量子アルゴリズムを用いた多項式GCDの計算

武田 邦敬 (Kunihiro TAKEDA)*　　甲斐 博 (Hiroshi KAI)†
愛媛大学理工学研究科　　愛媛大学工学部

野田 松太郎 (Matu-Tarow NODA)‡
愛媛大学工学部

1　はじめに

Shor は 1994 年、量子コンピュータ [3] を用いて素因数分解と離散対数問題を多項式時間で解くアルゴリズムを発見した [5]。これは、量子コンピュータの実現によって現在広く使われている RSA などの公開鍵暗号系の安全性が崩れることを意味し、大きな注目を集めた。そのため、Shor の発見以降、量子計算及び量子コンピュータの実現に向けた研究が盛んに行われるようになった。しかしながら、量子計算に対する研究は困難を極め、効果的な量子アルゴリズムの発見はまだ少ない。このような状況の中、通常は長い計算時間を必要とする数式処理のアルゴリズムに対する量子計算の可能性を探ることは、非常に興味深い。そこで、本研究では発見的 (ヒューリスティック) な多項式 GCD アルゴリズムである GCDHEU アルゴリズム [2] に対し、Grover による量子探索アルゴリズム [4] を応用することを考える。

2　量子計算の原理

量子計算は、重ね合わせの原理、ユニタリ変換、観測による状態の収縮といった量子力学の公理によって特徴付けされる。量子計算の概要を以下に簡単にまとめる。

2.1　重ね合わせの原理

古典コンピュータにおけるデータ表現の基本単位はビット (以下、古典ビット) である。これに対し、量子コンピュータにおけるデータ表現の基本単位を量子ビットと呼ぶ。両者の大きな違いは、古典ビットが 0 と 1 の離散的な状態しか取り得なかったのに対し、量子ビットはその重ね合わせ状態を取るという点である。量子ビットの取る状態を $|0\rangle,\ |1\rangle$ と表すと、その重ね合わせ状態は $|\alpha|^2 + |\beta|^2 = 1$ なる複素数 α, β を用いて

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

と書くことができる。ここで、$|\psi\rangle$ を状態ベクトル、α, β を (確率) 振幅という。状態ベクトル $|\psi\rangle$ は、正規直交基底 $|0\rangle,\ |1\rangle$ によって張られる複素 Hilbelt 空間上の単位ベクトルとして記述され、α, β は状態ベクト

*kunihiro@hpc.cs.ehime-u.ac.jp
†kai@cs.ehime-u.ac.jp
‡noda@cs.ehime-u.ac.jp

ルを観測したときに固有状態を得られる確率を与えるものである。即ち、式 (1) によって与えられる状態ベクトル $|\psi\rangle$ を観測すれば、確率 $|\alpha|^2$ で $|0\rangle$、$|\beta|^2$ で $|1\rangle$ という固有状態を得る。

量子ビットを n 個連結して量子レジスタを作ると、その状態ベクトルはテンソル積で記述される。$N = 2^n$ とおくと n 量子ビットは次のようになる。

$$|\psi_n\rangle = \bigotimes_{i=0}^{n-1} |\psi\rangle = \sum_{x\in\{0,1\}^n} \omega_x |x\rangle \tag{2}$$

ただし、$\sum_{x\in\{0,1\}^n} |\omega_x|^2 = 1$ である。n 古典ビットでは $\{0,1\}^n$ の状態の一つしか取ることができなかったが、n 量子ビットでは $N = 2^n$ 個の全ての重ね合わせ状態を取ることができ、これが量子計算の超並列性を可能にする。

2.2 ユニタリ変換

量子系の状態ベクトルの変化は、ユニタリ変換で表される。ユニタリ変換は Hilbelt 空間上におけるノルムを変えない可逆な線形変換であり、$U^\dagger U = UU^\dagger = I$ を満たす行列 U で記述される。ここで、$U^\dagger$ は U の転置共役行列である。以下に示す Hadamard 変換 H は、1 量子ビットに対する重要なユニタリ変換である。

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big), \quad |1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \tag{3}$$

固有状態 $|\mathbf{0}\rangle \equiv |00\cdots0\rangle$ にある n 量子ビットレジスタの各ビットに Hadamard 変換 H を施すと、等しい振幅を持つ $N = 2^n$ 個の重ね合わせ状態を作り出すことができる。

$$|\mathbf{0}\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n} |x\rangle \tag{4}$$

また、ある関数 $f(x) : \{0,1\}^n \to \{0,1\}^m$ に対し、次のようなユニタリ変換 U_f を考えることができる。

$$|a\rangle|b\rangle \xrightarrow{U_f} |a\rangle|b \oplus f(a)\rangle \tag{5}$$

ただし、$a \in \{0,1\}^n$, $b \in \{0,1\}^m$, $\oplus$ は $\{0,1\}^m$ 上の加算である。特に $b = 0$ とすると $|a\rangle|\mathbf{0}\rangle \xrightarrow{U_f} |a\rangle|f(a)\rangle$ となり、$f(a)$ の値を計算するする変換になる。さらに、重ね合わせ状態にある入力に対して U_f を施せば

$$|\mathbf{0}\rangle|\mathbf{0}\rangle \xrightarrow{H^{\otimes n}} \sum_{a\in\{0,1\}^n} |a\rangle|\mathbf{0}\rangle \xrightarrow{U_f} \sum_{a\in\{0,1\}^n} |a\rangle|f(a)\rangle \tag{6}$$

となり、全ての入力 $a \in \{0,1\}^n$ に対する関数 $f(a)$ の評価を一度に行うことができる。これが量子計算における超並列計算と呼ばれるものである。しかし、今式 (6) にある量子レジスタを観測して値を読み出そうとすると、状態ベクトルは固有状態に収束し、得られる値はただ一つとなることに注意しなければならない。

3 Grover のアルゴリズム

Grover の量子探索アルゴリズムは、N 個のデータからなる未整理のデータベースから、求める $M(1 \le M \le N)$ 個のデータの一つを探し出すアルゴリズムである。Grover のアルゴリズムではこの問題を次のように置き換える。$N = 2^n$ とし、N 個のデータを $x_0, x_1, \cdots, x_{N-1} \in \{0,1\}^n$ とラベル付けし、次のようなブラックボックス関数 $f(x) : \{0,1\}^n \to \{0,1\}$ を考える。

$$f(x) = \begin{cases} 1 & x\text{が解である} \\ 0 & x\text{が解でない} \end{cases}$$

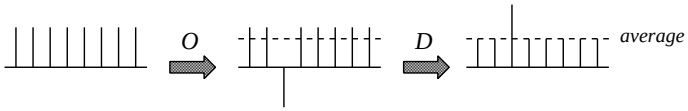

図 1: Grover 反復 $G = DO$ による振幅の変化

データベース検索の問題は $f(x) = 1$ を満たす $x \in \{0,1\}^n$ を見つけることに帰着する。古典コンピュータでは、解を見つけるために $f(x)$ を平均 $O(N/2M)$ 回参照する必要がある。一方、Grover のアルゴリズムでは、$O\left(\sqrt{N/M}\right)$ 回の操作でこれを完了する [4]。

Grover のアルゴリズムに用いられるユニタリ変換は以下の 2 つである。

1. 量子オラクル O $\quad |a\rangle \xrightarrow{O} (-1)^{f(a)}|a\rangle$
2. 拡散変換 $D = -H^{\otimes n} I_0 H^{\otimes n}$ ただし、$|\mathbf{0}\rangle \xrightarrow{I_0} -|\mathbf{0}\rangle$

量子オラクル O は式 (6) において $b = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$ とすることで可能となり、$f(x) = 1$ のときに振幅を反転させる変換になる。一方、拡散変換 D は振幅の平均値で折り返すような変換になる (図 1)。$G = DO$ を施すことを Grover のアルゴリズムにおける 1 ステップとする。これを Grover 反復という。

Grover のアルゴリズムを以下に示す。

1. 基底状態 $|\mathbf{0}\rangle$ に Hadamard 変換を施し、重ね合わせ状態を作る。
$$|\mathbf{0}\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle$$
2. 以下のユニタリ変換 $G = DO$(Grover 反復) を $R \approx O(\sqrt{N/M})$ 回繰り返す。
   (a) 量子オラクル O を施す。$f(x) = 1$ を満たす $|x\rangle$ の位相が π だけ回転する。
   (b) 拡散変換 $D = -HI_0H$ を施す。
$$\longrightarrow G^{\otimes R} \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle \approx |x_i\rangle, \quad f(x_i) = 1$$
3. 量子レジスタ観測して解を得る。

Grover のアルゴリズムは Grover 反復の適用回数によって成功確率が変化することが知られている。したがって、予め解の個数を知っておく必要があるが、Bøyer らは解の個数が不明の場合にも適用できるアルゴリズムを提案している [1]。

4 GCDHEU アルゴリズム

4.1 アルゴリズムの概観

Char,Geddes,Gonnet によって提案された多項式 GCD アルゴリズムである GCDHEU は、一つの大きな整数を評価点として $\mathbf{Z}$ 上で GCD を行い、1 点による補間から GCD を求めるヒューリスティックなアルゴリズムである [2]。このアルゴリズムの概略は以下の通りである。ただし、入力多項式 a, b は $\mathbf{Z}$ に関して原始的とする。

1. $vars \leftarrow a, b$ に含まれる変数;
2. **if** $vars = [\,]$ **then** return IGCD(a, b) **fi**;

3. $x \leftarrow vars[0]$;
4. $\xi \leftarrow 2 \times \min(|a|, |b|) + 2$;
5. $\gamma \leftarrow \mathrm{GCDHEU}(\phi_{x-\xi}(a),\ \phi_{x-\xi}(b))$;
6. $g \leftarrow \mathrm{genpoly}(\gamma, \xi, x)$;
7. **if** $\mathrm{pp}(g)|a$ and $\mathrm{pp}(g)|b$ **then** return $\mathrm{pp}(g)$
   else $\xi \leftarrow \mathrm{iquo}(\xi \times 73794, 27011)$ として 4 に戻る **fi**;

一つの評価点からの多項式の補間は、以下のような簡単なループで得ることができる。

```
procedure genpoly(γ, ξ, x)
  poly ← 0;
  e ← γ;
  for i from 0 while e ≠ 0 do
    g_i ← φ_ξ(e);
    poly ← poly + g_i × x^i;
    e ← (e − g_i)/ξ
  od;
  return poly
end;
```

GCDHEU の特徴は以下にまとめられる。

1. 評価点の取り方によって失敗するときがある。
2. 特に変数の数が少ない場合に高速である。
3. 再帰的な構造のため、多変数の場合に評価点 ξ の桁数が極端に増大する可能性がある。

4.2 GCDHEU に関する考察

GCDHEU は、動作は確率的であるが解の判定ができるという特色がある。この点に注目し、複数の評価点 ξ に対する GCDHEU を量子コンピュータ上で並列計算することを考える。問題となるのは、補間に失敗して評価点 ξ を取り替える操作がどの程度行われてるかである。そこで、次のような計算を行った。

> ランダムに選んだ多項式 F, G に対して GCDHEU を行い、評価点 ξ の取り替え回数を計測 (1000 回)。F, G は項数 10、次数 15 以下の原始的な多項式 $\tilde{F}, \tilde{G}, D$ に対し、$F = \tilde{F} \times D, G = \tilde{G} \times D$ として構成。

表 1: 評価点の取り替え回数の計測

ξ の取り替え回数	1 回	2 回	3 回	4 回
1 変数多項式	943	51	6	0
2 変数多項式	468	362	169	1
3 変数多項式	101	413	476	10

この結果を表 1 に示す。この計算の結果、評価点の取り替えは予想以上に少ないことがわかった。しかし、特に多変数多項式の場合、数回の評価点の取り替えでも評価点が極端に大きくなる問題がある。GCDHEU

表 2: $\xi^{(1)}, \xi^{(2)}$ 間における成功する評価点が含まれる割合

テスト回数	1000 回	10000 回
成功す評価点が含まれる割合	96.468%	96.436%

では、取り替え回数を抑えつつ桁数の増大を回避する狙いから、次のような黄金比を使った評価点の再設定を行っている。

$$\xi^{(1)} \leftarrow 2 \times \min(|a|, |b|) + 2; \quad \xi^{(k+1)} \leftarrow \mathrm{iquo}(\xi^{(k)} \times 73794, 27011); \quad (k \geq 1) \tag{7}$$

そこで、評価点の初期値と再設定された値の間にどの程度成功する評価点が含まれているかを見るため、次のような計算を行った。

> 前述の計算と同様にして選んだ 1 変数多項式に対して評価点 $\xi^{(1)}$ と $\xi^{(2)}$ の間の全ての整数について GCDHEU を行い、成功する評価点が含まれる割合を計測。

この結果を表 2 に示す。この結果から、$\xi^{(1)}$ と $\xi^{(2)}$ の間にも成功する評価点が多く存在することがわかる。したがって、評価点の取り替えを 1 刻みに増やしていくことで評価点の増大を回避できると考えられる。次に示す多項式 F, G について、GCDHEU を終了したときの評価点の桁数の比較を表 3 に示す。

$$\begin{aligned}
F &= 2zx^{21} + (16z^2 + 9)x^{20} + 13zy^2x19 + (4y^3 + 16z^2y)x^{18} + 5z^2y^2z16 + 5z^4y^3x^{14} + 7z^4x^{13} \\
&\quad + (6zy^9 + 8z^3y^7)x^{12} + (9zy^9 + 8z^2y^7 + 5zy^2)x^{11} + (9z^2y^8 + 2z^3y)x^9 + 16y^{11}x^8 \\
G &= (31y + 10)x^{21} + 12zy^2x^{19} + 15yx^{18} + 8z^3x^{17} + 9z^4yx^{14} + 17z^3y^3x^{13} + (5yz^7 + 3z^6y^4)x^{12} \\
&\quad + (9y^8 + 16z^6y^5)x^{10} + 6z^{11}x^9 + (11y^{13} + 16y^{11} + 11z^2y^8 + 7z^7y^5)x^8 \\
&\mathrm{GCD}(F, G) = x^8
\end{aligned}$$

表 3: GCDHEU を終了したときの評価点の桁数の比較

変数	x	y	z
式 (7) のように再設定した場合	2	51	528
1 刻みに再設定した場合	3	34	378

5 GCDHEU に対する量子アルゴリズム

以上の計算結果を踏まえ、1 刻みに選んだ複数の評価点 ξ に対する GCDHEU を量子コンピュータ上で並列計算を行う量子アルゴリズムを提案する。GCDHEU を並列処理する量子アルゴリズムを以下に示す。(ただし、規格化振幅に関する記述は省略)

1. 3 つの量子レジスタを用意する。$|index\ reg\rangle|\xi\ reg\rangle|answer\ reg\rangle$
2. Hadamard 変換によって重ね合わせ状態を作る。
$$|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle \longrightarrow \sum\nolimits_x |x\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle$$
3. 評価点 ξ の重ね合わせを生成する。
$$\longrightarrow \sum\nolimits_x |x\rangle|\xi_x\rangle|\mathbf{0}\rangle$$

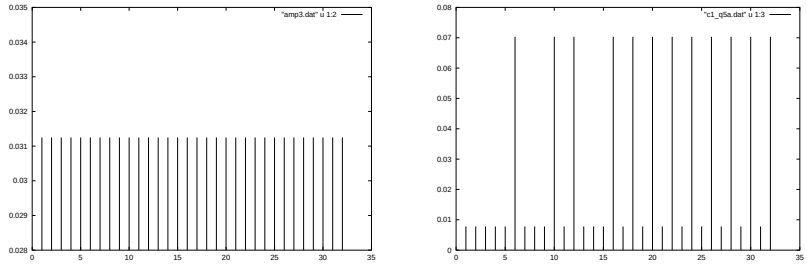

図 2: *answer* レジスタの振幅の変化。初期状態 (左) と、Grover のアルゴリズム適用後 (右)。

4. GCDHEU を計算して *answer* レジスタに格納する。
$$\longrightarrow \sum_x |x\rangle|\xi_x\rangle|\mathrm{GCDHEU}(a,b,\xi_x)\rangle$$

5. *answer* レジスタに対して Grover のアルゴリズムを適用する。ただし、解の個数が不明のため Bøyer らによるアルゴリズムを用いる。

6. *answer* レジスタを観測し、解を得る。

数式処理システム Risa/Asir 上で量子計算を行うシミュレータを作成した。以下の多項式 F, G に対し、5 量子ビット (32 並列) の量子計算シミュレータ上で評価点 $\xi = 108, 109, \cdots, 139$ において並列計算した場合の振幅の変化を図 2 に示す。等しい振幅の重ね合わせ状態にある *answer* レジスタに Grover のアルゴリズムを適用することによって正しい解の振幅が高くなり、高確率で正しい解を観測する。

$$\begin{aligned}
F &= 8x^{29} + 22x^{28} + 36x^{26} + 171x^{25} + 206x^{24} + 144x^{23} + 280x^{22} + 197x^{21} + 309x^{20} + 499x^{19} \\
&\quad + 320x^{18} + 763x^{17} + 917x^{16} + 312x^{15} + 800x^{14} + 1143x^{13} + 516x^{12} + 707x^{11} + 997x^{10} \\
&\quad + 754x^9 + 483x^8 + 322x^7 + 776x^6 + 144x^5 + 320x^4 + 144x^3 + 320x^2 \\
G &= 8x^{28} + 22x^{27} + 4x^{23} + 53x^{22} + 8x^{20} + 42x^{19} + 21x^{17} + 18x^{16} + 44x^{15} + 21x^{14} + 9x^{11} + 20x^{10} \\
\mathrm{GCD}(F,G) &= 4x^{15} + 11x^{14} + 21x^9 + 4x^7 + 21x^6 + 9x^3 + 20x^2
\end{aligned}$$

6 まとめ及び今後の課題

本研究では、Grover の量子探索アルゴリズムを使った GCDHEU に対する量子アルゴリズムを提案した。評価点の増大や補間失敗によるリスクを回避する目的で 1 刻みの評価点に対して並列計算を行った。しかし、提案したアルゴリズムは確率的なアルゴリズムに対する量子計算のアプローチに留まり、GCDHEU が抱える変数の数と次数に比例して評価点が極端に増大し急激な速度低下を起こすという問題を根本的に解決するものではない。今後、大きな整数に対する効率的な演算法など、数式処理を飛躍させるような量子アルゴリズムの発見が急務である。

参 考 文 献

[1] M.Bøyer, G.Brassard, P.Høyer and A.Tapp : Tight bounds on quantum searching, *Proc. PhysComp*, 1996 `http://xxx.lanl.gov/abs/quant-ph/9605034`

[2] B.W.Char, K.O.Geddes and G.H.Gonnet : GCDHEU: Heuristic Polynomial GCD Algorithm Based On Integer GCD Computation, *Journal of Symbolic Computation*, vol.7, pp.31-48, 1989

[3] D.Deutsch : Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer, *Proc. Royal Society London*, Vol.A400, pp.97-117, 1985

[4] L.K.Grover : A fast quantum mechanical for database search, *Proc. Annual ACM Symposium on Theory of Computing*, pp.212-219, 1996

[5] P.W.Shor : Algorithms for Quantum Computation: Discrete Logarithms and Factoring, *Proc. 35th Annual Symposium on Foundations of Computer Science*, pp.124-134, 1994