

量子コンピューティングにおける量子ゲートの模擬と 運動方程式の時間発展

大阪工業大学 情報科学部 情報システム学科
B19110 吉田就

2023年2月14日

目次

1	序論	3
1.1	背景	3
1.2	本研究の目的	4
1.3	本論文の構成	4
2	量子コンピュータの動作原理について	5
2.1	量子ビット (qubit)	5
2.1.1	量子ビットについて (ブラケット記法)	5
2.1.2	量子ビットについて (ブロッホ球)	7
2.1.3	量子ビットの測定について	8
2.1.4	2量子ビットの状態について	8
2.2	量子ゲート	9
2.2.1	1ビット量子ゲートについて	9
2.2.2	複数ビットの量子ゲートについて	11
2.2.3	2量子ビットゲートについて	12
2.2.4	量子回路の書き方	14
2.3	量子回路の応用	15
2.3.1	基底エンコーディング	15
2.3.2	振幅エンコーディング	15
2.3.3	量子ランダムアクセスメモリ (QRAM) と QRAM を用いた振幅エンコーディング	16
2.3.4	量子離散的フーリエ変換 (QFT)	17
2.3.5	量子位相推定 (QPE)	20
3	シュレーディンガー方程式の時間発展による量子ゲートの模擬	21
3.1	シュレーディンガー方程式	21
3.2	本研究のシュレーディンガー方程式の解き方	23
3.2.1	考えるモデル	23
3.3	ゲート模擬	25
3.3.1	SWAP ゲート	25
3.3.2	アダマールゲート	27
4	Harrow-Hassidim-Lloyd (HHL) アルゴリズム	29
4.1	量子アルゴリズムにおける HHL アルゴリズム	29
4.2	本研究における HHL アルゴリズム	30
4.2.1	Newmark β 法	31
4.2.2	HHL アルゴリズムを用いた運動方程式の時間発展	32
5	まとめ	33

1 序論

1.1 背景

量子コンピュータというテーマは10年程前には一般的にほとんど知られていなかったが、ここ数年、爆発的に知られるテーマとなってきた。その背景には、従来のコンピュータにおいて、ムーアの法則が停滞し、新たなコンピューティング技術の必要性が高まってきたことや、量子コンピュータの類を見ない計算能力、アルゴリズムの進展がある。ムーアの法則が停滞した理由として(図1)、近年は半導体に含まれるトランジスタの数、及びクロック周波数の速度向上については、トランジスタによる量子効果が発生するほどの技術的な限界に近づいてしまったからである。現在の量子コンピュータは小規模で計算能力も限定的ではあるが、問題の分割、小さい量子プログラムと統計処理、最適化を組み合わせる演算するアルゴリズムなど、将来に向けての試作機的位置づけとして、賢く使う方法が探索されている。実用的な量子コンピュータの実現が近づいて来たことによって、将来私たちは量子コンピュータを日常的に利用する日が来るかもしれない。

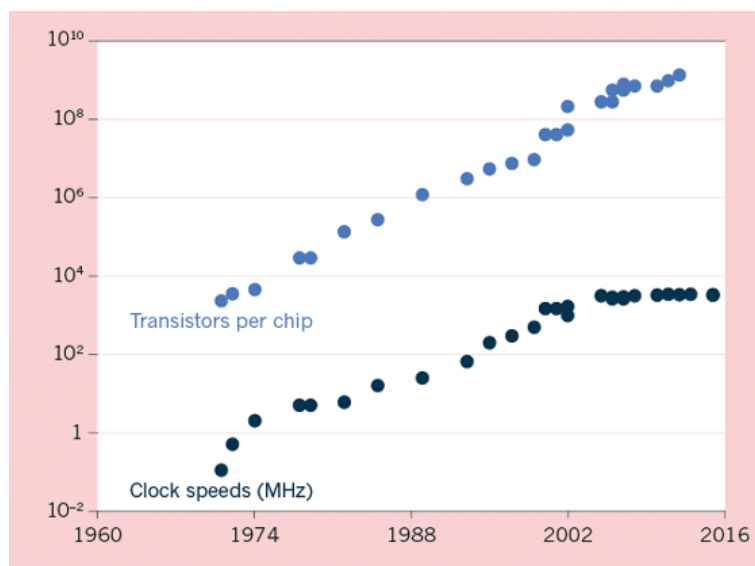


図 1: ムーアの法則 [1]

1.2 本研究の目的

本研究は量子コンピュータの動作原理の理解のためのシミュレートと、量子コンピューティングの応用例の計算を行った。どちらも現在の古典的なコンピュータを用いたものであるが、将来的な応用を念頭に置いたものである。まず、波動関数のふるまいをプログラミングを用いてシミュレートし、ポテンシャルを制御することで量子コンピュータの量子ゲートを2種類模擬した。次に、量子コンピューティングにおける1つのアルゴリズム (HHL アルゴリズム) を用いて運動方程式を解いた。

1.3 本論文の構成

第2章は量子コンピュータの動作原理について、解説する。必要な量子力学の知識とその数学的表現、最小単位である量子ビット、古典コンピュータの論理ゲートに値する量子ゲートについて、さらに量子回路の書き方とその応用について解説する。第3章では、シュレーディンガー方程式で量子ビットをシミュレートし、それぞれ量子ゲート2種類を模擬し、どのように模擬できたかを解説する。第4章では量子アルゴリズムの一つである HHL アルゴリズムを紹介し、それを用いて、古典コンピュータではねの運動方程式をシミュレートした。

2 量子コンピュータの動作原理について

本章では量子コンピュータを扱うにあたって、必要な基本的知識をまとめる。尚、文献 [4,5] を参照している。

2.1 量子ビット (qubit)

2.1.1 量子ビットについて (ブラケット記法)

まずは量子ビットからの説明をする。

通常のコンピュータが扱う情報の基本単位はビットとよばれ、0か1のどちらかのみであり、スイッチのオン・オフなどで表現される。一方量子コンピュータでは量子ビットが、計算の基本単位である。

構成される電子や光子などの素粒子によって、量子ビットは Newton 力学とは異なる力学、量子力学にしたがう。

素粒子のふるまいは、位置座標の値を x 、時間を t として、波動関数 $\psi(x, t)$ で記述される。波動関数を 2 乗したものは、素粒子の存在確率を表す。確率なので、規格化する。

$$\int_{-\infty}^{+\infty} |\psi(x, t)|^2 dx = 1 (\text{規格化条件}) \quad (1)$$

この波動関数が量子ビットに対応する。量子ビットは重ね合わせの原理より、複数の状態を同時に持つことができ、量子コンピュータは量子ビットの重ね合わされた高次元のベクトルを基本単位として演算することで、古典コンピュータよりも高速に計算が出来る。

波動関数 ψ はケット・ベクトル (状態ベクトル) と呼ばれる。列ベクトルで表記する事ができ、以下である。

$$|\psi\rangle = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$

例えば、 $|0\rangle, |1\rangle$ はそれぞれ列ベクトルとして

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ とすると}$$

状態ベクトルは状態を $|0\rangle, |1\rangle$ の 2 つの重ね合わせとして、2 つのケット・ベクトルの線形結合として、

$$c_1|0\rangle + c_2|1\rangle$$

で表される。 (c_1, c_2) は複素数

よって、2 つの状態のみの場合は、線形結合で表すと、 $|\psi\rangle = c_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = c_0|0\rangle + c_1|1\rangle$ となる。

上記のように、量子ビットの状態 $|\psi\rangle$ は古典ビット 0 に対応するベクトル $|0\rangle$ と 1 に対応する $|1\rangle$ を正規直交基底とする 2 次元の複素内積空間の単位ベクトルであり、大きさは

- $\langle \psi | \psi \rangle = (c_0^*, c_1^*) \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = |c_0|^2 + |c_1|^2 = 1$

であり、規格化条件より、全ての状態の係数が規格化されていなければならない。

2.1.2 量子ビットについて (ブロッホ球)

量子ビットの状態はブロッホ球と呼ばれる可視化手法も使用される。この手法は2つの状態ベクトル $|0\rangle$ と $|1\rangle$ と複素係数 c_0, c_1 で線形結合することは4つの実数の自由度を持つように思えるが、規格化条件により、実質的には3つである。これを単位球面上の点として表す手法である。複素数 c_0, c_1 を実数 ϕ, θ, γ を使って

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

と書き直す事が出来る。グローバル位相 γ を無視すれば、量子ビットの状態は図2のように緯度 θ ($-\pi \sim \pi$) と経度 ϕ ($0 \sim 2\pi$) で指定される球面上の1点として表す事が出来る。 $|\psi\rangle = |0\rangle$ の時は、 $\theta = 0, \phi = 0$ とした場合に相当する。また、 $|\psi\rangle = |1\rangle$ の時は、 $\theta = \pi, \phi = 0$ とした場合に相当する。基底ベクトル $|0\rangle, |1\rangle$ はZ軸上の点で表され、この正規直交基底の取り方はZ基底 (計算基底) と呼ばれる。単位球面上を表現する計算基底は他にもあり、X軸上の正規直交座標 $|+\rangle, |-\rangle$ も存在し、

$$|+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (3)$$

はアダマール基底 (X基底) と呼ばれる。

$$|+i\rangle := \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), |-i\rangle := \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \quad (4)$$

はY基底と呼ばれる。ブロッホ球を用いることで、量子ゲート (3.2節) を作用させる前後の状態を視覚的に捉えることが可能となる。

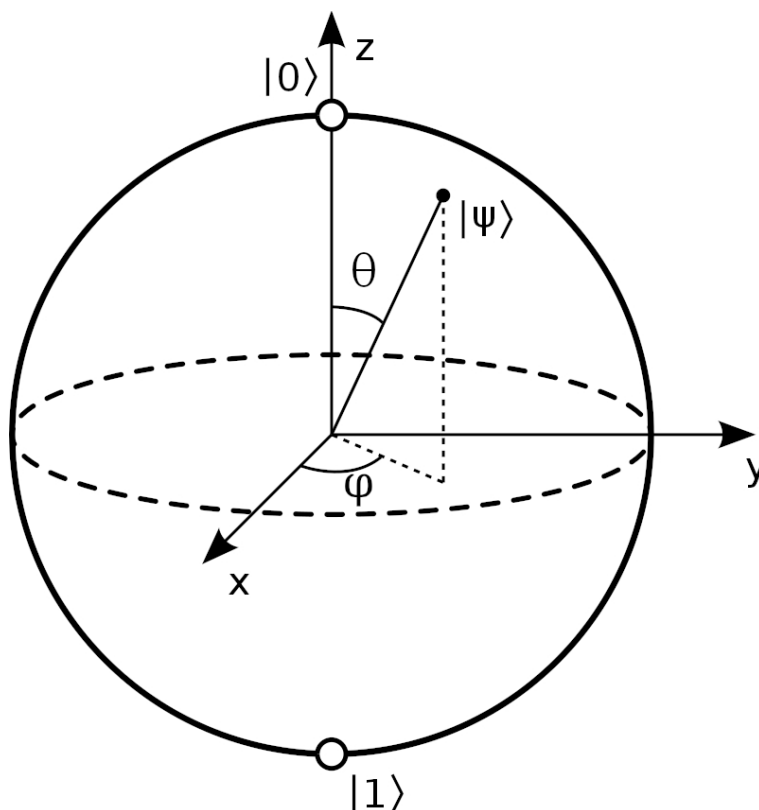


図 2: ブロッホ球 [2]

2.1.3 量子ビットの測定について

前節量子ビットの状態を表す2つの複素数 c_0, c_1 は規格化されていると説明した。規格化条件とは $|0\rangle$ と $|1\rangle$ のそれぞれの状態を観測する確率を定義し、発見する状態の全事象の確率の和が1でなければならないというを示している。 $(|0\rangle)$ の発見確率は $|c_0|^2$ 、 $(|1\rangle)$ の発見確率は $|c_1|^2$ この c_0, c_1 は確率振幅と呼ばれ、それぞれ $|c_0|^2, |c_1|^2$ が確率となる量である。確率とは具体的に、量子は量子力学より、重ね合わせ状態にあり、実際に観測することで状態が一つに決まる。そして、注意しなければならないのが、基底の取り方は $|0\rangle, |1\rangle$ 以外にも存在するから、どの基底で測定するのかということを決めておかなければならない。

2.1.4 2量子ビットの状態について

量子ビットには1ビットと1ビットの状態を複合させた状態を表現する事が出来る。原理的にはより多くのビットを混合させることが可能である。表記方法はテンソル積 (\otimes) を使うことで、複合された量子ビットを計算する事が出来る。例として、2つの量子ビット $(a_0|0\rangle_A + b_0|1\rangle_A)$ と $(a_1|0\rangle_B + b_1|1\rangle_B)$ の複合系は

$$(a_0|0\rangle_A + b_0|1\rangle_A) \otimes (a_1|0\rangle_B + b_1|1\rangle_B) := a_0a_1|00\rangle + a_0b_1|01\rangle + b_0a_1|10\rangle + b_0b_1|11\rangle \quad (5)$$

と書け、

$$|0\rangle \otimes |0\rangle = |0\rangle|0\rangle = |00\rangle \text{ である。} \quad (6)$$

列ベクトル表記では

$$\begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \otimes \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} = \begin{pmatrix} a_0a_1 \\ a_0b_1 \\ b_0a_1 \\ b_0b_1 \end{pmatrix} \quad (7)$$

と2次元ベクトル同士のテンソル積によって、要素が $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ の係数に対応する4次元ベクトルが得られる。

2.2 量子ゲート

2.2.1 1ビット量子ゲートについて

古典ビットにはANDゲート,ORゲート等存在し,量子ビットにもこれと同じような作用するゲートが存在する。すなわち、量子コンピュータは量子ビットに量子ゲート作用させることで、計算を行う。1量子ビットの演算に対しては、量子ビットの状態が2次元複素ベクトルで表すから、量子ゲート操作は 2×2 の複素行列である。

そして、 $|0\rangle$ と $|1\rangle$ の重ね合わせ状態である量子にゲートを作用させた後も、規格化条件は満たされていないなければならない。

量子ゲート作用を U を作用させると

$U \begin{pmatrix} a_0 \\ b_0 \end{pmatrix}$ と書けるから、 U^\dagger を U の随伴行列と定義し、規格化条件を課すと

$$(a_0^*, b_0^*) U^\dagger U \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} = 1$$

とならなければならない。

- 最も基本的なゲートであるパウリゲートは以下のような行列で表現できる。図3は後述の量子回路図におけるパウリゲートの表記方法である。

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (8)$$

$$\sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (9)$$

$$\sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (10)$$



図 3: 量子回路図におけるパウリゲート

実際に X ゲートを $|\psi\rangle = a_0|0\rangle + b_0|1\rangle$ に作用させると $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} = \begin{pmatrix} b_0 \\ a_0 \end{pmatrix}$ より

$X(|\psi\rangle) = b_0|0\rangle + a_0|1\rangle$ となり、状態 $|0\rangle$ と $|1\rangle$ の確率振幅を入れ替える事が出来る。

- 次にアダマールゲートは以下のように定義される。

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (11)$$

実際に $|0\rangle, |1\rangle$ に作用させると

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

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

となり、ブロッホ球上で基底変換を行うゲートである。図4は後述の量子回路図におけるアダマールゲートの表記方法である。



図 4: 量子回路図におけるアダマールゲート

- ブロッホ球上で、 X, Y, Z のそれぞれの軸で θ 回転させることに相当する量子ゲートは $R_j(\theta) = e^{(-i\theta\sigma_j/2)}$ である。

σ_j にはそれぞれパウリ行列が入る。($j = X, Y, Z$ のいずれか)

e の指数部に行列 (パウリゲート) が入っているが、行列 A に対する e^A とは以下を意味する

$$e^A = 1 + A + \frac{A^2}{2!} + \cdots + \frac{A^n}{n!} + \cdots$$

$$R_X(\theta) := e^{(i\theta\sigma_X/2)} = \begin{pmatrix} \cos \theta/2 & -i \sin \theta/2 \\ -i \sin \theta/2 & \cos \theta/2 \end{pmatrix} \quad (12)$$

$$R_Y(\theta) := e^{(i\theta\sigma_Y/2)} = \begin{pmatrix} \cos \theta/2 & -\sin \theta/2 \\ \sin \theta/2 & \cos \theta/2 \end{pmatrix} \quad (13)$$

$$R_Z(\theta) := e^{(i\theta\sigma_Z/2)} = \begin{pmatrix} e^{-\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \quad (14)$$

とそれぞれ求まる。

2.2.2 複数ビットの量子ゲートについて

複数量子ビットのゲート操作を表す行列のテンソル積は何番目の量子ビットに作用させるかということの表現に使用できる。

例えば、2量子ビットの系では1つ目の量子ビットに作用させるゲート U 、2つ目の量子ビットに作用させるゲート V とし、 $U \otimes V$ は

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \otimes \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} = \begin{pmatrix} a_1b_1 & a_1b_2 & a_2b_1 & a_2b_2 \\ a_1b_3 & a_1b_4 & a_2b_3 & a_2b_4 \\ a_3b_1 & a_3b_2 & a_4b_1 & a_4b_2 \\ a_3b_3 & a_3b_4 & a_4b_3 & a_4b_4 \end{pmatrix}$$

と 4×4 の行列で表す事が出来る。

2.2.3 2量子ビットゲートについて

- *CNOT* ゲート

2量子ビットゲートでは $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ の4つを基底とすると、制御 *NOT* ゲートは

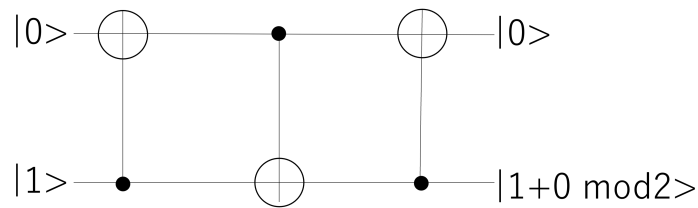
$$CNOT := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (15)$$

と書ける。このゲートは制御ビット、標的ビットを設定し、制御ビットが $|1\rangle$ の時に標的ビットの *NOT* をとるゲートである。

すなわち、

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ c_3 \\ c_2 \end{pmatrix}$$

となり、 $|1\rangle$ (制御ビット) $\otimes |0 \text{ or } 1\rangle$ 基底の確率振幅が入れ替わる。



制御ビット	$ 0\rangle \rightarrow 0\rangle$	$ 0\rangle \rightarrow 1\rangle$	$ 1\rangle \rightarrow 1\rangle$	$ 1\rangle \rightarrow 1\rangle$
標的ビット	$ 0\rangle \rightarrow 0\rangle$	$ 1\rangle \rightarrow 1\rangle$	$ 0\rangle \rightarrow 1\rangle$	$ 1\rangle \rightarrow 0\rangle$

図 5: CNOT ゲートと入出力の表

- SWAP ゲート

$$SWAP := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (16)$$

と書ける。このゲートはそれぞれの量子の確率振幅を入れ替えるゲートである。

すなわち、 $(a_0|0\rangle_A + a_1|1\rangle_A) \otimes (b_0|0\rangle_B + b_1|1\rangle_B)$ を SWAP ゲートに通すと

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_0b_0 \\ a_1b_0 \\ a_0b_1 \\ a_1b_1 \end{pmatrix} = \begin{pmatrix} a_0b_0 \\ a_0b_1 \\ a_1b_0 \\ a_1b_1 \end{pmatrix} = (b_0|0\rangle_A + b_1|1\rangle_A) \otimes (a_0|0\rangle_B + a_1|1\rangle_B) \text{ となる。}$$

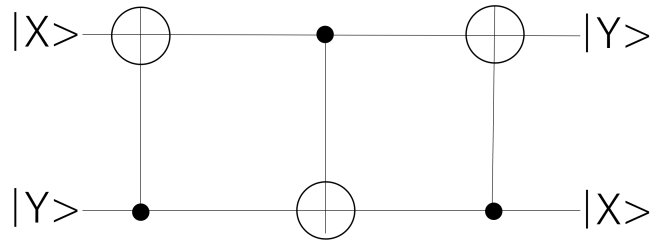


図 6: SWAP ゲート

2.2.4 量子回路の書き方

量子回路は左から右に読み、横線の1つ1つはそれぞれ1つの量子ビットを表している。

量子ビットはコピー不可のため、古典ビットのように分岐という処理は発生しない。図7の場合は3本横線が存在するため、3量子ビットに操作する量子回路である。左端の $|0\rangle$ は入力であり、最終的に右端の量子ゲートは量子ビットを測定するゲートである。量子回路については必ず入力と出力は1対1である。

図7の量子回路の過程を最初からたどると、
 $|000\rangle$

$$\xrightarrow{H_1, H_3} \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)|0\rangle\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)$$

$$\xrightarrow{CNOT_{1,2}} \frac{1}{2}(|000\rangle + |001\rangle + |110\rangle + |111\rangle)$$

$$\xrightarrow{CZ_{2,3}} \frac{1}{2}(|000\rangle + |001\rangle + |110\rangle - |111\rangle)$$

$$\xrightarrow{H_3} \frac{1}{2} \frac{1}{\sqrt{2}} (|00\rangle(|0\rangle + |1\rangle) + |00\rangle(|0\rangle - |1\rangle) + |11\rangle(|0\rangle + |1\rangle) - |11\rangle(|0\rangle - |1\rangle)) = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

となり、 $|000\rangle$ と $|111\rangle$ が50%の確率で現れる量子回路である。

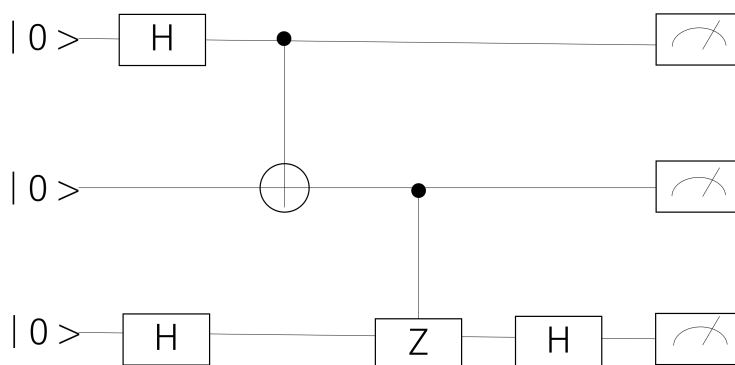


図7: 量子回路

2.3 量子回路の応用

この節より、古典コンピューティングにおける HHL アルゴリズムの理解のために必要な知識をまとめる。

2.3.1 基底エンコーディング

基底エンコーディングとは、古典データ n ビット列をそのまま量子ビットに置き換える。

例えば、実数 x を 4 桁のビット列で表すと

$|0\rangle := |0000\rangle$

$|1\rangle := |0001\rangle$

$|2\rangle := |0010\rangle$

$|3\rangle := |0011\rangle$

のようにエンコードする。これで多数の重ね合わせ状態を一度に計算出来ることが分かる。それぞれの確率振幅には情報が入力されていないため、計算の最終段階では欲しい結果の確率振幅を高め、間違っ
ていような答えを出力する確率を下げるという方法が必要である。

2.3.2 振幅エンコーディング

振幅エンコーディングとは古典データのエンコード先を確率振幅にエンコードする方法である。古典データからなる 2^n 次元のベクトル x を振幅エンコードすると

$$x = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{2^n-1} \end{pmatrix} \xrightarrow{\text{振幅エンコード}} \sum_{j=0}^{2^n-1} x_j |j\rangle \quad (17)$$

という量子状態に対応付けられる。ベクトル x は規格化されている。後述の HHL アルゴリズムでは振幅エンコーディングによって、古典データを量子コンピュータに入力することが不可欠である。

2.3.3 量子ランダムアクセスメモリ (QRAM) と QRAM を用いた振幅エンコーディング

量子ランダムアクセスメモリは、アドレスを表すレジスタもデータが出力されるレジスタも、共に量子ビットである。

あるバイナリデータ x_i のアドレス i に対応する量子ビット $|i\rangle_a$ と $|00 \cdots 0\rangle_r$ に初期化されたレジスタが与えられたとき、アドレス i に対応する x_i を量子ビット列 $|x_i\rangle_r$ として

$$|i\rangle_a |00 \cdots 0\rangle_r \xrightarrow{QRAM} |i\rangle_a |x_i\rangle_r$$

のように取り出す装置である。古典 RAM のような動作に加え、QRAM はアドレスとして重ね合わせ状態も受け付ける。

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} a_j |j\rangle_a$$

をアドレスとして入力すれば、重ね合わされた全てのアドレスに対応するデータがエンコードされた量子状態が

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} a_j |j\rangle_a |00 \cdots 0\rangle_r \xrightarrow{QRAM} \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} a_j |j\rangle_a |x_j\rangle_r \quad (18)$$

のように重ね合わせ状態として出力される。

さらに QRAM を使った振幅エンコーディングは、規格化された N 次元ベクトル v で表される古典データを振幅エンコーディングした

$$|v\rangle = \sum_i v_i |i\rangle$$

という状態を用意し、QRAM によって

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle_a |00 \cdots 0\rangle_r |0\rangle \xrightarrow{QRAM} \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} a_j |j\rangle_a |v_j\rangle_r |0\rangle$$

と基底エンコーディングで重ね合わせ状態のまま取り出す。そして、補助量子ビットの状態に応じた制御回転ゲート操作によって

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle_a |v_j\rangle_r (v_j |0\rangle + \sqrt{1 - v_j^2} |1\rangle)$$

という状態を作り、補助量子ビットを測定する。 $|0\rangle$ が測定されれば、状態は

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} v_j |j\rangle_a |v_j\rangle_r |0\rangle$$

となる。最後に QRAM の逆演算によって、 $|v_j\rangle_r$ を元に戻すと

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} v_j |j\rangle_a |00 \cdots 0\rangle_r |0\rangle$$

となり、古典データ v を振幅エンコーディングした値が得られる。これは 4.1 節の step3 で使用する。

2.3.4 量子離散的フーリエ変換 (QFT)

量子離散的フーリエ変換は後述の量子位相推定を行うために使用する。

離散的フーリエ変換

普通の離散的フーリエ変換は

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{i2\pi jk/N} x_j \quad (19)$$

である。より実践的には、 $N = 2^n$ 個の成分をもつ規格化された配列 $\{x_j\} (j = 0, \dots, 2^n - 1)$ に対して

$$y_k = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \exp\left(i\frac{2\pi jk}{2^n}\right) x_j \quad (20)$$

となる。配列を列ベクトル \mathbf{x} とすれば、この変換は行列として

$$\mathbf{y} = \frac{1}{\sqrt{2^n}} W \mathbf{x}, \quad W_{kj} = \left[\exp\left(i\frac{2\pi}{2^n}\right) \right]^{kj} \quad (21)$$

となる。

量子フーリエ変換 QFT

量子状態 $|x\rangle = \sum_{j=0}^{2^n-1} x_j |j\rangle$ から $|y\rangle = \sum_{k=0}^{2^n-1} y_k |k\rangle$ への変換と解釈する。(20) より

$$\begin{aligned} |y\rangle &= \sum_{k=0}^{2^n-1} \left[\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \exp\left(i\frac{2\pi jk}{2^n}\right) x_j \right] |k\rangle \\ &= \sum_{j=0}^{2^n-1} x_j \left[\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(i\frac{2\pi jk}{2^n}\right) |k\rangle \right] \end{aligned} \quad (22)$$

となるので、QFT は、正規直交基底ベクトルの変換として、

$$\begin{aligned} |j\rangle &\rightarrow |k\rangle \\ |j\rangle &\rightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(i\frac{2\pi jk}{2^n}\right) |k\rangle \end{aligned} \quad (23)$$

となる。

$N = 2$ のとき

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}} \sum_{k=0}^1 e^{i\frac{2\pi}{2} \cdot 0 \cdot k} |k\rangle = \frac{1}{\sqrt{2}} (e^0 |0\rangle + e^0 |1\rangle) = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \quad (24)$$

$$|1\rangle \rightarrow \frac{1}{\sqrt{2}} \sum_{k=0}^1 e^{i\frac{2\pi}{2} \cdot 1 \cdot k} |k\rangle = \frac{1}{\sqrt{2}} (e^0 |0\rangle + e^{i\pi} |1\rangle) = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \quad (25)$$

となるので、これはアダマールゲートである。

$N = 4$ のとき

$$|0\rangle \rightarrow \frac{1}{\sqrt{2^2}} \sum_{k=0}^3 e^{i\frac{2\pi}{4} \cdot 0 \cdot k} |k\rangle = \frac{1}{2} (e^0|0\rangle + e^0|1\rangle + e^0|2\rangle + e^0|3\rangle) = \frac{1}{2} (|0\rangle + |1\rangle + |2\rangle + |3\rangle)$$

$$|1\rangle \rightarrow \frac{1}{\sqrt{2^2}} \sum_{k=0}^3 e^{i\frac{2\pi}{4} \cdot 1 \cdot k} |k\rangle = \frac{1}{2} (e^0|0\rangle + e^{i\frac{2\pi}{4}}|1\rangle + e^{i\frac{4\pi}{4}}|2\rangle + e^{i\frac{6\pi}{4}}|3\rangle) = \frac{1}{2} (|0\rangle + i|1\rangle - |2\rangle - i|3\rangle)$$

$$|2\rangle \rightarrow \frac{1}{\sqrt{2^2}} \sum_{k=0}^3 e^{i\frac{2\pi}{4} \cdot 2 \cdot k} |k\rangle = \frac{1}{2} (e^0|0\rangle + e^{i\frac{4\pi}{4}}|1\rangle + e^{i\frac{8\pi}{4}}|2\rangle + e^{i\frac{12\pi}{4}}|3\rangle) = \frac{1}{2} (|0\rangle - |1\rangle + |2\rangle - |3\rangle)$$

$$|3\rangle \rightarrow \frac{1}{\sqrt{2^2}} \sum_{k=0}^3 e^{i\frac{2\pi}{4} \cdot 3 \cdot k} |k\rangle = \frac{1}{2} (e^0|0\rangle + e^{i\frac{6\pi}{4}}|1\rangle + e^{i\frac{12\pi}{4}}|2\rangle + e^{i\frac{18\pi}{4}}|3\rangle) = \frac{1}{2} (|0\rangle - i|1\rangle - |2\rangle + i|3\rangle)$$

となる。

2進数表記

10進数 k を 2進数表記する。

$$k = k_1 2^{n-1} + k_2 2^{n-2} + \cdots + k_n 2^0 \quad (26)$$

とすると, $k = (k_1 k_2 \cdots k_n)_{(2)}$ となる. $|k\rangle = |k_1 k_2 \cdots k_n\rangle = |k_1\rangle |k_2\rangle \cdots |k_n\rangle$ とする. また, 2進数の小数点以下も

$$\frac{k}{2^n} = \frac{k_1}{2^1} + \frac{k_2}{2^2} + \cdots + \frac{k_n}{2^n} \equiv (0.k_1 k_2 \cdots k_n)_{(2)} \quad (27)$$

と表記する. この表記は

$$(0.k_\ell k_{\ell+1} \cdots k_m)_{(2)} = \frac{k_\ell}{2^1} + \frac{k_{\ell+1}}{2^2} + \cdots + \frac{k_m}{2^{m-\ell+1}} \quad (28)$$

となる.

QFT

(23) は,

$$\begin{aligned} |j\rangle &\rightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(i2\pi j \frac{k}{2^n}\right) |k\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^1 \cdots \sum_{k_n=0}^1 \exp\left(i2\pi j (k_1 2^{-1} + k_2 2^{-2} + \cdots + k_n 2^{-n})\right) |k_1 k_2 \cdots k_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \left(\sum_{k_1=0}^1 \exp\left(i2\pi j k_1 2^{-1}\right) |k_1\rangle \right) \otimes \cdots \otimes \left(\sum_{k_n=0}^1 \exp\left(i2\pi j k_n 2^{-n}\right) |k_n\rangle \right) \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle + \exp\left(i2\pi \frac{j}{2^1}\right) |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + \exp\left(i2\pi \frac{j}{2^n}\right) |1\rangle \right) \end{aligned} \quad (29)$$

となる. $|j\rangle$ についても展開するが, $e^{i2\pi} = 1$ であることを用いると,

$$\begin{aligned} |j\rangle &= |j_1, j_2, \cdots, j_n\rangle \\ &= \frac{1}{\sqrt{2^n}} (|0\rangle + \exp(i2\pi(0.j_n)_{(2)}) |1\rangle) \otimes (|0\rangle + \exp(i2\pi(0.j_{n-1}j_n)_{(2)}) |1\rangle) \otimes \\ &\quad \cdots \otimes (|0\rangle + \exp(i2\pi(0.j_1 \cdots j_{n-1}j_n)_{(2)}) |1\rangle) \end{aligned} \quad (30)$$

(30) より, 制御回転ゲートとして,

$$R_m = \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^m} \end{pmatrix} \quad \text{すなわち} \quad \begin{cases} |0\rangle \rightarrow |0\rangle \\ |1\rangle \rightarrow e^{i2\pi/2^m} |1\rangle \end{cases} \quad (31)$$

とするものを定義すると, Hadamard ゲートと制御回転ゲートで量子フーリエ変換回路を構成することができる. R_m は回転なので, 操作する順によらない. $|j\rangle$ の展開順によっては, 量子ビットを最後に逆順にするために, スワップゲートを通す必要がある.

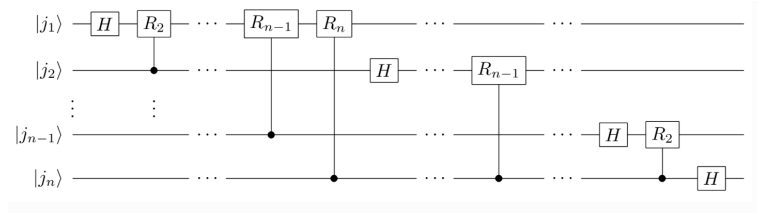


図 8: 量子フーリエ変換 (QFT) の量子回路

2.3.5 量子位相推定 (QPE)

量子位相推定は、ユニタリ行列の固有値を求めるアルゴリズム、多くの量子アルゴリズムの基礎になる重要なサブルーチンである。ユニタリー演算子 U に対する固有ベクトルを $|x\rangle$ 、固有値を $e^{i2\pi\phi}$ とすると、

$$U|x\rangle = e^{i2\pi\phi}|x\rangle \quad (32)$$

固有値を求めることを位相 ϕ を求めることに置き換えて考える。 $\phi \in [0, 1]$ であり、2進数表現は、 $0.\phi_1\phi_2\cdots\phi_n$ とする。

$$U^{2^k}|x\rangle = e^{i2\pi 2^k\phi}|x\rangle \quad (k = 0, 1, 2, \dots, n-1) \quad (33)$$

である。

- n 個の量子補助ビット $|0\rangle$ を加えて、 $|00\cdots 0\rangle \otimes |x\rangle$ を用意する。(解答用紙レジスター)
- 補助ビットに Hadamard ゲートを通して混合状態にする。
- n 個の補助ビットそれぞれに、 $U^{2^0}, U^{2^1}, \dots, U^{2^{(n-1)}}$ を作用させると、解答用紙レジスターは、量子フーリエ変換の出力状態になっている。つまり、

$$|\phi_1\phi_2\cdots\phi_n\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_n\rangle$$

をフーリエ変換したものになる。

- 逆量子フーリエ変換を行なって各レジスターの値を読み取ると $\phi_1\phi_2\cdots\phi_n$ が出力される。 2^n で割ると位相が計算されたことになる。

量子位相推定では任意の $|\psi\rangle$ はユニタリ行列 U の固有値 $e^{i\lambda_i}$ に対応する固有ベクトル $|\nu_i\rangle$ で

$$|\psi\rangle = \sum_i a_i |\lambda_i\rangle |\nu_i\rangle \quad (34)$$

と常に展開でき、量子位相推定サブルーチンは n 個の補助量子ビットを用いて

$$|00\cdots 0\rangle |\psi\rangle \xrightarrow{QRAM} \sum_i a_i |\lambda_i\rangle |\nu_i\rangle \quad (35)$$

と変換する。これにより、 $|\psi\rangle$ の重ね合わせの中にあるそれぞれの固有ベクトル $|\nu_i\rangle$ に対応した固有値を n 個の補助量子ビットに取り出せたから、最後に補助量子ビットを測定すると確率 $p_i = |a_i|^2$ で、いずれかの固有値ベクトル $|\nu_i\rangle$ とその固有値 λ_i が分かることになる。

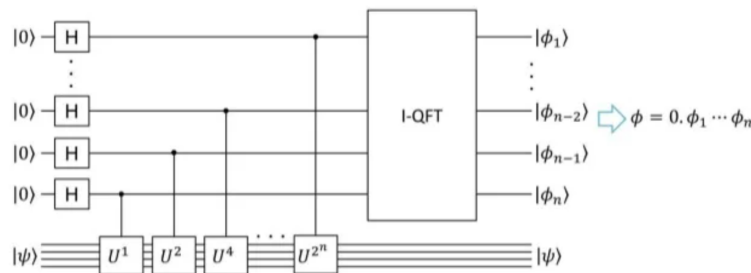


図 9: 量子位相推定 (QPE) の量子回路 [10]

3 シュレーディンガー方程式の時間発展による量子ゲートの模擬

この章では2章の波動関数をシュレーディンガー方程式を時間発展させることにより、波動関数のふるまいを可視化、制御する例を示す。そして、量子ゲートであるSWAPゲート、アダマールゲートの実装を行う。

波動関数の時間発展はシュレーディンガー方程式によって決まる。本章ではまずこのシュレーディンガー方程式について説明し、その後のプログラミングで実装するにあたって必要であるシュレーディンガー方程式の離散化を行う。その際にプログラミングで必要になる条件、各値の設定について述べる。

最後に、可視化された波動関数のふるまいをポテンシャルによって量子ゲートに似せたふるまいが実現できたので報告する。本章では文献 [8] を参照している。

3.1 シュレーディンガー方程式

シュレーディンガー方程式

波動関数 $\psi(x, t)$ の時間変化を決定する運動方程式をシュレーディンガー方程式とよび

$$i\hbar \frac{\partial}{\partial t} \psi(x, t) = H\psi(x, t) = \left[-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x, t) \right] \psi(x, t) \quad (36)$$

で表される。 H (ハミルトニアン)は運動エネルギーとポテンシャルエネルギー $v(x, t)$ の和であり、 \hbar はプランク定数 h から $\hbar = \frac{h}{2\pi}$ 、 m は量子の質量、 x は量子の位置座標、 t は時間である。

シュレーディンガー方程式の時間発展演算子

本章では、シュレーディンガー方程式を数値的に解くので、あらかじめシュレーディンガー方程式を離散化する。しばらく $\psi(x, t)$ を $\psi(t)$ と表記して時間発展の離散化について述べる、時間部分は

$$\frac{d\psi(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\psi(t + \Delta t) - \psi(t)}{\Delta t} \sim \frac{\psi(t + \Delta t) - \psi(t)}{\Delta t}$$

であり、シュレーディンガー方程式 (48) より、微小な時間 Δt に対して、波動関数の時間変化は

$$\psi(t + \Delta t) = \left(1 - \frac{iH\Delta t}{\hbar} \right) \psi(t) \quad (37)$$

と表す事が出来る (1次精度)。

ここで、波動関数の時間発展を表す演算子を $U(\Delta t)$ とおくと、(47) は

$$\psi(t + \Delta t) = U(\Delta t)\psi(t)$$

であり、

$$U(\Delta t) = \left(1 - \frac{iH\Delta t}{\hbar} \right)$$

と書ける。時間変化

$t + \Delta t$ を t' と書き換えて時間発展の演算子を

$$U(\Delta t) = U'(t', t)$$

とする。波動関数 ψ を $t_0 \rightarrow t = t_0 + \Delta t \rightarrow t' = t + \Delta t$ と時間発展させると

$$\psi(t') = U(t', t)\psi(t) = U(t', t)U(t, t_0)\psi(t_0)$$

$$= U(t', t_0)\psi(t_0)$$

より

$$U(t', t_0) = U(t', t)U(t, t_0)$$

と $U(t', t_0)$ は微小な時間発展の演算子の積として表される。そして

$$U(t', t_0) = \left(1 - \frac{iH\Delta t}{\hbar}\right)U(t, t_0)$$

から

$$\frac{U(t', t_0) - U(t, t_0)}{\Delta t} = \frac{iH}{\hbar}U(t, t_0)$$

となり、 Δt の極限で

$$\frac{1}{U(t', t_0)} \frac{dU(t, t_0)}{dt} = \frac{iH}{\hbar}$$

と表され、この両辺を積分すると

$$U(t, t_0) = e^{-\frac{i}{\hbar}H(t-t_0)} \quad (38)$$

が求められる。すなわち、シュレーディンガー方程式の時間発展は波動関数 $\psi(x_0, t_0)$ を

$$\psi(x, t) = U(t, t_0)\psi(x, t_0) = e^{-\frac{i}{\hbar}H(t-t_0)}\psi(x, t_0) \quad (39)$$

として書き表すことが出来る。

3.2 本研究のシュレーディンガー方程式の解き方

3.2.1 考えるモデル

前項のシュレーディンガー方程式を解く。この方程式をプログラムで解くことで、波動関数を求め、量子コンピュータにおける重ね合わせの原理の可視化、そして、ポテンシャル項の設定で量子の存在確率がどのように変化するかを検証し、量子ゲートをシミュレートする。以下に使用した条件を記載する。

- 2次精度

時間方向1次精度では精度が悪く、 Δt を小さくする必要がある。

そこで、波動関数 $\psi(x, t)$ と $\psi(x, t + \Delta t)$ から $\psi(x, t + 2\Delta t)$ を求め、時間方向2次精度で解く。簡略化し、順に $\psi(x, t_0), \psi(x, t), \psi(x, t')$ と表記する。

$$i\hbar \frac{\psi(x, t_2) - \psi(x, t_0)}{2\Delta t} = H\psi(x, t_1) \quad \text{より}$$
$$\psi(x, t_2) = \frac{2\Delta t H}{i\hbar} \psi(x, t_1) + \psi(x, t_0) = \frac{2\Delta t}{i\hbar} \left[-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right] \psi(x, t_1) + \psi(x, t_0)$$

となる（実際には空間微分を含めた右辺になる）。 $\psi(x, t_1)$ の2階微分項は位置方向2次精度で

$$\frac{\partial^2}{\partial x^2} \psi(x_n, t_1) = \frac{\psi(x_{n+1}, t_1) + \psi(x_{n-1}, t_1) - 2\psi(x_n, t_1)}{(\Delta x)^2} \quad (40)$$

として、解く。

そして、問題となるのは、初期値を設定して、その最初のステップである。 $\psi(x, t_0)$ から、まず $\psi(x, t_1)$ を求めないといけない。そこで、式(39)を使用して、 $\psi(x, t_1)$ を求める。

よって、

$$\psi(x, t_1) = \exp\left(-\frac{i}{\hbar} H \Delta t\right) \psi(x, t_0) \quad (41)$$

となることから、これを2次まで展開した

$$\psi(x, t_1) \sim \left[1 + \left(-\frac{i}{\hbar} H \Delta t\right) + \frac{1}{2} \left(-\frac{i}{\hbar} H \Delta t\right)^2 \right] \psi(x, t_0) \quad (42)$$

を用意して始める。

- クーラン条件

本研究はシュレーディンガー方程式を陽解法で解くので、安全な時間発展を行うために解析空間を離散化する $\Delta x, \Delta t$ が「クーラン条件」を満たす必要がある。クーラン条件を満たさない場合、解が発散してしまうことが知られている。離散化したモデルで波動を扱う場合はクーラン条件を常に満たす必要がある。

運動方程式の解を求める場合、ある時間ステップ Δt を用い、そのときの Δt は、実際の波動が隣の節点（あるいは格子）に到達するまでの時間よりも小さな時間ステップでなくてはならない。

式で表すと

$$\frac{\Delta x}{\Delta t} = C \quad (43)$$

C はクーラン数と呼ばれ、1 より小さくなる必要がある ($|C| < 1 \Delta x > \Delta t$)。

- 周期的境界条件

本研究の解析空間は $\Delta x = 0.1$ として、格子数は 100 である。 $x = [0, 10]$ の区間を解く。波動関数は波で、境界条件を決める必要があるから、周期的条件を使用する。

本研究の周期的境界条件では $x = 0$ の前の座標を $x = 9.9$ とし、 $x = 9.9$ の次の座標を $x = 0$ として扱う。

- 値の設定

プログラムでは、時間、位置等のシュレーディンガー方程式を解くにあたって必要な値は以下のよう設定する。 x_n は各格子に対応する ($n = 0 \sim 99$)

時間間隔 $\Delta t = 0.001$

座標 $\Delta x = 0.1$

(クーラン条件を満たす)

ディラック定数 $\hbar = 1.0$

質量 $m = 1.0$

波動関数の初期値 $\psi(x_n, 0) = (e^{-(x_n-4.0)^2} + e^{-(x_n+6.0)^2} + e^{-(x_n-14.0)^2}) / \sqrt{\int_0^{10} |\psi(x_n, 0)|^2 dx}$ $x_n = n * \Delta x$

波動関数の初期値はそのまま単体の波だけだと境界 ($x = 0, x = 9.9$) でノイズが入ってしまうから、境界外にも周期的に波があると想定した。

- ポテンシャルの設定

ポテンシャルを変更し、波動関数のふるまいを制御することで、量子ゲートを模擬するから、次項でポテンシャルの設定について適宜述べる。

3.3 ゲート模擬

量子を可視化した結果を図に示し、これからポテンシャルの設定によって、量子の確率振幅がどのように変わるのかを議論し、SWAP ゲート, アダマールゲートの模擬を行う。

3.3.1 SWAP ゲート

- ポテンシャルがないとき ($V(x)=0$)

ポテンシャル項が無い時の波動関数の確率推移は量子がポテンシャルに閉じ込められることはないため、初期において波動は $\pm x$ の 2 方向に進み始める。周期的境界条件を設定しているので、2 つに分かれた波は再び重ね合う様子を繰り返すことが分かる。図 10 に例を示す。確率振幅は量子がその位置に存在する確率である。

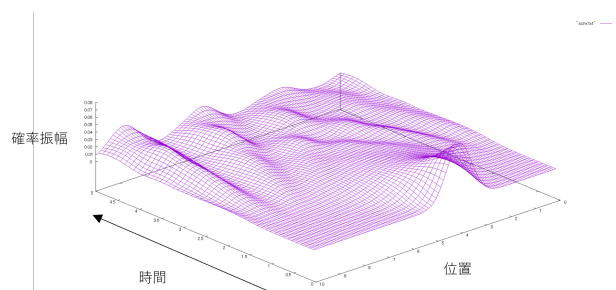


図 10: 波動関数の確率推移 ($V(x)=0$)

- $V(x)$ を変更しない時

2重井戸型ポテンシャルを図 12 のように設定すると、確率が $x = 4.0$ 付近から $x = 6.0$ 付近に推移し、またしばらくして元に戻るようなふるまいになる。これは量子状態を

$|0\rangle =$ 量子が 4.0 付近に存在する

$|1\rangle =$ 量子が 6.0 付近に存在する、と仮定するとポテンシャル内の量子状態 $|\psi\rangle$ は初期状態 $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$ から $|\psi\rangle = c_1|0\rangle + c_0|1\rangle$ と、確率振幅が $|0\rangle, |1\rangle$ で *SWAP* しているといえる。しかし、その後も再びに *SWAP* し、繰り返す。よって、1 回目の *SWAP* でポテンシャルを変更し、*SWAP* 状態を維持することを試みた

$$V(x) = -2.0e^{-(x-4.0)^2} - 2.0e^{-(x-6.0)^2} (0 \leq t < 20)$$

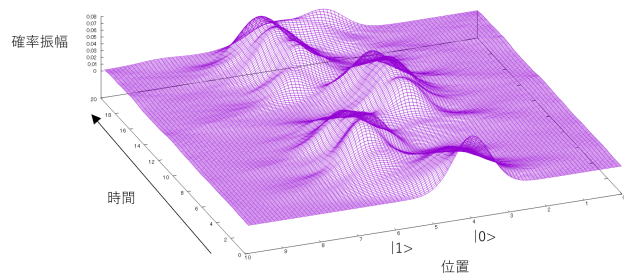


図 11: 波動関数の確率推移

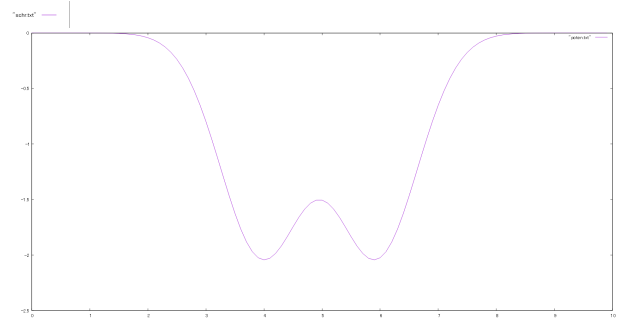


図 12: ポテンシャル

- $V(x)$ を途中で変更したとき

ポテンシャルは、 $t = 5.0$ で図 14(紫 変更前(図 12)、緑 変更後 $V(x) = -2.0e^{-(x-6.0)^2}$) より、ポテンシャルを変更している。実際には磁場などを用いて、量子の移動可能範囲を制限することに相当する。図 12 からポテンシャルを変更する事で、量子ビットは一回目のSWAP から再びSWAP する事はない。これはSWAP 回路を量子ビット $|\psi\rangle$ に作用させているといえる。今回は $|0\rangle$ 、 $|1\rangle$ の 2 状態からなる量子ビット $|\psi\rangle$ は 2 次元ベクトルで定義され

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}$$

である。

よって、ポテンシャルを変更する事で量子ゲート (SWAP)

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_0 \end{pmatrix} \text{ をシミュレートできた。}$$

$$V(x) = \begin{cases} -2.0e^{-(x-4.0)^2} - 2.0e^{-(x-6.0)^2} & (0 \leq t < 5.0) \\ -2.0e^{-(x-6.0)^2} & (5.0 \leq t \leq 20) \end{cases}$$

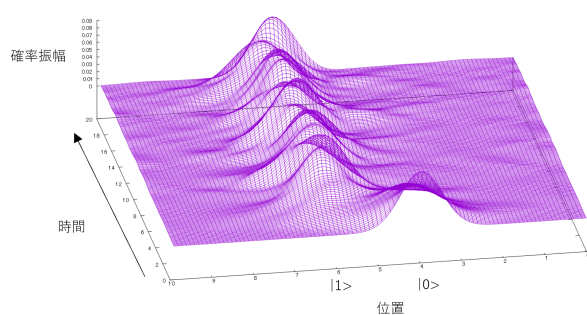


図 13: 波動関数の確率推移

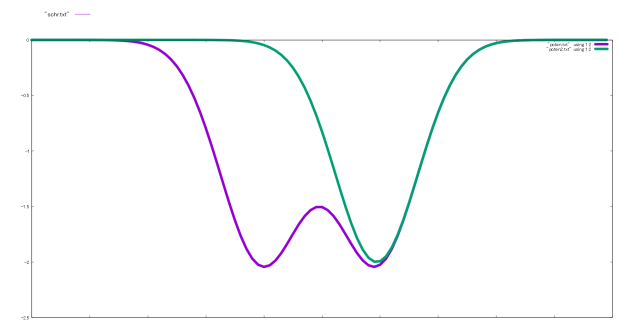


図 14: ポテンシャル変更比較

3.3.2 アダマールゲート

- $V(x)$ を変更しない時

ポテンシャルを図 16 のように設定すると、確率振幅は波のように広がり、 $|0\rangle, |1\rangle$ に量子を固定することが出来ない。そして、 $|0\rangle, |1\rangle$ の距離が近ければ、量子の波が片方に染み出してしまうため、前節の *SWAP* ゲートの場合とは別に出来るだけ $|0\rangle, |1\rangle$ の距離を遠く設定し、 $|0\rangle, |1\rangle$ の確率振幅が等しい時にポテンシャルを深く設定し、その状態を維持する必要がある。

$$V(x) = -0.1e^{-(x-2.5)^2} - 0.1e^{-(x-7.5)^2} (0 \leq t < 20)$$

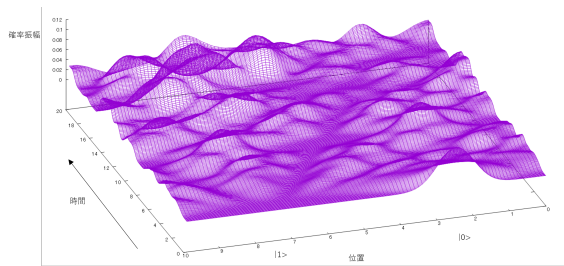


図 15: 波動関数の確率推移

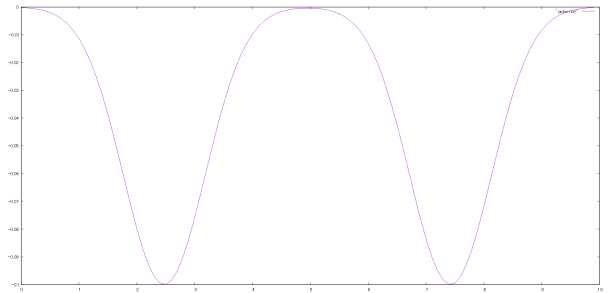


図 16: ポテンシャル

- $V(x)$ を途中で変更したとき

ポテンシャルは、 $t = 7.5$ で図 18(紫 変更前 (図 16)、緑 変更後 ($V(x) = -7.0e^{-(x-7.5)^2} - 7.0e^{-(x-2.5)^2}$)) より、ポテンシャルを変更している。図 16 からポテンシャルを変更する事で、量子ビットの $|0\rangle, |1\rangle$ 状態の確率振幅は等しくなる。これはアダマールゲートを量子ビット $|\psi\rangle = 1.0|0\rangle + 0|1\rangle$ に作用させているといえる。よって、ポテンシャルを変更する事で量子ゲート (アダマール)

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \text{ をシミュレートできた。}$$

$$V(x) = \begin{cases} -0.1e^{-(x-2.5)^2} - 0.1e^{-(x-7.5)^2} & (0 \leq t < 7.5) \\ -7.0e^{-(x-7.5)^2} - 7.0e^{-(x-2.5)^2} & (7.5 \leq t \leq 20) \end{cases}$$

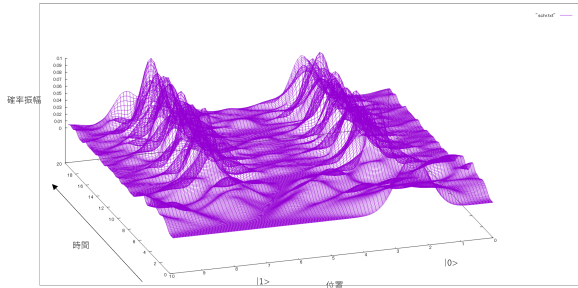


図 17: 波動関数の確率推移

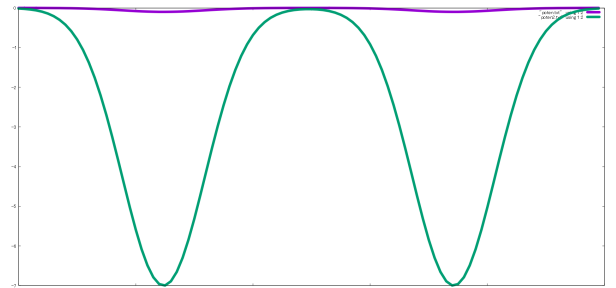


図 18: ポテンシャル変更比較

4 Harrow-Hassidim-Lloyd (HHL) アルゴリズム

Harrow-Hassidim-Lloyd (HHL) アルゴリズム [6] は Harrow-Hassidim-Lloyd によって提案された連立一次方程式を高速に「解く」アルゴリズムである。本研究では HHL アルゴリズムを用いて、古典的な計算機を使ってではあるが、3つのばね運動を解き、その有用性を確認した。本章ではまず量子コンピュータにおける HHL アルゴリズムについて説明し、次に実際に HHL アルゴリズムを用いて解いた方法を述べる。本章では文献 [3,4,6,7,9] を参照している。

4.1 量子アルゴリズムにおける HHL アルゴリズム

量子アルゴリズムにおける HHL アルゴリズムは逆行列を効率的に求める手法として、量子位相推定サブルーチンを使用する。このアルゴリズムは $N \times N$ の疎で正則な行列 A と N 次元ベクトル \mathbf{b} を用いて

$$Ax = \mathbf{b} \tag{44}$$

という線形方程式の解 \mathbf{x} を求めるアルゴリズムである。HHL アルゴリズムの計算ステップは以下の4つで構成される。

- step 1 \mathbf{b} を振幅エンコーディングした $|b\rangle := \sum_i b_i |i\rangle$ を用意する。
- step 2 制御 e^{iAt} を使う量子位相推定サブルーチンにより A の固有値 λ_i をレジスタ量子ビットに取り出し、 $\sum_i \beta_i |v_i\rangle$ を作る。 $(\beta_i$ は $|b\rangle$ を λ_i に 対応する固有ベクトル $|v_i\rangle$ で $|b\rangle = \sum_i \beta_i |v_i\rangle$ と展開した時の展開係数)
- step 3 2.3.3 項の方法と同様に、 $|\lambda\rangle$ を制御部とする制御回転ゲートを補助量子ビットに作用させ $\lambda^{-1}|0\rangle + \sqrt{1 - \lambda^{-2}}|1\rangle$ に変換する。
- step 4 補助量子ビットを測定し、逆演算によって不要な量子ビットを $|0\rangle$ に戻しておく。測定結果が 0 のとき、残りの量子ビットの状態は、 $\sum_i \beta_i \lambda_i^{-1} |v_i\rangle = |A^{-1}b\rangle$ が得られる。測定結果が $|1\rangle$ の時は step 1 に戻る。

この HHL アルゴリズムが量子計算に適していると言われる理由は、行列・ベクトルの計算と量子力学が持つ線形代数の構造は相性が良いからである。

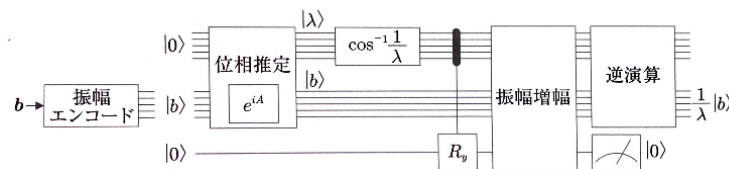


図 19: HHL アルゴリズムの量子回路

4.2 本研究における HHL アルゴリズム

本研究ではニュートン力学の運動方程式を解く問題に HHL アルゴリズムを適用した。2 階の常微分方程式として、ばねの運動をモデルとした。抵抗力があり、外力もある。式を次のように設定した。 M は質量、 $a = \frac{dv}{dt} = \frac{d^2x}{dt^2}$ は加速度、 $v = \frac{dx}{dt}$ は速度、 x 位置、 C は過衰係数、 k は剛性、 f は外力
ばね運動の方程式

$$Ma + Cv + kx = f \quad (45)$$

を異なるパラメータで三つ同時に解くことを考える。

そして、式 (37) を行列形式で式 (34) に落とし込んで、古典的に

$$\mathbf{x} = A^{-1}\mathbf{b} \quad (46)$$

として解く。これを量子計算として、 $|b\rangle$ から解状態 $|A^{-1}b\rangle$ を作ることに類推する。 A の固有値が λ_i のとき、 A^{-1} の固有値は $1/\lambda_i$ である。行列 A 、ベクトル \mathbf{b} 、 \mathbf{x} の設定は 3.3 節で述べる。

以下 (38) の解法部分についてまとめる。 A はエルミート行列であるとする。

step 1 A の固有値 λ_k と固有ベクトル w_k を求める。 $(A|w_k\rangle = \lambda_k|w_k\rangle)$ を解く.)

step 2 w_k は規格化する。固有値は逆数を計算しておく。

step 3 次の計算を行う。

$$\begin{aligned} |x\rangle &= A^{-1}|b\rangle = A^{-1}|b\rangle \sum_k \langle w_k|w_k\rangle \\ &= \sum_k \langle w_k|b\rangle A^{-1}|w_k\rangle = \sum_k \langle w_k|b\rangle \frac{1}{\lambda_k}|w_k\rangle \end{aligned} \quad (47)$$

式 (45) を行列形式に落とし込み、時間方向 2 次精度で解くために次に述べる Newmark β 法を用いる。

4.2.1 Newmark β 法

Newmark β 法 [7] は、微分方程式の時間発展の精度を2次以上で行うためにパラメータ β, γ を用いて差分方程式を書き直す定式化である。いま、速度の組 (v_n, x_n) を Δt の時間を進めて $((v_{n+1}, x_{n+1}))$ を求めることを考える。 t を時間として、 Δt 秒後の加速度を $a_{\Delta t}$ とすると

$$a_\gamma = (1 - \gamma)a_n + \gamma a_{n+1} \quad (48)$$

で速度は

$$v_{n+1} = v_n + a_\gamma \Delta t \quad (49)$$

より

$$v_{n+1} = v_n + (1 - \gamma)\Delta t a_n + \gamma \Delta t a_{n+1} \quad (50)$$

となる。再度

$$a_\gamma = (1 - 2\beta)a_n + 2\beta a_{n+1} \quad (51)$$

と

$$x_{n+1} = x_n + (v_n + a_\gamma \Delta t)\Delta t \quad (52)$$

から位置は

$$x_{n+1} = x_n + \Delta t v_n + \frac{\Delta t^2}{2}(1 - 2\beta)a_{n+1} + 2\beta a_n \quad (53)$$

となる。 β, γ の設定は最後 $\gamma = 0.5, \beta = 0.25$ と設定する。次に式 (45) を $M a_n, M a_{n+1}$ について解き、式 (50) の両辺に M をかけ、 $M a_n, M a_{n+1}$ に代入すると、加速度の項が消えた式になる。式 (53) についても同様に計算すると、最終的に式 (50) と式 (53) はそれぞれ速度 v_{n+1} と位置 x_{n+1} の2元1次方程式になるから、行列形式に変換すると

$$\begin{pmatrix} M + C\gamma\Delta t & K\gamma\Delta t \\ C\beta(\Delta t^2) & M + K\beta(\Delta t^2) \end{pmatrix} \begin{pmatrix} v_{n+1} \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} M - (1 - \gamma)\Delta t C & -(1 - \gamma)K\Delta t \\ M\Delta t - \frac{1}{2}(\Delta t)^2(1 - 2\beta C) & M - \frac{1}{2}(\Delta t)^2(1 - 2\beta K) \end{pmatrix} \begin{pmatrix} v_n \\ x_n \end{pmatrix} + \begin{pmatrix} (1 - \gamma)f_n + \gamma f_{n+1}\Delta t \\ (\frac{1}{2}(1 - 2\beta)f_n + \beta f_{n+1})(\Delta t^2) \end{pmatrix} \quad (54)$$

となる。この式は $\begin{pmatrix} v_n \\ x_n \end{pmatrix}$ から $\begin{pmatrix} v_{n+1} \\ x_{n+1} \end{pmatrix}$ へ時間発展させる行列形式になっている。本研究ではプログラム上では中心差分として $\gamma = 0.5, \beta = 0.25$ と設定した。

式 (54) の形で時間発展が書けることと、量子コンピューティングでは大規模行列やベクトルの取り扱いが優れていることから、本研究での式 (46) を複数のモデルや複数の初期条件を同時に解く方法へ応用した。

4.2.2 HHL アルゴリズムを用いた運動方程式の時間発展

本研究では外力 f を 0 とした M, C, K のそれぞれパラメータが異なる 過減衰 (添え字下 1)、臨界減衰 (添え字下 2)、減衰振動 (添え字下 3) について、HHL アルゴリズムを使用し、4.2.1 項でばね運動を行列形式に変換し、(46) 式を $(v_n^{(i)}, x_n^{(i)})$ から $(v_{n+1}^{(i)}, x_{n+1}^{(i)})$ への発展として次のように書く。ここで i はモデルの初期条件を変えたときの組を示すものとする。

$$M^{(i)} \begin{pmatrix} v_{n+1}^{(i)} \\ x_{n+1}^{(i)} \end{pmatrix} = A^{(i)} \begin{pmatrix} v_n^{(i)} \\ x_n^{(i)} \end{pmatrix} + B^{(i)} \quad (55)$$

$$M^{(i)} = \begin{pmatrix} M^{(i)} + C^{(i)}\gamma\Delta t & K^{(i)}\gamma\Delta t \\ C^{(i)}\beta(\Delta t^2) & M^{(i)} + K^{(i)}\beta(\Delta t^2) \end{pmatrix}$$

$$A^{(i)} = \begin{pmatrix} M^{(i)} - (1-\gamma)\Delta t C^{(i)} & -(1-\gamma)K^{(i)}\Delta t \\ M^{(i)}\Delta t - \frac{1}{2}(\Delta t^2)(1-2\beta C^{(i)}) & M^{(i)} - \frac{1}{2}(\Delta t^2)(1-2\beta K^{(i)}) \end{pmatrix}$$

$$B^{(i)} = \begin{pmatrix} (1-\gamma)f_n^{(i)} + \gamma f_{n+1}^{(i)}\Delta t \\ (\frac{1}{2}(1-2\beta)f_n^{(i)} + \beta f_{n+1}^{(i)})(\Delta t^2) \end{pmatrix}$$

ここで、 $i = 1, 2, 3$ の 3 つを同時に解くためには (本研究では外力は 0 とする)

$$\begin{pmatrix} M^{(1)} & 0 & 0 \\ 0 & M^{(2)} & 0 \\ 0 & 0 & M^{(3)} \end{pmatrix} \begin{pmatrix} v_{n+1}^{(1)} \\ x_{n+1}^{(1)} \\ v_{n+1}^{(2)} \\ x_{n+1}^{(2)} \\ v_{n+1}^{(3)} \\ x_{n+1}^{(3)} \end{pmatrix} = \begin{pmatrix} A^{(1)} & 0 & 0 \\ 0 & A^{(2)} & 0 \\ 0 & 0 & A^{(3)} \end{pmatrix} \begin{pmatrix} v_n^{(1)} \\ x_n^{(1)} \\ v_n^{(2)} \\ x_n^{(2)} \\ v_n^{(3)} \\ x_n^{(3)} \end{pmatrix} \quad (56)$$

式 (56) を式 (44) に対応させ、4.2 節の *step1-3* を実行し、初期値は全て $v(0) = 0, x(0) = 0.5$ に設定し、速度、位置をシミュレートした。図 21 (紫 過減衰 $M=1.0, C=4.0, k=6.0$, 緑 臨界減衰 $M=1.0, C=4.0, k=4.0$, 水色 減衰振動 $M=1.0, C=1.0, k=4.0$) ではパラメータの異なる三つのばね運動が時間方向 2 次精度でそれぞれ時間とともに減衰し、それぞれの一般解との誤差は小さい。

本研究のプログラミングでは 3 つのモデルが異なるばね運動を同時に解いた。さらに、モデルは同じで、初期条件のみを変えること (図 21) で同じ運動を同時に解く事ができ、同時に解く数を増やす場合は、プログラム上で、ベクトルを拡張するだけでよい。本研究では古典的に解いたが、式 (56) を量子コンピュータのデータにエンコードできれば同時に解くことが可能となる。

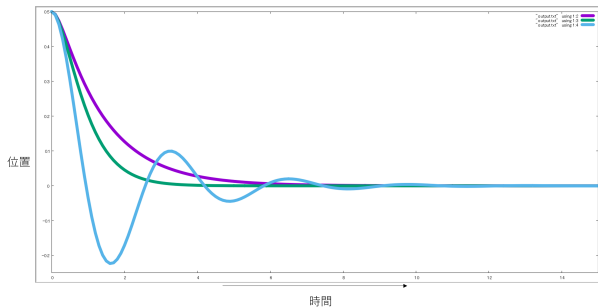


図 20: HHL アルゴリズムを用いた 3 つのばね運動の時間発展



図 21: 過減衰の初期条件のみを変えた例

5 まとめ

量子コンピュータの仕組みを理解し、応用例を示すため(古典コンピュータを用いて)2つのシミュレーションを行った。

[1] 波動関数のふるまい(量子の状態)をシュレーディンガー方程式を時間発展させて解いた。ポテンシャルを制御することで、量子ゲート(SWAPゲート、アダマールゲート)の模擬を行った。

[2] 常微分方程式(ばねの運動方程式)を2次精度で解いた。

HHL アルゴリズムに Newmark β 法を用いた。複数のモデルや初期条件を同時に解く実例を示した。

参考文献

- [1]N2i Journal、<https://blog.n2i.jp/entry/2017/06/09/113849>
- [2]Image of ブロツホ球、<https://japaneseclass.jp/img/ブロツホ球>
- [3]Interface、Python で体験！量子コンピュータ、CQ 出版株式会社、2022 年 6 月号
- [4] 嶋田義皓、量子コンピューティング、オーム社、2021
- [5] 宮野 健次郎・古沢 明、量子コンピュータ入門、日本評論社、2016
- [6] A.W. Harrow, A.Hassidim, S.Lloyd, Phys. Rev. Lett. 103(2009), 150502
- [7] N.M. Newmark. J.Eng.Mech.Div.85(1959),p67-94.
- [8] 佐川 弘幸・吉田 宣章、量子情報理論、シュプリンガー・ジャパン株式会社、2009
- [9] 真貝 寿明、徹底攻略常微分方程式、共立出版、2019
- [10] 中田 真秀、<https://www.slideshare.net/NakataMaho/ss-117321322>