

量子コンピューターの基礎

藤井 啓祐

近年、巨大 IT 企業や国家プロジェクトなどで量子コンピューターが話題になっている。複数のグループから量子コンピューターをクラウドで使えるような環境が提供されつつある。本稿では、このような研究開発が加速する量子コンピューターについて、その歴史や現状、量子ビットと従来のコンピューターにおけるビットとの違い、そして量子ビットに対する基本的な演算とそれらによって構成される量子アルゴリズムを解説する。最後に、小規模な量子コンピューターが実現しつつあることを踏まえ、それをうまく利用するための最近の研究のトレンドや今後の展望について総説する。

キーワード：量子コンピューター、量子アルゴリズム、量子情報

1. はじめに：量子コンピューターの歴史と現状

近年、Google、IBM、Microsoft、Intel といった巨大 IT 企業が軒並み量子コンピューターのハード開発へ参入して話題になっている。まだまだ規模的には小さいものの、IBM は量子コンピューターをクラウドで公開し、一部誰でも無料で使える環境を提供している（詳しくは本特集の今道貴司氏、井床利生氏、ルディー・レイモンド氏の記事を参照）。

最近になって急速に進展しているように思われるかもしれないが、量子コンピューターの研究は、30 年以上も前から着実に進められてきた。1980 年代、“Information is Physical” というスローガンのもと情報と物理との再統合が模索されるなか、素粒子物理学に関する研究でノーベル賞を受賞したファインマンは、（量子力学に従う）自然を効率よくシミュレーションしたければ量子力学のルールで動作するコンピューターを作らないといけないという指摘をする [1]。その後、1985 年にドイツが現在の万量子コンピューターにつながる量子チューリングマシンを定式化する [2]。さらに、1994 年にショアによる素因数分解アルゴリズムの発見 [3] によって、量子コンピューター研究は多くの物理学者および計算機科学者の注目を集めるようになった。1990 年代後半には量子ビットやそれに対する量子演算の実証実験も行われ、量子コンピューター研究の第一次ブームが到来する。その後 2000 年代に入って、実験的な難しさや、簡単にできるような理論研究がやり尽くされたことによって、量子コンピューター研究

は下火になる。しかし、量子誤り訂正符号理論の進展による実験的要求の緩和や、それを基礎物理研究へと応用する地道な基礎研究は着実に進んでいた。実験においても、量子ビットのエラー率の低減による高忠実化やスケラビリティの改善など基礎研究が進められていた。それら基礎研究が再び注目を集めるようになったのは、2014 年にマルチネス (UCSB) のグループから発表された超高忠実度（低雑音）の量子ビットと量子演算の実現であろう [4]。すでに D-Wave の量子アニーリングマシンを購入してその性能検証をしていた Google は、2014 年にマルチネスをグループごとに取り込み、回路型の量子コンピューターのデバイス研究を開始し、世界のほかのグループも追従する形で量子コンピューターの実現に向けて本腰を入れることになった。

Google や IBM はすでに 10–20 量子ビットを実現しており [5, 6]、次の数年で 50–100 量子ビットの量子コンピューターが実現する見込みである。これら巨大 IT 企業だけでなく、2013 年にはリゲッティが IBM から独立して Rigetti Computing を起業し、現在では 19 量子ビットの量子コンピューターに至っている [7]。超伝導量子ビット方式だけではなくイオンを用いたイオントラップ型量子コンピューターについても、米国の大学からは IonQ という企業が立ち上がり、ヨーロッパの大学からは Alpine Quantum Technologies が立ち上がった。IBM に続いて Google や IonQ もクラウドで量子コンピューターを提供する予定であり、今後複数の量子コンピューターを小規模ではあるが使える環境が整いつつある。

本稿では、このような量子コンピューターの基礎、量子ビットの記述や基本量子演算、そして量子アルゴリズムについて解説する。また、最近注目を集めている、小規模な量子コンピューター上でも動くよう

ふじい けいすけ
 京都大学大学院理学研究科物理学・宇宙物理学専攻
 〒 606-8502 京都府京都市左京区北白川追分町
 fujii.keisuke.2s@kyoto-u.ac.jp

に設計された古典・量子ハイブリッドアルゴリズムについても簡単に紹介する。残念ながら、これらの比較的小規模の量子コンピューターにおいて実用的な問題を古典コンピューターよりも安価にそして高速に解くことは容易ではないだろう。最後に、大規模な量子コンピューターの実現に向けた取り組みとそこで必須になる量子誤り訂正と精度保証された誤り耐性量子コンピューターについて紹介したい。

2. 量子計算の基礎

2.1 量子力学と量子ビット

従来の計算機（以降古典コンピューターと呼ぶことにする）では、スイッチのオン・オフや電圧の高・低など、二つの異なる状態を用いてビット0と1を物理的に表現することによって計算が行われる。古典コンピューター上では必ず0もしくは1のどちらか一方の状態をとることが要求されている。しかし、不思議なことに量子力学の世界では、二つの異なる状態のどちらかがまだ確定していない量子的重ね合わせ状態が許されている。重ね合わせ状態は量子力学に特有の現象で、われわれの日常の直感に反するのでなかなか想像しにくい。本来0と1のどちらかしかとらないものが量子の世界では不思議なことに、0から1への変換が途中で止まってしまい0なのか1なのかよくわからない中間的な、実にあやふやな状態が許されているのだ。このため、一つの量子力学的なビット（量子ビット）は、0と1という離散化された整数で表現することができず、0と1がどの程度の重みでの重ね合わせ状態になっているかという情報を二つの複素数 α と β を用いて2次元のベクトルとして表現することになる：

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

ここで用いた記号 $|\cdot\rangle$ はディラックのブラケット表記というもので、列ベクトル（ヒルベルト空間の元）をケット $|\cdot\rangle$ 、行ベクトル（双対空間の元）をブラ $\langle\cdot|$ で記述する。このルールに従うと内積は $\langle\phi|\psi\rangle$ と書ける。

$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ と $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ は計算に用いる二つの基本となる状態で、2次元の複素ベクトル空間の直交基底になっている。光子の偏光、電子・核スピン、原子の電子状態、量子化された電気回路の状態など、量子力学で記述される二つの状態であれば物理的にはどのようなものでもよい（超伝導量子ビットについては詳しくは本特集の川畑史郎氏の記事を参照）。二つの複

素数 α と β は、どの程度の重みで0と1が重ね合わさっているかを表す複素数である複素確率振幅と呼ばれ、規格化条件 $|\alpha|^2 + |\beta|^2 = 1$ を満たす。量子ビットを測定したときに0,1の測定結果を得る確率は、この複素確率振幅の絶対値の2乗

$$p_0 = |\alpha|^2, \quad p_1 = |\beta|^2, \quad (2)$$

で与えられる（ボルン則と呼ばれる）。

0と1が確率的に与えられるならば、確率分布を用いてビットを記述し、乱数を用いることで量子計算を古典コンピューターでもシミュレーションできると言うかもしれない。しかし、測定前の量子状態を記述するのは確率分布ではなく複素確率振幅であることに注意しなければならない。確率ではなく、ある意味確率よりももっと原始的である複素確率振幅が物理法則によって時間発展していくのが量子力学であり、それを計算に利用するのが量子計算である。

2.2 1量子ビットの基本演算

古典計算がAND, OR, NOT, NANDなどの論理演算から構成されるのと同様、量子計算も基本的な演算から構成される。一つの量子ビットの量子状態は規格化された2次元複素ベクトルとして表現されるので、それに対する演算は 2×2 の複素行列によって表現される。また、規格化条件（確率の保存）を守りたいので、ユニタリー行列が物理的に許された演算となる。一つの量子ビットに作用する基本的な量子演算であるパウリ演算子を導入する：

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad (3)$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (4)$$

X は古典ビットの反転（NOT）に対応し $X|0\rangle = |1\rangle$ 、 $X|1\rangle = |0\rangle$ のように作用する。重ね合わせのある量子ビットの場合は、 $|0\rangle$ と $|1\rangle$ の位相も情報として保持できるので、位相を反転 $Z|1\rangle = -|1\rangle$ させる Z 演算子も定義されている。 $Y = iXZ$ 演算子は位相の反転とビットの反転を組み合わせたもの（全体にかかる複素数 i を除いて）であると考えることができる。

もう少し複雑な演算を導入しておこう。アダマール演算と呼ばれる演算は

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (5)$$





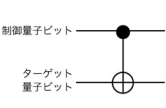

パウリX演算		$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
パウリZ演算		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
アダマール演算		$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
T演算		$\begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}$
CNOT		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
制御U演算		$\Lambda(U) = 0\rangle\langle 0 \otimes I_d + 1\rangle\langle 1 \otimes U$

図1 基本的な量子演算とその量子回路図

で与えられ、初期化された状態 $|0\rangle$ に作用させると $H|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ のように重ね合わせ状態が得られる。この重ね合わせ状態を測定すると 0 と 1 の測定結果が 1/2 の確率でランダムに出力される。しかし、測定するまでは 0 であるか 1 であるかは全く確定していない状態であることを注意しておこう。仮に 0 か 1 かがこの時点で確率 1/2 で確定していたとしてみる。さらにアダマール演算を作用させると 0 の場合も、1 の場合も重ね合わせ状態が得られ、 $H|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ もしくは $H|1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ が得られる。それぞれの場合に再び測定すると 1/2 の確率で 0 と 1 が得られるので、1 回目のアダマール演算の後に 0 か 1 かが確率 1/2 で確定しているとする、2 回目のアダマール演算を作用させたとき結局 0 と 1 が確率 1/2 で得られることになってしまう。しかし、量子力学に基づいて計算すると $HH|0\rangle = |0\rangle$ となるので、アダマール演算を 2 回作用させると必ず 0 の測定結果を得ることになる (IBM Q で簡単に実行できるので実際にやってみるのがよいだろう)。つまり、一度重ね合わせ状態が再び一つの状態に戻ることができる。このような現象は干渉と呼ばれ、波の性質をもつ複素確率振幅に特有の性質であるといえよう。

量子ビットの複素確率振幅は複素数なので、もっと複雑な演算が定義できる。たとえば、 $|0\rangle$ と $|1\rangle$ の相対的な位相を任意の角度 θ で回転させる

$$e^{-i(\theta/2)Z} = \cos(\theta/2)I - i \sin(\theta/2)Z = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}, \quad (6)$$

なども量子演算となる。

一つの量子ビットに対する任意の演算、つまり 2×2 のユニタリー行列はアダマール演算 H とこの Z 回転に分解することができる。さらに、任意の回転ではなく $\theta = \pi/4$ の Z 回転があればそれと H を組み合わせれば任意の 1 量子ビット演算を構成できることも知られている [8]。よって、 $\{H, T = e^{-i(\pi/8)Z}\}$ は 1 量子ビットの演算のための基本演算であるといえる。これらの量子演算とその量子回路図を図 1 に示す。

2.3 複数量子ビットの記述

1 量子ビットだけでは、2 次元ベクトルのユニタリー変換しか扱うことができないため、量子計算本来の威力は発揮できない。量子ビットが複数ある状況を取り扱う必要がある。 n 個の古典ビットの状態は n 個の 0, 1 の数字によって表現され、そのパターンの総数は 2^n 個ある。量子力学では、これらすべてのパターンの重ね合わせ状態が許されているので、どのビット列がどのような重みで重ね合わせになっているかという 2^n 個の複素確率振幅で記述される：

$$|\psi\rangle = c_{00\dots 0}|00\dots 0\rangle + c_{00\dots 1}|00\dots 1\rangle + \dots + c_{11\dots 1}|11\dots 1\rangle = \begin{pmatrix} c_{00\dots 0} \\ c_{00\dots 1} \\ \vdots \\ c_{11\dots 1} \end{pmatrix}. \quad (7)$$

ただし、複素確率振幅は規格化 $\sum_{i_1, \dots, i_n} |c_{i_1 \dots i_n}|^2 = 1$ されているものとする。このように n 量子ビットの重ね合わせ状態は、 n に対して指数的に大きい 2^n 次元の複素ベクトル空間で記述する必要があり、ここに古典ビットと量子ビットの違いが顕著に現れる。この n 量子ビットの量子状態を測定するとビット列 $i_1 \dots i_n$ が確率

$$p_{i_1 \dots i_n} = |c_{i_1 \dots i_n}|^2, \quad (8)$$

でランダムに得られる。先に紹介した 1 量子ビット演算を n 量子ビットのうちの k 番目の量子ビットに作用させる場合は、それぞれの複素確率振幅を

$$c'_{i_1 \dots j_k \dots i_n} = \sum_{i_k} (A)_{j_k, i_k} c_{i_1 \dots i_k \dots i_n}, \quad (9)$$

のようなルールで変換する。ここで A はある 1 量子ビット演算を表す 2×2 ユニタリー行列で、 $(A)_{j,i}$ は A の j 行 i 列の要素を表す。このようにして作った (ベクトル空間のテンソル積もしくは行列のクロネック

カー積と呼ばれる [8]) 2^n 次元に作用する行列もやはりユニタリー行列である. たとえば, 初期化された n 量子ビット $|00\dots0\rangle$ のすべての量子ビットにアダマール演算 $H^{\otimes n}$ を作用させると

$$H^{\otimes n}|00\dots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i_1\dots i_n} |i_1\dots i_n\rangle, \quad (10)$$

のようにすべてのビット列の均等な重ね合わせ状態が得られる.

2.4 複数量子ビットの演算

複数量子ビットの記述ができたので, 複数量子ビットに作用する演算を定義しよう. 2 量子ビットに作用する演算として制御 NOT 演算 (CNOT)

$$\Lambda(X) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X \quad (11)$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (12)$$

がある. \otimes はテンソル積 (行列であればクロネッカー積) を意味する. 4 行 4 列の行列成分はそれぞれ $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ 成分に対応する. つまり, 一つ目の量子ビットが $|0\rangle$ の場合は何もせず (恒等演算 I が作用), $|1\rangle$ の場合に二つ目の量子ビットに X を作用させる. 一つ目の量子ビットを制御量子ビット, 二つ目の量子ビットをターゲット量子ビットと呼ぶ. 制御 NOT 演算の作用は, \oplus を mod 2 の足し算として,

$$\Lambda(X)|ij\rangle = |i(i \oplus j)\rangle, \quad (13)$$

と書けるので, 古典計算における排他的論理和 (XOR) を可逆にしたものである. たとえば, 一つ目の量子ビットを $|0\rangle$ と $|1\rangle$ の重ね合わせ状態にし, 二つ目の量子ビットを $|0\rangle$ として

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad (14)$$

に CNOT を作用させると,

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad (15)$$

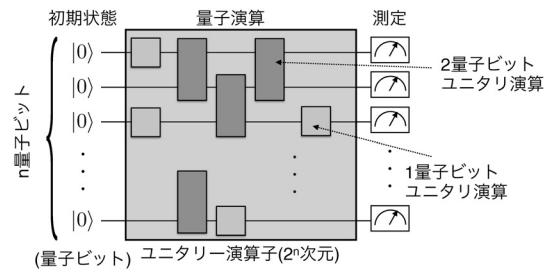


図2 量子回路図

が得られる. この状態は

$$|\psi\rangle \otimes |\phi\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle), \quad (16)$$

のように二つ目の量子ビットを分離して記述することができず, 二つ目の量子ビット間に量子的な相関 (エンタングルメント) が形成された状態である.

1 量子ビット演算の場合と同様に n 量子ビット系の k 番目と l 番目の量子ビットへの 2 量子ビット演算の作用は

$$c'_{i_1\dots j_k\dots j_l\dots i_n} = \sum_{i_k, i_l} (B)_{j_k j_l, i_k i_l} c_{i_1\dots i_k\dots i_l\dots i_n}, \quad (17)$$

となる. ただし $(B)_{j_k j_l, i_k i_l}$ は 4×4 行列で表現される 2 量子ビット演算の $j_k j_l$ 行 $i_k i_l$ 列成分 (行や列のインデックスを 2 進数表記している) である.

この CNOT 演算を先に紹介した H 演算及び T 演算と図 2 の量子回路図のように組み合わせることによって, 万能量子計算, すなわち任意の $2^n \times 2^n$ ユニタリー行列を任意の精度で構築できることが知られており, これらの基本演算は万能量子演算セット (universal set of gates) と呼ばれている.

制御演算を一般化することによって一般のユニタリー演算 U について制御ユニタリー演算 $\Lambda(U)$ を

$$\Lambda(U) = |0\rangle\langle 0| \otimes I_d + |1\rangle\langle 1| \otimes U, \quad (18)$$

のように定義することができる. U が作用する次元を d 次元とし, I_d は d 次元の恒等演算子 (単位行列) である.

3. 量子アルゴリズムの基礎

3.1 アダマールテスト

量子回路と古典回路の違いの詳細については本特集の高橋康博氏の記事を参照していただくことにして, ここでは量子アルゴリズムについて簡単に紹介する. まずユニタリー演算 U とその固有状態 $|\psi\rangle$ が与えられたときに, U の固有値を推定するアダマールテストについて説明する. この部分は, 一つひとつ計算を追って

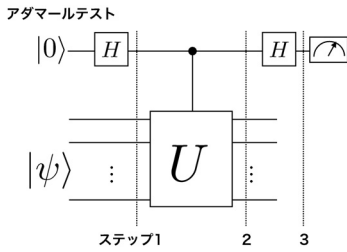


図3 アダマルテストの量子回路図

いくと固有値と確率が対応していることが理解できる。そして、このアダマルテストを位相推定というサブルーチンを使って発展させると、素因数分解問題を効率よく解くことができるショアのアルゴリズムに行き着くことができる。ただし、ショアの素因数分解アルゴリズムまでの議論は通常教科書の1章分に相当する内容を圧縮して説明している。このため、この部分がわからなくても気にすることなく次節の古典量子ハイブリッドアルゴリズムへと読み進んでいただきたい。

図3のような量子回路で定義されるアダマルテストと呼ばれる量子計算について考えてみよう。ただし、簡単のために量子状態 $|\psi\rangle$ をユニタリー演算 (行列) U の固有値 $e^{i\lambda}$ の固有状態 (固有ベクトル) とする：

$$U|\psi\rangle = e^{i\lambda}|\psi\rangle. \quad (19)$$

(一般の場合にも容易に拡張できるが議論を簡単にするため固有状態としておく。) ステップ1までの計算で

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\psi\rangle, \quad (20)$$

が得られる。ステップ2で制御 U 演算を作用させることによって、固有値が位相として得られる：

$$\frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes U|\psi\rangle), \quad (21)$$

$$= \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle + e^{i\lambda}|1\rangle \otimes |\psi\rangle), \quad (22)$$

$$= \frac{1}{\sqrt{2}}(|0\rangle + e^{i\lambda}|1\rangle) \otimes |\psi\rangle. \quad (23)$$

最後のステップ3のアダマル演算によって

$$\left(\frac{1+e^{i\lambda}}{2}|0\rangle + \frac{1-e^{i\lambda}}{2}|1\rangle \right) \otimes |\psi\rangle, \quad (24)$$

が得られる。一つ目の量子ビットを測定すると測定結果 $m = 0, 1$ を得る確率は

$$p_m = \left| \frac{1 + (-1)^m e^{i\lambda}}{2} \right|^2 = \frac{1 + (-1)^m \cos \lambda}{2}, \quad (25)$$

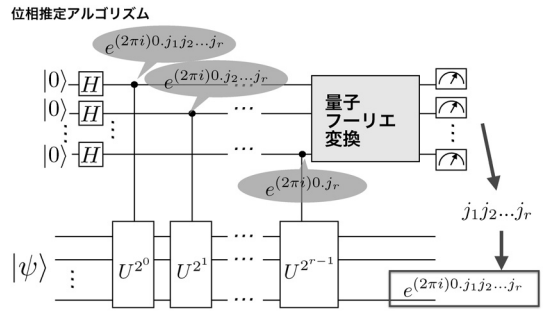


図4 位相推定量子アルゴリズム

となる $|\psi\rangle$, U , $\langle\psi|$ はそれぞれ 2^n 次元の列ベクトル, $2^n \times 2^n$ 行列, 2^n 次元の行ベクトルなので、このアダマルテストを古典コンピューター上で愚直に計算すると指数的に大きなメモリーの確保と演算回数が必要になる。一方で、量子コンピューターでは、確率分布 p_m のもとで m がサンプルされる。 $\cos \lambda$ をある誤差 ϵ で推定したい場合は、その逆数 $1/\epsilon$ の多項式回程度サンプルすればよいことになる。実際には固有ベクトルを状態として準備することは難しいが、アダマルテストを何回も繰り返すことによってどれかの固有ベクトル成分に状態が収束していき、その固有ベクトルに対する固有値が推定できる。しかし、サンプリング回数を指数的に増やすと効率が悪くなってしまいますので、固有値の推定精度を指数的に向上させることはできない。

3.2 位相推定

プローブとして r 個の量子ビットを確保し固有値の位相 λ を精度よく推定する方法が位相推定アルゴリズムである [9]。図4のように r 量子ビットをプローブとして用意し、アダマルテストと同様にアダマル演算の後、制御ユニタリー演算を作用させよう。ただし、 k 番目 ($k = 0, 1, \dots, r-1$) のプローブ量子ビットは制御 U^{2^k} 演算をすることにする。ユニタリー演算 U の固有値の位相 λ を r ビットの2進小数を用いて

$$\lambda = (2\pi)0.j_1 j_2 \dots j_r, \quad (26)$$

と書いておく。アダマルテストと同様に k 番目のプローブ量子ビットには $e^{i\lambda 2^k}$ の位相情報が獲得されるので

$$\bigotimes_{k=0}^{r-1} \frac{|0\rangle + e^{i(2\pi)0.j_r \dots j_r} |1\rangle}{\sqrt{2}}, \quad (27)$$

のような状態が得られていることになる。固有値の位相が2進小数表示で1ビットずつシフトしたものが各プローブ量子ビットに格納されている。この r 個の

プローブ量子ビットに対してフーリエ変換の量子版

$$\bigotimes_{k=0}^{r-1} \frac{|0\rangle + e^{i(2\pi)^0 \cdot j_{r-k} \dots j_r} |1\rangle}{\sqrt{2}} \rightarrow |j_1 \dots j_r\rangle, \quad (28)$$

を作用させると、プローブ量子ビットにビット列 $j_1 \dots j_r$ が得られ、位相の 2 進小数が得られる。これが位相推定アルゴリズムである。

ユニタリー演算の固有値を推定できる位相推定アルゴリズムはさまざまな量子アルゴリズムのサブルーチンとして使われている。次に紹介する素因数分解 [3], 分子の安定状態のエネルギーを計算する量子化学計算 [10], 量子メトロポリスサンプリング [11], 線型方程式を解く HHL アルゴリズム [12] やそれを利用した量子サポートベクトルマシン [13] などがその例である。

3.3 素因数分解

位相推定アルゴリズムの応用例としてショアによる素因数分解アルゴリズムを紹介しよう [3, 9]。 N を素因数分解したい整数だとしよう。 N と互いに素な整数 x を見つけてくる。適当に x を選びユークリッドの互除法で共約数が見つければ素因数分解ができることになるし、見つからなければ互いに素ということになる。以降の議論では表記を簡単にするため、 $\{|0\rangle, \dots, |N-1\rangle\}$ の基底を用いて書くことにする。 x と N を用いてユニタリー演算

$$U_x = \sum_y |xy \pmod N\rangle\langle y|, \quad (29)$$

を定義する。このユニタリー演算の場合、

$$U_x^{2^k} = \sum_y |x^{2^k} y \pmod N\rangle\langle y|, \quad (30)$$

なので、冪剰余 $x^{2^k} \pmod N \equiv e$ を先に計算しておけば U を 2^k 回作用させることなく、直接 $U_x^{2^k} = \sum_y |ey \pmod N\rangle\langle y|$ を作用させればよい。さて、位数 t を $x^t = 1 \pmod N$ を満たす整数と定義すると、固有状態のラベル $0 \leq s \leq t-1$ を用いて、 U_x の固有状態は、

$$|u_s\rangle = \frac{1}{\sqrt{t}} \sum_{k=0}^{t-1} e^{-2\pi i(s/t)k} |x^k \pmod N\rangle. \quad (31)$$

固有値は、

$$U_x |u_s\rangle = e^{2\pi i(s/t)} |u_s\rangle, \quad (32)$$

であることがわかる。位相推定をする入力状態は、 $|1\rangle = \sum_{s=0}^{t-1} |u_s\rangle$ になることから、 $|1\rangle$ を入力させると、位相推定によって確率的に固有状態 $|u_s\rangle$ が得られる。位相推定アルゴリズムを用いると、 s/t を効率

よく推定することができ、連分数展開を用いて有理数で書き直すと t の候補が得られる。 $|1\rangle$ からランダムに $|u_s\rangle$ を選ぶと高い確率で s と t が互いに素になるので、位数 t を得ることができる。そして、この位数から N の約数を見つけることができる。これがショアによる素因数分解量子アルゴリズムである [3, 9]。

素因数分解は NP 完全問題ではないものの、古典コンピュータを用いて多項式時間で解を得る方法が見つかっていない。この素因数分解問題の難しさに基づいて RSA 暗号などの暗号システムが構築されている。ファインマンの指摘のように電子などが関与する、そもそも量子力学で問題が記述されるような場合において量子コンピュータが有利であるのはある意味当然といえる。しかし、素因数分解問題のような非常に重要で、かつ一見量子力学とは全く関係のない問題において量子コンピュータが指数的な計算の加速をもたらしたことは、量子コンピュータをいっきにメジャーな学問領域へと押し上げ、さまざまな周辺分野から研究者の参入を促した。

4. 古典量子ハイブリッドアルゴリズム

最近の実験の進展に伴って (実験の解説の詳細については本特集の川畑史郎氏の記事を参照)、量子アルゴリズムの設計思想にも少し変化がみられる。前節で説明した位相推定アルゴリズムはやや複雑な量子回路であるといえる。量子回路の深さ (ステップ数) は問題のサイズに対して多項式的に深くなっていく。また制御 U 演算は必ずしも物理的に隣り合っているわけではない量子ビット間にも作用させないといけない。物理的に実現される量子演算が隣接する 2 量子ビット間に作用するものであれば、量子ビットをスワップして近接するところまで移動させるコストがさらに必要になるだろう。

現在、実験的に実現している量子演算の雑音レベルはまだ非常に高く、0.1~10% 程度のエラーを含む。このため、深い (ステップ数の多い) 量子回路を実行すると、エラーに埋もれてしまって有意義な結果は得られないだろう。これらの問題を克服する方法として、量子回路の深さを浅くすることによって、多少のエラーを含んだ量子演算でも動作する量子アルゴリズムの設計が試みられている。中でも、変分的に固有値を推定する変分量子固有値計算 (VQE: variational quantum eigensolver) が注目されている [14, 15]。

ユニタリー行列 U が効率よく記述されるエルミート行列 H を用いて $U = e^{-iH}$ と与えられているとする。そして、知りたい固有値はこのエルミート行列 H の最

小固有値であるとする。これは物理的にはよくある状況であり、 H は系のエネルギーを定義するハミルトニアン、 H の最小固有値はもっとも安定にある基底状態のエネルギーになる。VQE はこのような問題設定のもと、量子回路 $U(\{\theta_i\})$ のパラメータ $\{\theta_i\}$ を変分的に更新することによって、できるだけ H の最小固有値の固有ベクトルになるような量子状態を準備し最小固有値を近似する方法である。重要なことは、位相推定における複雑な制御 U 演算をする代わりに、量子ビットを直接測定し、それを繰り返しサンプルすることによってエルミート行列 H の平均値 (エネルギー) を求めることになる。量子回路の深さの代わりにサンプル回数を増やそうというアプローチであり、量子演算のエラーを考えると深い回路は計算をダメにしてしまうが、浅い回路を何回も独立にサンプルすることは容易にできる。個々の量子ビットの測定結果から上記のエルミート行列の平均値を計算するには古典コンピューターによる事後処理が必要になるが、この部分は古典コンピューターでも効率よく計算できる。このようにして、量子コンピューターにしか実行できない部分と古典コンピューターでも実行できる部分を切り分けた、古典・量子ハイブリッドアルゴリズムが盛んに研究されている。また、規模的には非常に小さいが、VQE を用いて水素化ヘリウムや水素化ベリリウムの基底エネルギーの量子化学計算が実証されている [15]。

量子アニーリング [16, 17] (詳しくは本特集の大関真之氏の記事を参照) もイジング型ハミルトニアン H に対する低エネルギー状態を量子ダイナミクスを用いて準備し、エネルギーを推定するという点で上記の問題と思想を共有しているといえる。量子断熱操作に該当する部分を回転角などの制御できるパラメータが付与された変分量子回路に置き換えるという試みも最近なされており、QAOA (quantum approximated optimization algorithm) と呼ばれている [18]。前述の量子を必然的に含む量子化学計算などの問題設定とは異なり、目標とする問題は組合せ最適化問題である。多くの場合、このような組合せ最適化問題は NP 完全問題を含む難しい問題になっている。万能量子コンピューターがあったとしても、NP 完全問題を効率よく解くことは難しいと考えられており、これら量子マシン上で動くヒューリスティックアルゴリズムが、古典コンピューター上のそれよりもよりよい近似を与えるかどうか、その計算時間にスケーリングにおいて優位性があるかどうかはまだよくわかっていない。今後、これらの量子ヒューリスティックアルゴリズムに優位性が

あるのか、あるのであればどういう問題に限定した場合は、理論と実験の両面からさらなる研究が求められる。

5. アナログ量子計算とデジタル化への道

前述のエラーを含む量子演算による量子アルゴリズムは残念ながらサイズに対してスケールしない。たとえば、0.1%のエラー確率に到達したとしても、50 量子ビット、20 ステップ、合計 1,000 個の基本演算をするとまったくエラーが発生しない確率は 0.36 程度でありギリギリ有意義な答えが得られるか否かの分かれ目である。いわば、アナログ情報として保持した量子状態に対して、アナログエラーをエンジニアリングによってできる限り抑え込む、というアナログコンピューターの様相を呈している。このような、エラーを含む有限サイズの小規模な量子計算は、近似量子計算 (approximated quantum computing) もしくは NISQ (noisy intermediate-scale quantum computing) [19] と呼ばれている。この領域において、実用的な問題設定で古典コンピューターに対する優位性があるかどうかは全くわからないし、エラーを許容する分、古典でシミュレーションすることもいくぶん容易になるだろう。また、このような理論と実験の距離に近い領域の量子コンピューターの理解が深まることによって、古典コンピューターを用いた量子コンピューターのシミュレーション方法も進化すると思われる。回路の深度が浅いものであれば 50 量子ビットを超える量子ビット数であってもシミュレーションできることが最近報告された [20, 21]。いずれにせよ、この領域は大規模な量子コンピューターの実現に向けた重要なマイルストーンであると考えられており、実際に動く量子コンピューターの理解、量子アルゴリズムの設計、そして大規模なハードウェア開発のための問題点をあぶり出すために大いに活用されると期待される。

究極的には、近似量子計算や量子アニーリングなどのアプローチはアナログ計算にすぎない。近似量子計算の場合は、万能な量子演算を有するので、理想的に動作すれば古典コンピューターに対する優位性は保証されているが、物理法則は指数個の複素数アナログパラメータを寸分狂いなく制御することを許さないだろう。量子アニーリングの場合も然りである。もちろんアナログ計算だから使いものにならないというわけでもなく、風洞実験が自動車や飛行機の設計に役立っているのと同様、特定領域においてはこれらアナログ量子計算が役に立つ局面もあると予想される。しかし、組合せ最適化問題などの問題を対象とするときには注意が必要である。なぜならば、理想的な無限精度のアナロ

グ計算を仮定してしまうと、量子をもち出さなくても NP 完全問題よりもっと難しい PSPACE 問題を多項式時間で解くことができることは昔からよく知られている [22]. しかし、このような用途で構築されたアナログコンピューターは、現在には残っていない. この問題は、古くは量子コンピューターの黎明期にランダウアによって指摘され [23], 有限の精度の素子から理論的に精度保証ができるデジタル量子コンピューターの重要性が認識されてきた. 素因数分解アルゴリズムを発見したショアは、自ら量子誤り訂正理論も構築し、アナログ情報をもつ量子状態をアナログエラーから守る誤り耐性量子コンピューターを切り開いた [8, 24]. 量子誤り訂正で保護された量子コンピューターを実現するためには、1 万から 1 億の量子ビットを集積化する必要がある. 近似量子計算以外にも、近未来的に登場するであろう 50–100 量子ビットの量子コンピューターを用いた量子誤り訂正技術の原理実証も、誤り耐性量子コンピューターに向けた重要なマイルストーンであると考えられている. 今後の進展に注目したい.

参考文献

- [1] R. P. Feynman, “Simulating physics with computers,” *International Journal of Theoretical Physics*, **21**, pp. 467–488, 1982.
- [2] D. Deutsch, “Quantum theory, the Church–Turing principle and the universal quantum computer,” In *Proceedings of the Royal Society of London A*, **400**(1818), pp. 97–117, 1985.
- [3] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, 1994.
- [4] R. Barends, 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 J. M. Martinis, “Superconducting quantum circuits at the surface code threshold for fault tolerance,” *Nature*, **508**, pp. 500–503, 2014.
- [5] C. Neill, P. Roushan, K. Kechedzhi, S. Boixo, S. V. Isakov, V. Smelyanskiy, R. Barends, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. Fowler, B. Foxen, R. Graff, E. Jeffrey, J. Kelly, E. Lucero, A. Megrant, J. Mutus, M. Neeley, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, H. Neven and J. M. Martinis, “A blueprint for demonstrating quantum supremacy with superconducting qubits,” arXiv: 1709.06678, 2017.
- [6] <https://www-03.ibm.com/press/us/en/pressrelease/53374.wss> (2018 年 4 月 18 日閲覧)
- [7] J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong, P. Karalekas, C. B. Osborn, A. Papageorge, E. C. Peterson, G. Prawiroatmodjo, N. Rubin, C. A. Ryan, D. Scarabelli, M. Scheer, E. A. Sete, P. Sivarajah, R. S. Smith, A. Staley, N. Tezak, W. J. Zeng, A. Hudson, B. R. Johnson, M. Reagor, M. P. da Silva and C. Rigetti, “Unsupervised machine learning on a hybrid quantum computer,” arXiv: 1712.05771, 2017.
- [8] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.
- [9] A. Y. Kitaev, “Quantum measurements and the Abelian stabilizer problem,” *Electronic Colloquium on Computational Complexity (ECCC)*, **3**, 1996.
- [10] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love and M. Head-Gordon, “Simulated quantum computation of molecular energies,” *Science*, **309**, pp. 1704–1707, 2005.
- [11] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin and F. Verstraete, “Quantum metropolis sampling,” *Nature*, **471**, p. 87, 2011.
- [12] A. W. Harrow, A. Hassidim and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Physical Review Letters*, **103**, article number: 150502, 2009.
- [13] P. Rebentrost, M. Mohseni and S. Lloyd, “Quantum support vector machine for big data classification,” *Physical Review Letters*, **113**, article number: 130503, 2014.
- [14] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik and J. L. O’Brien, “A variational eigenvalue solver on a photonic quantum processor,” *Nature Communications*, **5**, article number: 4213, 2014.
- [15] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow and J. M. Gambetta, “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets,” *Nature*, **549**, pp. 242–246, 2017.
- [16] T. Kadowaki and H. Nishimori, “Quantum annealing in the transverse Ising model,” *Physical Review E*, **58**, article number: 5355, 1998.
- [17] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren and D. Preda, “A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem,” *Science*, **292**, pp. 472–475, 2001.
- [18] E. Farhi, J. Goldstone and S. Gutmann, “A quantum approximate optimization algorithm,” arXiv: 1411.4028, 2014.
- [19] J. Preskill, “Quantum computing in the NISQ era and beyond,” arXiv: 1801.00862, 2018.
- [20] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, T. Magerlein, E. Solomonik and R. Wisnieff, “Breaking the 49-qubit barrier in the simulation of quantum circuits,” arXiv: 1710.05867, 2017.
- [21] Z. Chen, Q. Zhou, C. Xue, X. Yang, G.-C. Guo and G.-P. Guo, “64-qubit quantum circuit simulation,” arXiv: 1802.06952, 2018.
- [22] A. Schönhage, “On the power of random access machines,” In *International Colloquium on Automata, Languages, and Programming*, pp. 520–529, 1979.
- [23] S. Lloyd, “Obituary: Rolf Landauer (1927–99),” *Nature*, **400**, p. 720, 1999.
- [24] K. Fujii, *Quantum Computation with Topological Codes: From Qubit to Topological Fault-tolerance*, Springer, 2015.