

## 量子和回路の効率化とシミュレーションによるデコヒーレンス耐性の解析

長谷川 淳<sup>†</sup> 丹羽 純平<sup>†</sup>  
由良 文孝<sup>††</sup> 今井 浩<sup>†,††</sup>

科学や工学の分野では積分計算が重要である。量子和アルゴリズムは、積分計算を量子コンピュータ上で高速に行うアルゴリズムである。量子和アルゴリズムを用いると、積分計算が、古典の最も速い決定的なアルゴリズムよりも指数的に速くなり、古典の最も速い確率的なアルゴリズムよりも平方根的に速くなる。そのため、最近このアルゴリズムに関して盛んに研究がなされている。しかし、現在の量子和に関する研究はアルゴリズムの計算量の解析が中心であり、量子計算シミュレータを用いてアルゴリズムの実際の振る舞いは調べることは行われていない。

本研究は、量子和回路のデコヒーレンスエラーへの耐性を量子シミュレータを用いて評価した。さらに、量子和回路を改良しエラーに強い回路の構成を行い、従来の量子和回路と比較しエラーに対する特性の違いも示し、提案した量子回路の有用性を評価した。

### Improvements of Quantum Summation Circuits and Analyses of Robustness for Decoherence Error by Simulation

JUN HASEGAWA,<sup>†</sup> JUMPEI NIWA,<sup>†</sup> FUMITAKA YURA<sup>††</sup>  
and HIROSHI IMAI<sup>†,††</sup>

It is important to calculate numerical integrals in science and engineering. There is a quantum summation algorithm to calculate these fast on a quantum computer. This algorithm is exponentially faster than the best known classical deterministic algorithms and quadratically faster than the best known classical probabilistic algorithms. Thus, there have been many studies of the quantum summation algorithm. However, these studies have focused on analyzing the complexity of the algorithm and no one has investigated the actual behaviors of this algorithm by a quantum computational simulator.

In this paper, we estimated the robustness for decoherence errors of this quantum summation circuits. Moreover, we constructed quantum summation circuits robust for decoherence error by improving circuits, showed the difference of the behaviors in the presence of decoherence errors between on the existing circuits and on the improved circuits, and evaluated the usefulness of our improved quantum summation circuits.

#### 1. はじめに

1994年 Peter Shor<sup>10)</sup> は、素因数分解と離散対数問題が量子コンピュータ上で効率的に解くことができることを示した。翌年 Grover<sup>3)4)</sup> は量子コンピュータ上でのもう一つの重要なアルゴリズムであるデータ探索アルゴリズムを示した。このアルゴリズムを用いると、データベースの中から正しい解を見つけるのに、古典コンピュータよりも平方根的に速く求めることが

可能である。さらに、このデータ探索アルゴリズムには多くの応用があり、量子数え上げアルゴリズム<sup>1)</sup>はその一つの例となっている。量子数え上げアルゴリズムはデータベースの中の解の個数を数える平方根的に速いアルゴリズムであり重要なことはこのアルゴリズムが Grover のデータ探索アルゴリズムと Shor の量子フーリエ変換から構成されていることである。

近年、量子コンピュータ上で連続問題を数値的に計算しようとする研究が多くなされている。その一つが量子和アルゴリズムである。このアルゴリズムは近似的に積分を解くアルゴリズムで古典のコンピュータよりも平方根的に速く解くことができる。しかし、量子和アルゴリズムに対する最近の研究は、アルゴリズムの計算量の解析が中心であり、デコヒーレンスエラーへの耐性は調べられていなかった。

この量子和アルゴリズムを実装する方法には 2 種

<sup>†</sup> 東京大学大学院情報理工学系研究科コンピュータ科学専攻  
Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo

<sup>††</sup> 科学技術振興事業団今井量子計算機構プロジェクト  
ERATO Project Quantum Computation and Information, Japan Science and Technology Corporation

類存在する<sup>6)</sup>。量子数え上げアルゴリズムによるものと量子平均アルゴリズム<sup>4)</sup>によるものである。両者のアルゴリズムにはオーダーの観点からは違いがない。しかし、量子数え上げアルゴリズムを用いた方が回路の深さが短くなるという利点があるが必要な量子ビット数が多くなってしまふという欠点がある。我々は、量子ビットを削減する手法を用いて、この量子数え上げアルゴリズムの欠点を解消し、効率的な量子和回路を構成する。

本研究では、量子数え上げアルゴリズムによる量子和回路を量子計算シミュレーションシステム<sup>8)</sup>上で実装し、このアルゴリズムのエラーへの耐性を調べている。そして、Parkerら<sup>9)</sup>による半古典的な位相推定とDraper<sup>2)</sup>による量子フーリエ変換を用いた加算という2つのテクニックを用いることで、必要な量子ビット数の少ない、エラーに強い量子和回路を構成する。さらに、2つのテクニックによって改良した量子和回路と従来の方法によって構成した量子和回路でのエラーのあるときの解の得られる確率分布を比較し、両者の間の特性の違いを示す。最後に、改良した量子和回路の確率分布の特性を利用して正しい解を得やすくする手法を提示し、改良した量子和回路の有用性を評価す。

## 2. 準備

量子数え上げアルゴリズムを用いた量子和回路の構成に必要なアルゴリズムを紹介する。また、量子和回路の改良方法である半古典的な位相推定と量子フーリエ変換を用いた加算の手法も説明する。

### 2.1 Grover のデータ探索アルゴリズム

$N$  個の整列されていないデータの中で“正しい”データが一つあるとする。正しいデータを見つけるためには、古典コンピュータ上では明らかに  $O(N)$  回の操作が必要となる。それに対して、量子コンピュータ上では  $O(\sqrt{N})$  回の操作だけで正しいデータを見つけるアルゴリズムが存在する。これが Grover<sup>3)</sup> のアルゴリズムである。

#### 2.1.1 量子オラクル

$N$  個の要素による探索空間の中に  $t$  個の解があると仮定する。ここで与えられた入力探索問題の解かどうかを判断できる量子オラクルを考える。関数  $b$  を次のように定義する。

$$b(x) = \begin{cases} 1 & x \text{ が解のとき} \\ 0 & x \text{ が解でないとき} \end{cases} \quad (1)$$

ただし、入力  $x$  は 0 から  $N-1$  の範囲の整数である。このとき、量子オラクルは関数  $b$  とオラクル量子ビット  $|q\rangle$  を用いて次のようなユニタリ演算子  $O$  として書ける。

$$|x\rangle|q\rangle \xrightarrow{O} |x\rangle|q \oplus b(x)\rangle. \quad (2)$$

これは、入力  $x$  が解であるときにオラクル量子ビッ

ト  $|q\rangle$  の状態を反転するものとなっている。

#### 2.1.2 手法

Grover のデータ探索アルゴリズムは、欲しい解の振幅を増幅していくことで正しい解を得る。振幅を増幅するために、グローバール演算子  $G$  を反復適用する。グローバール演算子  $G$  は、量子オラクル  $O$  と、 $2|\psi\rangle\langle\psi| - I$  で構成されている。ただし、 $|\psi\rangle = H^{\otimes n}|0\rangle^{\otimes n}$  である。このときグローバール演算子  $G$  を  $R \approx [\pi\sqrt{2^n}/4]$  回適用し観測すると、正しい解を高い確率で得ることができる。

このグローバール演算子  $G$  はグローバール空間上での回転としても表すことができる。データ探索で探す「正しいデータの状態」を  $|w\rangle$ , 「正しくないデータの状態」を  $|r\rangle$  としたとき、正規直交基底  $\{|w\rangle, |r\rangle\}$  によって張られる空間をグローバール空間という。グローバール演算子  $G$  はこのグローバール空間上での回転

$$G = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \quad (3)$$

とかくことができる。ただし、 $\sin\frac{\theta}{2} = \sqrt{\frac{t}{N}}$ ,  $\cos\frac{\theta}{2} = \sqrt{\frac{N-t}{N}}$  である。

### 2.2 量子数え上げアルゴリズム

$N$  個の要素の中に  $t$  個の解が存在するとしたとき、その解の個数  $t$  を求めたい。Brassard らは  $O(\sqrt{tN})$  回の操作だけで解の個数を求める、量子数え上げアルゴリズムを考案した<sup>1)</sup>。このアルゴリズムは、Grover のアルゴリズムと量子フーリエ変換から構成されている。

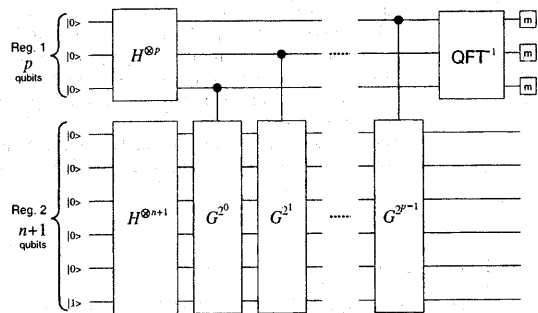


図1 グローバールのデータ探索回路。グローバール演算子  $G$  は  $O(\sqrt{N})$  回実行される。

図1が量子数え上げ回路である。第1レジスタの  $p$  量子ビットは位相  $\theta$  を推定するために用いられ、第2レジスタの  $n+1$  量子ビットは  $N = 2^n$  個の要素とオラクル量子ビットを表現するために用いられる。量子数え上げアルゴリズムは、グローバール演算子  $G$  の回転角  $\theta$  を位相推定アルゴリズムを用いることで求めている。

量子数え上げアルゴリズムは、位相推定に必要な量子ビット数  $p$  を大きくすることで解の精度を上げることができる。  $p \geq n/2 + 5$  にすると、正しい解を得る

確率が 90% となることが知られている<sup>7)</sup>。したがって、本研究では  $p = n/2 + 5$  として実験を行っている。

### 2.3 量子和アルゴリズム

Abrams と Williams は量子コンピュータ上で数値積分を高速に解く量子和アルゴリズムを示した<sup>6)</sup>。彼らは量子和アルゴリズムに対して 2 種類の実装方法を提案した。一つが量子数え上げアルゴリズム<sup>1)</sup>による実装で、もう一つが量子平均アルゴリズム<sup>4)</sup>による実装である。本研究では、量子数え上げアルゴリズムを用いて量子和を計算する。量子数え上げアルゴリズムを用いる方が、回路の深さが短くなるという利点があるが必要な量子ビット数は増えることになる。

#### 2.3.1 定義

$d$  次元実数関数の積分を考えたとき、一般性を失うことなしに、この関数を  $g(x_1, x_2, \dots, x_d)$  に変換することができる。ただし、 $x_i \in [0, 1]$  かつ  $g(x_1, x_2, \dots, x_d) \in [0, 1]$  である。したがって、積分

$$I = \int_0^1 \int_0^1 \dots \int_0^1 g(x_1, x_2, \dots, x_d) dx_1 dx_2 \dots dx_d \quad (4)$$

を計算すればよい。さらに、この関数  $g$  を範囲  $[1, M]$  の積分値  $a_i$  上で定義された  $d$  次元実数関数  $f(a_1, a_2, \dots, a_d)$  に近似する。ただし、

$$f(a_1, a_2, \dots, a_d) = g\left(\frac{a_1}{M}, \frac{a_2}{M}, \dots, \frac{a_d}{M}\right) \quad (5)$$

である。よって、近似的に  $I$  を計算するためには、和

$$S = \frac{1}{M^d} \sum_{x_1=1}^M \sum_{x_2=1}^M \dots \sum_{x_d=1}^M f(a_1, a_2, \dots, a_d) \quad (6)$$

を計算すればよい。Abrams と Williams は量子コンピュータ上でこの和  $S$  を高速に解くアルゴリズムとして量子和アルゴリズムを提案した。

#### 2.3.2 量子数え上げを用いた量子和の計算

量子和アルゴリズムを用いて式 (6) の和  $S$  を計算する。量子数え上げアルゴリズムを用いるためには、実数値関数  $f(a_1, a_2, \dots, a_d)$  を真偽値関数に変換する必要がある。そのため、範囲  $[1, Q]$  の積分値をとるパラメータ  $q$  を導入する。ただし、 $Q$  は欲しい精度によって決まる値である。真偽値関数  $b$  はパラメータ  $q$  を用いて次のように定義される。

$$b(a_1, a_2, \dots, a_d, q) = \begin{cases} 1 & \text{if } q \leq f(a_1, a_2, \dots, a_d) * Q \\ 0 & \text{if } q > f(a_1, a_2, \dots, a_d) * Q \end{cases} \quad (7)$$

この真偽値関数  $b$  を量子オラクルとして量子数え上げアルゴリズムを適用すればよい。この量子オラクルは、実数関数  $f$  のグラフに格子点を考えたときにグラフより下にある格子点の個数を数え上げることを意味している。そして得られた格子点の数から近似的に積分を計算する。

#### 2.3.3 量子和アルゴリズムの計算量解析

$n \gg \epsilon^{-1}$  と仮定すると、量子和アルゴリズムのオー

ダーは

$$\begin{array}{ll} \epsilon^{-1} & \text{量子クエリー,} \\ \epsilon^{-1} \log n & \text{演算回数,} \\ \log n & \text{量子ビット数} \end{array}$$

となる<sup>11)</sup>。

### 2.4 半古典的な位相推定

Parker ら<sup>9)</sup> は、Shor の因数分解アルゴリズム<sup>10)</sup> 中の位相推定に必要な量子ビットを大幅に削減する方法を提案した。一般的な Shor のアルゴリズムの因数分解回路の概要は図 2 のようになる。

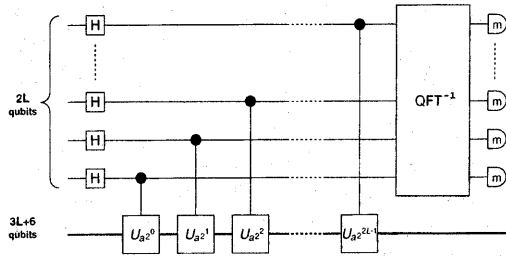


図 2 因数分解に対する位相推定回路。  $m$  と名付けられたゲートは観測である。

Parker らはこのアルゴリズムの位相推定に用いられている制御ビットの数を削減し、図 3 のように制御ビットを 1 量子ビットにした。

図 2 と図 3 を比較すると、図 3 では毎回制御ビットを観測しその結果を次の操作に用いている。

### 2.5 QFT を用いた加算

通常のキャリーを用いた加算では、2 つの  $n$  量子ビットの数を加算するのに、少なくとも  $3n$  量子ビットが必要となる。それに対して、Draper<sup>2)</sup> は量子フーリエ変換を用いることで、キャリービットを削減し加算に必要な量子ビットを削減する方法を提案した。図 4 がこの量子フーリエ変換を用いた加算の回路である。ただし、 $\phi(b)$  は  $b$  に量子フーリエ変換をかけたものである。

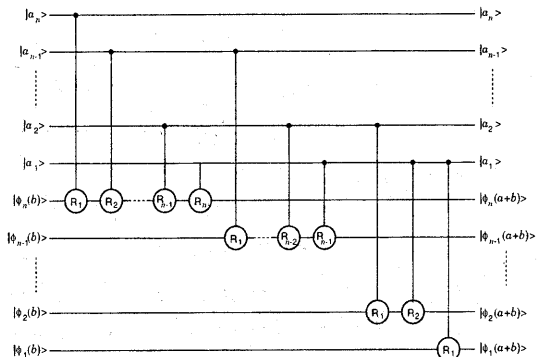


図 4 量子フーリエ変換を用いた加算

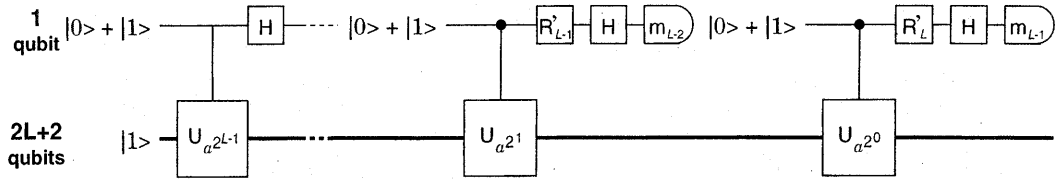


図3 1 制御ビットによる位相推定回路.  $R'_j$  直前の観測結果から与えられる:

$$R'_j = \begin{pmatrix} 1 & 0 \\ 0 & \phi'_j \end{pmatrix}, \text{ where } \phi'_j = \exp[-2\pi i \sum_{k=2}^j m_{j-2}/2^k].$$

フーリエ変換を行うと加算情報を位相に移すことができる。したがって、このフーリエ変換を用いた加算方法では位相を回転させることで、位相の上での加算を実現している。この量子フーリエ変換を用いた加算は、回路を並列化することで  $O(\log n)$  回の操作で実行できる。

### 2.6 デコヒーレンスエラー

我々はデコヒーレンスエラーモデルとして depolarizing channel を考える。このモデルでは、回路の深さが1増えるごとに各量子ビットに独立にエラーが発生する。それぞれの量子ビットには確率  $d$  でエラーが発生し、確率  $1-d$  では状態は変化しない。エラーが発生した場合には、エラー演算子  $\sigma_x, \sigma_y, \sigma_z$  がそれぞれ等確率 ( $\frac{1}{3}$ ) で適用される。したがって、デコヒーレンスエラーの影響は、回路の深さと量子ビット数の積、すなわち回路のサイズに比例することになる。

## 3. 量子和回路の構成

### 3.1 積分

量子和回路のエラーへの耐性を評価するために、次の関数の積分を考える。(図5(a))

$$y = f(x) = \frac{1}{6}x - \frac{1}{3} \quad (8)$$

$$I = \int_2^4 f(x) dx = \int_2^4 \left(\frac{1}{6}x - \frac{1}{3}\right) dx \quad (9)$$

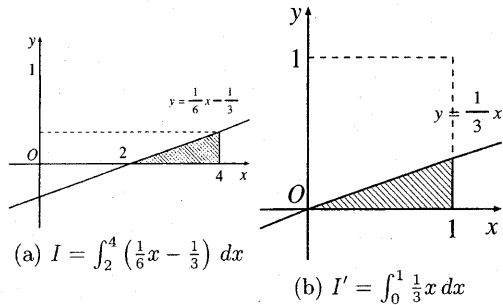


図5 量子和で計算する積分

この関数を量子和アルゴリズムを用いて積分するためには、この関数の定義域と値域を  $[0, 1]$  に収める必

要がある。そこで、次のような平行移動を行う。

$$I = \int_0^2 f(x+2) dx \quad (10)$$

$$= \int_0^2 \frac{1}{6}x dx \quad (11)$$

さらに、 $x$  軸方向にスケールリングする。

$$I' = \int_0^1 \frac{1}{3}x dx \quad (12)$$

ただし、 $I = 2I'$  である。(図5(b)) したがって、この積分  $I'$  を計算すればよい。積分  $I'$  を計算する量子和アルゴリズムの真偽値関数  $b$  は

$$b(x, y) = \begin{cases} 1 & y \leq \frac{1}{3}x \\ 0 & y > \frac{1}{3}x \end{cases} \quad (13)$$

である。この関数  $b$  は次の真偽値関数  $b'$

$$b'(x, y) = \begin{cases} 1 & 3y \leq x \\ 0 & 3y > x \end{cases} \quad (14)$$

と等しい。よって、 $b'$  を計算する量子和回路を構成する。

### 3.2 従来の量子和回路

図6は  $b'$  を計算する量子和回路の量子オラクルである。

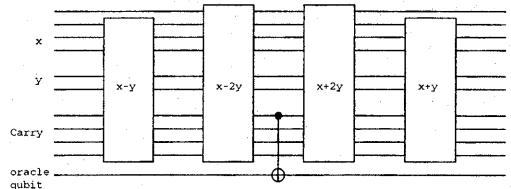


図6 積分  $I' = \int_0^1 \frac{1}{3}x dx$  を計算する量子オラクル, 加減算にはキャリーを使用

この量子オラクルは、実際には  $3y \leq x$  となるものの個数を数えるものではなく、 $3y > x$  となるものの個数を数えるものとなっている。なぜなら、引き算の結果のキャリーを制御ビットして状態を反転させるので、 $3y > x$  すなわち  $x - 3y < 0$  となるものの個数を数える方が都合がよい。したがって、本研究の量子和の計算ではグラフを考えたときに、グラフより上側の格子点の個数を量子和回路によって数え、全体の格子点の個数から量子和回路によって得られた個数を引くことで、実際に求めたい格子点の個数を求めるとい

た手順を踏んでいる。

### 3.3 改良した量子和回路

2.4の1制御ビットによる位相推定を用いると、図7のように量子和回路の位相推定に必要な量子ビット数を  $p-1 = n/2 + 4$  量子ビット減らすことができる。図7のグローバール演算子  $G$  は図8の形をしている。この図8の中の量子オラクル  $O$  を量子フーリエ変

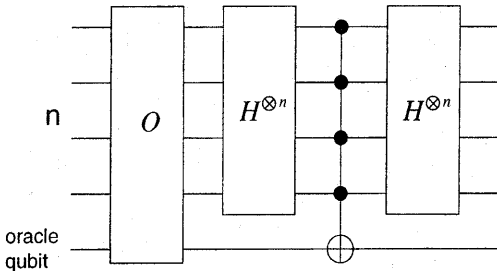


図8 グローバール演算子  $G$

換を用いた加算によって実装したものが図9である。図6と違い加減算にキャリーを必要としないため、少ない量子ビットで構成されている。

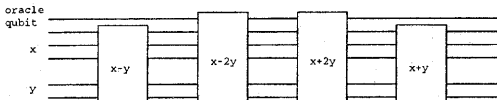


図9 量子フーリエ変換を用いた量子オラクル  $O$

## 4. 実験結果

量子とアルゴリズムのエラーへの耐性を調べるために、積分  $I' = \int_0^1 \frac{1}{3}x dx$  を計算する量子和回路の振る舞いを調べた。考える格子点のサイズは  $4 \times 4$  とする。したがって、 $n=4, p=n/2+5=7$  とする。真の解の個数を  $t$ 、量子和回路から得られる解の個数を  $\bar{t}$  としたとき、この積分  $I'$  での真の解は  $t=11$  となる。このときの積分結果は  $\frac{16-11}{16} = \frac{5}{16} = 0.3125$  であり、真の解は  $I' = \int_0^1 \frac{1}{3}x dx = \frac{1}{6} = 0.1333$  である。また、本研究の実験結果は100回の試行の平均をとったものとなっている。

### 4.1 従来の量子和回路と改良した量子和回路の比較

始めに従来の回路を用いて積分  $I' = \int_0^1 \frac{1}{3}x dx$  を計算する量子和回路の振る舞いを調べた。図10は、エラーのない場合での従来の量子和回路におけるそれぞれの解の個数を得られる確率を示している。横軸はこの量子和回路を観測して得られる解の個数を表し、縦軸はそれぞれの解を得る確率を表している。

エラーがない場合は、正しい解を得る確率は90%となっている。また、デコヒーレンスエラー率が  $d =$

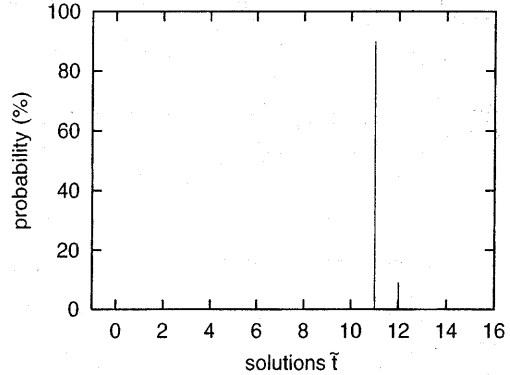


図10 エラーのない量子和回路でのそれぞれの解を得る確率

$10^{-5}$  のときのそれぞれの解の得られる確率は図11となった。図11から正しい解  $t=11$  を得る確率が

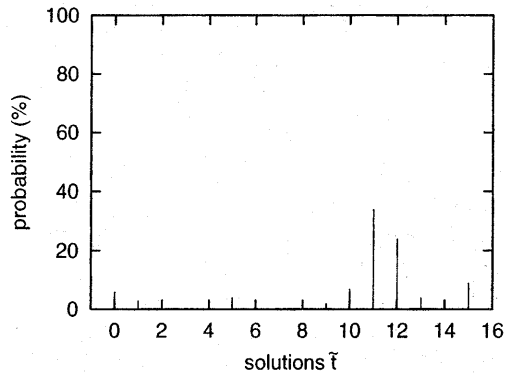


図11 デコヒーレンスエラー率が  $d = 10^{-5}$  のときの従来の量子和回路のそれぞれの解を得る確率

35% となった。さらに従来の量子和回路を用いるとエラーが入ったときの確率分布は山が崩れる形の分布をすることが分かる。

次に、3.3で構成した改良した量子和回路の振る舞いと比較する。エラーがない場合での両者の振る舞いは一致するので、デコヒーレンスエラーが発生したときの量子和回路の振る舞いを比較する。改良した量子和回路にデコヒーレンスエラー率  $d = 10^{-5}$  のエラーが発生したときのそれぞれの解の得られる確率分布は、図12のようになった。この改良した量子和回路では、デコヒーレンスエラー率が  $d = 10^{-5}$  のときでも正しい解を得る確率が50%ある。したがって改良した量子和回路の方がエラーへの耐性があることが分かる。また、図11と図12を比べると確率分布の仕方が異なっていることが分かる。改良した量子和回路では誤った解0を得る確率が大きくなっている。一般的には、量子回路にエラーが入ったときの振る舞いは、

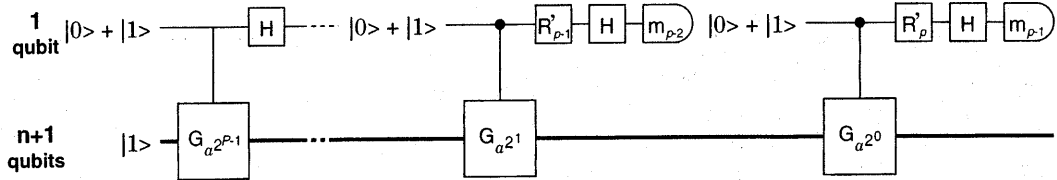


図 7 1 制御ビットによる量子和回路の位相推定

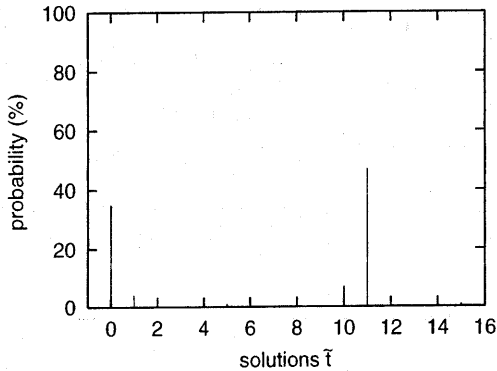


図 12 デコヒーレンスエラー率が  $d = 10^{-5}$  のときの改良した量子和回路でのそれぞれの解を得る確率

図 11 のように確率分布の山が崩れるのが普通であり、図 12 のように 0 付近に別のピークが出現するという現象は特殊なものである。この現象は、扱う問題設定には依存せず、改良した量子和回路にエラーが発生した場合に起こることが確認されている。

またデコヒーレンスエラー率を変化させたときの、正しい解を得る確率の遷移を比べたものが図 13 である。この図を見ると、改良した量子和回路の方が従来の量子和回路に比べて少しずつであるが高い確率で正しい解を得ていることが分かる。したがって、改良した量子和回路の方がデコヒーレンスエラーに対して少し強いことが分かった。

従来の量子和回路と改良した量子和回路を、回路の構成という点から比較したものが表 1 とである。

表 1 積分  $I = \int_0^1 \frac{1}{3} x dx$  を計算する量子和回路の比較

	深さ	量子ビット数	サイズ
従来の量子和回路	$1.0 \times 10^4$	18	$1.8 \times 10^5$
改良した量子和回路	$1.5 \times 10^4$	6	$9.0 \times 10^4$

この表を見ると、改良した量子和回路が従来のものと比べて、非常に少ない量子ビットで構成されていることが分かる。実際の量子コンピュータや量子計算シミュレータ上で量子回路を実装する場合、必要な量子ビット数が少ないことは大きな利点である。したがって、この観点から改良した量子和回路の有用性が大き

いことが示される。また必要な量子ビット数が少ないので、深さと量子ビット数の積である回路のサイズに関しても改良した量子和回路の方が小さなものとなっている。デコヒーレンスエラーの影響は回路のサイズに比例するので、デコヒーレンスエラーに対する耐性という面からも、改良した量子和回路の方が有用であることが分かる。また、表 1 と図 13 を比べたとき回路のサイズに比例するほどのデコヒーレンスエラーの影響の差は見られない。これは、量子数え上げアルゴリズムを用いた量子和回路自体がもともとそれほどエラーに強い回路ではないことを示唆している。

#### 4.2 改良した量子和回路でのエラーにおける確率分布の利用

図 12 から改良した量子和回路では、エラーが起これると 0 付近の誤った解を得る確率が高くなることが分かった。そこで、この 0 付近の解を得るという性質を利用する手法を提案する。

量子和アルゴリズムは、素因数分解などとは異なり得られた解を検算することができない。そのため、量子和アルゴリズムで正しい解を得るためには、量子和アルゴリズムを複数回試行し一番多く得た解を「正しい解」と見なすことになる。すなわち相対的な確率が大きいものが正しい解とみなされることになるのである。今改良した量子和回路にエラーがある状況下では 0 付近に誤りのピークが出現することが分かっている。この 0 付近の解を取り除くことで正しい解の相対的な確率を上げることを考える。図 14 は、図 12 の解のうち  $t = 0$  を得る確率を除いたときのそれぞれの解の得る相対確率を示している。この結果を見ると、正しい解  $t = 11$  を得る相対確率が約 25% 上昇している。したがって、この 0 付近の誤りの解を除くという手法を使うことで、エラーが入ったときに 0 付近に誤まりのピークを持つという特性を持つ改良した量子和回路は、従来の量子和回路に比べてさらに正しい解を得やすい、すなわちエラーにより強いことが分かった。

この 0 付近の誤りの解を除くという手法は、量子和回路が計算する積分値が 0 や 1 に近いときには使えない。しかし、積分値が 0 や 1 に近いときにも積分する関数を平行移動させてから計算することで 0 付近の解を誤りの解とみなすことができる、この手法を用いることが可能となる。

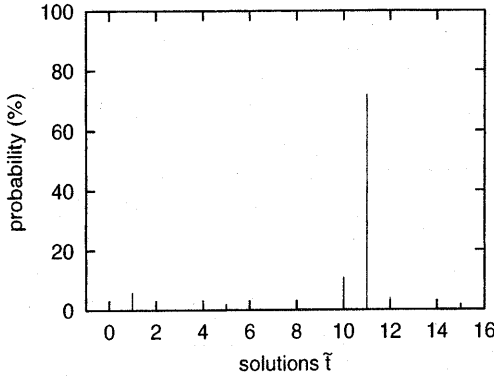


図 14 デコヒーレンスエラー率が  $d = 10^{-5}$  のときの 0 を得る確率を除いた相対確率

## 5. おわりに

量子計算シミュレータを用いることで、初めて量子和アルゴリズムのエラーに対する振る舞いを調べた。従来の手法での量子和回路では、積分  $I = \int_0^1 \frac{1}{3}x dx$  を計算するときは、デコヒーレンスエラー率が  $d = 10^{-5}$  までなら正しい解を得られることが分かった。また、半古典的な位相推定と量子フーリエ変換による加算という手法を用いて量子和回路を改良した。この改良した量子和回路は、従来のものと比べて非常に少ない量子ビットで構成することが可能であり、その点で非常に有用である。また必要な量子ビットが少ないため回路サイズが小さくなりデコヒーレンスエラーに対しても強い回路となっている。さらに、両者の回路のエラーに対する振る舞いを比較すると、エラーに対する性質が異なることも分かった。改良した量子和回路は、エラーが発生すると 0 付近の誤った解を得る確率が大きくなるという特殊な性質を持っており、この性質を利用し 0 付近の誤った解を取り除くことで正しい解を得る相対確率を向上させることができる。したがって、改良した量子和回路はさらにエラーに対して強いものといえ、我々の改良した量子和回路がデコヒーレンスエラーに関しても非常に有用なものとなっていることが示された。

今後の課題としては、改良した量子和回路にエラーが発生すると 0 付近の解を得る確率が大きくなる原因を理論的に解明する必要がある。現在の研究で、この 0 付近の解を得やすくなるのは Grover 空間上にエラーが発生した場合に起因していることが分かっている。より詳しい解析結果は WAAC03<sup>5)</sup> で発表する予定である。また、今回の実装では量子和回路を量子数え上げアルゴリズムを用いて実装しているが、量子平均アルゴリズムを用いて実装したときと比較することも挙げられる。

## 参考文献

- 1) BRASSARD, G., HØYER, P. and TAPP, A. Quantum Counting, Automata, languages and programming (Aalborg, 1998), Lecture Notes in Computer Science in Comput. Sci., 1443, Springer, Berlin, 1998 (May 1998), arXiv:quant-ph/9805082 v1.
- 2) DRAPER, T. G. Addition on a Quantum Computer (Aug 2000), arXiv:quant-ph/0008033 v1.
- 3) GROVER, L. K. A Fast Quantum Mechanical Algorithm for Database Search, 23th Annual ACM Symposium on the Theory of Computing (STOC) (may 1996), arXiv:quant-ph/9605043.
- 4) GROVER, L. K. A framework for fast quantum mechanical algorithms, STOC '98 (Dallas, TX), New York (1999), ACM, arXiv:quant-ph/9711043.
- 5) HASEGAWA, J., NIWA, J., YURA, F. and IMAI, H. Analysis and Extensions of Quantum Search and Counting Against Decoherence, 2003 Japan-Korea Joint Workshop on Algorithms and Computation (WAAC03) (2003), in preparation.
- 6) HEINRICH, S. Quantum Summation with an Application to Integration, J. Complexity 18 (2002), Vol. 1 (May 2001), arXiv:quant-ph/0105116 v1.
- 7) NIELSEN, M. A. and CHUANG, I. L. *Quantum Computation and Quantum Information*, Cambridge University press (2000).
- 8) NIWA, J., MATSUMOTO, K. and IMAI, H. General-purpose parallel simulator for quantum computing (2002), Phys. Rev. A **62**, 062317, December 2002.
- 9) PARKER, S. and PLENIO, M. B. Efficient Factorization with a Single Pure Qubit and log  $N$  Mixed Qubits, *Phys. Rev. Lett.*, **85** (2000), 3049–3052, arXiv:quant-ph/0001066.
- 10) SHOR, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM Journal on Computing*, **26**, 5 (1997), 1484–1509.
- 11) TRAUB, J. F. and WOŹNIAKOWSKI, H. Path Integration on a Quantum Computer (Sep 2002), arXiv:quant-ph/0109113 v2.

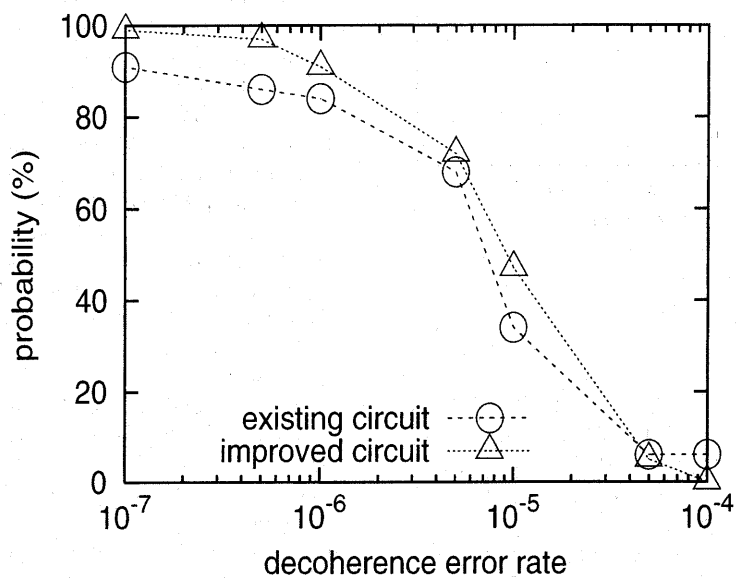


図 13 デコヒーレンスエラー率が  $d = 10^{-5}$  のときの改良した量子回路でのそれぞれの解を得る確率