量子コンピュータを用いた数値積分計算について

On a Numerical Integration Using a Quantum Computer

加 藤 公 一

要 約 数値積分計算の量子アルゴリズムを紹介する．このアルゴリズムの発見によって，主に離散的な問題に利用されていた量子アルゴリズムが，連続的な問題にも利用できることが示された．このことは，量子コンピュータの更なる応用の可能性を示すものである．計算手法においては，従来の量子アルゴリズムでは所望の結果が高い確率で直接得られるというデジタルな計算が中心であった．それに対し，本稿で紹介するアルゴリズムは，所望の結果が確率の値として得られるというノンデジタルな計算が特徴である．

Abstract We explain a quantum algorithm for a numerical integration. This algorithm showed that the quantum algorithm is also applicable to a continuous problem, though it had been mainly applied to discrete problems. This fact indicates its potential for further application of a quantum computer.

From the view of its method, the existing algorithms are mainly classified as a digital computation, in which the result can be obtained directly with a high probability. On the other hand, the algorithm we introduce in this survey can be distinguishingly considered as a non-digital computation, in which the desired result is obtained as a value of a probability.

1. はじめに

量子計算理論や量子情報理論といった分野がコンピュータ科学における新しい分野として注目されている．それらは，量子力学的振る舞いをする系を，計算や情報通信に生かそうという研究である．

従来のコンピュータ（以下「古典的コンピュータ」と呼ぶ）は，シリコンに不純物を入れた半導体を使って電子の流れを制御することを基本原理として計算を行っている．それに対して，量子力学的振る舞いを示す粒子に様々な物理的操作を行うことで計算ができることがわかってきた．そのような計算機構は量子コンピュータと呼ばれている．現在量子コンピュータは，実験室レベルでは非常に限られた計算について成果をあげているが[17]，まだ実用段階には至っていない．また，この量子コンピュータはいくつかの問題について，古典的コンピュータよりも高速に計算できることが知られている．

本稿では量子コンピュータを使ったアルゴリズムの中でも，特に多重積分の計算をするアルゴリズムを紹介する．これは，従来のモンテカルロ法と比べて二乗オーダでの計算量の向上があることがわかっている．また多重積分計算は，多くの分野で応用されており，現在知られている量子アルゴリズムの中で特に応用可能性が期待できる．

本稿の構成は以下のとおりである．2章では，量子コンピュータとはどのようなものであるかを解説し，それが考えられるようになった経緯を概説する．3章では，量子コンピュータを使った有名なアルゴリズムをいくつか紹介する．4章では数値積分の問題を定義し，その古典的コンピュータによる解法を説明する．5章では，量子コンピュータの数学的モデルを説明する．量子コンピュータによる数値積分アルゴリズムは6章で説明する．最後に7章で，他の関

連する結果をまとめ，今後の展望を述べる．

2. 量子コンピュータとその背景

特定の問題を解くためにコンピュータの処理速度を向上できないかという問題は，古くからコンピュータ科学にとって重要な研究テーマとなってきた．量子コンピュータを使って問題を解こうというアプローチは，新たなハードウェアアーキテクチャを利用しようという考えに基づくものだが，その構造の特殊性から，その上でのアルゴリズムの研究も広く行われている．

量子系を計算に利用しようという動きのきっかけは Benioff と Feynman だとされている．Benioff[2]は量子系を使って Turing マシン相当のものを構築可能であることを示した．Feynman[8]は，量子力学において必要とされる計算量が，従来のコンピュータでは手に負えないほど大きくなることから，その計算に量子系そのものを使うことを提案した．

実際に量子系を使って高速な演算ができる可能性を示したのが Deutsch[5]である．その後，Shor[36]による素因数分解アルゴリズムの発見により，この分野の研究は注目されるようになる．このアルゴリズムによって，整数の素因数分解は古典的コンピュータで知られているアルゴリズムよりも指数的に高速で解けることがわかった．量子コンピュータ研究の歴史的経緯については石井による解説[19]が詳しいので参照されたい．

量子コンピュータに関する研究が多くの研究者によって精力的に行われているのは，その考え方の自然さが，研究者に広く受け入れられているからとも考えられる．量子力学の原理が非常に特殊なもので，専門の研究者までもが「直感と異なる」と指摘しているのに対して，それを利用して計算することが自然であるというのは奇異に感じるかもしれない．しかし，これは従来の問題解決手法の自然な拡張になっているという点で，自然なのである．この点について以下に説明する．

ひとつには，古典的コンピュータにおいても乱数を使ったアルゴリズムが有効であることが知られていたことがある．つまり，仮想的にサイコロを振ったりコインを投げたりしながらその結果に応じて次の処理内容を決めるようなアルゴリズムが，決定的なアルゴリズム（サイコロを振ったりせずにロジックのみで次の処理を決定するアルゴリズム）と比べて，高速に問題を解決するケースが多く知られている[27][32]．例えば，モンテカルロ法を使った多重数値積分はその典型的な例である．それに対して，量子コンピュータでは量子力学的原理により，観測結果が確率的に分布するなどという性質がある．これは，本質的に乱数を内包していることを意味しており，古典的コンピュータで考えられてきた乱数を使ったアルゴリズムの自然な延長として，量子コンピュータの機構をとらえることもできる．

二つ目には半導体の細密化による量子効果の問題が考えられる．つまり，コンピュータの CPU（中央演算装置）の計算速度を上げていこうと考えると，その回路を細密化し多くの論理ゲートを詰め込むということが従来から行われてきた．しかし，この手法の限界として，結線の大きさがある限界より小さくなると量子効果が無視できなくなり，電子が量子力学的挙動を示すようになるという問題が挙げられる．一方，量子コンピュータは量子力学的挙動を積極的に使おうというアプローチであり，この CPU の細密化の自然な延長上にあると考えることもできる．

以上のように，量子コンピュータという発想は大きなパラダイム転換として考えられている一方，従来のコンピュータ科学の研究の自然な延長上にあると考えることもできる．このよう

な自然さがこの分野の将来性を示唆し，そのことがこの分野の研究が活発になっている要因であると考えられる．

3. 有名な量子アルゴリズム

いくら新しいハードウェアが考えられたからといって，その上で動くアプリケーションがなければその価値は評価されない．応用を考慮した量子コンピュータ上のアルゴリズムがいくつか考えられているが，そのうちの三つについて本章で概説する．それぞれのアルゴリズムの詳細についてはここでは説明しないので，各原論文か，量子計算の教科書[29][12]を適宜参照されたい．また，一般向けの解説書[38]はさほど前提知識がなくても読めるように工夫されているのでこちらも参照されたい．

まず，量子アルゴリズムの代表格といえば，Shor のアルゴリズム[36]があげられる．Shor は，量子コンピュータがあれば素因数分解が従来より桁違いに速く計算できることを示した．大きな整数の素因数分解は，現在知られている古典的アルゴリズムでは現実的な時間では解けない（素因数分解したい整数の桁数に対して指数時間かかる[23][22]）．それが，量子コンピュータでは十分に速く（多項式時間で）解けることが示された．素因数分解の困難さは，現在社会で広く使われている RSA 暗号の安全性を保障するものであり，それが一瞬で解けるということになると RSA 暗号は役に立たなくなる．

次に有名なアルゴリズムとしては，Grover のデータベース検索アルゴリズム[9][11][10]があげられる．古典的アルゴリズムでは，n 個のデータが与えられてその中から印のつけられたデータを知るには，それをひとつずつチェックしていくしか手段はなく，計算量は $O(n)$ となる．その一方で，Grover の量子検索アルゴリズムでは計算量は $O(\sqrt{n})$ で済むことが示された．

確かに上記二つのアルゴリズムは，量子アルゴリズムの研究の上で大きな歴史的マイルストンであり，その後の研究に多くの影響を与えたことは言うまでもない．しかし，量子コンピュータが実現できたとき，それらのアルゴリズムが十分に社会的な利益を与えるかという点では疑問が残る．それらのアルゴリズムを利用できるアプリケーションは非常に限定的であり，量子コンピュータを実社会へ適用するためには，さらなる応用の可能性を示す必要がある．

ところが最近になり，量子コンピュータがそのような離散的な問題だけではなく，連続的な問題にも応用できることがわかってきた．特に多重積分の効率化に量子コンピュータが役立つことが Abrams と Williams[1]によって示された．多重積分の効率化は大きな社会的貢献が期待できる．実際，多重積分は各種シミュレーションや金融工学の世界でも頻繁に使われる計算要素であり，それが従来よりも高速に行えるのであれば社会へ与える影響も大きいと考える．筆者は，このアルゴリズムこそがもっとも社会的ニーズに近いものだと考えている．

量子コンピュータを使った数値積分計算アルゴリズム（以下「量子積分アルゴリズム」と呼ぶ）のアイデアを最初に考えたのは Abrams と Williams だが，その後 Novak[30]と Heinrich[14]は，いくつかの関数クラスについて数学的な解析を行い，その計算誤差と計算量の関係についてより詳細な結果を与えた．

また，積分計算と量子計算の親和性の高さも注目すべきである．モンテカルロ法による積分計算は，完全な乱数が得られれば十分に効率的であることが理論的に知られていたが，その一方でコンピュータで計算する限界として擬似乱数の問題があった．つまり，擬似乱数は完全な乱数ではないため，計算効率の面で問題が生じた．そのため，ランダム性の中に規則性を入れ

ることで例えば特定の関数に対する収束を速くする[41][28]，周期を長くする[26]などの工夫が行われてきた．ところが，量子積分アルゴリズムは自然にランダム性を内包しており，乱数の生成に気を使う必要がない．

また量子積分アルゴリズムは，その計算過程においても特殊性がある．既存の素因数分解アルゴリズムや検索アルゴリズムなどの量子アルゴリズムがいわゆるデジタルな計算をするのに対して，量子積分アルゴリズムはノンデジタルに結果が得られることが特徴的である．量子計算でノンデジタルという言葉を使い始めたのは Litvinov ら[24]である．Litvinov らは得たい計算結果が量子状態の係数として得られる計算をノンデジタルな計算と呼び，そのノンデジタルな枠組みの中での四則演算のアルゴリズムを示した．量子積分アルゴリズムもそのノンデジタルな計算の枠組みの中での計算である．

以上いくつかのアルゴリズムを紹介したが，量子コンピュータを使ったアルゴリズムはそれほど多くは見つかっていない．なぜ量子アルゴリズムを見つけるのが困難なのかについては，Nielsen と Chuang[29]による説明が示唆的で興味深いので紹介する．彼らによると，a）量子アルゴリズムは常に古典的アルゴリズムに勝たなければいけないということと，b）量子でなくてもアルゴリズムの設計自体が実は難しいということを理由としてあげている．以上のようなことと，古典的アルゴリズムと比べて研究の歴史が浅いことにより，現在ではあまり多くのアルゴリズムが見つかっていないのだと思われる．

4. 古典的積分計算

ここでは，数値積分問題の定式化と，その古典的アルゴリズムについて説明する．例えば，金融工学の分野では，金融商品の価格計算に数百〜数千次元の積分計算が必要になることがあり[39]，それを高速に計算できないかというのは以前からの課題である．

積分の数値計算問題とは，与えられた d 変数関数 $f(x_1, \cdots, x_d)$ に対して

$$\int_{a_1}^{b_1} \cdots \int_{a_d}^{b_d} f(x_1, \cdots, x_d) dx_1 \cdots dx_n \tag{1}$$

を近似的に求めようという問題である．特に，適当なスケーリングをすることにより

$$a_1 = a_2 = \cdots = a_d = 0,\ b_1 = b_2 = \cdots = b_d = 1 \tag{2}$$

としても一般性を失わない．つまり，

$$I[f] = \int_0^1 \cdots \int_0^1 f(x_1, \cdots, x_d) dx_1 \cdots dx_n \tag{3}$$

を近似的に計算する問題を考えることとする．

もっとも素朴なアルゴリズムとしては，長方形近似が知られており，

$$S[f] = \frac{1}{n^d} \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} \cdots \sum_{i_d=1}^{n} f\left(\frac{i_1}{n}, \frac{i_2}{n}, \ldots, \frac{i_d}{n}\right) \tag{4}$$

によって近似される．これは n^d 個の関数値の和を含むため，低次元では有効であるが，多次元では計算量が膨大になる．実際，この場合の計算誤差を ε 以内に抑えるためには，計算量は $O(\varepsilon^{-d/2})$になることが知られている．これは，d が大きくなると実用的ではない．

そこで多次元における積分計算を効率的に計算するために考えられたのが，モンテカルロ法である．乱数列 $\mathbf{r}_1, \cdots, \mathbf{r}_N$ が与えられたとして，

$$S'[f] = \sum_{i=1}^{N} f(\mathbf{r}_i) \tag{5}$$

を計算するのが，モンテカルロ法である．この場合の計算誤差は $O(1/\sqrt{N})$であることが知られている[4]．逆に誤差 ε を固定すると，計算量は $O(\varepsilon^{-2})$になる．これは次元に依存しないので，高次元では長方形近似よりも効率がよい．

モンテカルロ法で積分値を計算するには，乱数が重要な役割を果たす．完全な乱数を古典的コンピュータで高速に発生させるのは不可能であり，そのため数学的な数列計算による擬似乱数が用いられてきた．しかし，ハードウエアの発達や応用分野の発展などにより，より高次元の多重積分の計算が求められるようになり，初期に考えられた擬似乱数発生アルゴリズムでは不十分な場合が多くなってきた．擬似乱数の不十分さは，多くはその周期の短さに原因がある．つまり，式(5)で N が乱数の周期を超えてしまうと，それ以上計算を続けても近似誤差が向上しない．

そのような理由により長周期乱数の必要性が増す中で，非常に長周期な乱数を発生させるアルゴリズムとして有名なのがメルセンヌ・ツイスター[26][25]である．これはメルセンヌ数という数の数論的性質を用いた乱数発生アルゴリズムである．周期は $2^{19937}-1$ と非常に長いのが特徴的であり，乱数生成のための計算量が非常に小さいのも特徴的である．

一方で，乱数にある程度の規則性を持たせることで収束の速さを高める方向の研究もなされてきた．低食い違い列（low discrepancy sequence, LDS）[41][28]は高次元に対しても分布の等質性が高いという特殊な性質をもつ．この数列は一般の擬似乱数よりは規則性が高いため，これを使った積分計算は準モンテカルロ法と呼ばれる．これによる計算誤差は $O((\log N)^d/N)$であることが知られている．

5. 量子計算のモデル

ここでは，量子コンピュータによる計算過程の数学的モデルについて説明する．数学的なモデルの背後にある物理的な意味づけについては，ここでは細かく説明しないが，次章で説明するアルゴリズムの理解のためには数学的なモデルを理解するだけでも十分であると考える．詳細については教科書[29][12]や解説論文[7][35]を参照されたい．

量子計算の過程は，初期値，量子ゲート，測定という三つの要素によって構成される．初期値は古典的コンピュータにおける入力にあたる．量子ゲートは古典的には論理回路にあたり，主に計算と呼ばれるのはこの過程である．m ビットの古典的計算においては，$\{0, 1, \cdots, 2^m-1\}$ の「いずれかの」状態に対して演算を行うのに対し，m キュービット（量子ビット）の計算においては，$\{0, 1, \cdots, 2^m-1\}$が「ある割合で重ね合わされた」状態に対して演算が行えることが特徴である．つまり，内部的にはキュービット数に対して指数的に多くの計算を行える

ことになる．なぜ「内部的」なのかと言うと，得られたすべての計算結果を最終的に取得することができないからである．ここで，計算の三つ目の要素である測定が重要になる．実際に計算結果を得るためには測定が必要であるが，測定の結果は確率的な振る舞いをする．つまり重ね合わせのブレンド率によって，$\{0, 1, \cdots, 2^m - 1\}$のどの整数値が測定されるかが，確率的に決まる．このことにより，内部的には指数的に多くの計算ができていても，実際に計算結果として得られる量は限られてしまう．

まず，量子状態のなかでももっとも単純な 1 キュービットの状態とは，内積付き 2 次元複素ベクトル空間$\mathbb{C}^2$上の点として，

$$a_0 |0\rangle + a_1 |1\rangle, \quad a_0, a_1 \in \mathbb{C}, |a_0|^2 + |a_1|^2 = 1 \tag{6}$$

で表される．ここで$|\cdot\rangle$は Dirac のブラケット記法と呼ばれ，$|\psi\rangle$のように一般の記号が入ったときは一般の複素ベクトルを表し，$|k\rangle (k \in \mathbb{Z}, k \geq 0)$のように非負整数が入ったときはベクトル空間の正規直交基底の k 番目の要素を表す．1 キュービットのケースでは 2 次元複素ベクトル空間なので，特に

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{7}$$

と思ってよい．

式(6)は$|0\rangle$と$|1\rangle$が重ね合わされた状態だと見ることができる．実際これに対して，測定を行うと確率$|a_0|^2$で$|0\rangle$が観測され，確率$|a_1|^2$で$|1\rangle$が観測される．この 1 キュービットの状態に対して 2 次元の複素ベクトル空間$\mathcal{H}_1 = \mathbb{C}^2$上の点を対応させる．このキュービットは量子コンピュータにおける情報量の最小単位である．古典的コンピュータでのビットでは，0 か 1 のどちらかの状態しか表せないのに対し，キュービットは 0 と 1 がある割合で混ざり合った状態を表すことができるのである．

次に，m キュービット状態は，m 個のテンソル積[*1]として

$$\mathcal{H}_m = \underbrace{\mathcal{H}_1 \otimes \mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_1}_{m} \tag{8}$$

と定義される．各$\mathcal{H}$の基底は$|0\rangle$，$|1\rangle$なので，$\mathcal{H}_m$の基底は

$$\left\{ |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_m\rangle \;\middle|\; i_j \in \{0, 1\} \right\} \tag{9}$$

で表される．$\mathcal{H}$は2^m次元の複素ベクトル空間になる．このとき，$|i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_m\rangle$の$\otimes$は省略する．また，$k$ を二進表記すると$(i_1 i_2 \cdots i_m)_2$になるときには，これを$|k\rangle$と表記する．つまり

$$|i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_m\rangle = |i_1\rangle |i_2\rangle \cdots |i_m\rangle = |k\rangle \qquad (k = (i_1 i_2 \cdots i_m)_2 \text{ のとき}) \tag{10}$$

である．これが古典的コンピュータでいう m ビット（m 個の 0 と 1 の組み合わせ）に対応すると思えばよい．ただし，量子コンピュータではこれらが基底となっており，これらの線形結合も状態として考えることができる．つまり，一般には m キュービットの状態は

$$\sum_{k=0}^{2^m-1} a_k \left|k\right\rangle, \qquad \sum_{k=0}^{2^m-1} \left|a_k\right|^2 = 1 \tag{11}$$

で表される．これはつまり，2^m-1 個の状態 $|0\rangle$, $|1\rangle$, $\cdots$, $|2^m-1\rangle$ が，ブレンド率 $|a_k|^2$ で重ね合わされたと思うことができる．この $|a_k|$ の値は，後述の測定時の確率に関係してくる．

ここでは，最小限の構成要素として W, P_θ, X の 3 種類の量子ゲートを説明する．古典的コンピュータでは AND ゲートと NOT ゲートがあれば任意の論理回路が構成できるのと同様に，任意の量子回路はこれら三つのゲートから構成可能である．W と P_θ は一つのキュービットに作用するのに対し，X は二つのキュービットに作用する．古典的回路と異なり，一つのキュービットを分岐させることができないことも注意すべきである．これは，不明な量子状態を複製することはできないというクローン不可能定理[6][40]によるものである．以上の制約をわかりやすくするため，図 1 に量子回路の例を示す．

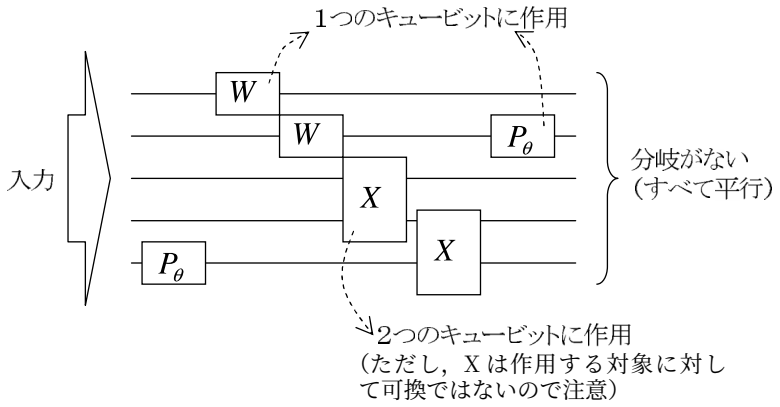

図 1　量子回路の例

次にそれぞれのゲートについて，数学的な意味を説明する．まず Walsh–Hadamard 変換 W から定義する．これは，1 キュービットの状態に対して

$$W\left|0\right\rangle = \frac{\left|0\right\rangle + \left|1\right\rangle}{\sqrt{2}}, \qquad W\left|1\right\rangle = \frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}} \tag{12}$$

によって定義される．次に一般の m キュービット状態への j 番目の要素に対する作用 $W^{(j)}$ を考えることができる．つまり，

$$W^{(j)}\left|i_1\right\rangle \cdots \left|i_m\right\rangle = \left|i_1\right\rangle \cdots \left|i_{j-1}\right\rangle W\left|i_j\right\rangle \left|i_{j+1}\right\rangle \cdots \left|i_m\right\rangle \tag{13}$$

によって定義される．これは，「平均的に重ね合わせた状態」を生成するためによく使われる．実際 $W_m = W^{(m)} \cdots W^{(2)} W^{(1)}$ とすると，

$$W_m |0\rangle = 2^{-m/2} \sum_{k=0}^{2^m-1} |k\rangle \tag{14}$$

となり，これは $|0\rangle, |1\rangle, \cdots, |2^m-1\rangle$ を平均的に重ね合わせた状態である．

次にフェーズシフト P_θ と呼ばれるゲートを定義する．これは，1 キュービットに対しては

$$P_\theta |0\rangle = |0\rangle , \qquad P_\theta |1\rangle = e^{i\theta} |1\rangle \tag{15}$$

と定義される．次に m キュービットに対する作用 $P_\theta^{(j)}$ は $W^{(j)}$ と同様に定義する．

三つ目の基本ゲートとしては，量子 XOR（あるいは制御 NOT とも呼ばれる）を定義する．これは 2 キュービット状態については，

$$\begin{aligned} X |0\rangle |0\rangle &= |0\rangle |0\rangle , & X |0\rangle |1\rangle &= |0\rangle |1\rangle , \\ X |1\rangle |0\rangle &= |1\rangle |1\rangle , & X |1\rangle |1\rangle &= |1\rangle |0\rangle \end{aligned} \tag{16}$$

で定義される．これは m キュービットに対しては，j 番目と k 番目の要素に対する作用 $X^{(j,k)}$ として，

$$X^{(j,k)} |i_1\rangle \cdots |i_m\rangle = |i_1\rangle \cdots |i_{j-1}\rangle |\tilde{i}_j\rangle |i_{j+1}\rangle \cdots |i_{k-1}\rangle |\tilde{i}_k\rangle |i_{k+1}\rangle \cdots |i_m\rangle \tag{17}$$

で定義される．ただし，

$$|\tilde{i}_j\rangle |\tilde{i}_k\rangle = X |i_j\rangle |i_k\rangle \tag{18}$$

で表されるものとする．

以上のように定義されたゲート $W^{(j)}$，$P_\theta^{(j)}$，$X^{(j,k)}$ をいくつか作用させたものが量子回路と呼ばれ，その作用の回数が計算量である．初期状態としては，複素ベクトル空間 $\mathcal{H}^m$ の基底を取り，それに量子回路を作用させたものがその結果として得られる状態である．これらの作用はすべて，$\mathbb{C}^{2^m}$ に対するユニタリ行列[*2] で表現されている．よって，量子回路は行列 $W^{(j)}$, $P_\theta^{(j)}$, $X^{(j,k)}$ の掛け算によって表現され，その積もまたユニタリ行列である．

初期状態として何らかの基底状態をとって，それに量子ゲートをいくつか作用させることで量子計算は進行する．もしそのようにして得られた状態の情報（つまりすべての係数）を得られれば，m キュービットで 2^m 重の並列計算ができたことになるが，前述のように実際はそう単純ではない．量子状態についての情報を得るには測定が必要で，その測定によって得られる情報はわずかである．以下にその測定について説明する．

測定の説明のため，量子状態のレジスタ表現を与えられた m_1, m_2（$m_1 + m_2 = m$）に対して

$$
\begin{aligned}
|k\rangle &= (|i_1\rangle \ldots |i_{m_1}\rangle)(|i_{m_1+1}\rangle \cdots |i_{m_1+m_2}\rangle) \\
&= |j_1\rangle\,|j_2\rangle \in \mathcal{H}_{m_1} \otimes \mathcal{H}_{m_2} \qquad (0 \le j_p \le 2^{m_p} - 1)
\end{aligned}
\tag{19}
$$

のように定義する．ただしここで，j_p は各対応するビット（m_p キュービット分）を二進数表現だとみなして対応する整数をとったものである．つまり，

$$
\begin{aligned}
j_1 &= (i_1 i_2 \cdots i_{m_1})_2 = \sum_{k=0}^{m_1-1} 2^k i_{m_1-k} \\
j_2 &= (i_{m_1+1} i_{m_1+2} \cdots i_{m_1+m_2})_2 = \sum_{k=0}^{m_2-1} 2^k i_{m_1+m_2-k}
\end{aligned}
\tag{20}
$$

である．また，このレジスタ表現のうち $|j_1\rangle$ を第 1 レジスタ，$|j_2\rangle$ を第 2 レジスタとそれぞれ呼ぶ．以下では第 1 レジスタに対する測定を考える．

ここで，初期状態 $|k\rangle=|j_1\rangle|j_2\rangle$ に対して量子ゲートがいくつか作用した結果，状態 $|\psi\rangle$が得られたとする．このとき，$|\psi\rangle$に対する測定が行えるためには

$$
|\psi\rangle = \left(\sum_{l=0}^{2^{m_1}-1} a_l\,|l\rangle \right) |j_2\rangle \tag{21}
$$

の形で表される必要がある．この量子状態を第 1 レジスタによって測定するとは，この観測によって考えられる事象が 2^{m_1} 個あって，それらを $0, 1, \cdots, 2^{m_1}-1$ と名付けると，

「確率 $|a_l|^2$ で事象 l が発生する」

ということを意味する．特にレジスタが 1 つの場合（第 2 レジスタがない場合），最終状態は

$$
|\psi\rangle = \sum_{l=0}^{2^m-1} a_l\,|l\rangle \tag{22}
$$

で表され，$0 \le l \le 2^m-1$ について事象 l が確率 a_l で発生する．これは前述したとおり，量子状態が確率的ブレンド率 $|a_l|^2$ の重ね合わせに対応していることを示している．

以上のことをまとめると，量子計算というのは次のようなステップで行われる．

1）初期状態：$|k\rangle=|j_1\rangle|j_2\rangle \in \mathcal{H}_m$ を自然な基底から選ぶ

2）計算：量子ゲート $U_1, \cdots, U_n$ を順に作用させて $|\psi\rangle=U_n \cdots U_2 U_1 |k\rangle$を得る

3）測定：$|\psi\rangle=(\sum_{l=0}^{2m_1-1} a_l\,|l\rangle)|j_2\rangle$ に対して測定を行い，確率 a_l で事象 l を得る

このことを図式化すると図 2 のようになる．

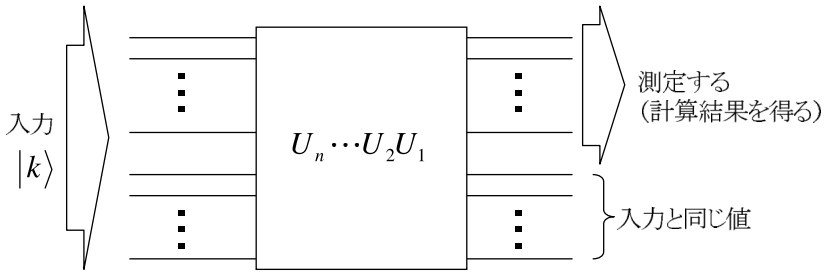

図 2　量子計算の概念図

6. 量子積分アルゴリズム

長方形近似による積分計算では，関数の定義域を各座標軸方向に等分して，そのメッシュ点のすべてについて関数値を計算する.そのメッシュ点は次元が高くなると急激に多くなるため，そこで考えられた工夫がモンテカルロ法であった．これは，計算対象の関数値を間引くが，間引き方に十分なランダム性があれば近似できるというアイデアである．一方，量子積分アルゴリズムでは長方形近似と同様に間引かずにすべてのメッシュ点について内部的な計算を行う．これは，キュービット数の指数倍の量の計算が内部的に可能であるという量子コンピュータの特徴を利用している．すべてのメッシュ点における関数値が，途中の量子状態として表れ，その状態を加工することによって所望の積分値を得るのである．

最終的な計算結果を得るには測定が必要である．このアルゴリズムの場合，結果がノンデジタルに得られるのが特徴である．ノンデジタルな計算とは，求めたい計算結果が量子状態の係数として得られるものであり，これは観測時の確率として認知される．例えて言えば，ある実数値 r(ただし $0 \le r \le 1$) を求めたいときに，計算結果として「表の出る確率が r であるようなコイン」が得られるような装置を想像すればよい．コインの場合，その確率を誤差 ε で求めるには，中心極限定理（詳細は統計学の教科書[37][31]を参照）によりコインを $O(\varepsilon^{-2})$回投げる必要がある．

しかし量子状態については，振幅増大法[3]と呼ばれる手法により，係数を選択的に大きくすることが可能である．コインの例でいうと，確率 r で表が出るコインが得られたら，その確率を高めて $ar\,(a>1)$とすることができると思えばよい．確率が増大されたコインを投げて ar を推定し，その値を a で割って r を求めたほうが計算効率がよいのである．実際この振幅増大法により，量子状態のある測定による観測確率 r を誤差 ε で推定するには，$O(\varepsilon^{-1})$程度の観測を繰り返せば十分であることが知られている[1].

各軸で領域 I^d を等分したメッシュ点の集合を D とする．ここで全単射

$$\tau:\ \{0,1,\ldots,2^m-1\} \to D \tag{23}$$

を考える．簡単にするため，例えば $m=dl$ とし，

$$\begin{aligned}\tau : i &= (i_1 i_2 \cdots i_m)_2 \\ &\longmapsto \left((0.i_1 i_2 \cdots i_l)_2, (0.i_{l+1} i_{l+2} \cdots i_{2l})_2, \ldots, (0.i_{(d-1)l+1} i_{(d-1)l+2} \cdots i_{dl})_2\right)\end{aligned} \tag{24}$$

とすればよい．つまり，i の二進数表記を l 桁ごとに区切って，それぞれを 0.･･･で始まる二進数表記だと思うことで，全単射が定義できる．

以下関数 f を計算するゲートの構成法は既知であるとする．ここで f を計算する装置 Q とは，f の値域 D に対して写像 $\tau:\{0,1,\cdots,2^n-1\}\to D$ があらかじめあって，

$$\begin{aligned} Q_f|0\rangle|j\rangle &= \sqrt{1-f(\tau(j))}\,|0\rangle|j\rangle + \sqrt{f(\tau(j))}\,|1\rangle|j\rangle \\ Q_f|1\rangle|j\rangle &= -\sqrt{f(\tau(j))}\,|0\rangle|j\rangle + \sqrt{1-f(\tau(j))}\,|1\rangle|j\rangle \end{aligned} \tag{25}$$

を満たすものとする．簡単にするため，f の値域は$[0,1]$とする．スケーリングにより，この仮定は本質的ではない．

ここで，1 キュービットと m キュービットのレジスタからなる量子状態 $|0\rangle|0\rangle$を用意し，第 2 レジスタに対して Walsh–Hadamard 変換 W_m を作用させ，その後 Q_f を作用させると，

$$\frac{1}{2^{m/2}}\sum_{j=0}^{2^m-1}\left[\sqrt{1-f(\tau(j))}\,|0\rangle|j\rangle + \sqrt{f(\tau(j))}\,|1\rangle|j\rangle\right] \tag{26}$$

となる．これは 2^m個の関数値 $f(\tau(0)),\cdots,f(\tau(2^m-1))$を同時に保持している状態である．メッシュ点における関数値をすべて同時に計算していると見ることができる．測定のため第 2 レジスタに対して Walsh–Hadamard 変換 W_m を作用させると，

$$\frac{1}{2^{m/2}}\sum_{j=0}^{2^m-1}\left[\sqrt{1-f(\tau(j))}\,|0\rangle + \sqrt{f(\tau(j))}\,|1\rangle\right]|0\rangle \tag{27}$$

となる．

次に計算結果を得るためには測定を行う必要がある．この場合，第 1 レジスタを測定したときに，事象$|1\rangle$が起こる確率は，

$$\frac{1}{2^{m/2}}\sum_{j=0}^{2^m-1}|f(\tau(j))| \tag{28}$$

となっている．これは，積分値の近似値となっている．つまり，これは「確率 $I[f]$（の近似値）で表を向くコイン」を手に入れたことに相当する．この係数を近似誤差 ε とすると何度も測定するという素朴なアルゴリズムだと計算量は $O(\varepsilon^{-2})$ になるが，実際には $O(\varepsilon^{-1})$で済むことを以下に説明する．

まず ε とは別に，暫定的に得たい近似誤差 δ を定める．例えば $\delta=1/2$ とする．測定を繰り返すことで近似誤差 δ を得る．このための計算量は，中心極限定理により $O(\delta^{-2})$である．ここで得られた近似値を E とし，$\tilde{f}(x)=f(x)-(E-\delta/2)$とする．このときこの $\tilde{f}$を f として上記のアルゴリズムを繰り返す．ただし，今度は最後の測定の前に振幅増大法によって確率を高めることができる．なぜなら近似誤差の定義により $0\le I[\tilde{f}]\le\delta$ であるので，これを δ^{-1} 倍すればよい．

以上のアルゴリズムを帰納的に表現すると以下のようになる．

$\widetilde{f}_0 = f$, $E_0 = 0$ とする．

$k = 0$ から $\delta^k \leq \varepsilon$ となるまで以下を繰り返す

$\widetilde{f}_k = \widetilde{f}_{k-1} - E_{k-1}$ とする．

$|0\rangle|0\rangle$に $I_1 \otimes W_m$ と $Q_{\widetilde{f}_{k-1}}$ を作用させる．

その結果を振幅増大法により測定確率を δ^{-k} 倍し，

測定を繰り返すことでその近似値 E_k を求める．

（このときの近似誤差は δ^k になる．）

ここで，振幅増大法による計算量は繰り返しの各回ごとに $O(\delta^{-k})$になる．繰り返しが終了する k を l とすると，$\delta^l \approx \varepsilon$ であり，振幅増大法の全体計算量は $O(\delta^{-l}) = O(\varepsilon^{-1})$となる．測定の繰り返しについては，計算量は常に $O(\delta^{-2})$であり，全体計算量のオーダには寄与しない．したがって，このアルゴリズムの計算量は $O(\varepsilon^{-1})$である．

次に振幅増大法について説明する．$|0\rangle$にある量子アルゴリズム X を作用させて得られる量子状態 $|\psi\rangle = X|0\rangle$とし，これは直交する量子状態 $|\alpha\rangle$と $|\beta\rangle$の線形和で

$$|\psi\rangle = \sqrt{a}\,|\alpha\rangle + \sqrt{1-a}\,|\beta\rangle \tag{29}$$

と表せたとする．このとき $|\alpha\rangle$の係数を増大させることを考える．

$$\begin{aligned} U &= -X(I - 2|0\rangle\langle 0|)X^{\dagger}(I - 2|\beta\rangle\langle\beta|) \\ &= (I - 2|\psi\rangle\langle\psi|)(I - 2|\beta\rangle\langle\beta|) \end{aligned} \tag{30}$$

とすると，

$$\begin{aligned} U|\alpha\rangle &= (1-2a)|\alpha\rangle - 2\sqrt{a(1-a)}\,|\beta\rangle \\ U|\beta\rangle &= \sqrt{2a(1-a)}\,|\alpha\rangle + (1-2a)|\beta\rangle \end{aligned} \tag{31}$$

となる．ここで特に$\sqrt{a} = \sin\theta$ となる θ（ただし $0 \leq \theta \leq \frac{\pi}{2}$）を考えると次の式が得られる．

$$\begin{aligned} U|\alpha\rangle &= \cos 2\theta\,|\alpha\rangle - \sin 2\theta\,|\beta\rangle \\ U|\beta\rangle &= \sin 2\theta\,|\alpha\rangle + \cos 2\theta\,|\beta\rangle \end{aligned} \tag{32}$$

これは，$|\alpha\rangle$，$|\beta\rangle$ を基底と見ると，回転行列を作用していることに対応する．したがって，

$$\begin{aligned} U^k|\psi\rangle &= U^k(\sin\theta\,|\alpha\rangle + \cos\theta\,|\beta\rangle) \\ &= \sin(2k+1)\theta\,|\alpha\rangle + \cos(2k+1)\theta\,|\beta\rangle \end{aligned} \tag{33}$$

となる． このときにできるだけ $|\alpha\rangle$の係数が大きくなるように繰り返し数 k を定めればよい．$\sqrt{a}$の値がある値 b によって上から押さえられるとき，つまり$\sqrt{a} = \sin\theta \leq b$ がわかっている時は，θ'を $\sin\theta' = b$ ととり，$(2k+1)\theta' \approx \frac{\pi}{2}$となるように k をとるのが最適である．実際このと

き $|\alpha\rangle$の係数のとりうる値の範囲は，$\sin(2k+1)\theta \leq \sin(2k+1)\theta' \approx 1$ となり，振幅が最大に増幅されたことになる．このときの計算量は $O(k)=O(1/b)$である．

7. お わ り に

量子コンピュータを使った積分計算のアルゴリズムを紹介した．ここで紹介したのは Abrams と Williams[1]によるものだが，その後いくつかの関数クラスについて数学的な解析が行われてきた．特に Hölder クラスや Sobolev クラスなどの高階偏導関数の連続性が知られている関数クラスについては，計算量が数学的に証明されている[30][14]．その結果を用いると，初期条件つき微分方程式や楕円型偏微分方程式も量子コンピュータで高速に解けることが知られている[15][20][21]．これらの研究の流れについては Heinrich による解説論文[13][16]を参照されたい．この方向の研究が発展することによって，物理学や金融工学などにおける具体的な問題についても研究が進むよう期待したい．

量子積分アルゴリズムを実用化するには，まず何よりも量子コンピュータの実用化が必要であるが，アルゴリズムの面からはノイズの影響の検討が必要である．つまり，本稿では数学的に理想的な状況を仮定して議論を進めたが，実装上は途中の量子状態が理論値より多少ずれることを考慮する必要がある．そのずれを考慮したうえでの頑健性の確保が今後の課題である．

また実現されていないハードウェア上のアルゴリズムを考えて何の役に立つのかという批判もあるかと思われるが，そのハードウェアを作る価値があるかどうかの判断基準はその上で何ができるようになるかにも依存することを考慮すべきである．つまり，社会的に十分に役に立つアプリケーションが提示されれば，ハードウェアの実装の研究は加速されると考えられる．そういった視点からも，量子積分アルゴリズムに端を発した一連の量子コンピュータの数値計算への応用の研究に期待したい．

＊1　テンソル積

テンソル積は一般の環に対して定義される概念ではあるが，本稿で扱っている代数的対象は行列のみであるので行列に限定して説明する．行列に対しては，テンソル積は Kronecker 積として与えられる．行列

$$A=\begin{pmatrix} a_{1,1} & \cdots & a_{1,l} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \cdots & a_{k.l} \end{pmatrix}, \quad B=\begin{pmatrix} b_{1,1} & \cdots & b_{1,n} \\ \vdots & \ddots & \vdots \\ b_{m,1} & \cdots & b_{m.n} \end{pmatrix}, \tag{34}$$

が与えられたとき，A と B の Kronecker 積とは

$$A\otimes B=\begin{pmatrix} a_{1,1}B & \cdots & a_{1,l}B \\ \vdots & \ddots & \vdots \\ a_{k,1}B & \cdots & a_{k.l}B \end{pmatrix} \tag{35}$$

$$= \begin{pmatrix} a_{1,1}b_{1,1} & \cdots & a_{1,1}b_{1,n} & & a_{1,l}b_{1,1} & \cdots & a_{1,l}b_{1,n} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ a_{1,1}b_{m,1} & \cdots & a_{1,1}b_{m,n} & & a_{1,l}b_{m,1} & \cdots & a_{1,l}b_{m,n} \\ & \vdots & & \ddots & & \vdots & \\ a_{k,1}b_{1,1} & \cdots & a_{k,1}b_{1,n} & & a_{k,l}b_{1,1} & \cdots & a_{k,l}b_{1,n} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ a_{k,1}b_{m,1} & \cdots & a_{k,1}b_{m,n} & & a_{k,l}b_{m,1} & \cdots & a_{k,l}b_{m,n} \end{pmatrix} \tag{36}$$

によって定義される．つまり，A が $k \times l$ 行列で B が $m \times n$ 行列であるとき，$A \otimes B$ は $km \times ln$ 行列になる．また，この Kronecker 積には結合法則が成り立つことが知られている．つまり

$$(A \otimes B) \otimes C = A \otimes (B \otimes C) \tag{37}$$

は常に成り立つ.

例 1

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \tag{38}$$

のとき

$$A \otimes B = \begin{pmatrix} 1 \cdot \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} & 2 \cdot \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \\ 3 \cdot \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} & 4 \cdot \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{pmatrix} \tag{39}$$

例 2

$$u = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad v = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad w = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{40}$$

のとき,

$$u \otimes v \otimes w = (u \otimes v) \otimes w = \begin{pmatrix} 1 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} \otimes w = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \otimes w = \begin{pmatrix} 0 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ 1 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{41}$$

これは実際,

$$u \otimes (v \otimes w) = u \otimes \begin{pmatrix} 0 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ 1 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{pmatrix} = u \otimes \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{42}$$

により，結合法則が成り立っていることが確認できる．

例 3（本文中式(9)の説明）3 キュービットの系で，第 5 基底$|5\rangle$を考える．5＝$(101)_2$ なので式(10)の定義により，

$$|5\rangle = |1\rangle\,|0\rangle\,|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{43}$$

$$= \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{44}$$

となる．同様にして計算すると，$|k\rangle$は一番上を第 0 要素として上から数えて第 k 要素のみを 1 とし，それ以外を 0 としたベクトルに対応することがわかる．これは式(9)で説明した事実に対応する．

＊2　ユニタリ行列

複素正方行列

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,k} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \cdots & a_{k,k} \end{pmatrix} \tag{45}$$

に対して随伴行列 $A^{\dagger}$ を

$$A^{\dagger} = \bar{A}^{T} = \begin{pmatrix} \bar{a}_{1,1} & \cdots & \bar{a}_{k,1} \\ \vdots & \ddots & \vdots \\ \bar{a}_{1,k} & \cdots & \bar{a}_{k,k} \end{pmatrix} \tag{46}$$

で表す．ただし，ここで $\bar{}$ は複素共役を表す．A がユニタリ行列であるとは，

$$AA^{\dagger} = A^{\dagger}A = I \tag{47}$$

（ここで I は単位行列）という条件を満たすことである．

量子状態は各成分の絶対値の二乗が観測時の事象の発生確率を表すのであった．つまり，

$$|\psi\rangle = \sum_{j=1}^{2^m-1} b_j \, |j\rangle \tag{48}$$

のとき，確率が満たすべき条件により

$$\sum_{j=1}^{2^m-1} |b_j|^2 = 1 \tag{49}$$

となる．ψ は単位行列である．これが変換 A によって保たれること，つまり $A|\psi\rangle$ がまた単位行列になるための必要十分条件が，A がユニタリ行列であることである．量子ゲートがすべてユニタリ行列で表され，その結果回路全体がユニタリ行列で表されることは，この制約によるものである．

参考文献

［1］ D. S. Abrams and C. P. Williams. Fast quantum algorithms for numerical integrals and stochastic processes, 1999. arXiv : quant-ph/9908083.

［2］ P. Benioff. The computer as a physical system : A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *J. of Stat. Phys*., 22（5）：563-591, 1980.

［3］ G. Brassard, P. Hφyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation, 2000. arXiv : quant-ph/0005055.

［4］ P. J. Davis and P. Rabinowitz. *Methods in Numerical Integration*. Academic Press, New York, 1975.

［5］ D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. of the Royal Society of London Ser*. A, A 400 : 97-117, 1985.

［6］ D. Dieks. Communication by EPR devices. *Phys. Lett*. A, 92（6）：271-272, 1982.

［7］ A. Ekert, P. Hayden, and H. Inamori. Basic concepts in quantum computation, 2000. arXiv : quant-ph/0011013.

［8］ R. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21 : 467-488, 1982.

［9］ L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28 th Annual ACM Symp. on the Theory of Computing*, pp. 212-219. ACM Press New York, 1996. arXiv : quant-ph/9605043.

［10］ L. K. Grover. A frame work for fast quantum mechanical algorithms. In *Proc. 30 th Annual ACM Symp. On the Theory of Computing*, pp. 53-62. ACM Press New York, 1998. arXiv : quant-ph/9711043.

［11］ L. K. Grover. Quantum computers can search rapidly by using almost any transformation. *Phys. Rev. Lett.*, 80（19）：4329-4332, 1998.

［12］ J. Gruska. *Quantum Computing*. McGraw-Hill, 1999.

［13］ S. Heinrich. From Monte Carlo to quantum computation. quant-ph/0112152.

［14］ S. Heinrich. Quantum integration in Sobolev classes. *J. Complexity*, 19 : 19-42, 2003.

［15］ S. Heinrich. The quantum query complexity of elliptic PDE, 2005. quant-ph/0512241.

［16］ S. Heinrich. Numerical analysis on a quantum computer. *Lecture Notes in Computer Science*, 3743 : 28-39, 2006.

［17］ IBM. IBM ' s test-tube quantum computer makes history. `http://domino.research.ibm.com/comm/pr.nsf/pages/news.20011219_quantum.html`, December 2001.

［18］ 伊理正夫．一般線形代数．岩波書店，2003.

［19］ 石井茂．量子コンピュータへの誘い．日経 BP 社，2004.

［20］ B. Z. Kacewicz. Randomized and quantum algorithms yield a speed-up for initial-value problems. *J. Complexity*, 20（6）：821-834, 2004.

［21］ B. Z. Kacewicz. Improved bounds on the randomized and quantum complexity of initial-value problems. *J. Complexity*, 21（5）：740-756, 2005.

［22］ A. Lenstra and H. Lenstra eds. *The development of the number field sieve*, Vol. 1554. Springer-Verlag, 1993.

［23］ A. Lenstra, H. Lenstra, M. Manasse, and J. Pollard. The number field sieve. In *Proc. 22 nd ACM Annual Symp. on the Theory of Computing*, pp. 564-572, 1990.

［24］ G. Litvinov, V. Maslov, and G. Shpiz. Nondigital implementation of the arithmetic of real numbers by means of quantum computer media. *Math. Notes*, 70 : 53-60, 2001.

［25］ M. Matsumoto. Mersenne twister home page. `http://www.math.sci.hiroshima-u. ac.jp/~m-mat/MT/mt.html`.

［26］ M. Matsumoto and T. Nishimura. Mersenne Twister : A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, 8（1）：3-30, 1998.

［27］ R. Motowani and P. Raghavan. *Randomized Algorithms*. Cambridge Univ. Press, 1995.

［28］ H. Niederreiter. *Monte Carlo and Quasi-Monte Carlo Methods*. Springer-Verlag, 1998.

［29］ M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Infomation*. Cambridge Univ. Press, 2000.

[30] E. Novak. Quantum complexity of integration. *J. Complexity*, 17 : 2–16, 2001.
[31] 岡部靖憲．確率・統計．応用数学基礎講座．朝倉書店，2002.
[32] C. H. Papadimitriou. *Computational Complexity*. Addison–Wesley, 1994.
[33] 斎藤正彦．線形代数入門．東京大学出版会，1970.
[34] 佐竹一郎．線型代数．裳華房，1974.
[35] P. Shor. Introduction to quantum algorithms, 2000. arXiv : quant–ph/0005003.
[36] P. W. Shor. Algorithms for quantum computation : Discreate logarithms and factoring. In *35 th Annual Symp. on Foundations of Computer Science*. IEEE Computer Society Press, 1998. arXiv : quant–ph/9508027.
[37] 竹内彰通．統計．共立講座 21 世紀の数学．共立出版，1997.
[38] 竹内繁樹．量子コンピュータ．ブルーバックス．講談社，2005.
[39] P. Willmott, S. Howison, and J. Dewynne. *The Mathematics of Financial Derivatives– A Student Introduction–*. Cambridge University Press, 1995. 日本語訳：デリバティブの数学入門，伊藤幹夫・戸瀬信之訳.
[40] W. Wootters and W. Zurek. A single quantum cannot be cloned. *Nature,* 299 : 802–803, 1982.
[41] C. Xing and H. Niederreiter. A construction of low–discrepancy sequences using global function fields. *Acta Arithmetica,* 73（1）：87–102, 1995.

注）arXiv : quant–ph/？？？？？？？（？は数字）と記述のある論文は，
http://arxiv.org/abs/quant–ph/？？？？？？？から入手可能である．

執筆者紹介　加　藤　公　一（Kimikazu Kato）
1998 年，東京大学数理科学研究科修士課程修了．同年日本ユニシス(株)入社. 以来, 通販業界向け顧客分析システムや CAD/CAM システムの開発に従事．現在，先端技術部に所属すると同時に東京大学情報理工学系研究科社会人博士課程に在学中.