研究ノート

量子コンピュータによる素因数分解

佐藤昌平

Prime Factoring by Quantum Computer

Shohei SATO

ABSTRACT

Quantum computers are expected to show tremendous computing power based on quantum mechanics parallelism. This paper investigates and explains how this tremendous computing power is realized in the view of a mathematical model of quantum computation. Though the physical realization of quantum computers is another important subject, it is in its infant stage and is not dealt with in this paper.

KEYWORDS : quantum computer, prime factoring

1．はじめに

量子コンピュータが現行のコンピュータに比べて優れているであろうと想定されているのは，その量子力学原理に基づく並列計算能力である。本研究ノートはその並列計算能力がどの様に実現されるかを量子コンピュータの数学的モデルの観点から調査し解説する。本研究ノートの多くの部分は文献［1］に負っている。

尚，量子コンピュータの物理的な実現手段はまだ揺籃期にあり［4］，重要な課題であるが，今回の研究ノートの調査対象としていない。

2．素因数分解アルゴリズム

量子コンピュータが能力を発揮するであろう具体的アルゴリズムが示されている問題は現在のところそれ程多くはない。その中で以前新聞等でも報じられ，よく知られているのは，大きな整数の素因数分解問題であろう。これが話題となったのはRSA公開鍵暗号方式の安全性が，大きな整数を素因数分解することの困難性に依拠しており，量子コンピュータが実現できれば実用的な時間内に素因数分解できる（即ち第三者が暗号を解読できる）可能性が示されたからである。

（2002年8月にManindra Agrawal, Neeraj Kayal, Nitin SaxenaによってAKSアルゴリズムが発表されるまでは，素因数分解はP問題（多項式時間で解ける）ではないのではないかと推測されていた。AKSアルゴリズムにより素因数分解はP問題であることが示されたが，その時点のアルゴリズムは実用的な意味で効率的とは考えられていない［3］。）

量子コンピュータによる合成数Nの素因数分解はおおよそ次の3段階からなるアルゴリズムを利用する。

受理日：平成17年10月11日

①N より小さく，N と互いに素な数 x をランダムに選ぶ。
②x modulo N のオーダー r を見つける。
（x modulo N のオーダー r とは　$x^r = 1 \bmod N$ となる最小の r のこと。）
③オーダー r を使って，N の因数を探す。この r に基づく因数を発見できなければ，①へ戻る。
①～③を繰り返すことにより，Nの素因数分解できる確率を好きなだけ1に近づけることができる。このうち①と③は従来から効率的（多項式時間で）に計算できる事が知られており，本研究ノートでは言及しない。問題は②である。P. W. Shor は1994年に量子コンピュータモデルを使って②を効率的に実行するアルゴリズムを発表し，大きな反響をひきおこした。

3．量子情報処理の数学的モデル

量子情報処理の物理は量子力学原理に基づくものであるから，その数学的モデルも量子力学原理に基づく。本節では物理的意味を示すことなく結果としての数学的モデルの概要を示す。

3．1　量子ビット（Qビット）

量子情報処理では情報の基本単位を量子ビット（Qビット）と呼ぶ。1Qビットの状態は数学的には次のように2次元の複素列ベクトル $|\varphi\rangle$ で表現される。

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$$

ここに α，β は $|\alpha|+|\beta|=1$ を満たす複素数。$|0\rangle$,$|1\rangle$ は基底ベクトルで，その列ベクトル表示は

$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$，$|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$ である。$|\varphi\rangle$ を列ベクトル表示すれば $|\varphi\rangle = \begin{pmatrix}\alpha\\\beta\end{pmatrix}$ となる。

人がQビットの値を知る為にはその状態を“観測”しなければならない。観測の結果は“0”か“1”であって，α，β を知る事はできない。観測の結果0，1を得る確率は各々 $|\alpha|$,$|\beta|$ である。

3．2　複合量子ビット

n 個のQビットからなるシステムの状態 $|\varphi\rangle$ は，個々のQビットの状態を $|\varphi_i\rangle$，$(i=1\sim n)$ とするとそれらのテンソル積で表される。

$$|\varphi\rangle = |\varphi_1\rangle \otimes |\varphi_2\rangle \otimes \cdots \otimes |\varphi_n\rangle \qquad (\ = |\varphi_1\rangle|\varphi_2\rangle\cdots|\varphi_n\rangle = |\varphi_1\varphi_2\cdots\varphi_n\rangle \ \text{等とも書く})$$

$n=2$ の場合のテンソル積をベクトル表示すると

$$|\varphi\rangle = \begin{pmatrix}\alpha_1\alpha_2\\ \alpha_1\beta_2\\ \beta_1\alpha_2\\ \beta_1\beta_2\end{pmatrix} \qquad \text{ここに } |\varphi_1\rangle = \begin{pmatrix}\alpha_1\\\beta_1\end{pmatrix},\ |\varphi_2\rangle = \begin{pmatrix}\alpha_2\\\beta_2\end{pmatrix} \text{ である。}$$

3．3　単一量子ビットの状態変化

閉じた量子システム（Qビット）の状態の変化はあるユニタリ行列 U によって記述される。
例えば単一Qビットの状態 $|0\rangle$，$|1\rangle$，$|\varphi\rangle$ に次の X なるユニタリ行列を作用させると

$$X|0\rangle = \begin{pmatrix}0&1\\1&0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix} = |1\rangle,\ X|1\rangle = \begin{pmatrix}0&1\\1&0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix} = |0\rangle,\ X|\varphi\rangle = \begin{pmatrix}0&1\\1&0\end{pmatrix}\begin{pmatrix}\alpha\\\beta\end{pmatrix} = \begin{pmatrix}\beta\\\alpha\end{pmatrix}$$

となる。上より X は通常の論理代数における否定（NOT）に相当するものである事がわかる。もうひ

とつ H で表されるユニタリ行列の例（アダマール行列）を示しておく。

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

3．4　複合量子ビットの状態変化

ユリタリ行列 A，B が各々Qビット $|a\rangle$，$|b\rangle$ に作用する時，次が成り立つ。

$$(A\otimes B)(|a\rangle\otimes|b\rangle)=A|a\rangle\otimes B|b\rangle$$

4．オーダー r を探す量子アルゴリズム

2．節②で述べた x modulo N のオーダー r を探す量子アルゴリズムを以下に示し解説する。ここで N は素因数分解したい整数，x は N より小なる N と互いに素なる整数である。N は L ビット長とする。$t=2L+m$ ビットの第一の量子レジスタと L ビット長の第二の量子レジスタを用意する。ここで m はオーダー r を探し当てられる確率に関する考察から決まる値で，凡そ（10進）1桁の数と考えてよい。本研究ノートでは $m=1(t=2L+1)$ としておく。

ステップ1：初期状態の設定

◇第一のレジスタの状態を $|0\rangle$ に，第二のレジスタの状態を $|1\rangle$ に初期設定する。その複合状態は

$$|0\rangle|1\rangle$$

である。

［解説］

第一のレジスタは t ビット長であるから，その基底ベクトルは 2^t 個だけある。その基底ベクトルを $|0\rangle,|1\rangle,\cdots|2^t-1\rangle$ で表す。個々の $|j\rangle$ は t 個のQビット t のテンソル積である。単一Qビットの状態も，複合Qビットの状態も $|0\rangle,|j\rangle$ で表す事があり紛らわしいが記号の簡潔さを重視してそのような記法も使う。第二のレジスタについても同様である。

ステップ2：第一レジスタ状態に基づく重ね合わせの生成

◇第一のレジスタ内容に基づき基底ベクトルの重ね合わせを生成する。生成後のシステムの状態は次の通りである。

$$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|1\rangle$$

［解説］

この重ね合わせは第一のレジスタの各Qビットに H 行列を作用させる事により得られる。何故なら，第一レジスタの状態は $|0\rangle$ であり，この時各単一Qビットの状態は $|0\rangle$ でありこれに H 行列を作用させるとその結果は前述したように

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

となるから，t ビット全体の状態は次のようになるからである。

$$H|0\rangle\otimes H|0\rangle\cdots H|0\rangle=\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)^{t}=\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle$$

この様に H 行列を t 回作用させる事により，2^t 個の状態の重ね合わせが実現できることに注意。量子並列計算を理解するポイントの一つである。

ステップ３－１：第二のレジスタ状態を用い $x^j \bmod N$ を作りこむ。
◇先ず j の２進数表示を $j_t\, j_{t-1}\cdots j_1$ とする。つまり，$j=j_t 2^{t-1}+j_{t-1}2^{t-2}+\cdots+j_1 2^0$。次に

$$x^j \bmod N=(x^{j_t 2^{t-1}} \bmod N)(x^{j_{t-1}2^{t-2}} \bmod N)\cdots(x^{j_1 2^0} \bmod N) \qquad \cdots(3.1)$$

なる事に注意する。また上式右辺の各項は $(x \bmod N)$ を２乗，更にそれを２乗，・・・して得られることに注意する。つまり $(x \bmod N)$ を計算する回路があれば，上式は効率的に計算できる。第二レジスタの状態 $|1\rangle$ に上述の計算を作用させた後のシステムの状態は次のようになる。

$$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|x^j \bmod N\rangle \qquad \cdots(3.2)$$

L ビットの第二レジスタに上記の状態を設定すると量子力学原理に基づいて全体の状態として 2^t 個の状態の重ね合わせが自動的に生ずる。これも量子並列計算を理解するポイントの一つである。

ステップ３－２：ステップ３－１で得られた結果の式を変形する。

◇ $$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|x^j \bmod N\rangle=\frac{1}{\sqrt{r2^t}}\sum_{s=0}^{r-1}\sum_{j=0}^{2^t-1}\exp(2\pi isj/r)|j\rangle|u_s\rangle, \qquad \cdots(3.3)$$

$$\text{ここに } u_s=\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp(-2\pi isk/r)|x^k \bmod N\rangle \text{ である。} \qquad \cdots(3.4)$$

［解説］
式(3.4)を式(3.3)の右辺に代入すれば

$$\frac{1}{r\sqrt{2^t}}\sum_{s=0}^{r-1}\sum_{j=0}^{2^t-1}\exp(2\pi isj/r)|j\rangle\sum_{k=0}^{r-1}\exp(-2\pi isk/r)|x^k \bmod N\rangle$$

$$=\frac{1}{r\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle\sum_{s=0}^{r-1}\sum_{k=0}^{r-1}\exp(2\pi is(j-k)/r)|x^k \bmod N\rangle \qquad \cdots(3.5)$$

ここで，$j=p_j r+q_j\ (0\le q_j<r)$ とすると

$$\sum_{s=0}^{r-1}\exp(2\pi is(j-k/r)=r \qquad \text{for} \qquad q_j=k$$

$$=0 \qquad \text{for} \qquad q_j\neq k$$

となり，$q_j=k$ に対しては

$$x^k \bmod N = x^{q_j} \bmod N = (x^{q_j} \bmod N)(x^{p_j r} \bmod N) = x^j \bmod N$$

となることを注意すると式(3.5)から，式(3.3)の左辺が得られる。

ステップ4：逆量子離散フーリエ変換

◇式(3.3)を逆量子離散フーリエ変換すると次を得る。

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|2^t s/r\rangle|u_s\rangle$$

[解説]

式(3.3)の右辺に含まれる

$$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\exp(2\pi isj/r\,|j\rangle=\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\exp(2\pi ij(s/r))|j\rangle \qquad \cdots(4.1)$$

は $|s/r\rangle$ の量子離散フーリエ変換とよばれるものであり，これを逆変換すると次を得る。

$$\frac{1}{\sqrt{2^t}}\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\sum_{l=0}^{2^t-1}\exp(-2\pi ijl/2^t)\exp(2\pi ij(s/r))|l\rangle=\frac{1}{2^t}\sum_{j=0}^{2^t-1}\sum_{l=0}^{2^t-1}\exp(2\pi ij(s/r-l/2^t))|l\rangle \qquad \cdots(4.2)$$

ここで，理想的な条件（ある s/r に対し $s/r-l/2^t=0$ を満足する l が存在する）下では

$$\sum_{j=0}^{2^t-1}\exp(2\pi ij(s/r-l/2^t))=2^t \qquad \text{for} \qquad s/r-l/2^t=0$$

$$=\frac{1-\exp(2\pi i(s/r-l/2^t)2^t)}{1-\exp(2\pi i(s/r-l/2^t))}=0 \qquad \text{for} \qquad s/r-l/2^t\neq 0$$

となる事に注意する。

すると式(4.2)は

$$(4.2)=|2^t s/r\rangle$$

となる。従って式(3.3)逆量子フーリエ変換すると

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|2^t s/r\rangle|u_s\rangle \qquad \cdots(4.3)$$

第一レジスタ容量は t ビットに制限されているので一般的には上記の理想的条件は満たされずある誤差が生ずる。そのため式(4.3)s/r は誤りの可能性を含む推定値となるので，詳細な議論は省くが式(4.3)を

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\left|2^t\frac{\tilde{s}}{r}\right\rangle|u_s\rangle \qquad \cdots(4.3)'$$

と書き直す。

ステップ５：第一レジスタを“観測”する。

◇式(4.3)'の第１レジスタ部分はr個の状態を重ね合わせになっているが，これを観測するとそのうちの一つの状態がランダムに読み出される。読み出された値を$\tilde{k}$とする。観測した事により，状態の重ね合わせは瞬時に解消（収縮）し，観測された状態に確定される。

［解説］

$\tilde{k}$はtビットに制限されているから一般的には$2^t s/r$の近似値である。ここで，$\tilde{k}$は得られるが，s,rの各々の値はまだ得られていないことに注意。また$r\le N\le 2^L \Rightarrow 2r^2\le 2^{L+1}$から，次が成り立つことに注意する。

$$\left|\frac{s}{r}-\frac{\tilde{k}}{2^t}\right|\le\frac{1}{2^t}=\frac{1}{2^{2L+1}}\le\frac{1}{2N^2}\le\frac{1}{2r^2} \qquad \cdots(5.1)$$

ステップ６：$\tilde{k}$からオーダーrを求める。

◇　$\tilde{k}/2^t$を連分数展開し，その近似の中で式(5.1)を満足するs/r（即ちsとr）を求める。そのrが，$x^r=1(\mathrm{mod}\,N)$を満足すればこのrが求めるオーダーrである。

［解説］

一般的にzをある有理数，p/qも有理数とし，次が成り立つ場合

$$\left|\frac{p}{q}-z\right|\le\frac{1}{2q^2} \qquad \cdots(6.1)$$

p/qはzの近似連分数展開である。これを式(5.1)に適用すると$\tilde{k}/2^t$の連分数展開としてあるs/rが求まる。$0\le s,r\le N$に注意すると式(5.1)を満たすs,rの組はひとつである事が次にようにして分る。

式(5.1)を満たす２組のs,rがあるとし，その値を各々s,rとs',r'とすると，

$$\left|\frac{s}{r}-\frac{s'}{r'}\right|\le\left|\frac{s}{r}-\frac{\tilde{k}}{2^t}\right|+\left|\frac{s'}{r'}-\frac{\tilde{k}}{2^t}\right|\le\frac{1}{N^2} \qquad \cdots(6.2)$$

一方上式の最左辺は$sr'-s'r\neq 0$ならば

$$\left|\frac{s}{r}-\frac{s'}{r'}\right|=\left|\frac{sr'-s'r}{rr'}\right|\ge\frac{|sr'-s'r|}{N^2}\ge\frac{1}{N^2} \qquad \cdots(6.3)$$

よって式(6.2)，(6.3)から

$$\left|\frac{sr'-s'r}{rr'}\right|=\frac{1}{N^2}\Rightarrow r=r'=N\Rightarrow|sr'-s'r|=N|s-s'|=1$$

これは矛盾であるから，$s/r=s'/r'$である。

観測のときにランダムに選ばれた s,r が共役数 d をもっている場合，即ち

$$s = s'd, \qquad r = r'd$$

の時，結果として得られるのは r' であり，r そのものではない。これをチェックするため，得られた r が $x^r \equiv 1 \bmod N$ を満足するか否かをチェックするのである。

以上ステップ1～6により，ある確率でオーダー r が求められる。r を求められなかった場合にはステップを繰り返すことにより，成功確率を1に近づける事ができる。素因数分解アルゴリズム，並びに成功確率の解析の詳細については文献［1］の5章に詳しいので参照されたい。

5．あとがき

量子コンピュータの並列計算は量子力学的状態の重ねあわせと，テンソル積により表現される複合状態が自動的に生成されることを巧妙に利用している。また，計算の結果を読み出すときには，重ね合わせた状態のどれが読み出されるか事前には確率的にしかわからない。これを考慮して望む結果を最終的に得るには精妙なアルゴリズムが必要である。

本ノートで紹介した素因数分解アルゴリズムから受ける印象は，量子コンピュータ用のアルゴリズム（プログラム）が有効に働く分野（問題）は案外狭いのではないか（・・・かってのアナログコンピュータのように)。また結果読み出しに確率的要素が不可避的（?）にはいってしまうことを考えると，現行のソフトウエアシステムで大問題になっている，“検証/デバッグ”がより困難になり，大規模・複雑なプログラムは作成できないような印象を受ける。勿論量子コンピュータ開発は，揺籃期にあるわけであり，まだ可能性を断定的にいうのは早いわけだけれど。

参考文献

［1］M.A.Nielsen and I.L.Chuang, “Quantum Computation and Quantum Information,” Cambridge University Press, 2000
［2］西野哲朗“量子コンピュータ入門”，東京電機大学，1997
［3］内山成憲“素数判定アルゴリズム”，電子情報通信学会誌 Vol. 86，No. 9，pp.703～708，2003年9月
［4］川畑史郎“量子コンピュータ入門”，電子情報通信学会誌 Vol. 88，No. 9，pp.760～762，2005年9月

（佐藤　昌平：四国大学　情報科学研究室）