Utility scale quantum computing
量子ユーティリティー授業

量子位相推定

Quantum Phase Estimation

2025/07/28

林 穂高

自己紹介

- **名前：林 穂高**
- **東京大学 工学部**（~2025-03）
  - **卒論：確率微分方程式の細胞モデルをシミュレーション**
- **日本**IBM **コンサルティング事業本部** (2025-04~)
- **量子：大学の講義やイベント**（Quantum Challenge 2024, QFF2024**など**）
  - QGSS2025**も参加しました！**

目次

目次

量子位相推定(Quantum Phase Estimation)の目的

量子位相推定（QPE）は、ユニタリ行列 U とその固有状態 $|\psi\rangle$ に対し、
固有値に紐づいた実数値 θ を求めるアルゴリズムである。

$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$$

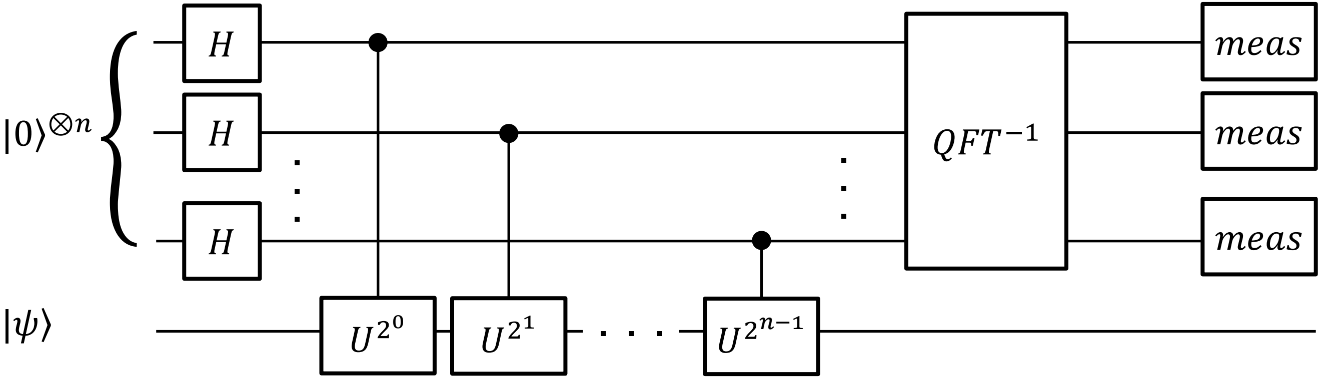
Uはユニタリ行列 → その固有値は絶対値1の複素数

応用

- ハミルトニアンのエネルギー固有値と固有状態の数値計算
- Shorのアルゴリズム：QPEと根本は同じ！

QPEの全体像

位相推定アルゴリズムは、$|\psi\rangle$で初期化したレジスタのほかにn量子ビットの計算レジスタを用意し、以下の量子回路として実装する。



1. 計算ビットすべてにアダマールゲートを施す。
2. 制御Uゲートを$|\psi\rangle$に施す。
3. 計算ビットに逆QFTを施す。
4. 計算ビットを測定する。

目次

量子フーリエ変換(Quantum Fourier Transformation)

量子フーリエ変換（QFT）は、バイナリ整数の計算基底を離散フーリエ変換のような形式へ変換する量子アルゴリズムである。

$$U_{QFT}|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j \frac{k}{2^n}} |k\rangle$$

n ビット整数 j の量子状態 $|j\rangle$ に対するQFT

（古典）離散フーリエ変換：$F(k) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} e^{2\pi i k \frac{x}{N}} f(x)$

Shorのアルゴリズムや QPEなど、主要なアルゴリズムの**重要なサブルーチン**
高速な実装は $O(n\log n)$ が知られている
- ➢ 古典は指数時間かかる → **量子計算で指数的加速**（ただし観測できない）

QFTのテンソル積による表現

n ビット整数 k を、バイナリ表現として $k = k_{n-1}k_{n-2} \ldots k_1 k_0\ _{(2)}$ と書くとき、$|j\rangle$ に対しQFTを実行した後の状態はテンソル積表現で書ける。

$$U_{QFT}|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j \frac{k}{2^n}} |k\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j \frac{2^{n-1}k_{n-1}+2^{n-2}k_{n-2}+\cdots+2k_1+k_0}{2^n}} |k_{n-1}k_{n-2} \ldots k_1 k_0\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \otimes_{l=0}^{n-1} \left(e^{2\pi i j \frac{k_l 2^l}{2^n}} |k_l\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}} \otimes_{l=0}^{n-1} \left(\sum_{k_l \in \{0,1\}} e^{2\pi i j \frac{k_l 2^l}{2^n}} |k_l\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i j \frac{2^{n-1}}{2^n}} |1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i j \frac{2^{n-2}}{2^n}} |1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i j \frac{2^0}{2^n}} |1\rangle\right)$$

QFTのテンソル積による表現

n ビット整数 k を、バイナリ表現として $k = k_{n-1}k_{n-2}\dots k_1k_0\ _{(2)}$ と書くとき、
$|j\rangle$ に対しQFTを実行した後の状態はテンソル積表現で書ける。

$$U_{QFT}|j\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi i j\frac{k}{2^n}}|k\rangle$$

k を2進数表現

$$= \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi i j\frac{2^{n-1}k_{n-1}+2^{n-2}k_{n-2}+\cdots+2k_1+k_0}{2^n}}|k_{n-1}k_{n-2}\dots k_1k_0\rangle$$

$$= \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} \otimes_{l=0}^{n-1} \left(e^{2\pi i j\frac{k_l 2^l}{2^n}}|k_l\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}} \otimes_{l=0}^{n-1} \left(\sum_{k_l\in\{0,1\}} e^{2\pi i j\frac{k_l 2^l}{2^n}}|k_l\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i j\frac{2^{n-1}}{2^n}}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i j\frac{2^{n-2}}{2^n}}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i j\frac{2^0}{2^n}}|1\rangle\right)$$

QFTのテンソル積による表現

n ビット整数 k を、バイナリ表現として $k = k_{n-1}k_{n-2}\dots k_1k_{0\ (2)}$ と書くとき、
$|j\rangle$ に対しQFTを実行した後の状態はテンソル積表現で書ける。

$$U_{QFT}|j\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi i j\frac{k}{2^n}}|k\rangle$$

$$= \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi i j\frac{2^{n-1}k_{n-1}+2^{n-2}k_{n-2}+\cdots+2k_1+k_0}{2^n}}|k_{n-1}k_{n-2}\dots k_1k_0\rangle$$

$$= \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}\bigotimes_{l=0}^{n-1}\left(e^{2\pi i j\frac{k_l 2^l}{2^n}}|k_l\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}}\bigotimes_{l=0}^{n-1}\left(\sum_{k_l\in\{0,1\}} e^{2\pi i j\frac{k_l 2^l}{2^n}}|k_l\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i j\frac{2^{n-1}}{2^n}}|1\rangle\right)\otimes\left(|0\rangle + e^{2\pi i j\frac{2^{n-2}}{2^n}}|1\rangle\right)\otimes\cdots\otimes\left(|0\rangle + e^{2\pi i j\frac{2^0}{2^n}}|1\rangle\right)$$

Σ 内の各項を
テンソル積で表現

QFTのテンソル積による表現

n ビット整数 k を、バイナリ表現として $k = k_{n-1}k_{n-2} \dots k_1 k_0\ _{(2)}$ と書くとき、$|j\rangle$ に対しQFTを実行した後の状態はテンソル積表現で書ける。

$$U_{QFT}|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j \frac{k}{2^n}} |k\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j \frac{2^{n-1}k_{n-1}+2^{n-2}k_{n-2}+\cdots+2k_1+k_0}{2^n}} |k_{n-1}k_{n-2} \dots k_1 k_0\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \bigotimes_{l=0}^{n-1} \left(e^{2\pi i j \frac{k_l 2^l}{2^n}} |k_l\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}} \bigotimes_{l=0}^{n-1} \left(\sum_{k_l \in \{0,1\}} e^{2\pi i j \frac{k_l 2^l}{2^n}} |k_l\rangle\right)$$

$k_l \in \{0,1\}$の和の積へ分解

$$= \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i j \frac{2^{n-1}}{2^n}} |1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i j \frac{2^{n-2}}{2^n}} |1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i j \frac{2^0}{2^n}} |1\rangle\right)$$

QFTのテンソル積による表現

n ビット整数 k を、バイナリ表現として $k = k_{n-1}k_{n-2} \dots k_1 k_0\ {}_{(2)}$ と書くとき、$|j\rangle$ に対しQFTを実行した後の状態はテンソル積表現で書ける。

$$U_{QFT}|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j \frac{k}{2^n}} |k\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j \frac{2^{n-1}k_{n-1}+2^{n-2}k_{n-2}+\cdots+2k_1+k_0}{2^n}} |k_{n-1}k_{n-2} \dots k_1 k_0\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \otimes_{l=0}^{n-1} \left(e^{2\pi i j \frac{k_l 2^l}{2^n}} |k_l\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}} \otimes_{l=0}^{n-1} \left(\sum_{k_l \in \{0,1\}} e^{2\pi i j \frac{k_l 2^l}{2^n}} |k_l\rangle\right)$$

Σと⊗を展開

$$= \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i j \frac{2^{n-1}}{2^n}} |1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i j \frac{2^{n-2}}{2^n}} |1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i j \frac{2^0}{2^n}} |1\rangle\right)$$

逆量子フーリエ変換

QFTの実装に対応するユニタリ行列 U_{QFT} に対して、共役転置をとった行列は U_{QFT} の逆行列となり、これに対応する演算を**逆量子フーリエ変換**という。

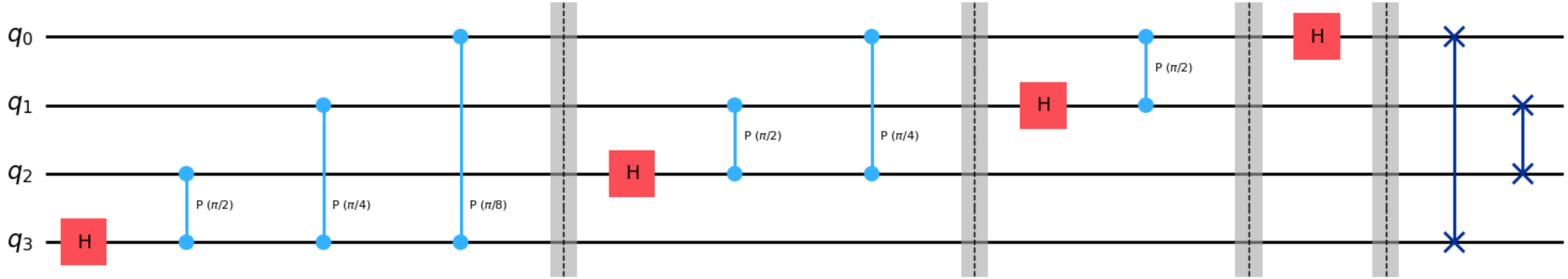
U_{QFT}はユニタリ　→　$U_{QFT}^{\dagger} = U_{QFT}^{-1}$: 逆QFT

$$U_{QFT}^{-1}(U_{QFT}|j\rangle) = U_{QFT}^{-1}U_{QFT}|j\rangle = |j\rangle$$

逆QFTはQFTの計算結果をもとに戻す

QFTの実装

参考に、4量子ビット QFTの量子回路を示す。



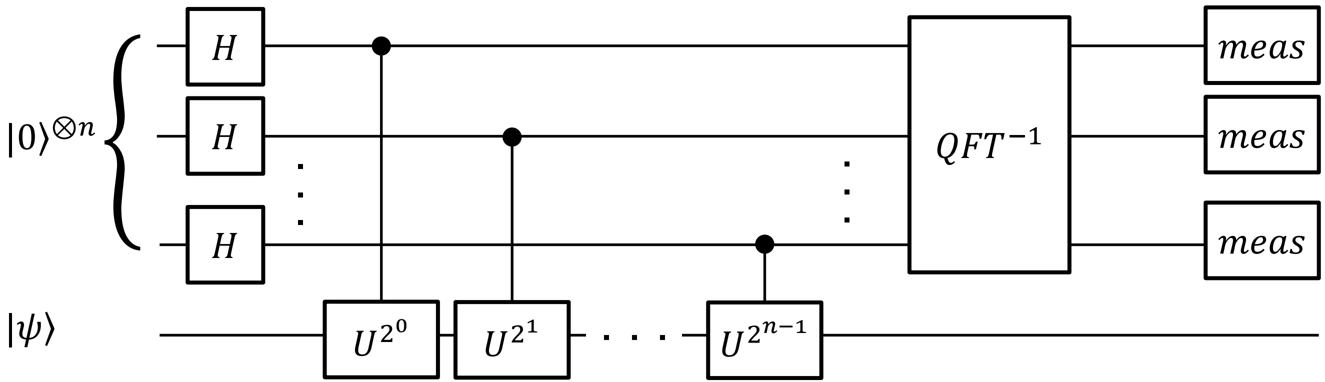
4量子ビット QFTの量子回路

n 量子ビットに対し、$O(n^2)$の実装

目次

位相推定アルゴリズムの実装

位相推定アルゴリズムは以下の手順で行われる。

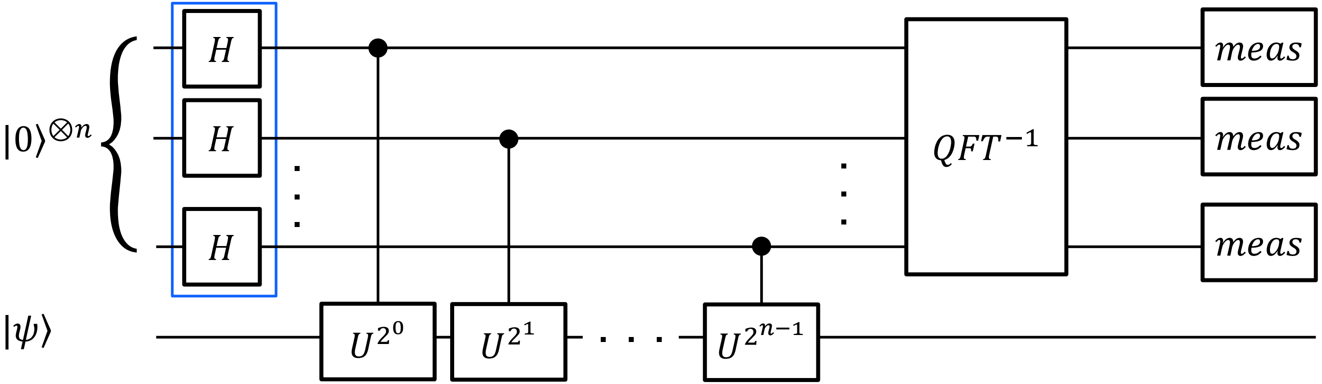


【手順】
1. 計算ビットすべてにアダマールゲートを施す。
2. 制御Uゲートを$|\psi\rangle$に施す。
3. 計算ビットに逆QFTを施す。
4. 計算ビットを測定する。

$$\boxed{U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle}$$

位相推定アルゴリズムの実装

1. 計算ビットすべてにアダマールゲートを施す。



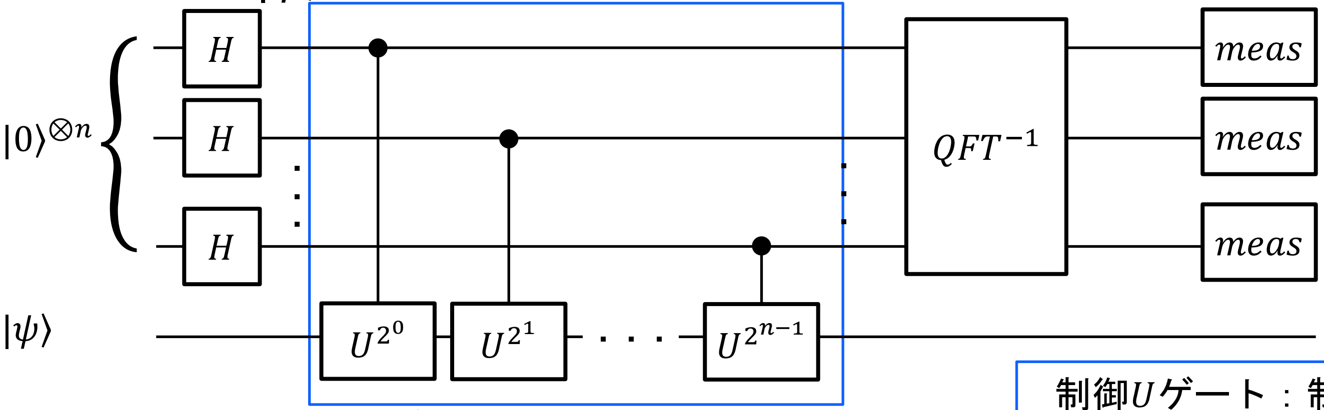
$$|\psi\rangle|0\rangle^{\otimes n}$$

$$\rightarrow\ |\psi\rangle(H|0\rangle)^{\otimes n} = |\psi\rangle\{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\}^{\otimes n}$$

$$= |\psi\rangle\{\frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)\otimes(|0\rangle + |1\rangle)\otimes \ldots \otimes(|0\rangle + |1\rangle)\}$$

位相推定アルゴリズムの実装

2. 制御Uゲートを$|\psi\rangle$に施す。



$$|\psi\rangle\{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\}^{\otimes n}$$

制御Uゲート：制御ビットが1のときのみ標的ビットにUをかける

$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$$
$$U^2|\psi\rangle = e^{2\pi i\theta}U|\psi\rangle = e^{2\pi i\theta\times 2}|\psi\rangle$$
$$\vdots$$
$$U^k|\psi\rangle = e^{2\pi i\theta\times k}|\psi\rangle$$

上から j 番目の計算ビットに注目：

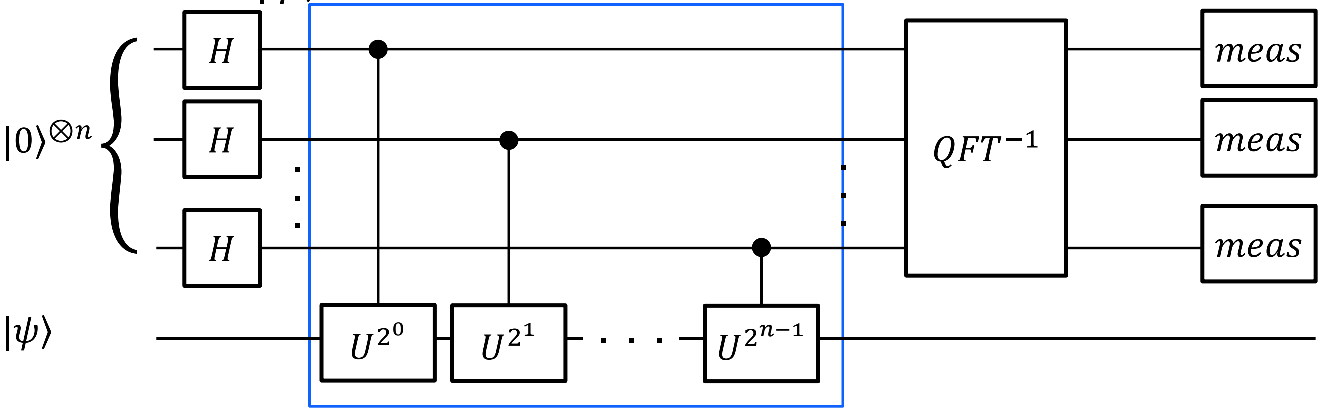
$$|\psi\rangle\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}(|\psi\rangle|0\rangle + |\psi\rangle|1\rangle)$$

$$\rightarrow \frac{1}{\sqrt{2}}(|\psi\rangle|0\rangle + U^{2^j}|\psi\rangle|1\rangle) = \frac{1}{\sqrt{2}}(|\psi\rangle|0\rangle + e^{2\pi i\theta\times 2^j}|\psi\rangle|1\rangle)$$

$$= |\psi\rangle\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\theta\times 2^j}|1\rangle)$$

位相推定アルゴリズムの実装

2. 制御Uゲートを$|\psi\rangle$に施す。



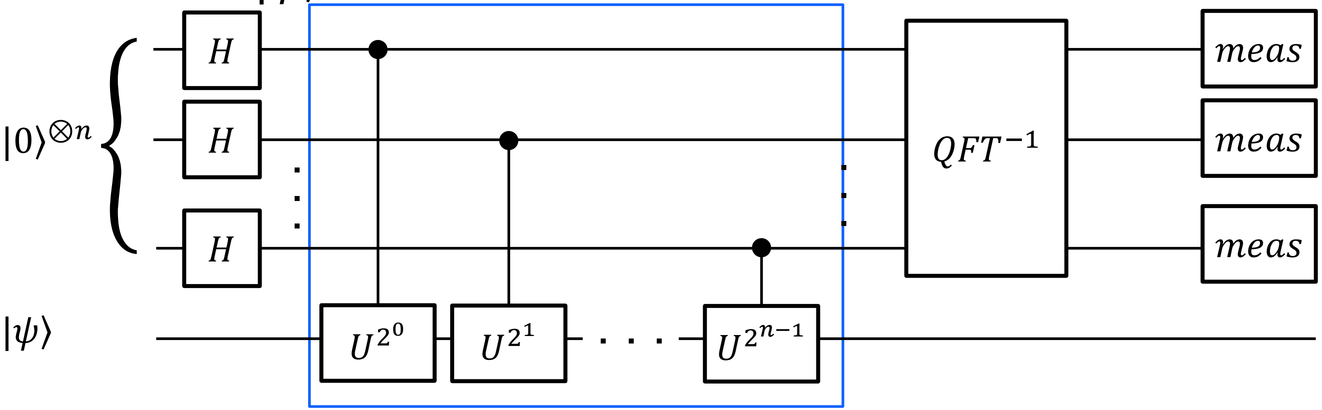
上から j 番目の計算ビットに注目：$|\psi\rangle\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\theta\times 2^j}|1\rangle)$

$j = 0,1, \dots, n-1$の結果を統合する：

$$|\psi\rangle\{\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i\theta\times 2^{n-1}}|1\rangle)\otimes(|0\rangle + e^{2\pi i\theta\times 2^{n-2}}|1\rangle)\otimes \dots \otimes(|0\rangle + e^{2\pi i\theta\times 2^0}|1\rangle)\}$$

位相推定アルゴリズムの実装

2. 制御Uゲートを$|\psi\rangle$に施す。



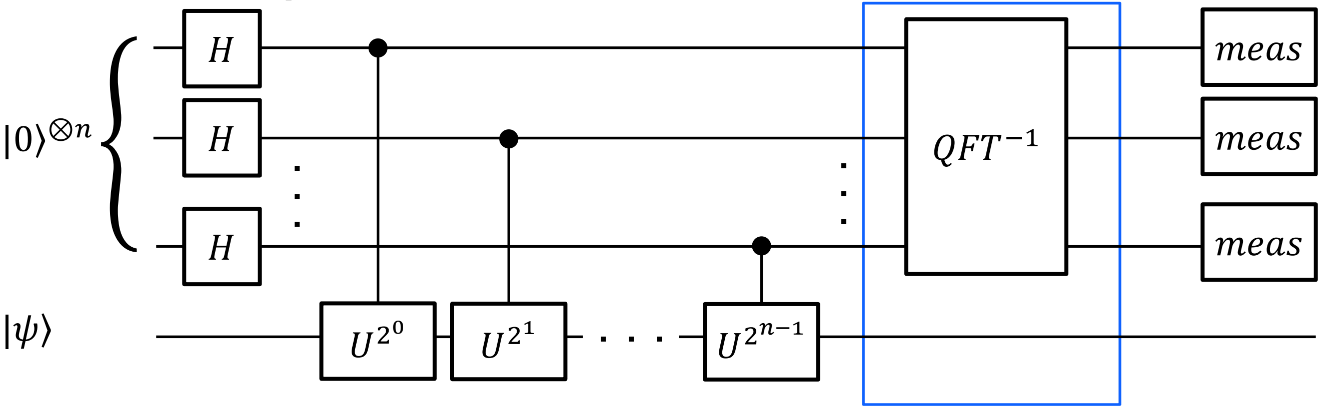
$$|\psi\rangle\{\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i\theta\times 2^{n-1}}|1\rangle)\otimes(|0\rangle + e^{2\pi i\theta\times 2^{n-2}}|1\rangle)\otimes \dots \otimes(|0\rangle + e^{2\pi i\theta\times 2^{0}}|1\rangle)\}$$

ここで、$\varphi = 2^n\theta$とおくと、$\theta = \varphi/2^n$ であるから、

$$|\psi\rangle\{\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i\varphi\frac{2^{n-1}}{2^n}}|1\rangle)\otimes(|0\rangle + e^{2\pi i\varphi\frac{2^{n-2}}{2^n}}|1\rangle)\otimes \dots \otimes(|0\rangle + e^{2\pi i\varphi\frac{2^{0}}{2^n}}|1\rangle)\}$$

位相推定アルゴリズムの実装

3. 計算ビットに逆QFTを施す。



$$|\psi\rangle\{\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i\varphi\frac{2^{n-1}}{2^n}}|1\rangle)\otimes(|0\rangle + e^{2\pi i\varphi\frac{2^{n-2}}{2^n}}|1\rangle)\otimes \dots \otimes(|0\rangle + e^{2\pi i\varphi\frac{2^0}{2^n}}|1\rangle)\}$$

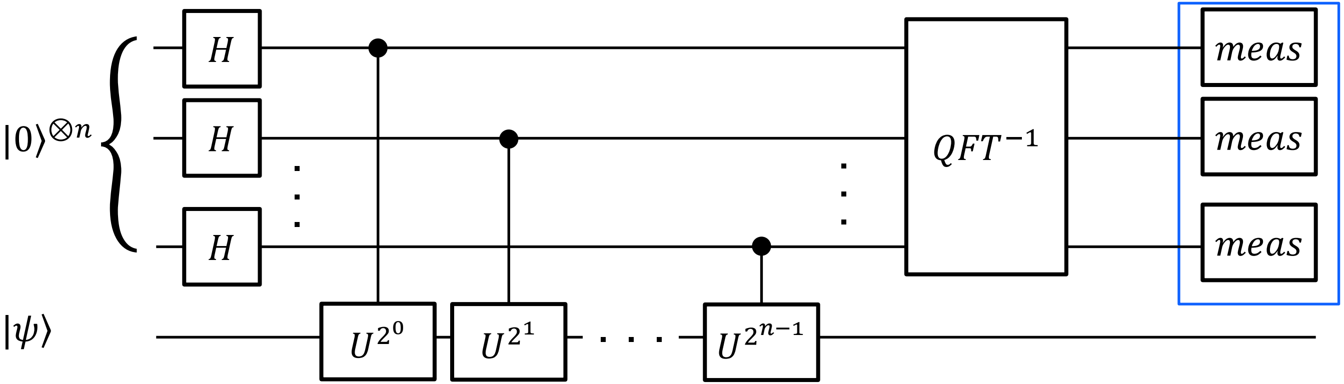
計算ビットの状態は、$|\varphi\rangle = |2^n\theta\rangle$ **に対しQFTをかけたもの**と同じ！

→QFTの逆の演算をすることで、$|\varphi\rangle$ を得られる。

$$|\psi\rangle \otimes U_{QFT}^{-1}\{\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i\varphi\frac{2^{n-1}}{2^n}}|1\rangle)\otimes(|0\rangle + e^{2\pi i\varphi\frac{2^{n-2}}{2^n}}|1\rangle)\otimes \dots \otimes(|0\rangle + e^{2\pi i\varphi\frac{2^0}{2^n}}|1\rangle)\} = |\psi\rangle|2^n\theta\rangle$$

位相推定アルゴリズムの実装

4. 計算ビットを測定する



$$|\psi\rangle|2^n\theta\rangle$$

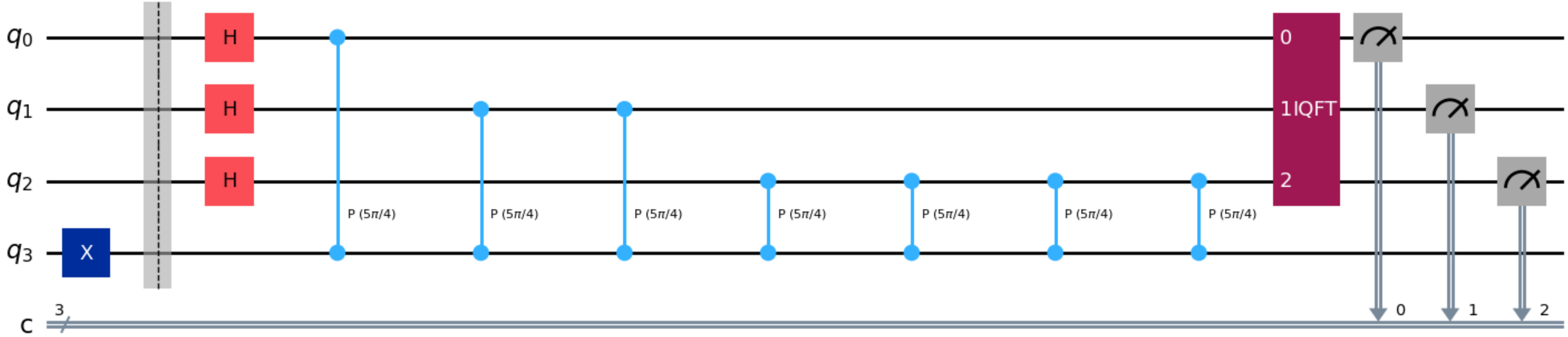
計算ビットを測定し、$2^n\theta$ を得る。

→測定された値を 2^n で割ることで、θ（の近似値）を得る！

位相推定アルゴリズムの実装

Qiskitを用いてQPEの実験をしてみよう。わかりやすく P ゲートを用いる。

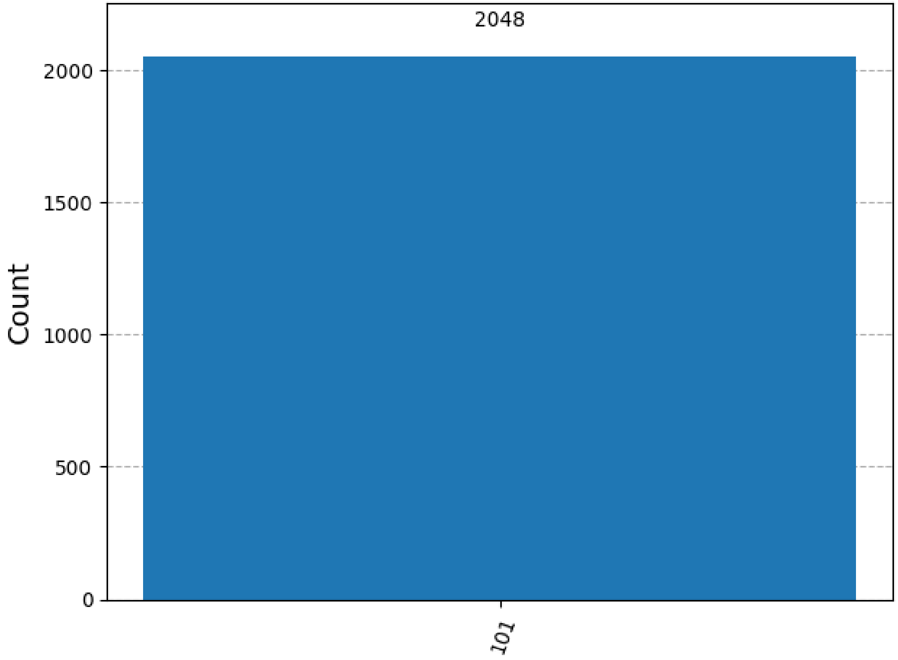
$$P(\lambda)|0\rangle = |0\rangle \rightarrow \theta = 0$$
$$P(\lambda)|1\rangle = e^{i\lambda}|1\rangle \rightarrow \theta = \frac{\lambda}{2\pi}$$



$\lambda = \frac{5}{4}\pi \rightarrow \theta = \frac{5}{8}$　$|\psi\rangle = |1\rangle$ の実装

位相推定アルゴリズムの実装

前ページの量子回路の測定結果は全て５であり、位相推定の結果として $\theta = 5/8$ を得る。



シミュレータでの実験結果

測定結果：$5 \rightarrow \theta = \frac{5}{2^3} = \frac{5}{8}$

まとめ

量子位相推定（QPE）は、$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$ を満たす実数 θ を求めるアルゴリズムである。

- U^{2^j} ゲートが高速に実装できるとき、多項式時間の計算量

- 逆QFTにより、位相の情報を計算基底として取り出し、一度の測定で解を得る
  - ➢ Shorのアルゴリズムなどでも用いられる方法
  - ➢ 主要なアルゴリズムの量子超越のカギになると期待

最後に：もっと勉強したい方へ


Quantum Tokyo へようこそ

学習コンテンツ

Qiskit の始め方

IBM Quantum Plaform 教材 日本語訳

IBM Research Blog 日本語版

(旧) Qiskitテキストブック 日本語版

(旧)Qiskitテキストブック(Qiskitコース) 日本語版

(旧) Qiskitドキュメント・チュートリアル 日本語版リンク集

IBM Quantum Challenge

Qiskit Global Summer School（Qiskit夏の学校） 資料 日本語版

Quantum Tokyo 過去イベント資料

Qiskitコミュニティー関連イベント案内

その他： IBM Quantum の便利なツール

Contents

1.紹介

2. Computation as a diagram

3. 単一量子ビットゲート

4. 複数量子ビット回路ともつれ状態

5. まとめ

量子ビット、量子ゲート、量子回路

沼田 祈史(19 Apr 2024)
Translated by Kifumi Numata

講義ノートのPDFは近日公開予定です。一部のコード スニペットは静的イメージであるため、非推奨になる可能性があることに注意してください。

この実験を実行するための QPU 時間は 約5秒です。

1.紹介

ビット、ゲート、および回路は、量子コンピューティングの基本的な構成要素です。量子ビットと量子ゲートを用いた回路モデルによる量子計算を学び、重ね合わせ、測定、エンタングルメントの復習も行います。

このレッスンでは、次のことを学びます。

- 単一量子ビットゲート
- ブロッホ球
- 重ね合わせ
- 測定
- 2量子ビットゲートとエンタングルメント状態

この講義の最後には、ユーティリティスケールの量子コンピューティングに不可欠な回路の深さについて学びます。

2. Computation as a diagram

量子ビットを使用する場合もビットを使用する場合も、入力を必要な出力に変換するためにそれらを操作する必要があります。少数のビット用の非常に単純なプログラムでは、このプロセスを **回路図** と呼ばれる図で表すと便利です。
左下の図が古典回路の例、右下の図が量子回路の例です。 どちらの場合も、左側に入力があり、出力が右側で、その間に演算が記号によって表されます。演算に用いられる記号は、主に歴史的な背景から「ゲート」と呼ばれます。

3. 単一量子ビットゲート

3.1 量子状態とブロッホ球

量子ビットの状態は、$|0\rangle$と$|1\rangle$ の重ね合わせの状態で表します。任意の量子ビットは以下のように表します。

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

ここで α と β は、$|\alpha|^2 + |\beta|^2 = 1$ を満たす複素数で、確率振幅と呼ばれます。

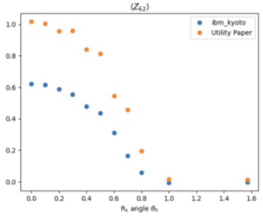
IBM Quantum Learning

「Utility-scale quantum computing」の日本語解説

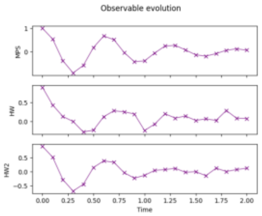
元のコース：https://quantum.cloud.ibm.com/learning/en/courses/utility-scale-quantum-computing

Jupyter notebookの和訳：
https://quantum-tokyo.github.io/introduction/courses/utility-scale-quantum-computing/overview-ja.html

I. Nature paper (127 qubits x 60 entangling gates)



II. 1D Transverse Ising model (70 qubits x 80 entangling gates)



III. The largest GHZ state challenge