

初期状態にずれがある場合の
グローバーのアルゴリズムの振る舞いについて

9BSP1118 村岡海人

目次

1	はじめに	2
1.1	研究の背景	2
1.2	研究の目的	2
1.3	本論文の構成	3
2	基本的内容	3
2.1	量子ビット	3
2.2	量子計算	4
2.3	量子並列性	7
2.4	グローバーのアルゴリズム	8
3	本論	14
3.1	回転演算子の組み合わせとしてのアダマール演算子	14
3.2	y 軸方向にずれ δ_y がある場合	14
3.3	z 軸方向にずれ δ_z がある場合	17
3.4	y 軸方向、 z 軸方向にずれ δ_y, δ_z がある場合	17
4	まとめと結論	21

1 はじめに

1.1 研究の背景

量子コンピュータは量子力学を利用して計算を行うコンピュータである。そもそもコンピュータとは物理現象を利用することで計算を行う道具であるが、現代の通常のコンピュータ (古典コンピュータ) では古典的な電磁気現象を利用している。量子力学を用いた情報処理は量子情報という分野で調べられており、量子コンピュータでは古典コンピュータによる情報処理以上の情報処理を行うことができる。

量子情報の基本単位は量子ビットと呼ばれているが、これは古典情報の基本単位であるビットを量子力学に対応させたものである。古典計算では 0 と 1 の 2 つの異なる状態についての操作を行うことで計算を行う。一方、量子力学では、2 つの異なる状態の量子力学的な重ね合わせが許されており、量子ビットの状態は 0 と 1 の 2 つの状態の重ね合わせとして表現されている。量子ビットは様々な物理系で実現されており、例えば超伝導を利用した電気回路における量子ビットや、真空中に作ったポテンシャルに原子を捕まえてその電子を使った量子ビットなどがある。

量子コンピュータを構成する量子演算は、量子系における時間発展を利用することになる。量子ビットの状態は自由度 2 のベクトルで表現され、量子演算はそのベクトルに作用する行列で表現される。量子コンピュータには量子状態が入力として与えられ、量子演算を作用させた後の状態が出力される。そのため、量子コンピュータの記述には線形代数を利用することができる。

量子コンピュータは大量の量子ビットの集合であり、量子アルゴリズムを表すユニタリ演算子を状態に作用させることによって計算が行われる。量子アルゴリズムとは、量子コンピュータのためのアルゴリズムである。これまでに、古典コンピュータ用のアルゴリズムより計算量の小さい量子アルゴリズムがいくつか提案されてきた。例えば、グローバーのアルゴリズムは、整列化されていない N 個のデータベースから特定のデータを探索する量子アルゴリズムである。古典的なアルゴリズムである線形探索アルゴリズムでこの問題を解くには $O(N)$ の計算量を必要とするが、グローバーのアルゴリズムでは $O(\sqrt{N})$ の計算量で計算を行える。

量子コンピュータを実現する上で最も深刻な問題の 1 つとして、量子ビットの操作にずれが生じることにより、計算結果にずれが生じることがある。量子ビットの操作はシュレーディンガー方程式の解であり、量子ゲートを表すユニタリ演算子は時間 t 、ハミルトニアンを H とすると、量子ゲートを表すユニタリ演算子は $U = \exp(-iHt/\hbar)$ と表すことができる。しかし、実際に量子ゲートを作用させる際、ハミルトニアン H のパラメータや、時間 t がずれてしまうことがある。これはユニタリ演算子を作用して作りたかった状態と異なった状態が得られてしまう。量子計算は大量の量子ゲートの組み合わせとして実現されるため、1 つ 1 つの量子ゲートにずれがないことが重要になってくる。

1.2 研究の目的

本研究では、初期状態を準備する操作にずれが含まれる場合に、グローバーのアルゴリズムがどれだけ機能するか調べることを目的とした。

1.3 本論文の構成

本論文の構成は次のようになっている。2 章では本研究に関する基本的事項を示す。3 章では本研究の内容と結果を示す。また、各節で結果から読み取れることを議論できることをまとめた。4 章では本研究のまとめと今後の展望について示している。

2 基本的内容

本章では、本研究に必要な基本事項のまとめを行う。2.1 節では単一量子ビットと多量子ビットの説明を行う。2.2 節では量子情報における重要な演算子や量子回路 2.3 節では量子アルゴリズムの中で利用される量子並列性について説明する。2.4 節では本研究の対象である量子アルゴリズムのグローバーのアルゴリズムについて説明する。

2.1 量子ビット

この節では量子計算の基本要素である量子ビットとその状態について説明する。

2.1.1 単一量子ビット

まず、量子ビットの説明をする。古典ビットの 0 と 1 の 2 つの状態に対応した、状態 $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ と $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ がある。このように、量子ビットの状態は 2 次元複素ベクトル空間のベクトルで表される。ここで、量子状態を表すために、ブラケット記法を用いた。量子ビットと古典ビットの違いは、量子ビットは $|0\rangle$ と $|1\rangle$ の重ね合わせ状態を取り得ることである。これは次のように $|0\rangle$ と $|1\rangle$ の線型結合として、

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2.1)$$

$$|\alpha|^2 + |\beta|^2 = 1 \quad (2.2)$$

と表される。量子ビットの状態 $|\psi\rangle$ は 2 次元の複素ベクトル空間 \mathbb{C}^2 の元である。ここで、 α, β は複素数であり、 $|0\rangle, |1\rangle$ は \mathbb{C}^2 の正規直交基底を成している。古典計算では、古典ビットを調べてそれが 0, 1 のいずれの状態にあるかを知ることができる。例えば、古典コンピュータでメモリの内容を取り出す際には、常にこの性質を利用している。一方、単一量子ビットに対しては、その量子状態 $|\psi\rangle$ を調べてその量子状態を決めることはできない。つまり、ある 1 つの量子ビット $|\psi\rangle$ に対して状態を変化させることなく α, β の値を決めることはできない。なぜなら、 $|\psi\rangle$ に対して、 $|0\rangle, |1\rangle$ のいずれにあるかを調べるための計算基底による測定を行うと確率 $|\alpha|^2$ で $|0\rangle$ 、確率 $|\beta|^2$ で $|1\rangle$ が得られ、状態が変わってしまうからである。なお、式 (2.2) は確率の和が 1 であることを示しており、以下ではこれを正規化条件と呼ぶ。

量子ビットは自由度が 2 の多くの系で実現されている。例えば、単一光子の異なる偏光、一様磁界中の核スピン、単一原子における電子軌道の 2 つの状態などが挙げられる。原子モデルでは電子はいわゆる基底状態または励起状態に存在しそれぞれは $|0\rangle, |1\rangle$ に対応している。

図 1 のような幾何学的表現 (ブロッホ球) が量子ビットを考える上で有用である。式 (2.2) より、式 (2.1) を次のように書き換える。

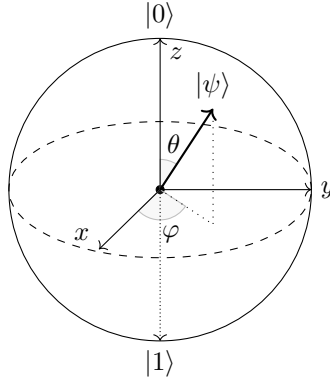


図1 量子ビットのブロッホ球表示

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \quad (2.3)$$

ここで、 θ, φ は実数である。図1に示すように、 θ, φ は3次元単位球面上の点に対応する。これは単一量子ビット状態を視覚化する便利な方法である。単一量子ビットの操作はブロッホ球上の描像で記述できる。

2.1.2 多量子ビット

多量子ビットの状態について考えてみる。簡単のため、2個の量子ビットがあるとする。これが古典ビットならば4つの取り得る状態00, 01, 10, 11がある。同様に2個の量子ビットの系には $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ で表される計算基底がある。2個の量子ビットを記述する状態ベクトルは、

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle \quad (2.4)$$

で与えられる。ここで、 $\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}$ はそれぞれの基底の複素確率振幅である。単一量子ビットの場合と同様に、計算基底による測定結果 $x (= 00, 01, 10, 11)$ は確率 $|\alpha_x|^2$ で得られ、計算基底による測定後の量子ビットの状態は $|x\rangle$ となる。正規化条件は $\sum_{x \in \{0,1\}^2} |\alpha_x|^2 = 1$ で表される。ここで、記号 $\{0,1\}^2$ は各ビットの値が0または1をとる長さ2のあり得るビット列の集合を意味しており、 $\sum_{x \in \{0,1\}^2}$ はその集合に属する全てのビット x に対して和をとることを意味する。

一般に n 個の量子ビットを考えると、この系の計算基底は $|x_1 x_2 \cdots x_n\rangle$ の形をしており、この系の量子状態は 2^n 個の振幅で表される。ここで、 $x = x_1 x_2 \cdots x_n$ $x \in \{0,1\}^n$ は各文字が0または1であり、長さ n のビットの集合を表す。

2.2 量子計算

この節では量子計算について説明する。量子力学を利用して行う計算を量子計算と呼ぶ。量子計算は量子的な論理演算の組み合わせによって実現される。これは古典コンピュータが論理演算の組み合わせで計算が行われているのと同様である。本節では、量子コンピュータにおけるいくつかの基本的な演算および、それを可視化する量子回路について説明する。

入力	出力
0	1
1	0

表 1 古典 NOT ゲートの真理値表

2.2.1 単一量子ビット演算

量子ビットに対する論理演算を説明するために、まず古典コンピュータについて考える。古典コンピュータ回路は配線と論理演算より構成される。配線は回路中で情報を運び、論理演算は情報を別の情報に変換して操作する。例えば、古典的単一ビット論理演算である NOT ゲートを考える。NOT 演算子の働きは表 1 に示す真理値表で定義され、 $0 \rightarrow 1$ 及び $1 \rightarrow 0$ つまり、状態 $0, 1$ の入れ替えが行われる。真理値表とは入力と出力の対応を表すための表である。

量子ビットに対しても NOT 演算子を定義することができる。量子ビットへの操作は線形変換であり、単一量子ビットに作用する量子ゲートは 2×2 の行列で記述できる。特に量子ビットに対する NOT ゲートは次のように定義される行列 X で表される。

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (2.5)$$

この行列はパウリ X と呼ばれる。また、量子状態 $\alpha|0\rangle + \beta|1\rangle$ をベクトル表記で次のように表す。

$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (2.6)$$

ここで、上の要素 α は $|0\rangle$ に対する振幅、下の要素 β は $|1\rangle$ に対する振幅に相当する。量子ビットに対する NOT 演算子の作用は、

$$X \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} \quad (2.7)$$

で与えられる。このパウリ X はブラケット表記で次のようにも記述できる。

$$X = |0\rangle\langle 1| + |1\rangle\langle 0| \quad (2.8)$$

量子演算の行列に対する制約について議論する。量子状態 $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ に対して正規化条件 $|\alpha|^2 + |\beta|^2 = 1$ が必要であるが、演算子を作用させた後の状態 $|\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$ に対しても正規化条件が成立しなければならない。この条件を満たすのは、単一量子ビットゲートを記述する行列 U がユニタリ行列、つまり $U^\dagger U = U U^\dagger = I$ を満たす場合である。ここで U^\dagger は U のエルミート共役、 I は恒等演算子である。

単一量子ビットに対する量子演算子として、パウリ Z 、パウリ Y 、アダマール演算子 H を定義する。

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1| \quad (2.9)$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = i|1\rangle\langle 0| - i|0\rangle\langle 1| \quad (2.10)$$

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 1|) \quad (2.11)$$

アダマール演算子を $|0\rangle$ または $|1\rangle$ に作用させると、

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad (2.12)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (2.13)$$

のように変換される。以上の演算子については、簡単な計算によって $X^2 = Y^2 = Z^2 = H^2 = I$ であることがわかる。

2.2.2 1 量子ビットにおける任意の回転演算子

1 量子ビットに対する任意のユニタリ演算子はいくつかの回転演算子の組み合わせで表すことができる。まず、 x, y, z 軸周りに回転させる演算子 $R_x(\theta), R_y(\theta), R_z(\theta)$ を以下のように定義する。

$$R_x(\theta) = e^{-i\theta X/2} = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) X = \begin{pmatrix} \cos\frac{\theta}{2} & -i \sin\frac{\theta}{2} \\ -i \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \quad (2.14)$$

$$R_y(\theta) = e^{-i\theta Y/2} = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) Y = \begin{pmatrix} \cos\frac{\theta}{2} & -i \sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \quad (2.15)$$

$$R_z(\theta) = e^{-i\theta Z/2} = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) Z = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \quad (2.16)$$

すると、1 量子ビットの任意のユニタリ回転演算子は、 R_y と R_z で分解できる。実数 $\gamma, \phi, \theta, \lambda$ を用いて、

$$U(\theta, \phi, \lambda) = e^{i\gamma} R_z(\phi) R_y(\theta) R_z(\lambda) = e^{i(\gamma - \frac{\theta}{2} - \frac{\phi}{2})} \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda} \sin\frac{\theta}{2} \\ e^{i\phi} \sin\frac{\theta}{2} & e^{i(\lambda+\phi)} \cos\frac{\theta}{2} \end{pmatrix} \quad (2.17)$$

と定義すると、 $\theta, \phi, \lambda, \gamma$ を適切に選ぶことで任意のユニタリ行列を表すことができる。上記で用いられる $e^{i(\gamma - \frac{\theta}{2} - \frac{\phi}{2})}$ は全体位相と呼ばれ、ブロッホ球上の回転操作や回転角に直接関わることはないので、本論では任意の回転行列として

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda} \sin\frac{\theta}{2} \\ e^{i\phi} \sin\frac{\theta}{2} & e^{i(\lambda+\phi)} \cos\frac{\theta}{2} \end{pmatrix} \quad (2.18)$$

のようにパラメタ表示されるユニタリ行列を考える [1]。

2.2.3 多量子ビット演算

複数の量子ビットからなる多量子ビットについての量子演算について考える。まず、2 量子ビットに作用する制御 NOT 演算子 U_{CNOT} を説明する。このゲートは、制御量子ビットと標的量子ビットに作用し、もし制御量子ビットの状態が $|0\rangle$ ならば標的量子ビットには何もせず、 $|1\rangle$ ならば標的量子ビットに NOT 演算子を作用させる。1 つ目の量子ビットを制御量子ビット、2 つ目の量子ビットを標的量子ビットとすると、制御 NOT 演算子 U_{CNOT} の作用は、

$$\begin{aligned} |00\rangle &\xrightarrow{U_{\text{CNOT}}} |00\rangle \\ |01\rangle &\xrightarrow{U_{\text{CNOT}}} |01\rangle \\ |10\rangle &\xrightarrow{U_{\text{CNOT}}} |11\rangle \\ |11\rangle &\xrightarrow{U_{\text{CNOT}}} |10\rangle \end{aligned} \quad (2.19)$$

となる。

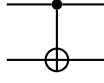


図2 制御 NOT 演算子

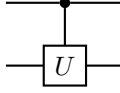


図3 制御ユニタリ演算子

最後に、2量子ビットに作用する制御 U 演算子を説明する。ここで U は任意のユニタリ行列である。このとき、制御 NOT 演算子の自然な拡張として制御 U 演算子を定義できる。もし、制御量子ビットが $|0\rangle$ ならば、標的量子ビットには何もせず、 $|1\rangle$ ならば標的量子ビットに任意のユニタリ演算子 U を作用させる。1つ目の量子ビットを制御ビット、2つ目の量子ビットを標的量子ビットとすると、制御 U 演算子 $\Lambda(U)$ は、

$$\begin{aligned}
 |00\rangle &\xrightarrow{\Lambda(U)} |00\rangle \\
 |01\rangle &\xrightarrow{\Lambda(U)} |01\rangle \\
 |10\rangle &\xrightarrow{\Lambda(U)} |1\rangle \otimes U|0\rangle \\
 |11\rangle &\xrightarrow{\Lambda(U)} |1\rangle \otimes U|1\rangle
 \end{aligned} \tag{2.20}$$

となる。

2.2.4 量子回路

この節では量子操作を表現する量子回路図について、例を挙げて説明する。量子回路図は水平線は各量子ビットを表しており、左から右に向けて読む。図2は2量子ビットに作用する制御 NOT 演算子を表しており、上の黒点が制御量子ビット、下の記号が標的量子ビットに NOT 演算子がかかる様子を表している。同様に図3は制御ユニタリ演算子の量子回路図である。 U は単一量子ビットに作用するユニタリ演算子であり、箱の中に U を書いた記号で表されている。

2.3 量子並列性

この節では多くの量子アルゴリズムの基本的性質である量子並列性について説明する。1ビットに作用する関数を $f(x) : \{0, 1\} \rightarrow \{0, 1\}$ とする。この関数を量子コンピュータで計算する便利な方法として、状態 $|xy\rangle (x \in \{0, 1\}, y \in \{0, 1\})$ から出発する2量子ビットの量子コンピュータを考える。適切な論理ゲートを用いるとこの状態は $|x\rangle |y \oplus f(x)\rangle$ に変換できる。ここで、 \oplus は2を法とする和を表す。この変換を行うためには、最初の量子ビットを制御量子ビット、2番目の量子ビットを標的ビットとすればよい。任意の x, y に対して $U_f |xy\rangle = |x\rangle |y \oplus f(x)\rangle$ となるユニタリ変換 U_f を考える。もし、 $y = 0$ ならば第2の標的量子ビットの最終状態の値は $f(x)$ である。重ね合わせ状態 $(|0\rangle + |1\rangle)/\sqrt{2}$ を制御量子ビットに入れて、 U_f を作用すると

$$\frac{|0\rangle |f(0)\rangle + |1\rangle |f(1)\rangle}{\sqrt{2}} \tag{2.21}$$

を得る。あたかも 2 つの x に対する $f(x)$ を計算したように見えるので、この特徴を量子並列性と呼ぶ。

任意の量子ビット数 n に対する関数についての量子並列性の議論は、アダマール変換と呼ばれる演算を用いて以下に行える。アダマール変換 W とは、 n 個の量子ビットの状態が全て $|0\rangle$ に初期化された $|0\rangle^{\otimes n}$ に対して、全量子ビットにアダマール演算子を作用させる変換のことである。 $n = 2$ の時、 $|0\rangle^{\otimes 2}$ にアダマール変換 $W = H^{\otimes 2}$ を作用させると以下の状態が得られる。

$$\begin{aligned} W |0\rangle^{\otimes 2} &= (H \otimes H) |00\rangle \\ &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \\ &= \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{\sqrt{2^2}} \end{aligned} \quad (2.22)$$

この操作を任意の n に一般化すると全ての $|0\rangle$ 状態に初期化された n 個の量子ビットにアダマール変換 $W = H^{\otimes n}$ を作用させることで、

$$\begin{aligned} W |0\rangle^{\otimes n} &= (H \otimes \dots \otimes H) |0 \dots 0\rangle \\ &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \otimes \dots \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \\ &= \frac{1}{\sqrt{2^n}} \sum_x |x\rangle \end{aligned} \quad (2.23)$$

という状態が得られる。すなわち、アダマール変換により全ての計算基底の均等な重ね合わせを作り出すことができる。

n ビットの入力 x と 1 ビットの出力を有する関数 $f(x)$ の量子並列計算は次のように行う。2 量子ビットの時と同様に、 $n + 1$ 個の量子ビットの状態 $|x\rangle |0\rangle$ に作用するユニタリ変換 U_f は、 $U_f |x\rangle |0\rangle = |x\rangle |f(x)\rangle$ とする。 $n + 1$ 個の量子ビット状態 $|0\rangle^{\otimes n} |0\rangle$ を準備して、最初の n 個の量子ビットにアダマール変換を作用し、次に U_f を作用すると、

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle \quad (2.24)$$

の状態が得られる。 U_f を 1 回しか作用していないにも関わらず、量子並列性により、 f の全ての可能な値を同時に計算することができる [2]。

この量子並列性を用いた量子アルゴリズムの例として、ドイチュ・ジョザのアルゴリズムや本研究の対象である、量子アルゴリズムの 1 つであるグローバーのアルゴリズムなどがある。

2.4 グローバーのアルゴリズム

本節では、探索問題を解く量子アルゴリズムの一種であるグローバーのアルゴリズムを説明する。

2.4.1 概要

グローバーのアルゴリズムは以下のような流れで行う。 N 個のデータに対して $O(\sqrt{N})$ 回の計算量で解を見出すことができる。ソートされていないデータベースを扱う古典的な探索アルゴリズムとして線形探索アルゴリズムがある。線形探索アルゴリズムは N 個のデータに対して端から 1 つずつ順番に解であるか調べていくアルゴリズムである。このアルゴリズムの計算量は $O(N)$ である。よって、この古典アルゴリズムと比較するとグローバーのアルゴリズムの方が高速である。

グローバーのアルゴリズムが対象とするのは、 N 個のデータベースから M 個の解を探索する問題である。ここでは簡単のため $M = 1$ とする。 n を量子ビット数とすれば、 $N = 2^n$ に対応し、探索するデータのラベルとして n 桁のビット列 $x = x_1 \cdots x_n$ を採用することができる。

グローバーのアルゴリズムは以下のように行われる。

1. 全ての状態の重ね合わせ状態 $|s\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ を用意する
2. 演算子 U_w (解に対する反転操作) を作用させる
3. 演算子 U_s ($|s\rangle$ を対象軸にした反転操作) を作用させる
4. 2、3 を k 回繰り返す
5. 計算基底による測定

ただし、 k は量子ビット数に応じて決まるある非負整数であり、2,3 の操作を k 回繰り返すことによって、解が見つかる確率が大きくなる。

2.4.2 アルゴリズムの流れ

以下では w を解とする。

■操作 1(初期状態の準備) まず初めに、全ての状態の重ね合わせ状態 $|s\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ を用意する。初期状態 $|0\rangle^{\otimes n} = |0 \cdots 0\rangle$ に対してアダマール変換 $W = H^{\otimes n}$ を作用させると、

$$\begin{aligned}
 |s\rangle &= H^{\otimes n} |0\rangle^{\otimes n} \\
 &= (H \otimes \cdots \otimes H) |0 \cdots 0\rangle \\
 &= \frac{1}{\sqrt{2^n}} (|0\rangle + |1\rangle) \otimes \cdots \otimes (|0\rangle + |1\rangle) \\
 &= \frac{1}{\sqrt{2^n}} (|00 \cdots 00\rangle + |00 \cdots 01\rangle + \cdots + |11 \cdots 10\rangle + |11 \cdots 11\rangle) \\
 &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle
 \end{aligned} \tag{2.25}$$

という状態が得られる。ここで、

$$\sum_{x=0}^{2^n-1} |x\rangle = |00 \cdots 00\rangle + |00 \cdots 01\rangle + \cdots + |11 \cdots 10\rangle + |11 \cdots 11\rangle \tag{2.26}$$

である。

■操作 2(解に対する反転操作) 次に解に対する反転操作を作用させる。つまり、入力 $|x\rangle$ に対して x が解なら -1 をかけて位相を反転し、そうでないならば何もかけないような単一量子ゲート U_w を作用させる。 U_w は以下のように表される。

$$\begin{cases} U_w |x\rangle = |x\rangle & (x \neq w) \\ U_w |w\rangle = -|w\rangle \end{cases} \tag{2.27}$$

$$U_w = I - 2|w\rangle\langle w| \tag{2.28}$$

これを用いると、 $|s\rangle$ に U_w を作用させることで、

$$\begin{aligned}
U_w |s\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x=0, x \neq w}^{2^n-1} U_w |x\rangle + \frac{1}{\sqrt{2^n}} U_w |w\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{x=0, x \neq w}^{2^n-1} |x\rangle - \frac{1}{\sqrt{2^n}} |w\rangle \\
&= \left(|s\rangle - \frac{1}{\sqrt{2^n}} |w\rangle \right) - \frac{1}{\sqrt{2^n}} |w\rangle \\
&= |s\rangle - \frac{2}{\sqrt{2^n}} |w\rangle
\end{aligned} \tag{2.29}$$

という状態が得られる。

■操作 3($|s\rangle$ を対象軸にした反転操作) 最後に、 $|s\rangle$ を対象軸にした反転操作 U_s を定義する。

$$U_s = 2 |s\rangle \langle s| - I \tag{2.30}$$

$$\begin{cases} U_s |w\rangle = 2 |s\rangle \langle s|w\rangle - |w\rangle = \frac{2}{\sqrt{2^n}} |s\rangle - |w\rangle \\ U_s |s\rangle = 2 |s\rangle \langle s|s\rangle - |s\rangle = |s\rangle \end{cases} \tag{2.31}$$

式 (2.29) に U_s を作用させると、

$$\begin{aligned}
U_s U_w |s\rangle &= |s\rangle - \frac{2}{\sqrt{2^n}} \left(\frac{2}{\sqrt{2^n}} |s\rangle - |w\rangle \right) \\
&= \frac{2^n - 4}{2^n} |s\rangle + \frac{2}{\sqrt{2^n}} |w\rangle \\
&= \frac{2^n - 4}{2^n \sqrt{2^n}} \sum_{x=0, x \neq w}^{2^n-1} |x\rangle + \left(\frac{2^n - 4}{2^n \sqrt{2^n}} + \frac{2}{\sqrt{2^n}} \right) |w\rangle \\
&= \frac{2^n - 4}{2^n \sqrt{2^n}} \sum_{x=0, x \neq w}^{2^n-1} |x\rangle + \frac{3 \cdot 2^n - 4}{2^n \sqrt{2^n}} |w\rangle
\end{aligned} \tag{2.32}$$

という状態が得られる。

上記の 2 つの反転操作 U_w, U_s を k 回繰り返す。2.4.4 項で述べるが、 $O(\sqrt{N})$ 回の繰り返しを行えば、十分に高い確率で解が得られる。

2.4.3 幾何学的な説明

$|w\rangle$ を解に対応する状態とし、 $|w\rangle$ に直交する状態 $|w^\perp\rangle$ を

$$|w^\perp\rangle \equiv \frac{1}{\sqrt{2^n - 1}} \sum_{x=0, x \neq w}^{2^n-1} |x\rangle \tag{2.33}$$

と定義する。すると初期状態 $|s\rangle$ は、

$$|s\rangle = \sqrt{\frac{2^n - 1}{2^n}} |w^\perp\rangle + \frac{1}{\sqrt{2^n}} |w\rangle \tag{2.34}$$

のように書き換えることができる。したがって、初期状態 $|s\rangle$ は解空間とその直交空間の成分に分けることができる。ここで、 $\cos \frac{\theta}{2} = \sqrt{\frac{2^n-1}{2^n}}$, $\sin \frac{\theta}{2} = \sqrt{\frac{1}{2^n}}$ とすると、式 (2.34) は

$$|s\rangle = \cos \frac{\theta}{2} |w^\perp\rangle + \sin \frac{\theta}{2} |w\rangle \quad (2.35)$$

と表すことができる。これを図示すると、図 4 のようになる。

次に、 $|s\rangle$ に U_w を作用させる。式 (2.29) より、

$$\begin{aligned} U_w |s\rangle &= |s\rangle - \frac{2}{\sqrt{2^n}} |w\rangle \\ &= \sqrt{\frac{2^n-1}{2^n}} |w^\perp\rangle + \frac{1}{\sqrt{2^n}} |w\rangle - \frac{2}{\sqrt{2^n}} |w\rangle \\ &= \sqrt{\frac{2^n-1}{2^n}} |w^\perp\rangle - \frac{1}{\sqrt{2^n}} |w\rangle \\ &= \cos \frac{\theta}{2} |w^\perp\rangle - \sin \frac{\theta}{2} |w\rangle \end{aligned} \quad (2.36)$$

と表すことができる。これを図示すると、図 5 のようになる。

最後に、 U_s を $U_w |s\rangle$ に作用させるが、以下の式が成り立つことに注意する。

$$\begin{aligned} \cos \left(\frac{3}{2} \theta \right) &= 4 \cos^3 \frac{\theta}{2} - 3 \cos \frac{\theta}{2} \\ &= 4 \left(\sqrt{\frac{2^n-1}{2^n}} \right) - 3 \left(\sqrt{\frac{2^n-1}{2^n}} \right) \\ &= \frac{2^n-4}{2^n} \sqrt{\frac{2^n-1}{2^n}} \end{aligned} \quad (2.37)$$

$$\begin{aligned} \sin \left(\frac{3}{2} \theta \right) &= 3 \sin \frac{\theta}{2} - 4 \sin^3 \frac{\theta}{2} \\ &= 3 \left(\frac{1}{\sqrt{2^n}} \right) - 4 \left(\sqrt{\frac{1}{2^n}} \right) \\ &= \frac{3 \cdot 2^n - 4}{2^n \sqrt{2^n}} \end{aligned}$$

式 (2.32) より、

$$\begin{aligned} U_s U_w |s\rangle &= \frac{2^n-4}{2^n} |s\rangle + \frac{2}{\sqrt{2^n}} |w\rangle \\ &= \frac{2^n-4}{2^n} \left(\sqrt{\frac{2^n-1}{2^n}} |w^\perp\rangle + \frac{1}{\sqrt{2^n}} |w\rangle \right) + \frac{2}{\sqrt{2^n}} |w\rangle \\ &= \frac{2^n-4}{2^n} \sqrt{\frac{2^n-1}{2^n}} |w^\perp\rangle + \frac{3 \cdot 2^n - 4}{2^n \sqrt{2^n}} |w\rangle \\ &= \cos \left(\frac{3}{2} \theta \right) |w^\perp\rangle + \sin \left(\frac{3}{2} \theta \right) |w\rangle \end{aligned} \quad (2.38)$$

と表すことができる。これを図示すると図 6 となる。

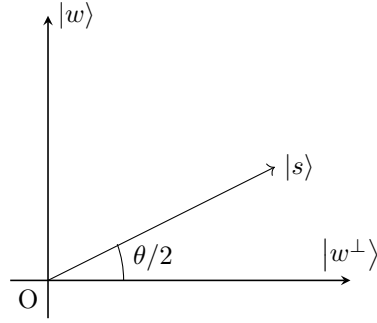


図4 $|w\rangle$ と $|w^\perp\rangle$ で張られる 2 次元空間内における、すべての状態の重ね合わせ状態 $|s\rangle$

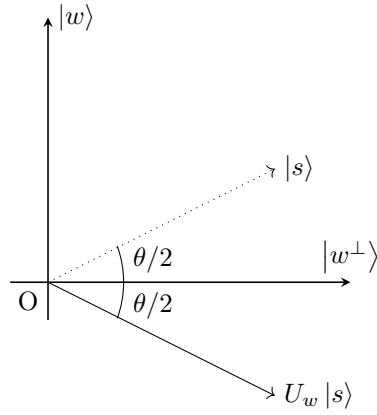


図5 $|w\rangle$ と $|w^\perp\rangle$ で張られる 2 次元空間内における、 $|s\rangle$ に対する U_w の作用

以上より、 $U_s U_w$ を作用させることで、 $|w\rangle, |w^\perp\rangle$ によって張られる 2 次元空間内で角度 θ だけ回転が行われる。これを繰り返すことで $|w\rangle$ の成分が大きくなるため、解が見つかる確率が上昇することがわかる [3] [4]。

2.4.4 最適な k の見積もり

最後に、 $U_s U_w$ を作用させる回数 k の最適な値 k_{theory} を見積もる。式 (2.38) より、 $U_s U_w$ を 1 回作用させると、

$$U_s U_w |s\rangle = \cos\left(\frac{3}{2}\theta\right) |w^\perp\rangle + \sin\left(\frac{3}{2}\theta\right) |w\rangle \quad (2.39)$$

となる。これを k 回作用させると、

$$(U_s U_w)^k |s\rangle = \cos\left(\frac{2k+1}{2}\theta\right) |w^\perp\rangle + \sin\left(\frac{2k+1}{2}\theta\right) |w\rangle \quad (2.40)$$

となる。これを用いると、 $|w\rangle$ が見つかる確率が 1 になる条件は以下のように書ける。

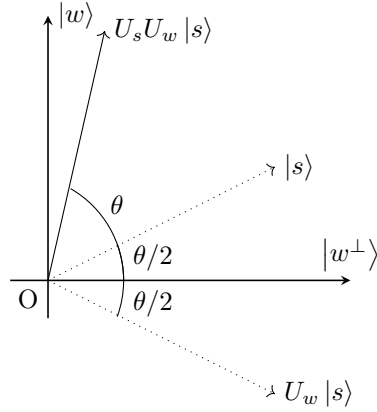


図6 $|w\rangle$ と $|w^\perp\rangle$ で張られる2次元空間内における、 $U_w |s\rangle$ に対する U_s の作用

$$\begin{aligned}\sin\left(\frac{2k+1}{2}\theta\right) &= 1 \\ \Leftrightarrow 2k+1 &= \frac{\pi}{\theta} \\ k &= \frac{\pi}{2\theta} - \frac{1}{2}\end{aligned}\tag{2.41}$$

よって、 $(U_s U_w)^k |s\rangle$ が $|w\rangle$ に最も近づくのは、 k が

$$k_{\text{theory}} \equiv \text{ClosestInteger}\left(\frac{\pi}{2\theta} - \frac{1}{2}\right)\tag{2.42}$$

の時である。ここで、 $\text{ClosestInteger}(x)$ は x に最も近い整数を表す。

最後に、計算量を見積もるため k_{theory} の上限を評価する。 $\theta > 0$ について成り立つ式、

$$\frac{\theta}{2} > \sin \frac{\theta}{2} = \frac{1}{\sqrt{N}}\tag{2.43}$$

を使うと、

$$\begin{aligned}\frac{\theta}{2} &> \frac{1}{\sqrt{N}} \\ \Leftrightarrow \frac{2}{\theta} &< \sqrt{N} \\ \Leftrightarrow \frac{\pi}{2\theta} &< \frac{\pi}{4}\sqrt{N}\end{aligned}\tag{2.44}$$

と変形することができるので、以下のように表すことができる。

$$k_{\text{theory}} < \left(\frac{\pi}{2\theta} - \frac{1}{2}\right) + 1 = \frac{\pi}{2\theta} + \frac{1}{2} < \frac{\pi}{4}\sqrt{N} + \frac{1}{2}\tag{2.45}$$

つまり、 k_{theory} は $O(\sqrt{N})$ である。これにより、グローバーのアルゴリズムの計算量が $O(\sqrt{N})$ であることがわかる [5]。

3 本論

本章では本研究の本題である、初期状態にずれがある場合のグローバーのアルゴリズムの振る舞いについて数値的に計算した結果を示す。数値計算では、Python を用いた量子コンピュータのシミュレータである Qulacs[6] を使用した。

3.1 回転演算子の組み合わせとしてのアダマール演算子

ここでは回転演算子の組み合わせでアダマール演算子を表す方法を示す。グローバーのアルゴリズムで初期状態を用意するために使用したアダマール演算子 H は、式 (2.18) を用いて回転演算子の積に分解すると、

$$\begin{aligned} H &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ &= \begin{pmatrix} \cos \frac{\pi}{4} & -e^{i\pi} \sin \frac{\pi}{4} \\ e^{i0} \sin \frac{\pi}{4} & e^{i\pi} \cos \frac{\pi}{4} \end{pmatrix} \\ &= U \left(\frac{1}{2}\pi, 0, \pi \right) \end{aligned} \quad (3.1)$$

と表すことができる。

本研究では、このアダマール演算子 H に y 軸方向のずれ δ_y , z 軸方向のずれ δ_z が生じている場合を考え、

- y 軸方向にずれ δ_y がある場合 [式 (3.2)]
- z 軸方向にずれ δ_z がある場合 [式 (3.4)]
- y 軸方向、 z 軸方向にずれ δ_y, δ_z がある場合 [式 (3.7)]

の 3 つの場合を扱った。

3.2 y 軸方向にずれ δ_y がある場合

まず、アダマール演算子 H の y 軸周りにずれ δ_y の値にとった時のアルゴリズムを量子コンピュータのシミュレータ Qulacs を用いて実装した。このシミュレーションでは、量子ビット数を n とする。ずれ δ_y は $\frac{\pi}{12}$ の間隔で 0 から 2π までの値にとった。よって、 δ_y を加えたアダマール演算子 H は式 (2.18) を用いて、

$$U \left(\frac{1}{2}\pi + \delta_y, 0, \pi \right) = \begin{pmatrix} \cos \left(\frac{\pi}{4} + \frac{\delta_y}{2} \right) & -e^{i\pi} \sin \left(\frac{\pi}{4} + \frac{\delta_y}{2} \right) \\ e^{i0} \sin \left(\frac{\pi}{4} + \frac{\delta_y}{2} \right) & e^{i\pi} \cos \left(\frac{\pi}{4} + \frac{\delta_y}{2} \right) \end{pmatrix} \quad (3.2)$$

と表すことができる。ずれ δ_y ごとに解が見つかる確率を計算基底で測定し、最適な U_s, U_w を作用させる回数 k 、理想的なグローバーのアルゴリズムでの最適な U_s, U_w を作用させる回数 k_{theory} [式 (2.45)] の時の解が見つかる確率の δ_y 依存性を数値計算で求めた。

図 7 は $(k/k_{\text{theory}}) - 1$ の δ_y 依存性を表している。量子ビット数は $n = 3, 5, 10$ の場合を表しており、これらの結果を重ねてプロットしている。図 7 を見るとわかるように、 $0 \leq \delta_y \leq \pi$ の範囲では $(k/k_{\text{theory}}) - 1$ は 0 から離れた値を取っているが、 $\delta_y = \frac{5\pi}{4}$ 付近で $(k/k_{\text{theory}}) - 1$ が 0 になる。つまり、 $\frac{5\pi}{4} < \delta_y \leq 2\pi$ の領域は、今考えているずれを含む初期状態に対するグローバーのアルゴリズムの最適な回数 k は理想的な回数 k_{theory} と同じになる。このことから、 $\frac{5\pi}{4} < \delta_y \leq 2\pi$ 間の δ_y の値にとったグローバーのアルゴリズムの最適な回数 k は理想的な回数 k_{theory} と同じになる。

図 8 はグローバーのアルゴリズムの解が見つかる確率の δ_y 依存性を表している。量子ビット数は $n = 3, 5, 10$ の場合を表しており、これらの結果を重ねてプロットしている。図 8 を見ると、解が見つかる確率は $\delta_y = \frac{\pi}{2}$ でほぼ 1 となり、その後減少し $\delta_y = \frac{5}{4}\pi$ で極小となり極小値は 0.14 となった。この値は解の状態の確率がほとんど増加しないことを示している。

図 8 において、 $\delta_y = \frac{\pi}{2}$ において解が得られる確率が 1 となっているが、これは以下のように説明できる。 $\delta_y = \frac{\pi}{2}$ のとき式 (3.2) より、

$$\begin{aligned} U\left(\frac{1}{2}\pi + \frac{1}{2}\pi, 0, \pi\right) &= \begin{pmatrix} \cos \frac{\pi}{2} & -e^{i\pi} \sin \frac{\pi}{2} \\ e^{i0} \sin \frac{\pi}{2} & e^{i\pi} \cos \frac{\pi}{2} \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{aligned} \quad (3.3)$$

となり、パウリ X に等しい。つまり、ずれ $\delta_y = \frac{\pi}{2}$ の値にとったとき、アダマール演算子はパウリ X と同じ作用を持つ。よって、解が見つかる確率が $\delta_y = \frac{\pi}{2}$ で 1.0 となる理由は、数値計算で使用したグローバーのアルゴリズムの解を $|11 \cdots 11\rangle$ と設定しているからであり、確率が 1.0 となるのは偶然である。これは図 8 が示すように、より n が大きい場合でも同様である。

図 9 は理想的な繰り返し回数 k_{theory} の解が見つかる確率の δ_y 依存性を表している。量子ビット数は $n = 3, 5, 10$ の場合を表しており、これらの結果を重ねてプロットしている。図 9 より、 δ_y を加えると、解が見つかる確率が下がっていき、 $\delta_y = \frac{\pi}{2}$ で解が見つかる確率がほぼ 0 になる。そこから、 $\delta_y = \frac{5}{4}\pi$ 程度まで解が見つかる確率は小さいままであり、グローバーのアルゴリズムが機能していない。さらに、 δ_y を大きくすると、 $\delta_y = 2\pi$ に向けて解が見つかる確率が大きくなっていく。しかし、サイズが大きくなるにつれて、グローバーのアルゴリズムが機能しない範囲が増えている。そのため、大きな n に対してグローバーのアルゴリズムを機能させるためには δ_y を十分に小さい値にとることが重要だと考えられる。

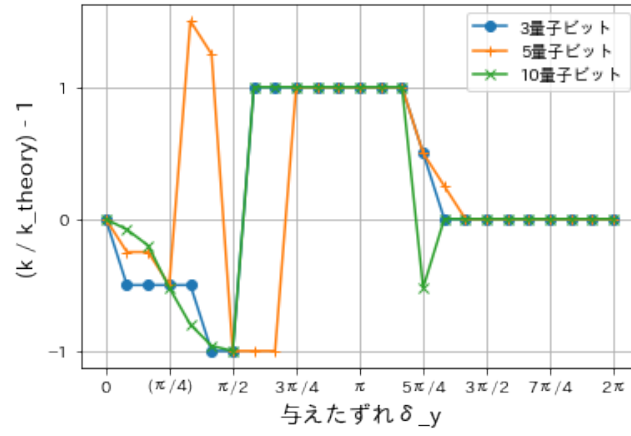


図7 $n = 3, 5, 10$ のときの、 $(k/k_{\text{theory}}) - 1$ の δ_y 依存性

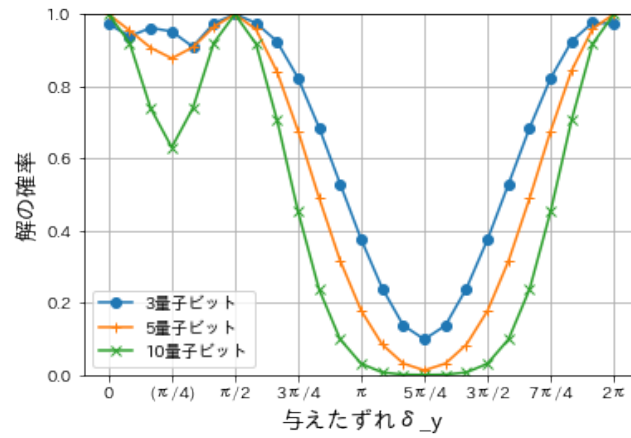


図8 $n = 3, 5, 10$ のときの、グローバラーのアルゴリズムの解が見つかる確率の δ_y 依存性

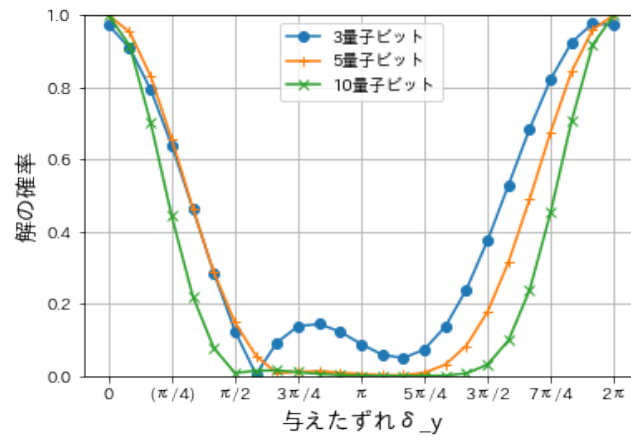


図9 $n = 3, 5, 10$ のときの、理想的な繰返し回数 k_{theory} のときの解が見つかる確率と δ_y 依存性

3.3 z 軸方向にずれ δ_z がある場合

次にアダマール演算子 H の z 軸周りにずれ δ_z を加えて数値計算を行った。3.2 節と同様に、このシミュレーションでは、量子ビット数 n を固定し、ずれ δ_z は $\frac{\pi}{12}$ の間隔で 0 から 2π までの値にとった。よって、 δ_z を加えたアダマール演算子 H は式 (2.18) を用いて、

$$U\left(\frac{1}{2}\pi, 0, \pi + \delta_z\right) = \begin{pmatrix} \cos\left(\frac{\pi}{4}\right) & -e^{i(\pi+\delta_z)} \sin\left(\frac{\pi}{4}\right) \\ e^{i0} \sin\left(\frac{\pi}{4}\right) & e^{i(\pi+\delta_z)} \cos\left(\frac{\pi}{4}\right) \end{pmatrix} \quad (3.4)$$

と表すことができる。ずれ δ_z ごとに解が見つかる確率を計算基底で測定し、最適な U_s, U_w を作用させる回数 k 、理想的なグローバーのアルゴリズムでの最適な U_s, U_w を作用させる回数 k_{theory} [式 (2.45)] の時の解が見つかる確率の δ_z 依存性を数値計算で求めた。

図 10 は $(k/k_{\text{theory}}) - 1$ の δ_z 依存性を表している。量子ビットは $n = 3, 5, 10$ の場合を表しており、これらの結果を重ねてプロットしている。図 10 より、 δ_y を加えても、繰り返し回数 k と理想的な回数 k_{theory} との差は見られなかった。この結果については、後でその原因を議論する。

図 11 はグローバーの解が見つかる確率の δ_z 依存性を $n = 3, 5, 10$ に対してプロットしたものであるが、解が見つかる確率の極大値に変化は見られなかった。また、図 12 は理想的な繰り返し回数 k_{theory} の時に解が見つかる確率の δ_z 依存性を $n = 3, 5, 10$ の場合にプロットしたものであるが、こちらも δ_z 依存性は見られなかった。

図 10, 11, 12 で示したようにずれ δ_z は解の確率を変えない。その理由は式 (3.4) は、

$$\begin{aligned} U\left(\frac{1}{2}\pi, 0, \pi + \delta_z\right) &= \begin{pmatrix} \cos\left(\frac{\pi}{4}\right) & -e^{i(\pi+\delta_z)} \sin\left(\frac{\pi}{4}\right) \\ e^{i0} \sin\left(\frac{\pi}{4}\right) & e^{i(\pi+\delta_z)} \cos\left(\frac{\pi}{4}\right) \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -e^{i(\pi+\delta_z)} \\ 1 & e^{i(\pi+\delta_z)} \end{pmatrix} \end{aligned} \quad (3.5)$$

となることから、ここに $|0\rangle$ を作用させると、

$$\begin{aligned} U\left(\frac{1}{2}\pi, 0, \pi + \delta_z\right) |0\rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -e^{i(\pi+\delta_z)} \\ 1 & e^{i(\pi+\delta_z)} \end{pmatrix} |0\rangle \\ &= \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \\ &= H |0\rangle \end{aligned} \quad (3.6)$$

となるからである。つまり、ずれ δ_z をどのような値にとったとしても、ずれ δ_z のあるアダマール演算子を $|0\rangle$ に作用させると、ずれのないアダマール演算子 H を $|0\rangle$ に作用させた結果と同じになる。これがずれ δ_z が k や確率を変えない理由である。

3.4 y 軸方向、 z 軸方向にずれ δ_y, δ_z がある場合

次に、アダマール演算子 H の y, z 軸周りの回転に、ずれ δ_y, δ_z の両方を加えて数値計算を行った。3.2 節、3.3 節と同様に、このシミュレーションでは、量子ビット数 n を固定し、ずれ δ_y, δ_z は等しくとり、 $\frac{\pi}{2}$ の間隔で 0 から 2π までの値にとった。

よって、 δ_y, δ_z を加えたアダマール演算子 H は式 (2.18) を用いて、

$$U\left(\frac{1}{2}\pi + \delta_y, 0, \pi + \delta_z\right) = \begin{pmatrix} \cos\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) & -e^{i(\pi+\delta_z)} \sin\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) \\ e^{i0} \sin\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) & e^{i(\pi+\delta_z)} \cos\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) \end{pmatrix} \quad (3.7)$$

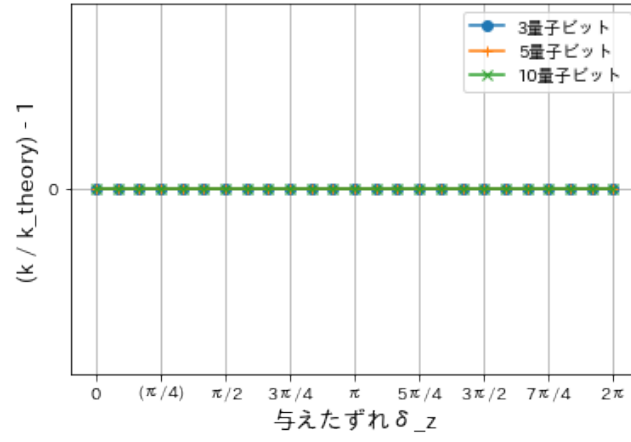


図 10 $n = 3, 5, 10$ のときの、 $(k/k_{\text{theory}}) - 1$ の δ_z 依存性

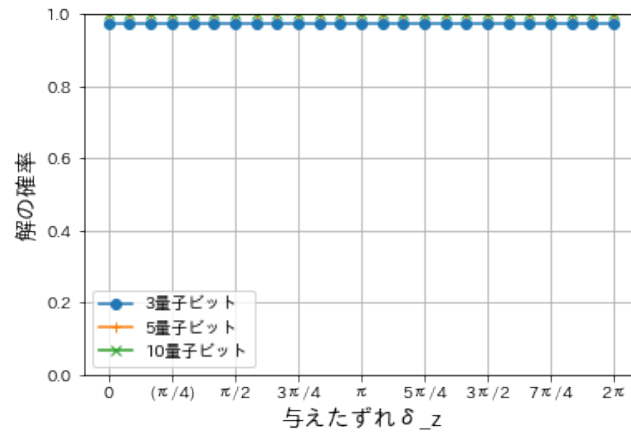


図 11 $n = 3, 5, 10$ のときの、グローバーのアルゴリズムの解が見つかる確率の δ_z 依存性

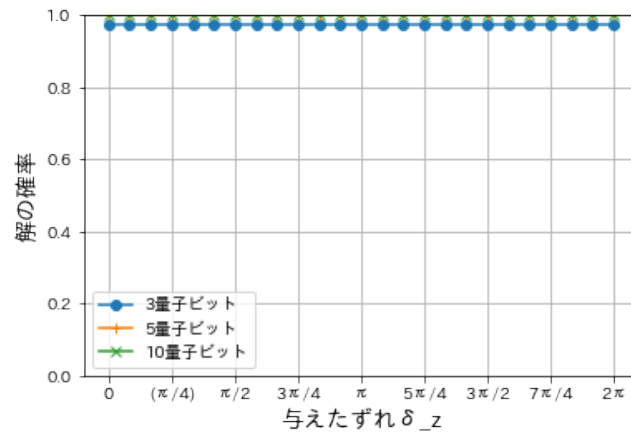


図 12 $n = 3, 5, 10$ のときの、理想的な繰り返し回数 k_{theory} のときの解が見つかる確率と δ_z 依存性

と表すことができる。ずれ δ_y, δ_z ごとに解が見つかる確率を計算基底で測定し、最適な U_s, U_w を作用させる回数 k 、理想的なグローバーのアルゴリズムでの最適な U_s, U_w を作用させる回数 k_{theory} [式 (2.45)] の時の解が見つかる確率の δ_y, δ_z 依存性を数値計算で求めた。

図 13, 14, 15 は 3.2 節と同様に $(k/k_{\text{theory}}) - 1$ 、グローバーの解が見つかる確率、理想的な繰り返し回数 k_{theory} の時の解が見つかる確率の δ_y, δ_z 依存性を表している。量子ビット数は $n = 3, 5, 10$ としており、 $\delta_y = \delta_z$ に対して計算を行った。しかし、図 13, 14, 15 は $\delta_z = 0$ の場合の結果 (図 10, 11, 12) と同様の結果となり、 δ_z に依存しない結果が得られた。

図 13, 14, 15 で示したようにずれ δ_y, δ_z を与えると、ずれ δ_y のみを与えた結果と同様の結果となる。その理由を考える。式 (3.2) を $|0\rangle$ に作用させると、

$$\begin{aligned} U\left(\frac{1}{2}\pi + \delta_y, 0, \pi\right) |0\rangle &= \begin{pmatrix} \cos\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) & -e^{i\pi} \sin\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) \\ e^{i0} \sin\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) & e^{i\pi} \cos\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) \end{pmatrix} |0\rangle \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} \cos\frac{\delta_y}{2} - \sin\frac{\delta_y}{2} & -e^{i\pi}(\cos\frac{\delta_y}{2} + \sin\frac{\delta_y}{2}) \\ \cos\frac{\delta_y}{2} + \sin\frac{\delta_y}{2} & e^{i\pi}(\cos\frac{\delta_y}{2} - \sin\frac{\delta_y}{2}) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} \cos\frac{\delta_y}{2} - \sin\frac{\delta_y}{2} \\ \cos\frac{\delta_y}{2} + \sin\frac{\delta_y}{2} \end{pmatrix} \end{aligned} \quad (3.8)$$

という状態になる。

一方、式 (3.7) を $|0\rangle$ に作用させると、

$$\begin{aligned} U\left(\frac{1}{2}\pi + \delta_y, 0, \pi + \delta_z\right) |0\rangle &= \begin{pmatrix} \cos\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) & -e^{i(\pi+\delta_z)} \sin\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) \\ e^{i0} \sin\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) & e^{i(\pi+\delta_z)} \cos\left(\frac{\pi}{4} + \frac{\delta_y}{2}\right) \end{pmatrix} |0\rangle \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} \cos\frac{\delta_y}{2} - \sin\frac{\delta_y}{2} & -e^{i(\pi+\delta_z)}(\cos\frac{\delta_y}{2} + \sin\frac{\delta_y}{2}) \\ \cos\frac{\delta_y}{2} + \sin\frac{\delta_y}{2} & e^{i(\pi+\delta_z)}(\cos\frac{\delta_y}{2} - \sin\frac{\delta_y}{2}) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} \cos\frac{\delta_y}{2} - \sin\frac{\delta_y}{2} \\ \cos\frac{\delta_y}{2} + \sin\frac{\delta_y}{2} \end{pmatrix} \end{aligned} \quad (3.9)$$

という状態を得る。よって、式 (3.2) を $|0\rangle$ に作用させた結果と式 (3.7) を $|0\rangle$ に作用させた結果は同じ値を取るため、ずれ δ_y, δ_z を与えても、ずれ δ_y のみを与えた結果と同様の結果となる。

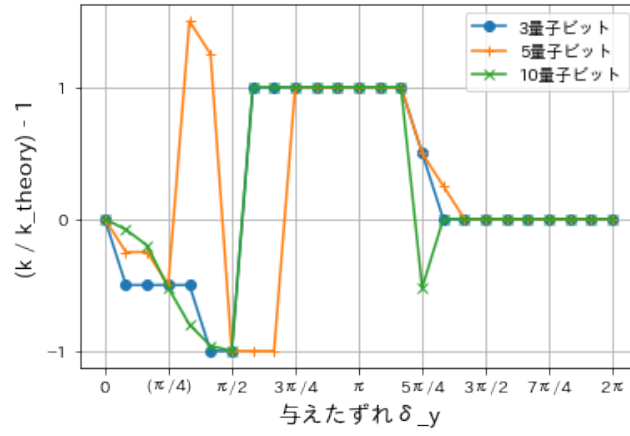


図 13 $n = 3, 5, 10$ のときの、 $(k/k_{\text{theory}}) - 1$ の δ_y, δ_z 依存性

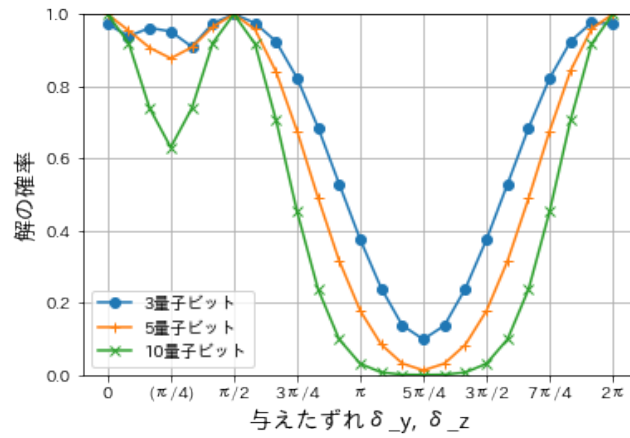


図 14 $n = 3, 5, 10$ のときの、グローバーのアルゴリズムの解が見つかる確率の δ_y, δ_z 依存性

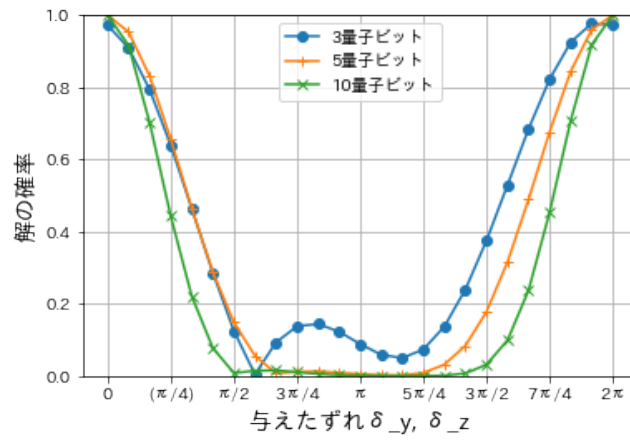


図 15 $n = 3, 5, 10$ のときの、理想的な繰り返し回数 k_{theory} のときの解が見つかる確率と δ_y, δ_z 依存性

4 まとめと結論

本研究では、初期状態を準備する操作にずれがある場合に、グローバーのアルゴリズムがどれだけ機能するかを、量子シミュレータ Qulacs[6] を用いて調べた。アダマール演算子は任意の回転行列に分解できるので、 y 軸方向のずれ、 z 軸方向のずれ、 y, z 軸方向のずれの 3 通りのずれを加えて数値計算を行った。

y 軸方向のずれ δ_y を加えると、解が見つかる確率が下がっていくが、 z 軸方向のずれ δ_z を加えても解が見つかる確率は変化しなかった。これは、作用させる空間の方向である $|0\rangle$ 基底の直交補空間の方向のずれ、つまり $|1\rangle$ 基底に z 軸方向のずれがあったとしても、アダマール演算子を作用させた結果は変わらないからである。

最後に、今後の展望について述べる。本研究では初期状態のずれとして y 軸方向のずれ、 z 軸方向のずれ、 y, z 軸方向のずれの 3 通りを行ったが、 y, z 軸方向のずれについては同時に δ_y と δ_z に同じ量を与えている。したがって、 δ_y と δ_z に異なる量を与えて、本研究で行なった数値計算を再度行う場合、違う結果になるかもしれない。さらに、本研究では初期状態を用意する際にアダマール演算子を $|0 \cdots 0\rangle$ に作用させ、解を $|1 \cdots 1\rangle$ として設定したが、もし $|1 \cdots 1\rangle$ に作用させた場合や、解をランダムに設定した場合なども今後調べていきたい。

謝辞

本論文を執筆するにあたり、多くの方々にご支援をいただきました。指導教員である伊與田英輝先生には、研究のテーマから研究を進める上で必要だった環境や資料まで多くのご指導をいただきましたこと、心から感謝申し上げます。

また、研究を支えてくださった家族や友人にも感謝の意を伝えたいと思います。ありがとうございました。

参考文献

- [1] 藤井啓祐, “量子コンピュータの基礎と物理の接点”, 物性研究・電子版 **6**, 064217 (2017).
- [2] I. L. Chuang, and M. A. Nielsen, “量子コンピュータと量子通信 I-量子力学とコンピュータ科学-”, (オーム社, 2004).
- [3] I. L. Chuang, and M. A. Nielsen, “量子コンピュータと量子通信 II-量子力学とコンピュータ科学-”, (オーム社, 2004).
- [4] 湊雄一郎, 比嘉恵一郎, 永井隆太郎, 加藤拓巳, “IBM Quantum で学ぶ量子コンピュータ” (秀和システム, 2021).
- [5] Qulacs, Quantum Native Dojo, <https://github.com/qulacs/quantum-native-doj>
- [6] Qulacs, Qulacs Document, <https://github.com/qulacs/qulacs>