解説

量子コンピュータ†

三原 孝志*¹ 西野 哲朗*²

1. はじめに

現在のコンピュータハードウェアは日々進歩し、高速化、多機能化してきている。しかし、逆に、現在のコンピュータアーキテクチャではハードウェアの処理能力をいくら高速化しても、効率的に解くことができない問題も明らかになってきている。例えば、ある問題を解く「最も良いアルゴリズム」を用いた時、 n 個のデータを処理する時に少なくとも 10^n 命令かかるとする。「最も良い」とは、現在存在する最良のアルゴリズムではなく、どのようにアルゴリズムを考えてもこれ以上良いアルゴリズムを作ることができないことを意味しているものとする。仮に $n=100$ とすると、1 命令を 10^{-9} 秒で実行するコンピュータでは 10^{91} 秒(約 10^{76} 億年)という気の遠くなるような時間が必要である。これは、高々 100 個のデータでも解くことが不可能であることを意味する。この問題を 1 日程度で解くためには 1 命令を 10^{-95} 秒で実行するコンピュータが必要であるが、現在のアーキテクチャに基づいたコンピュータをこのように高速化することは物理的に不可能である。

では、この種の問題を解くことをあきらめるのであろうか。現在、この種の問題は厳密解が効率的に得られなくても「実用上問題のない範囲内の解」を効率的に求めるにはどうすればよいかとい

う観点から研究が続けられている(近似アルゴリズム、確率アルゴリズムなど)。また一方で、現在のコンピュータより高性能な新しいアーキテクチャのコンピュータのモデルを提案している研究者もいる。量子コンピュータもそのうちの 1 つと位置づけられる。

量子コンピュータは、まず、Feynman により 1980 年代始めに提案された[12,13]。彼はノーベル物理学賞を受賞した著名な物理学者であり、物理学者の立場からコンピュータを考察した。すなわち、

現在のコンピュータは物理現象を効率的に模倣することができるか？

という質問を投げかけ、彼自身、

もし、量子物理学が古典物理学と異なれば、現在のコンピュータでは効率的に模倣することができない物理現象が存在する

という結論に達した。なぜなら、現在のコンピュータアーキテクチャは高度な量子効果を有効に利用してはいるが、計算の基本原則は 1930 年代に Turing により提案された決定性 Turing 機械と等価なものであり、決定性 Turing 機械は古典物理学の範囲内で説明できるものだからである。さらに、量子物理学が古典物理学と異なるか否かは物理学最大の未解決問題であるが、ほとんどの物理学者は異なると信じている。つまり、古典物理学だけではすべての物理現象を説明することができない限り、現在のコンピュータで効率的に模倣することができない物理現象が存在することになる。

† Quantum Computer
Takashi MIHARA and Tetsuro NISHINO

*1 九州東海大学工学部電子情報工学科
Department of Electronics and Information Technology,
School of Engineering, Kyushu Tokai University

*2 電気通信大学電子情報学科
Department of Communications and System Engineering,
The University of Electro-Communications

そこで、この種の物理現象を効率的に模倣するために、計算の基本原理に量子物理学を取り入れた新しいコンピュータとして、**量子コンピュータ**が考えられた。しかし、Feynman は量子コンピュータは現在のコンピュータより高性能になるであろうと予想はしたが、彼自身で量子コンピュータを数学的に形式化することはなかった。

その後、量子コンピュータは1985年に Deutsch により、量子 Turing 機械として初めて形式化された[7]。しかし、彼もまた物理学者であったため、物理学で用いられる記号が多く用いられており、計算機科学者には理解しにくいものであった。しかし、1993年に Bernstein と Vazirani が、計算機科学者にも分かり易い量子 Turing 機械の形式化を発表した[3]。さらに、1994年に Shor が素因数分解問題と離散対数問題が量子 Turing 機械で効率的に解くことができることを示した[19]。この2種類の問題は現在広く利用されている RSA 公開鍵暗号系に代表される公開鍵暗号系が安全であることを保証する問題であり、現在のコンピュータでは効率的に解くことができないと広く信じられている問題である。この結果の発表以降、物理学者は量子コンピュータの実現に向けて、また、計算機科学者は量子コンピュータが実現した時に利用できるアルゴリズムの開発について活発に研究を行っている[18]。

2. 量子並列計算

2.1 量子ビット

現在のコンピュータは、一般に、最小の情報量を2値、すなわちビット(値0と値1)で表現している。量子コンピュータでも情報は基本的にビットで表現するが、量子効果を利用するために通常のビットと異なる性質を持つので、量子コンピュータ上でのビットを**量子ビット**(*quantum bit* または *qubit*)と呼ぶ。量子ビットは通常のビットと異なり、値0、値1以外に値0と値1の重ね合わせの状態を取ることができる。

ビット自体は抽象的な概念である。しかし、コ

ンピュータを実現するためにはビットを物理的に表現する必要があるので、2状態の物理系を構成することにより実現している。通常のビットは電位差により値0と値1を表現する物理系として構成される。一方、量子ビットを構成する物理系としては、例えば、図1の水素原子が考えられる。良く知られているように水素原子は陽子1個(原子核)と電子1個から構成され、電子は原子核の周囲を回っている。さらに、電子は保持しているエネルギーの大きさに従い基底状態または励起状態に存在する。励起状態は複数存在するが、ここでは簡単のため1状態に限定すると、基底状態を値0、励起状態を値1と決める事により量子ビットを構成する事ができる。

今までの説明では、電位差で表現したビットと本質的に違いはなく、単に小さなもの(水素原子)で表現しただけではないかと思われるかもしれない。しかし、この小さいということが重要な点である。物理学者は古典物理学より量子物理学が物理現象をより忠実に説明できると確信している。しかし、日常生活では、これが量子的な現象であると認識できるものはほとんどなく、古典物理学で説明がつくものばかりである。なぜなら、日常目にするような巨視的な物理現象では量子効果が無視できるからである。つまり、電位差のような巨視的な物理現象によるビット表現では量子効果は無視することができる。

一方、水素原子の振る舞いという微視的な物理現象では量子効果が無視できないほど大きく影響してくる。そして、水素原子で表現したビットでは通常では思いもつかない2つの値の重ね合わせ、つまり、値0と値1の重ね合わせられた状態を表現することが可能になる。この考え方を一般化すると、量子コンピュータは量子並列計算機能を持て

図1 量子ビット

ることになり、この機能が量子コンピュータが従来のコンピュータより高性能になるであろうという可能性を示す機能になる(量子並列計算については次節で詳しく説明する)。

では、量子ビットは数学的にはどのように表現されるのであろうか。量子物理学では、物理系(の作用素)と物理状態はそれぞれ Hilbert 空間の行列(ただし、ユニタリ行列に制限される)とベクトルで表現される。すなわち、1 ビットは2次元 Hilbert 空間のある基底ベクトル(これを**計算の基底**(*computational basis*)と呼ぶ)で表現される。例えば、ビット値 0、1 に対応する物理状態をそれぞれ記号 $|0\rangle$ と $|1\rangle$ で表すと、

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

で表現される。

また、 n ビットは 2^n 次元 Hilbert 空間上の物理状態

$$|q_1, q_2, \dots, q_n\rangle = |q_1\rangle \otimes |q_2\rangle \otimes \dots \otimes |q_n\rangle$$

として表現される。ただし、 $q_i \in \{0, 1\}$ ($i = 1, 2, \dots, n$) であり、 $\otimes$ はテンソル積を表す。量子物理学の数学的枠組みについての詳細は、例えば、[15]を参考にいただきたい。

2.2 量子並列計算機能

前節で述べたように、水素原子ほどの小さなものになると量子効果が無視できなくなる。量子コンピュータではこの量子効果を有効に利用して、より高性能なコンピュータを実現しようとしている。特に、状態の重ね合わせを有効に取り込んだ計算機モデルである。

(物理)状態の重ね合わせ(*superposition*)とは、ある物理系に対し存在可能な物理状態が複数存在する時、それらの状態が線形に重ね合わされることをいう。これを数学的に述べると、ある空間の任意のベクトルは、基底ベクトルの線形重ね合わせとして表現できるということに対応する。水素原子では、値 0 と値 1 に対応する基底ベクトル $|0\rangle$ と $|1\rangle$ に対し、重ね合わされた状態に対応す

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

図2 1 ビットの重ね合わせ

るベクトルを $|\psi\rangle$ とすると、

$$|\psi\rangle = a|0\rangle + b|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$$

と表現される。ここで、 a, b は複素数であり、条件 $|a|^2 + |b|^2 = 1$ を満たす。この条件は(後で述べるが)、各係数の絶対値の2乗が対応する状態を得る確率を表すことに由来している。直観的には、 $|\psi\rangle$ は、図2のように、電子が基底状態と励起状態の間を行ったり来たりしている状態を表している。

また、3 ビットの場合には各ビット毎の重ね合わせ $|\psi_1\rangle, |\psi_2\rangle, |\psi_3\rangle$ を

$$|\psi_1\rangle = a_1|0\rangle + b_1|1\rangle$$

$$|\psi_2\rangle = a_2|0\rangle + b_2|1\rangle$$

$$|\psi_3\rangle = a_3|0\rangle + b_3|1\rangle$$

とすると、全体の重ね合わせ $|\Psi\rangle$ は

$$\begin{aligned} |\Psi\rangle &= (a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) \\ &\quad \otimes (a_3|0\rangle + b_3|1\rangle) \\ &= a_1a_2a_3|0,0,0\rangle + a_1a_2b_3|0,0,1\rangle \\ &\quad + a_1b_2a_3|0,1,0\rangle + a_1b_2b_3|0,1,1\rangle \\ &\quad + b_1a_2a_3|1,0,0\rangle + b_1a_2b_3|1,0,1\rangle \\ &\quad + b_1b_2a_3|1,1,0\rangle + b_1b_2b_3|1,1,1\rangle \end{aligned}$$

となる(図3参照)。

では、この状態の重ね合わせがどのように計算と結びつくのであろうか？まず、個々の基底状態(基底ベクトル)を1つのコンピュータと考え、3+2を計算することを考える。3と2は2進数でそれぞれ11と10であるから、基底ベクトルとして $|1,1,1,0,0,0,0\rangle$ を取って考える。ここで、最初の2ビットの値3と次の2ビットの値2が加算されて最後の3ビットにセットされるものとする。この時、

$$\begin{aligned}
|\psi_1\rangle &= a_1|0\rangle + b_1|1\rangle & |\psi_2\rangle &= a_2|0\rangle + b_2|1\rangle & |\psi_3\rangle &= a_3|0\rangle + b_3|1\rangle \\
\end{aligned}$$

$$|\Psi\rangle = (a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) \otimes (a_3|0\rangle + b_3|1\rangle)$$

図 3 3 ビットの重ね合わせ

$$| \underbrace{1,1}_3, \underbrace{1,0}_2, \underbrace{0,0,0}_0 \rangle \rightarrow | \underbrace{1,1}_3, \underbrace{1,0}_2, \underbrace{1,1,0}_5 \rangle$$

なる状態の変化を引き起こすことができれば $3+2=5$ が計算できたことになる(つまり、このような状態の変化を起こす物理系を構成したい)。これが量子コンピュータでの計算の原理である。

次に、複数の基底ベクトルが重ね合わされている時、同様に状態を変化させるとどうなるか考えてみよう。重ね合わされた初期状態を

$$\frac{1}{2^{1/2}}|1,1,0,1,0,0,0\rangle + \frac{1}{2^{1/2}}|1,1,1,0,0,0,0\rangle \text{ とし、}$$

$$\frac{1}{2^{1/2}}| \underbrace{1,1}_3, \underbrace{0,1}_1, \underbrace{0,0,0}_0 \rangle + \frac{1}{2^{1/2}}| \underbrace{1,1}_3, \underbrace{1,0}_2, \underbrace{0,0,0}_0 \rangle$$

$$\begin{aligned}
\rightarrow \frac{1}{2^{1/2}}| \underbrace{1,1}_3, \underbrace{0,1}_1, \underbrace{1,0,0}_4 \rangle \\
+ \frac{1}{2^{1/2}}| \underbrace{1,1}_3, \underbrace{1,0}_2, \underbrace{1,1,0}_5 \rangle
\end{aligned}$$

なる状態の変化が起こったとする。先ほどと同様に解釈すると、最初の基底ベクトルの変化は $3+1=4$ 、もう一方の基底ベクトルの変化は $3+2=5$ を表現すると解釈することができる。すなわち、重ね合わされた状態で計算(状態の変化)を実行すると、複数の計算が同時に実行されたとみなされる。このような計算を量子並列計算(*quantum parallel computation*)という。より一般的に、ある関数 $f(x_1, x_2, \dots, x_n)$ の計算を量子並列計算で実行する時には、

$$\sum_i \alpha_i |a_{i1}, a_{i2}, \dots, a_{in}, 0\rangle$$

$$\rightarrow \sum_i \alpha_i |a_{i1}, a_{i2}, \dots, a_{in}, f(a_{i1}, a_{i2}, \dots, a_{in})\rangle$$

と表される。

2.3 観測

前節で述べたように、いくつかの基底の重ね合わせを用いてある種の並列計算が実行できる。では、量子並列計算は理想的な並列計算と考えることができるだろうか？つまり、量子コンピュータは(理論的には)無限個のプロセッサを持つコンピュータなのだろうか？現時点ではその答えは不明である。はっきりと言えることは、物理的な観測という処理が大きな制約を与えているということである。

観測(*observation* または *measurement*)とは、量子物理学で新たに導入された物理学概念であり、微視的な物理状態から巨視的な物理量への変換を行うことを意味している。重ね合わされた状態を観測すると、量子物理学の法則に従ったある確率でどれか1つの状態(物理量)を得ることができる。水素原子の場合、重ね合わされた状態を

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

とすると、値0(基底状態)を得る確率は $|a|^2$ となり、値1(励起状態)を得る確率は $|b|^2$ となる。つまり、以前述べた条件 $|a|^2 + |b|^2 = 1$ は全体の確率が1に正規化されていることを意味している(図4参照)。

図4 水素原子の観測

次に、一般の観測を考えよう。任意の状態ベクトル

$$|\psi\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

に対し、共役なベクトル $\langle\psi|$ を

$$\langle\psi| = (a_1^*, a_2^*, \dots, a_n^*)$$

と記述する。ここで、 $a_i^* (i=1, 2, \dots, n)$ は a_i の複素共役である。この時、 $\langle\varphi| = (b_1^*, b_2^*, \dots, b_n^*)$ と $|\psi\rangle$ の内積は

$$\langle\varphi|\psi\rangle = (b_1^*, b_2^*, \dots, b_n^*) \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \sum_{i=1}^n b_i^* a_i$$

となる。

今、任意の状態ベクトル $|\varphi\rangle$ は正規化されており、

$$|\langle\varphi|\varphi\rangle|^2 = 1$$

が成り立つとする。さらに、基底ベクトルを $|e_i\rangle$ ($i=1, 2, \dots, n$) とし、計算が終了した時の状態の重ね合わせを $|\psi\rangle = \sum_{i=1}^n a_i |e_i\rangle$ とする。ここで、ある物理量 φ (対応するベクトル $|\varphi\rangle$) に対し、 $\varphi(|\psi\rangle)$ の $|\varphi\rangle$ 方向の成分を得る確率は

$$|\langle\varphi|\psi\rangle|^2 = \left| \sum_{i=1}^n a_i \langle\varphi|e_i\rangle \right|^2$$

であり、特に、 e_i を得る確率は $|a_i|^2$ である。

量子コンピュータは、いくつかの状態を重ね合わせるにより、複数の計算を同時に並列的に実行することができる。しかし、結果を得るためには観測を行う必要があり、一般には、同時に複数の結果を得ることができないという制約がある。

3. 量子 Turing 機械

従来のコンピュータの数学的モデルとして Turing 機械、回路やセルオートマトンなどが存在するように、量子コンピュータのモデルとしても量子 Turing 機械、量子回路や量子セルオートマトンなどが提案されている。ここでは、Deutsch により提案され、Bernstein と Vazirani により数学的に形式化された量子 Turing 機械の定義を与える [3, 7]。

量子 Turing 機械 M は、通常の Turing 機械と同様に、有限制御部、無限長のテープ、およびテープヘッドから構成される。以下では、 $\mathbf{C}$ 、 $\mathbf{Z}$ をそれぞれ複素数全体の集合、整数全体の集合とする。

定義 3.1 [3] 量子 Turing 機械 (quantum Turing machine, QTM) とは、7 項組 $M = (Q, \Sigma, \Gamma, U, q_0, B, F)$ のことをいう。ただし、

Q は状態の有限集合、

Γ はテープ記号の有限集合、

$B \in \Gamma$ は空白記号、

$\Sigma \subseteq \Gamma$ は入力記号の集合、

δ は状態遷移関数 (state transition function)

で、 $Q \times \Gamma \times \Gamma \times Q \times \{L, R\}$ から $\mathbf{C}$ への写像、

$q_0 \in Q$ は初期状態、

$F \subseteq Q$ は受理状態の集合とする。

ある時点における QTM M の状態を $p \in Q$ 、テープヘッドが置かれているマス目の番号を $h \in \mathbf{Z}$ 、テープに書き込まれた内容を $\alpha \in \Gamma^*$ (すなわち、 α は Γ の記号からなる列) とするとき、 M の様相 (configuration) を 3 項組 (p, h, α) で定義する。ま

た、状態遷移関数 $\delta(p, a, b, q, d) = A_m$ は、 M が状態 p で記号 a を読んでいる時 (M のこのときの様相を c_1 とする) に、テープ上でヘッドが走査しているマス目に記号 b を書き込み、状態を q に遷移させ、さらにヘッドを方向 $d \in \{L, R\}$ に 1 マス分動かす (M のこのときの様相を c_2 とする) という**事象の振幅** (amplitude) が A_m であることを表している (図 5 参照)。この時、 M が様相 c_1 から c_2 に遷移する確率は $|A_m|^2$ であると定義する。このため、 A_m を確率振幅ともいう。

この状態遷移関数 δ は、 M の様相の重ね合わせの線形空間における線形写像を定義し、この線形写像は次のような行列 M_δ で表される。 M_δ の各行、各列には M の様相が対応する。 c_1, c_2 を M の 2 つの様相とする時、 M_δ の c_1 行 c_2 列成分は c_2 から c_1 への M の 1 ステップの遷移に対応した δ の値 A_m である。

$$M_\delta = c_1 \begin{pmatrix} & c_2 \\ \cdots & A_m & \cdots \\ & \vdots & \end{pmatrix}$$

もし、このような遷移が M に存在しなければ、 M_δ の対応する成分は 0 である。この行列 M_δ を M の**時間推移行列** (time evolution matrix) という。つまり、状態遷移関数は時間推移行列に、様相は状態ベクトルに対応する。時刻 0 における状態ベクトルを $|\psi(0)\rangle$ 、 s 時間後の状態ベクトルを $|\psi(s)\rangle$ とし、QTM が 1 ステップの実行に要する時間を τ とすると、

$$|\psi(\tau t)\rangle = M_\delta |\psi(0)\rangle$$

となる。また、

図 5 状態遷移関数 δ

仮定：QTM においては、時間推移行列 M_δ は**ユニタリ行列**でなければならないものとする。

すなわち、 $M_\delta^\dagger$ を M_δ の転置共役行列、 I を単位行列とする時、 $M_\delta^\dagger M_\delta = M_\delta M_\delta^\dagger = I$ が成り立つものとする。この制約が QTM が単なる数学的モデルではなく、量子物理学を組み込んだモデルとなる理由である。

例えば、時間推移行列 M_δ を

$$M_\delta = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

様相 c_1, c_2, c_3, c_4 に対応する状態ベクトル (計算の基底) $|c_1\rangle, |c_2\rangle, |c_3\rangle, |c_4\rangle$ を

$$|c_1\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |c_2\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$|c_3\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |c_4\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

としよう。この時、計算前の様相 $|\psi_I\rangle$ を

$$|\psi_I\rangle = \alpha |c_1\rangle + \beta |c_3\rangle = \begin{pmatrix} \alpha \\ 0 \\ \beta \\ 0 \end{pmatrix}$$

とすると、計算後の様相 $|\psi_0\rangle$ は

$$\begin{aligned} |\psi_0\rangle &= M_\delta |\psi_I\rangle \\ &= \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} \alpha \\ 0 \\ \beta \\ 0 \end{pmatrix} \end{aligned}$$

$$= \begin{pmatrix} \alpha a_{11} + \beta a_{13} \\ \alpha a_{21} + \beta a_{23} \\ \alpha a_{31} + \beta a_{33} \\ \alpha a_{41} + \beta a_{43} \end{pmatrix}$$

$$= (\alpha a_{11} + \beta a_{13}) |c_1\rangle + (\alpha a_{21} + \beta a_{23}) |c_2\rangle$$

$$+ (\alpha a_{31} + \beta a_{33}) |c_3\rangle + (\alpha a_{41} + \beta a_{43}) |c_4\rangle$$

となる。

4. 素因数分解問題

Shor は、QTM が整数の素因数分解問題と離散対数問題を効率的に解くことができることを示した[19]。これらの問題は、現在のコンピュータでは効率的に解くアルゴリズムが知られていない問題であり、これらの問題の解法の困難さが RSA などの公開鍵暗号系の安全性のよりどころとなっている。ここでは、素因数分解問題を QTM で効率的に解くアルゴリズムの概略を説明する。なお、このアルゴリズムを拡張することにより、ある種の周期関数の周期を効率的に求めることができることが知られている[17]。

2つの素数 p と q の積 $N = p \cdot q$ が与えられた時、 N から p と q を求める問題が整数の**素因数分解問題**である。一般性を失うことなく、 N は奇数とする。いま、 $a-b$ が n で割り切れる時、 $a \equiv b \pmod{n}$ と書く (a と b は n を法として**合同**であるという[23]) ことにすると、

$$x^2 \equiv 1 \pmod{N}$$

を満たす自明でない解 x を求めることができれば、 N を素因数分解することができることは良く知られた結果である[6]。つまり、上記方程式は

$$(x-1)(x+1) \equiv 0 \pmod{N}$$

と変形できるため、自明でない解を X とすると、

$$\gcd(X \pm 1, N)$$

により p または q を求めることができる。ここで、 $\gcd(a, b)$ は a と b の最大公約数を表す。最大公約数はユークリッドの互除法により効率的に求めることができる。

例えば、851 を素因数分解することを考える。

$$482^2 \equiv 1 \pmod{851}$$

であるから

$$\gcd(482-1, 851) = 37$$

より $851 = 37 \times 23$ と素因数分解できる。

では、どうすれば X を効率的に求めることができるのだろうか？ Shor は、まず、ある正整数 y ($1 < y \leq N-1$) に対して、

$$y^r \equiv 1 \pmod{N}$$

を満たす r が求まり、かつ r が偶数のときには、 $x = y^{r/2}$ とすると

$$x^2 = (y^{r/2})^2 \equiv 1 \pmod{N}$$

となることから、

$$\gcd(y^{r/2} \pm 1, N)$$

により p または q を求めることができることを示した。さらに、QTM が効率的に高い確率で r を求めることができることを示した。

Shor の r を求めるアルゴリズムでは、量子 Fourier 変換を有効に用いている。行列の次元を q とすると、**量子 Fourier 変換** (quantum Fourier transform) A_q は

$$|a\rangle \xrightarrow{A_q} \frac{1}{q^{1/2}} \sum_{c=0}^{q-1} e^{2\pi i ac/q} |c\rangle$$

で定義される。ここで、 i は虚数単位 $i^2 = -1$ である。具体的には A_q は、 $\omega = e^{2\pi i/q}$ とすると ($\omega^q = 1$ である)、

$$A_q = \frac{1}{q^{1/2}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \omega^2 & \cdots & \omega^{q-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(q-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{q-1} & \omega^{(q-1)2} & \cdots & \omega^{(q-1)(q-1)} \end{pmatrix}$$

となり、高速 Fourier 変換で良く知られた行列になる。つまり、量子 Fourier 変換は、高速 Fourier 変換の量子版である。高速 Fourier 変換は q の多項式時間で実行できることが知られているが、QTM では量子並列計算を用いることにより

$\log_2 q$ の多項式時間で実行できる。従来のコンピュータでも、 q 台のコンピュータで並列計算を行えば $O(\log_2 q)$ 時間で実行できるが、 q が大きくなればこの方法は実用的でなくなる。これに対し、QTM の量子並列計算は 1 台のコンピュータ上で行われる。

Shor のアルゴリズムでは、まず、 $N^2 \leq q < 2N^2$ を満たし、かつ 2 の冪乗となる値 q を求める。そして、QTM の初期様相を、

$$|0, 0\rangle \rightarrow \frac{1}{q^{1/2}} \sum_{a=0}^{q-1} |a, 0\rangle$$

に設定する。以下では、様相 $|s_1, s_2\rangle$ の、 s_1 を第 1 レジスタ、 s_2 を第 2 レジスタと呼ぶことにする。

次に、 y ($1 < x \leq N-1$) をランダムに選び、 $a = 0, \dots, q-1$ に対し $y^a \pmod{N}$ の値を量子並列計算で計算し、その値を第 2 レジスタにセットする。この時、様相は以下ようになる。

$$\frac{1}{q^{1/2}} \sum_{a=0}^{q-1} |a, y^a \pmod{N}\rangle$$

ここで、第 2 レジスタを観測する。観測結果を Y とすると、任意の非負整数 h とある k に対し、 $Y \equiv y^k \equiv y^{k+hr} \pmod{N}$ が成り立つ。すなわち、 a として、 $a = k, k+r, k+2r, \dots, k+mr$ ($m = \lfloor (q-k-1)/r \rfloor$) なる値を取ることができるため、観測後の様相は、

$$\frac{1}{(m+1)^{1/2}} \sum_{h=0}^m |k+hr, Y\rangle$$

となる。係数 $1/(m+1)^{1/2}$ により、様相の大きさ (内積の絶対値) が 1 になるように正規化されている。

次に、量子 Fourier 変換 A_q を第 1 レジスタに対して実行する。簡単のため、本解説では q/r は整数とする (一般の場合は、Shor の原論文 [19, 20] を参照していただきたい)。この時、

$$\begin{aligned} & \frac{1}{(m+1)^{1/2}} \sum_{h=0}^m |k+hr, Y\rangle \\ & \xrightarrow{A_q} \frac{r^{1/2}}{q} \sum_{b=0}^{q-1} \sum_{h=0}^{q/r-1} e^{2\pi i(k+hr)b/q} |b, Y\rangle \\ & = \frac{r^{1/2}}{q} \sum_{b=0}^{q-1} e^{2\pi i k b/q} \sum_{h=0}^{q/r-1} e^{2\pi i h r b/q} |b, Y\rangle \end{aligned}$$

$$= \frac{1}{r^{1/2}} \sum_{j=0}^{r-1} e^{2\pi i k b/q} |j \frac{q}{r}, Y\rangle$$

となる。ここで、最後の式を導くために整数論で良く知られた性質

$$\sum_{j=0}^{n-1} e^{2\pi i j b/n} = \begin{cases} n, & b \text{ が } n \text{ の整数倍のとき} \\ 0, & \text{その他のとき} \end{cases}$$

を用いている [23]。この性質は物理的には、いくつかの波が干渉を起こし、山と山 (または谷と谷) が増幅しあい、山と谷にあたる波が打ち消しあうことを意味している。

最後に、第 1 レジスタを観測する。この時、ある j ($0 \leq j < r$) に対して、 $b = jq/r$ を得る。 b と q は既知なので、 $\gcd(j, r) = 1$ ならば r を求めることができるが、素数定理 [23] から高い確率で $\gcd(j, r) = 1$ となることがわかる。

Shor のアルゴリズムは、 $y^r \equiv 1 \pmod{N}$ を満たす r を効率的に求めるアルゴリズムである。ここで、 $f(x) = y^x \pmod{N}$ なる関数 $f(x)$ を考えると、任意の k に対して $f(k) = f(k+r)$ となる。このため、Shor のアルゴリズムはある種の周期関数の周期を効率的に求めるアルゴリズムに拡張することができる [17]。

5. おわりに

本解説では、量子コンピュータがいかなる発想の下で誕生した計算機モデルであるかを紹介し、さらに、その一形式化である量子 Turing 機械 (QTM) と、QTM を用いた素因数分解問題の効率的な解法について簡単に紹介した。

量子コンピュータの形式化としては、QTM 以外にも量子回路 [8, 25] や量子セルオートマトン [10, 11, 24] などが提案されている。これらのモデルに基づく量子コンピュータは、従来のコンピュータより高性能なのかという疑問に対し、現時点では、はっきりとした答えを述べることはできない。しかし、本解説で述べた素因数分解問題の他にも QTM が、従来のコンピュータよりも効率的に解くことができると予想される問題がいくつか

知られている[9,14,21,22]。さらに、従来の計算量クラスと量子計算量クラスの関係に関する研究も行われている[1,3,4,5,16]。

量子コンピュータは物理的に実現可能なのだろうか？量子コンピュータは実現不可能であると主張する研究者もいる。しかし、量子コンピュータの構成に利用できる基本的な量子ゲートや量子回路がすでに提案され(例えば、[2])、実験室レベルではあるが小規模な実現は行われている。つまり、数年先とは言わないまでも、数十年先には実用レベルの量子コンピュータが登場すると予想してもおかしくはない状況になっている。

参 考 文 献

- [1] L. M. Adleman, J. DeMarrais, and M.-D. A. Huang, "Quantum computability", *SIAM J. Comput.*, **26**, pp.1524-1540, 1997.
- [2] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. W. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary gates for quantum computation", *Phys. Rev.*, **A 52**, pp.3457-3467, 1995.
- [3] E. Bernstein and U. V. Vazirani, "Quantum complexity theory(Extended Abstract)", in: *Proc. of the 25th ACM Symp. on Theory of Computing*, pp. 11-20, 1993.
- [4] E. Bernstein and U. V. Vazirani, "Quantum complexity theory", *SIAM J. Comput.*, **26**, pp.1411-1473, 1997(文献[3]の Journal 版).
- [5] C. H. Bennett, E. Bernstein, G. Brassard, and U. V. Vazirani, "Strength and weaknesses of quantum computing", *SIAM J. Comput.*, **26**, pp.1510-1523, 1997.
- [6] H. Cohen, *A course in computational algebraic number theory*, Springer-Verlag, 1993.
- [7] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer", *Proc. R. Soc. London*, **A 400**, pp.97-117, 1985.
- [8] D. Deutsch, "Quantum computational networks", *Proc. R. Soc. London*, **A 425**, pp.73-90, 1989.
- [9] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation", *Proc. R. Soc. London*, **A 439**, pp.553-558, 1992.
- [10] C. Dürr and M. Santha, "A decision procedure for well-formed linear quantum cellular automata", in: *Proc. of the 37th Ann. Symp. on Theoretical Aspects of Computer Science*, pp.281-292, 1996.
- [11] C. Dürr and M. Santha, "A decision procedure for unitary linear quantum cellular automata", in: *Proc. of the 37th Ann. Symp. on Foundations of Computer Science*, pp.38-45, 1996.
- [12] R. P. Feynman, "Simulating physics with computers", *Int. J. Theor. Phys.*, **21**, pp.467-488, 1982.
- [13] R. P. Feynman, "Quantum mechanical computers", *Foundation of Physics*, **16**, pp.507-531, 1986.
- [14] L. K. Grover, "A fast quantum mechanical algorithm for database search", in: *Proc. of the 28th ACM Symp. on Theory of Computing*, pp.212-219, 1996.
- [15] A. メシア (小出昭一郎, 田村二郎 訳), 量子力学 I, 東京図書, 1971.
- [16] T. Mihara, "Are measurements effective on quantum computation?", *IEICE Trans. Inf. & Sys.*, **E79-D**, pp.382-384, 1996.
- [17] T. Mihara, "On the complexity of finding cycles in periodic functions using the quantum Turing machine", *IEICE Trans. Inf. & Sys.*, **E79-D**, pp.579-583, 1996.
- [18] 西野哲朗, 量子コンピュータ入門, 東京電機大学出版局, 1997.
- [19] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring", in: *Proc. of the 35th Ann. Symp. on Foundations of Computer Science*, pp.124-134, 1994.
- [20] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", *SIAM J. Comput.*, **26**, pp.1484-1509, 1997(文献[19]の Journal 版).
- [21] D. R. Simon, "On the power of quantum computation", in: *Proc. of the 35th Ann. Symp. on Foundations of Computer Science*, pp.116-123, 1994.
- [22] D. R. Simon, "On the power of quantum computation", *SIAM J. Comput.*, **26**, pp.1474-1483, 1997(文献[21]の Journal 版).
- [23] 高木貞治, 初等整数論講義 第2版, 共立出版, 1971.
- [24] J. Watrous, "On one-dimensional quantum cellular automata", in: *Proc. of the 36th Ann. Symp. on Foundations of Computer Science*, pp.528-537, 1995.
- [25] A. C. Yao, "Quantum circuit complexity", in: *Proc. of the 34th Ann. Symp. on Foundations of Computer Science*, pp.352-361, 1993.

(1997年12月19日 受付)

[問い合わせ先]

〒862-8652

熊本県熊本市渡鹿 9-1-1

九州東海大学 工学部

電子情報工学科

三原 孝志

TEL : 096-386-2661

FAX : 096-381-7956

E-mail : tmihara@ktmail.ktokai-u.ac.jp

著 者 紹 介

三原 孝志 (みはら たかし)

九州東海大学 工学部電子情報工学科

1982年 広島大学理学部物理学科卒業、
1997年 北陸先端科学技術大学院大学情
報科学研究科博士後期課程修了。同年 九
州東海大学工学部電子情報工学科講師、
現在に至る。量子計算量理論の研究に従
事。博士(情報科学)。

西野 哲朗 (にし の てつろう)

電気通信大学 電子情報学科

1982年 早稲田大学理工学部数学科卒
業、1984年 同大学院理工学研究科博士前
期課程修了。同年 日本アイ・ビー・エム
(株)入社、1987年 東京電機大学助手、
1992年 北陸先端科学技術大学院大学助
教授、1994年 電気通信大学助教授、現在
に至る。回路計算量理論、量子計算量理
論、計算論的学習理論などの研究に従
事。理学博士。