$$+\frac{e^{-i\theta}}{\sqrt{2}}\sum_{y=0}^{2^{m}-1}e^{-2iy\theta}|y\rangle_{m}|\Psi_{-}\rangle_{n+1}$$
(20)

が得られる. ここで、周波数解析等を思い起こして、離散フーリエ変換を行うことで振動数を取り出せることの類推から、式 (20) の m 量子ビットに付録 A の式 (60) の量子逆フーリエ変換演算子 F_m^{-1} を作用させた量子状態

$$\frac{e^{i\theta}}{\sqrt{2}} \sum_{y=0}^{2^{m}-1} e^{2iy\theta} F_{m}^{-1} |y\rangle_{m} |\Psi_{+}\rangle_{n+1}
+ \frac{e^{-i\theta}}{\sqrt{2}} \sum_{y=0}^{2^{m}-1} e^{-2iy\theta} F_{m}^{-1} |y\rangle_{m} |\Psi_{-}\rangle_{n+1}
= \frac{e^{i\theta}}{\sqrt{2}} \sum_{x=0}^{2^{m}-1} \alpha_{+}(x) |x\rangle_{m} |\Psi_{+}\rangle_{n+1}
+ \frac{e^{-i\theta}}{\sqrt{2}} \sum_{x=0}^{2^{m}-1} \alpha_{-}(x) |x\rangle_{m} |\Psi_{-}\rangle_{n+1}$$
(21)

の係数 $\alpha_+(x)$ 及び $\alpha_-(x)$ は, $2^m \frac{\theta}{\pi}$ 及び $2^m \left(1 - \frac{\theta}{\pi}\right)$ 付近にピークを持つ関数であると予想できる.もし,この付近に十分高いピークを持つとすれば,m 個の補助ビットの量子状態 $|x\rangle_m$ を観測すれば, θ または $\pi - \theta$ の近似値を得ることが出来るため, $S(f) = \sin^2(\theta) = \sin^2(\pi - \theta)$ から,数値積分の近似値を推定することが出来る.

実際,式 (21) の状態に対して,m ビット量子状態 $|x\rangle$ を観測したときに, θ または $\pi-\theta$ が $\pm\pi/2^m$ 以内の誤差で得られる確率の下限値 $\left(\frac{8}{\pi^2}\sim81\%\right)$ が, Brassard ら [20] により求められている。同様に,k を適当な整数とした時, θ または $\pi-\theta$ が $\pm\pi k/2^m$ 以内の誤差で得られる確率の下限は, $\frac{8}{\pi^2}\sum_{i=1}^k\frac{1}{(2i-1)^2}$ と求めることが出来る (例えば,k=5 とすれば成功確率を 95% 以上)。 θ または $\pi-\theta$ が $\pm\pi k/2^m$ 以内

の誤差で得られたとした時, S(f) の推定値 $\hat{S}(f)$ の推定誤差は, 誤差の伝播により,

$$|S(f) - \hat{S}(f)| \le 2\pi k \frac{\sqrt{S(f)(1 - S(f))}}{2^m} + k^2 \left(\frac{\pi}{2^m}\right)^2$$
(22)

と求められる.

以上の手法において、演算子 A の呼び出し回数 N は、演算子 \mathbf{Q} の中に A が 2 回含まれることを考慮すると、 $N=2^{m+1}-1$ 回となる. よって、この手法における誤差 $\varepsilon=|S(f)-\hat{S}(f)|$ と関数 f の呼出回数 N の関係は $N\sim O\left(\varepsilon^{-1}\right)$ であることがわかる. なお、以上の説明では簡単のため、数値積分の次元 d=1 としたが、以上の計算過程から明らかなように、量子振幅増幅法においても、モンテカルロ法と同様に、関数の呼出回数 N は次元 d に依存しないことに留意する.

以上で概要を与えた Brassard ら [20] の振幅推定 法では、多くの制御ユニタリ演算子 $^{\circ}\mathbf{Q}$ を用いている. 現在、利用可能な量子コンピュータ [2、3] では \mathbf{Q} 等の単なるユニタリ演算子に較べて、 $^{\circ}\mathbf{Q}$ 等の制御ユニタリ演算子は演算数が多く、ノイズが生じやすい傾向にあり、この傾向は今後もしばらくは続くと考えられる. このため、近い将来の量子コンピュータの活用に向けて、制御ユニタリ演算子 $^{\circ}\mathbf{Q}$ をなるべく使用しないアルゴリズムの開発は重要な課題である. 以下、制御ユニタリ演算子 $^{\circ}\mathbf{Q}$ を使わずに量子振幅を行う2種類のアルゴリズムを紹介する.

3.3 Abrams らの量子振幅推定法を用いた数値積分

本小節では、Abramsらの方法 [22] に基づき、量子位相推定法を使用せずに、量子振幅増幅法のみで振幅推定を行い、式 (10) の S(f) を求める手法について述べる。この手法では、量子位相推定を使わないため、3.2 節の方法で使われた制御ユニタリ演算子 $^{\mathbf{c}}\mathbf{Q}$ や量子逆フーリエ変換 F_m^{-1} を使う必要が無く、演算数の削減が期待される。ただし、後述するように、Abramsらの方法では、1 種類の振幅増幅演算子 \mathbf{Q} のみでなく、測定結果に依存して、様々な演算子を用いて振幅増幅を行う必要があることに留意する。

Abrams らの振幅推定法の大まかな概念を図 1 に示す。この手法は、Iteration によって、徐々に正確な振幅を推定していくアルゴリズムである。まず、S(f) を埋め込んだ初期状態に対して、ベルヌーイ試行を繰り返し、振幅 S(f) の大きさを適当な誤差範囲で定める。次に、この誤差範囲を振幅増幅法で拡大した上

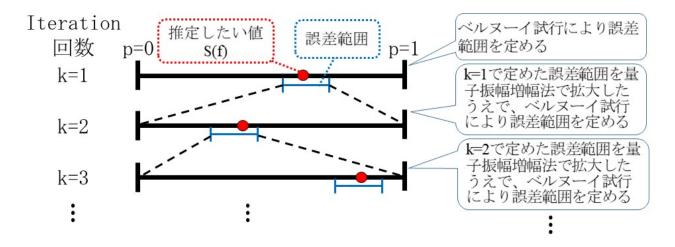


図1 Abrams らの振幅推定法 [22] の概念図. ベルヌーイ試行による誤差範囲の設定と, 振幅増幅法による誤差範囲の拡大を繰り返す Iteration を行うことで, 正確な振幅の推定値を求めていく.

で、ベルヌーイ試行を行い、より正確な誤差範囲を定める。 Abrams らの手法は、この誤差範囲の設定と振幅増幅法を繰り返す Iteration を行うことで、徐々に正確な振幅の推定値を求めていく手法である。 なお、後述するように、この手法において、関数の評価回数 N と誤差 ε の関係は、 $N\sim O\left(\varepsilon^{-1}\right)$ と Brassard らの振幅推定法と同程度である。

本小節では、まず、3.3.1 節において、Iteration の 1 回目のアルゴリズムを示し、続いて、3.3.2 節において k 回目の Iteration のアルゴリズムを示す。 最後に、3.3.3 節において、関数の評価回数 N と誤差 ε の関係式を導く.

3.3.1 Iteration の 1 回目

本手法では、関数 f(x) を計算するため、付録 A の式 (59) と類似した以下の演算子を仮定する.

$$\mathcal{R}_f |x\rangle_n |0\rangle = |x\rangle_n \left(f(x) |0\rangle + \sqrt{1 - f(x)^2} |1\rangle \right)$$
(23)

式 (23) と付録 A の式 (58) から構成される演算子

$$\mathcal{B}_f = (\mathcal{P}^{\dagger} \otimes \mathbf{I}_1) \mathcal{R}_f (\mathcal{P} \otimes \mathbf{I}_1)$$
 (24)

を計算の初期状態 $|0\rangle_{n+1}$ に作用させると

$$|\Psi_1\rangle_{n+1} = \mathcal{B}_f |0\rangle_{n+1}$$

= $\sum_{x=0}^{2^n-1} p(x)f(x) |0\rangle_{n+1} + \cdots$ (25)

が得られ, $|0\rangle_{n+1}$ の確率振幅として, S(f) が得られる.

ここで、式 (25) の量子状態 $|\Psi_1\rangle$ に対して、 N_1 回のベルヌーイ試行をした場合に、S(f) の推定値 $\hat{S}_1(f)$ がどの程度の誤差範囲で求まるかを算出する. 試行回数 N、確率 p のベルヌーイ分布に従う確率変数の誤差範囲 ε_p が、N が十分大きい場合に、適当な定数 c_1 と標準偏差 $\sqrt{\frac{p(1-p)}{N}}$ を用いて、

$$\varepsilon_p = c_1 \sqrt{\frac{p(1-p)}{N}} \tag{26}$$

と表されることを考慮すると、量子状態 $|\Psi_1\rangle$ を N_1 回測定した場合の $S(f)^2$ の推定誤差は、

$$c_1 \sqrt{\frac{S(f)^2 (1 - S(f)^2)}{N_1}} \tag{27}$$

で与えられる. 誤差の伝播公式を考慮すれば, 推定値 $\hat{S}_1(f)$ の推定誤差 $arepsilon_1$ は

$$\varepsilon_1 = \frac{c_1}{2} \sqrt{\frac{1 - S(f)^2}{N_1}} \le \frac{c_1}{2} \sqrt{\frac{1}{N_1}}$$
(28)

と求められる.

3.3.2 節で述べるように、Amrams らの手法では、こうして求められた誤差範囲 $\hat{S}(f)-\frac{\epsilon_1}{2} \leq S(f) \leq \hat{S}_1(f)+\frac{\epsilon_1}{2}$ を振幅増幅法を用いて拡大することで、更に精度のよい推定を行う手法である.

式 (28) により, ε_1 の値と c_1 の値を定めれば, Iteration 内で必要なベルヌーイ試行の回数 N_1 を求めることが出来る. c_1 の値は, 3.3.3 節で議論するように本手法の成功確率が有限になるように定める必要がある. また, 設定する誤差 ε_1 は, 最終的に求めたい誤差 ε に較べて遥かに大きい値をとることに留意する

(例えば, 最終的に求めたい誤差 $\varepsilon \sim 10^{-5}$ に対して $\varepsilon_1 = \frac{1}{2}$ など).

なお、以下では簡単のため、全ての Iteration でベルヌーイ試行の推定誤差を一定の値 ε_0 に設定するとして議論を進めていく.

3.3.2 Iteration の k 回目

ここでは、Abrams らの振幅推定手法 [22] の k 回目の Iteration の手続きを示す.

Iteration の k-1 回目までで、S(f) の推定値 $\hat{S}_{k-1}(f)$ が誤差 ε_0^{k-1} の範囲で求まっていると仮定する.この時、以下の手続きを行うことで、Iteration の k 回目において、ほぼ定数回のベルヌーイ試行を行うことで、振幅 S(f) の推定値 $\hat{S}_k(f)$ を誤差 ε_0^k の範囲で求めることが出来ることを示す.

Iteration の k 回目の手続きでは、 $\hat{S}_{k-1}(f)$ - $\frac{\varepsilon_0^{k-1}}{2} \leq S(f) \leq \hat{S}_{k-1}(f) + \frac{\varepsilon_0^{k-1}}{2}$ の範囲の確率振幅を振幅増幅法で拡大することで推定値 $\hat{S}_k(f)$ を求める.この目的のため、まず、以下の量 $f_k(x)$ 及び $D_k(f)$ を定義する:

$$f_k(x) = f(x) - \left(\hat{S}_{k-1}(f) - \frac{\varepsilon_0^{k-1}}{2}\right)$$

$$D_k(f) = \sum_{x=0}^{2^n - 1} p(x) f_k(x)$$
(29)

もし、こうして定義した $D_k(f)$ の推定値 $\hat{D}_k(f)$ を、誤差 ε_0^k の範囲で求めることが出来れば、S(f) の推定値 $\hat{S}_k(f)$ を誤差 ε_0^k で以下のようにして求めることが出来る:

$$\hat{S}_k(f) = \hat{S}_{k-1}(f) - \frac{\varepsilon_0^{k-1}}{2} + \hat{D}_k(f)$$
 (30)

式 (29) の $D_k(f)$ を推定するため,

$$\mathcal{R}_{f_k} |x\rangle_n |0\rangle = |x\rangle_n \left(f_k(x) |0\rangle + \sqrt{1 - f_k(x)^2} |1\rangle \right)$$
(31)

と付録 A の式 (58) から構成される演算子

$$\mathcal{B}_{f_k} = (\mathcal{P}^{\dagger} \otimes \mathbf{I}_1) \mathcal{R}_{f_k} (\mathcal{P} \otimes \mathbf{I}_1) \tag{32}$$

を定義する. 式 (32) を計算の初期状態 $|0\rangle_{n+1}$ に作用させると

$$|\Psi_{k}\rangle_{n+1} = \mathcal{B}_{f_{k}} |0\rangle_{n+1}$$

$$= \sum_{x=0}^{2^{n}-1} p(x) f_{k}(x) |0\rangle_{n+1} + \cdots$$
(33)

が得られ, $|0\rangle_{n+1}$ の確率振幅として, $D_k(f)$ が得られる.以下,被積分関数 f(x) の計算が引き算に較べて遥かに複雑であるとし, \mathcal{B}_f と \mathcal{B}_{f_k} の総演算回数を被積分関数 f の評価回数 N として数えることとする.

ここで, $0 \leq D(f) \leq \varepsilon_0^{k-1}$ であることを考慮して, 式 (33) の $|0\rangle_{n+1}$ の振幅増幅を行う. 以下, 便宜のため, $\sqrt{D_k(f)} = \sin\theta_k, (0 \leq \theta_k \leq \frac{\pi}{2})$ と置く. 振幅増幅演算子 \mathbf{Q}_k を

$$\mathbf{Q}_{k} = \mathbf{U}_{\Psi_{k}} \mathbf{U}_{\tilde{\Psi}_{k0}}$$

$$\mathbf{U}_{\tilde{\Psi}_{k0}} = \mathbf{I}_{n+1} - 2 |0\rangle_{n+1} \langle 0|_{n+1}$$

$$\mathbf{U}_{\Psi_{k}} = \mathcal{B}_{k} \left(\mathbf{I}_{n+1} - 2 |0\rangle_{n+1} \langle 0|_{n+1} \right) \mathcal{B}_{k}^{\dagger}$$
(34)

と定義すると、振幅増幅演算子 \mathbf{Q}_k を式 (33) の $|\Psi_k\rangle_{n+1}$ に j_k 回作用させることで、

$$\mathbf{Q}_{k}^{j_{k}} |\Psi\rangle_{n+1} = \sin((2j_{k}+1)\theta_{k}) |0\rangle_{n+1} + \cdots (35)$$

と, $|0\rangle_{n+1}$ の振幅を増幅させることが出来る. j_k の値は, 振幅が最も増幅するように,

$$(2j_k + 1)\sin^{-1}\varepsilon_0^k \sim \frac{\pi}{2} \tag{36}$$

と選ぶこととする. 式 (35) の量子状態に対して N_k 回の測定を行った時の $\sin^2((2j_k+1)\theta_k)$ の推定値がベルヌーイ分布に従うことから, 誤差の伝播公式を考慮すると, $D_k(f)=\sin^2\theta_k$ の推定誤差は, 適当な定数 c_k を用いて,

$$\varepsilon_k = \frac{c_k \cos\left(\frac{\sin^{-1}(D_k(f))}{2j_k + 1}\right)}{2(2j_k + 1)\sqrt{N_k}} \le \frac{c_k}{2(2j_k + 1)\sqrt{N_k}}$$
(37)

と表される.

式 (36) から, ε_0^k が十分小さければ,

$$j_k \sim O\left(\varepsilon_0^{-k}\right) \tag{38}$$

なので、誤差範囲を $\varepsilon_k=\varepsilon_0^k$ にするために必要なベルヌーイ試行の回数 N_k は、式 (37) から、k が大きい場合には、 c_k の因子を除いて、ほぼ定数 $^{\mathrm{iv}}$ になることがわかる.

以上により、ほぼ定数回のベルヌーイ試行により、k回目の Iteration において、S(f) の推定値 $\hat{S}_k(f)$ を誤差 ε_0^k で求める手続きを示した.

iv ここでは, k や ε に依存しないという意味での定数

3.3.3 関数の呼出回数と誤差の関係

ここでは、以上により概要を示した Abrams の振幅推定法 [22] における関数 f の呼出回数 N と誤差 ε の関係が、(対数因子を除いて) $N\sim O\left(\varepsilon^{-1}\right)$ により与えられることを示す。

m 回目の Iteration により、最終的に要求される精度 $\varepsilon_0^m = \varepsilon$ に達したとする.この時、Abrams の振幅推定法における関数 f の総呼び出し回数は、k 回目の Iteration における振幅増幅の回数 j_k とベルヌーイ試行の回数 N_k により、以下のように与えられる.

$$N = \sum_{k=1}^{m} (2j_k + 1)N_k \tag{39}$$

ここで、 j_k の ε 依存性は、式 (38) により、 j_k ~ $O\left(\varepsilon_0^{m-k}\varepsilon^{-1}\right)$ で与えられる. 以下、残りの部分、ベルヌーイ試行の回数 N_k と誤差 ε の関係を示す.

この目的のため、m 回目の Iteration の後の S(f) の推測値 $S_m(f)$ が、誤差 ε 以内に入る確率が一定の値 C_p 以上であることを要請するとする。簡単のため、各 Iteration 内の失敗確率を一定の値 $P_{\rm fail}$ となるとする。この時、m 回の Iteration 後の成功確率が、 C_p 以上であるとすると

$$(1 - P_{\text{fail}})^m \ge C_p \tag{40}$$

が成り立つ. ここで, 平均値 μ , 標準偏差 σ に従う確率変数 X に対する Chebyshev の不等式

$$P_{\text{fail}}(|X - \mu| \ge c_k \sigma) \le \frac{1}{c_k^2} \tag{41}$$

を考慮すると,式 (40) を満たすためには次の関係式 が成り立てば良いことがわかる.

$$\frac{1}{m}\ln\left(\frac{1}{C_p}\right) \sim \frac{1}{c_\nu^2} \tag{42}$$

ここで、各 Iteration の誤差 ε_0 と最終的な誤差 ε の 関係式

$$\varepsilon_0^m = \varepsilon \tag{43}$$

及び k 回目の Iteration におけるベルヌーイ試行の 誤差

$$\varepsilon_0 = c_k \sqrt{\frac{\sin^2((2j_k+1)\theta_k)\cos^2((2j_k+1)\theta_k)}{N_k}}$$
 (44)

を用いると式 (42) から、ベルヌーイ試行の回数 N_k と誤差 ε の関係が $N_k \sim O(\ln(\varepsilon^{-1}))$ 程度であることがわかる.以上により、f の全評価回数 N と最終的な誤差 ε の関係は、対数の因子を除いて $N \sim O(\varepsilon^{-1})$ 程度であることがわかる.

以上に示した Abrams らの振幅推定法 [22] は,量子位相推定法を使わない手法であるため,Brassard らの手法 [20] で必要であった制御ユニタリ演算子 $^{\mathbf{Q}}$ や逆フーリエ変換演算子 F_m が不要となっている.このため,Abrams らの手法は,近い将来の量子コンピュータの活用に向けて,Brassard らの手法と較べて,有効な手法である.一方で,(32)のように,測定結果に応じて,振幅増幅演算子 \mathbf{Q}_k を変更する必要があるため,実用上,演算子の構成に伴う困難が生じる可能性がある.以下,3.4 節では,1 種類の振幅増幅演算子のみを用いて,振幅推定を行う鈴木らの最近の研究 [21] について述べる.

3.4 鈴木らの振幅推定法を用いた数値積分

この小節では、鈴木らによる、振幅増幅法と最尤推定を組み合わせた振幅推定法 [21] について述べる. なお、本研究は、慶應義塾大学の IBM Q ネットワークハブにおいて、本稿の著者を含めた共同研究として行った研究である.

鈴木らの振幅推定法の概念図を図 2 に示す.大まかに言えば,鈴木らの手法は,推定したい量 S(f) と相関するような様々な量子状態について測定を行い,得られる情報を組み合わせることで,S(f) を推定する手法である.より具体的には,振幅増幅法を用いてS(f) に相関する様々な量子状態を作り,得られた測定結果から最尤推定により,S(f) の推定を行う手法である.

以下, 3.4.1 節において本手法の概要を, 3.4.2 節において本手法の関数の呼出回数と誤差の関係を示す.

3.4.1 手法の概要

以下,鈴木らの手法を用いて,式 (14) の $|\tilde{\Psi}_0\rangle_{n+1}$ の確率振幅 $\sqrt{S(f)}=\sin\theta$ を推定する方法を述べる. 本手法においても、3.3 節において概要を説明した Abrams らの手法 [22] と同様に、Iteration により、 $\sqrt{S(f)}$ を求める。k 回目の Iteration v^i では、式 (14)

v 例えば $\frac{1}{3}$ 等の適当な定数

 $^{^{\}mathrm{vi}}$ k 回目の振幅増幅回路の実行に際しては k-1 回目より前の結果を使用しないため、独立に実行することが可能であり、正確には Iteration では無いが、ここでは Abrams らの手法と対比させるため、Iteration と呼ぶこととした.

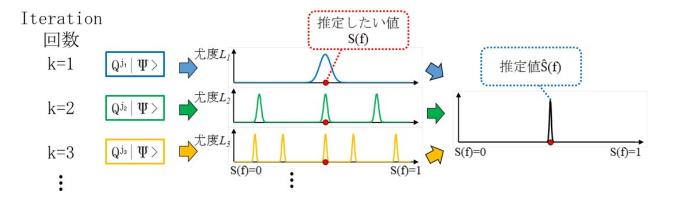


図 2 鈴木ら [21] の振幅増幅法の概念図. 各 Iteration では, j_k 回の振幅増幅を行った量子状態 $\mathbf{Q}^{j_k} | \Psi \rangle$ に対する測定結果をもとに, 尤度関数 L_k を作成する. 全ての Iteration の尤度関数の積から全体の尤度関数 L を求めることで, L を最大にする引数として, 推定値 $\hat{S}(f)$ が得られる.

に対して、 j_k 回の振幅増幅を行った状態

$$\mathbf{Q}^{j_k} |\Psi\rangle_{n+1} = \sin((2j_k + 1)\theta) |\tilde{\Psi}_0\rangle_{n+1} + \cos((2j_k + 1)\theta) |\tilde{\Psi}_1\rangle_{n+1}$$

$$(45)$$

に対して、 N_k 回のベルヌーイ試行を行うとする.この時 $|\tilde{\Psi}_0\rangle_{n+1}$ が観測される回数が h_k 回であったとすると, θ に対する尤度関数 $L_k(h_k;\theta)$ は (定数項を除いて) 以下の式で表される.

$$L_k(h_k; \theta) = \left[\sin^2((2j_k + 1)\theta)\right]^{h_k} \left[\cos^2((2j_k + 1)\theta)\right]^{N_k - h_k}$$
(46)

以上の手続きを k=1 から m まで、適当な振幅増幅 の回数 $\vec{j}=\{j_1,j_2,\cdots,j_m\}$ 及びベルヌーイ試行の回数 $\vec{N}=\{N_1,N_2,\cdots,N_m\}$ に対して行い、 $|\tilde{\Psi}_0\rangle_{n+1}$ が観測された回数が $\vec{h}=\{h_1,h_2,\cdots,h_m\}$ と得られたとする。この時、全観測結果に対する尤度関数は式 (46) を用いて、以下のように構成することが出来る.

$$L(\vec{h};\theta) = \prod_{k=1}^{m} L_k(h_k;\theta)$$
(47)

このように構成された尤度関数を用いて, θ の最尤推定を行う. 最尤推定値 $\hat{\theta}$ は, 尤度関数を最大化する θ として与えられる:

$$\begin{split} \hat{\theta} &= \underset{\theta}{\text{arg max}} \ L(\vec{h}; \theta) \\ &= \underset{\theta}{\text{arg max}} \ \ln L(\vec{h}; \theta) \end{split} \tag{48}$$

ここで、2番目の等式では対数関数 ln が単純増加関数であることを用いた. 以上により得られた最尤推

定値 $\hat{\theta}$ を用いて, S(f) の推定値を $\hat{S}(f) = \sin^2 \hat{\theta}$ と求めることが出来る.

ここで、 $0 \le \theta \le \pi/2$ の範囲では、 θ と $S(f) = \sin^2 \theta$ が 1 対 1 対応することから、尤度関数 L を S(f) の関数としても見なすことが出来ることを考慮して、 $L(\vec{h};S(f)) = L(\vec{h};\theta)$ と書くとすると、式 (48) による推定は

$$\hat{S}(f) = \underset{S(f)}{\operatorname{arg max}} \ln L(\vec{h}; S(f))$$
(49)

と見なすことも出来ることに留意する.

以上により、最尤推定法による S(f) の予測値 $\hat{S}(f)$ の推定方法について述べた. 以下では、本手法における関数 f の呼出回数 N と誤差 ε の関係について議論する.

3.4.2 関数の呼出回数と誤差の関係

ここでは、鈴木らの手法 [21] における、関数 f の呼出回数 N と誤差 ε の関係について、フィッシャー情報量を用いて議論する. なお、フィッシャー情報量の詳細については、統計学の教科書 [28] 等を参照されたい.

一般に、確率変数 x が、母数 a により決まる確率分布 L(x;a) に従うとき、フィッシャー情報量 $\mathcal{I}(a)$ は以下により与えられる.

$$\mathcal{I}(a) = \mathbb{E}\left[\left(\frac{\partial}{\partial a} \ln L(x; a)\right)^2\right]$$
 (50)

ここで、期待値 $\mathbb E$ は確率変数 x について取るものとする. a の任意の不偏推定量 $\hat a$ の推定誤差 $\varepsilon = \sqrt{\mathbb E[(\hat a - a)^2]}$ について、推定誤差の下限は、フィッシャー情報量を用いて、以下のクラメール—ラオ不等

式 [28] により与えられることが知られている:

$$\varepsilon^2 = \mathbb{E}[(\hat{a} - a)^2] \ge \frac{1}{\mathcal{I}(a)} \tag{51}$$

なお、最尤推定法を用いた推定では、サンプル数が大きくなるに連れて、推定誤差は漸近的に、式 (51) の下限に近づくことが知られている.

鈴木らの手法 [21] において,式 (46), (47) 及び (50) からフィッシャー情報量は

$$\mathcal{I}(a) = \frac{1}{S(f)(1 - S(f))} \sum_{k=1}^{m} N_k (2j_k + 1)^2 \quad (52)$$

と与えられる. また, 関数 f の呼出回数 N は, 以下の式により与えられる.

$$N = \sum_{k=1}^{m} N_k (2j_k + 1) \tag{53}$$

以上により、もし、振幅増幅の回数 $\vec{j}=\{j_1,\cdots,j_m\}$ 及びベルヌーイ試行の回数 $\vec{N}=\{N_1,\cdots,N_m\}$ を具体的に決めれば、最尤推定法による推定誤差 ε の下限と関数の呼出回数 N の関係を、式 (51), (52) 及び (53) を用いて導くことが出来る.

たとえば、全ての $k=1,\cdots,m$ に対して、ベルヌーイ試行の回数を一定回数 $N_k=N_0$ として、また、 j_k を指数関数的に増加する場合 $m_1=0,m_2=2^0,m_3=2^1,\cdots,m_m=2^{(m-2)}$ を考えると、推定誤差 ε の下限と関数の呼出回数 N の関係は $N\geq O\left(\varepsilon^{-1}\right)$ となることが示される.最尤推定法では、サンプル数が十分多いときにこの不等式の下限に漸近することが知られているが、鈴木ら [21] は、数値シミュレーションにより、 $N_0=100$ 回程度と選ぶことで、下限をほぼ達成することを確かめている.

本手法は、制御ユニタリ演算子 ^{c}Q や逆フーリエ変換演算子 F_m を使用しないため、3.2 節において概説した Brassard らの手法に較べて、演算数が減少することが期待される。実際に、鈴木ら [21] は、IBM の公開している量子コンピュータ向けの SDK(Software Development Kit) の Qiskit [19] を用いて、簡単な三角関数の積分について、本手法と Brassard らの手法の量子回路を実装し、C-NOT 演算子の個数が 5% 程度に削減することを確認している。

4 まとめ

本稿では、量子コンピュータの応用先の例として、 数値積分を行う3種類の量子アルゴリズムについて 紹介した. 数値積分で誤差 ε を達成するために, モンテカルロ法では $O\left(\varepsilon^{-2}\right)$ 回の関数の演算が必要だったのに対して, 量子コンピュータを使うことで, 演算回数が $O\left(\varepsilon^{-1}\right)$ 回程度に減少することを示した.

現在、金融を始めとして、各分野に量子コンピュータを使う取り組みが始まっており、今後、将来の量子コンピュータの活用に向けて、本稿で紹介した手法を基本として、各分野に特化したような更に洗練された手法の開発が期待される。

付録 A 本文中で使用するユニタリ演算子

本付録では、本文中で使用するユニタリ演算子をまとめる.

 $\operatorname{Hadamard}$ 演算子 H の 1 量子ビット計算基底 $|0\rangle$ への作用を

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{54}$$

と定義する.

Hadamard 演算子 $H^{\otimes n}$ の n 量子ビット計算基底 $\ket{0}_n$ への作用を

$$H^{\otimes n} |0\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n - 1} |x\rangle_n$$
 (55)

と定義する.

1量子ビット計算基底 $|0\rangle$, $|1\rangle$ への Z の作用を,

$$Z |0\rangle = |0\rangle$$

$$Z |1\rangle = -|1\rangle$$
(56)

n ビット量子状態 $|\Psi\rangle_n$ に作用する恒等演算子 I_n を

$$I_n |\Psi\rangle_n = |\Psi\rangle_n \tag{57}$$

と定義する.

式 (10) の確率関数 p(x) を計算するための n 量子ビット に作用する演算子を

$$\mathcal{P}|0\rangle_n = \sum_{x=0}^{2^n - 1} \sqrt{p(x)} |x\rangle_n \tag{58}$$

と定義し, f(x) を計算するための n+1 ビットに作用する 演算子を

$$\mathcal{R} |x\rangle_n |0\rangle = |x\rangle_n \left(\sqrt{f(x)} |0\rangle + \sqrt{1 - f(x)} |1\rangle \right) (59)$$

と定義する.

量子逆フーリエ変換を行う n ビットに作用する演算子を

$$F_n^{-1} |x\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n - 1} e^{-2\pi i x y/2^n} |y\rangle_n$$
 (60)

と定義する.

n量子ビットに作用するユニタリ演算子 U が与えられたとき, m ビットを制御ビットとして, m+n ビットに作用する制御ユニタリ演算子 $^{\mathbf{c}}U$ を

$$^{\mathbf{c}}U|x\rangle_{m}|y\rangle_{n} = |x\rangle_{m}U^{x}|y\rangle_{n} \tag{61}$$

と定義する.

引用文献

- [1] QUANTUM COMPUTING REPORT. https://quantumcomputingreport.com/ scorecards/qubit-count. Accessed:2019-05-29.
- [2] IBM Q Experience. https://quantumexperience.ng.bluemix.net/ qx/editor. Accessed:2019-05-29.
- [3] Quantum Cloud Services Rigetti. https://www.rigetti.com/qcs. Accessed:2019-05-29.
- [4] 慶應義塾大学. https://www.keio.ac.jp/ja/news/2018/5/22/27-44149/. Accessed:2019-05-29.
- [5] P. Glasserman. Monte Carlo Methods in Financial Engineering. Springer, 2004.
- [6] 湯前 祥二, 鈴木 輝好. モンテカルロ法の金融工学への 応用. 朝倉書店, 2000.
- [7] W. Krauth. Statistical Mechanics: Algorithms and Computations. Oxford University Press, 2006.
- [8] M. Creutz, L. Jacobs and C. Rebbi. Monte Carlo computations in lattice gauge theories. Physics Reports, 95(4):201–282, 1983.
- [9] 青木 慎也. 格子上の場の理論. シュプリンガー・ジャパン, 2012.
- [10] M. EricksonKirk et al. Technical basis for revision of the pressurized thermal shock (PTS) screening limit in the PTS rule (10 CFR 50.61). NUREG-1806, US Nuclear Regulatory Commission, 2007.
- [11] J. Katsuyama et al. Guideline on a structural integrity assessment for reactor pressure vessel based on probabilistic fracture mechanics. JAEA-Research, 2017.
- [12] C. M. Johnson *et al.* Monte Carlo explicitly correlated second-order many-body perturbation theory. The Journal of chemical physics, 145(15):154115, 2016.
- [13] C. M. Johnson *et al.* Monte Carlo explicitly correlated many-body Green's function theory. The

- Journal of chemical physics, 149(17):174112, 2018.
- [14] A. Montanaro. Quantum speedup of Monte Carlo methods. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 471(2181):20150301, 2015.
- [15] P. Rebentrost, B. Gupt and T. R. Bromley. Quantum computational finance: Monte Carlo pricing of financial derivatives. Physical Review A, 98:022321, 2018.
- [16] S. Woerner and D. J. Egger. Quantum risk analysis. npj Quantum Information, 5(1):15, 2019.
- [17] A. Martin *et al.* Towards pricing financial derivatives with an ibm quantum computer. arXiv preprint arXiv:1904.05803, 2019.
- [18] N. Stamatopoulos *et al.* Option pricing using quantum computers. arXiv preprint arXiv:1905.02666, 2019.
- [19] G. Aleksandrowicz *et al.* Qiskit: An open-source framework for quantum computing, 2019.
- [20] G. Brassard et al. Quantum amplitude amplification and estimation. Contemporary Mathematics, 305:53-74, 2002.
- [21] Y. Suzuki *et al.* Amplitude estimation without phase estimation. arXiv preprint arXiv:1904.10246, 2019.
- [22] D. S. Abrams and C. P. Williams. Fast quantum algorithms for numerical integrals and stochastic processes. arXiv:quant-ph/9908083, 1999.
- [23] A. Papageorgiou and J. F. Traub. Quantum Algorithms for Continuous Problems and Their Applications, pages 151–178. John Wiley & Sons, Ltd, 2014.
- [24] M. A. Nielsen and I. Chuang. Quantum computation and quantum information, 2002.
- [25] 石坂 智, 小川 朋宏, 河内 亮周, 木村 元, 林 正人. 量子 情報科学入門. 共立出版, 2012.
- [26] L. K. Grover. A fast quantum mechanical algorithm for database search. Proceedings of 28th Annual ACM Symposium on Theory of Computing, pages 212–219, 1996.
- [27] A. Y. Kitaev. Quantum measurements and the Abelian stabilizer problem. Electronic Colloquium on Computational Complexity, 3, 1996.
- [28] C. R. Rao. Linear statistical inference and its applications, volume 2. Wiley New York, 1973.