

チュートリアル講演

Hamiltonian シミュレーション

~量子計算の基礎と量子多体系シミュレーション~

水田 郁 (Kaoru Mizuta)

大阪大学 量子情報・量子生命研究センター
東大, 理研RQC 客員研究員
JSTさきがけ「量子協奏」

自己紹介

▶ 水田 郁 (Kaoru Mizuta) 世話人もやっています

経歴

2017-2022 修士・博士課程 @ 京都大学

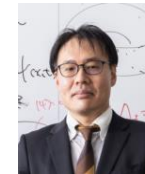
専門: 非平衡量子多体系の物性物理
(特に Floquet 系)



指導教員:
川上 則雄 教授

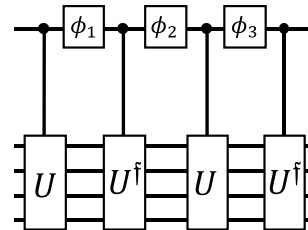
物性
物理

2022-2023 基礎科学特別研究員 @ 理研 RQC



PI:
藤井 啓祐 教授

2023-2026 助教 @ 東京大学



PI:
小芦 雅斗 教授

量子
計算

2026.04- 准教授 @ 大阪大学 QIQB

専門: 量子計算、特に量子系のシミュレーション

参考文献

▶ 量子計算の基礎

Michael A. Nielsen & Isaac L. Chuang 正直、quantum simulation の部分はあまり良くないです
“Quantum Computation and Quantum Information”

▶ Hamiltonian シミュレーション

- A. M. Childs, et al, PRX 11, 011020 (2021).
- A. Gilyén, arXiv:1806.01838 (2018)
- J. M. Martyn, et al., PRX Quantum 2, 040203 (2021).
- L. Lin, arXiv:2201.08309 (2022)
- 水田のノート, 日本語 (水田 HP より)

<https://k-mizuta.github.io/others.html>



本スライドもあり

研究会 HP

→ Time table

→ Hamiltonian シミュレーション

Outline

I. 導入

I-1. 量子計算の基礎

I-2. 量子多体系と Hamiltonian シミュレーション

II. 積公式アルゴリズム

III. Post-Trotter アルゴリズム

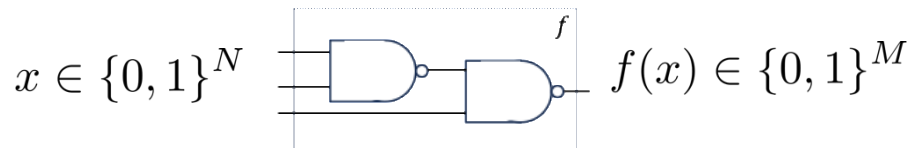
IV. 最近の進展と関連する問題

I-1. 量子計算の基礎

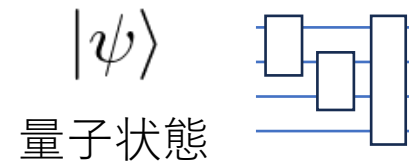
▶ 量子計算とは

- 量子力学に従う系を用いた計算 (量子ビット, 量子ゲート, 量子測定)

古典計算



量子計算



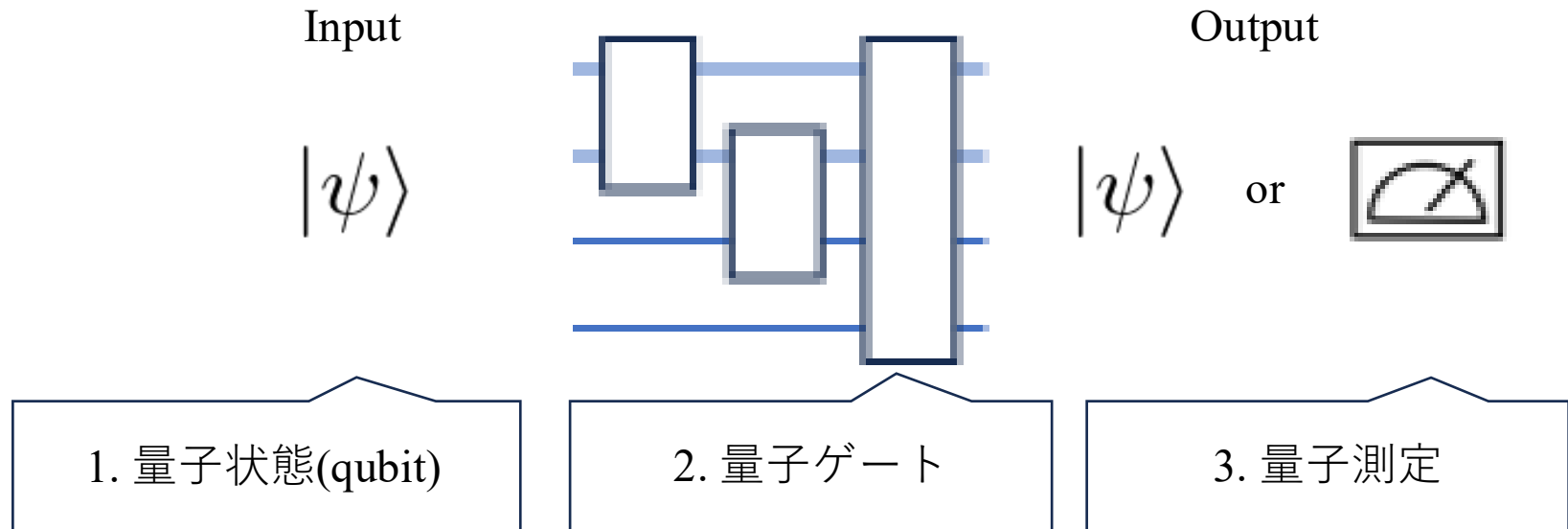
なぜ量子計算をする？

- 一部のタスクにおいて古典計算より高速で、
社会の役に立つかもしれない (と期待)
- 量子加速の有無, 程度を明らかにすることで
計算能力の観点から量子力学の本質を探る

I-1. 量子計算の基礎

▶ 量子計算の構成要素

- 量子力学に従う系を用いた計算 (量子ビット, 量子ゲート, 量子測定)



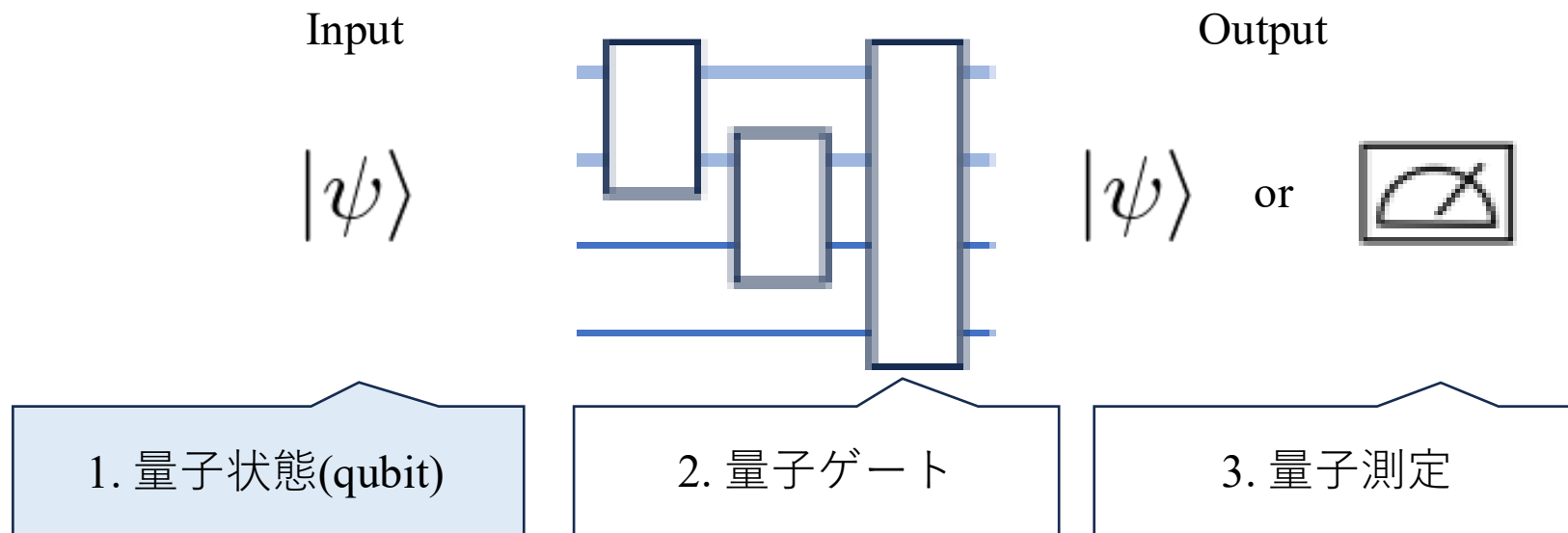
注. 以降は、誤り耐性量子計算(Fault-tolerant quantum computation, FTQC)

を想定して以上の基本操作は正確に実行できると仮定

I-1. 量子計算の基礎

▶ 量子計算の構成要素

- 量子力学に従う系を用いた計算 (量子ビット, 量子ゲート, 量子測定)



I-1. 量子計算の基礎

▶ 量子ビット

- 1 量子ビット (qubit): 量子的な 2 状態

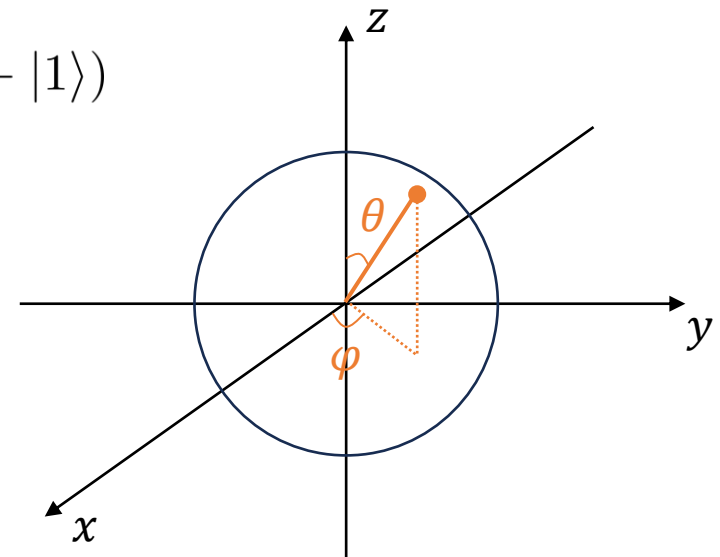
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad \langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1.$$

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

Bloch球表示

$$\alpha = \cos \theta, \quad \beta = e^{i\varphi/2} \sin \theta$$



I-1. 量子計算の基礎

▶ 量子ビット

- 1 量子ビット (qubit): 量子的な 2 状態

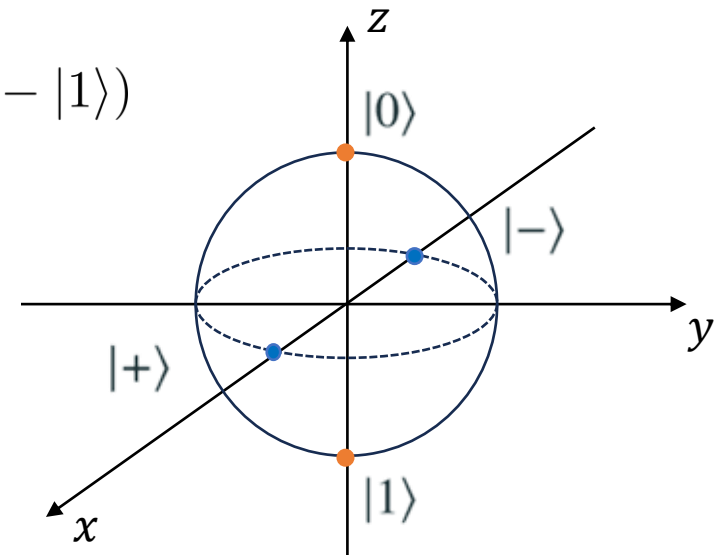
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad \langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1.$$

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

Bloch球表示

$$\alpha = \cos \theta, \quad \beta = e^{i\varphi/2} \sin \theta$$



I-1. 量子計算の基礎

▶ 量子ビット

- 複数量子ビット (n qubits)

2 qubit 状態 $|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle$

例. Bell 状態 (最大エンタングル) $|\Psi_+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$:

n qubit 状態

$$|\psi\rangle = \sum_{z_1, \dots, z_n=0}^1 \alpha_{z_1 \dots z_n} |z_1 \dots z_n\rangle \in \mathbb{C}^{2^n}$$



<https://www.r-ccs.niken.jp/fugaku/>

→ 指数的に大きなベクトル. 富岳でも ~45 qubit の保持が限界.

I-1. 量子計算の基礎

▶ 量子ゲート

- ユニタリー行列

$$|\psi'\rangle = U |\psi\rangle, \quad \exists U \in \mathbb{C}^{2^n \times 2^n}$$

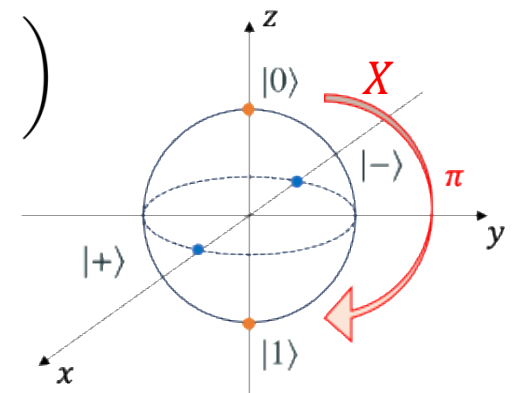
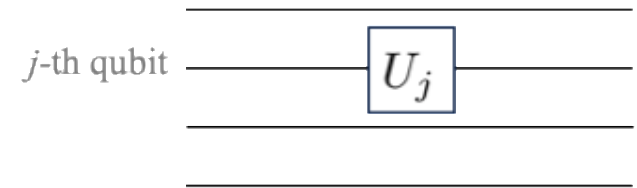
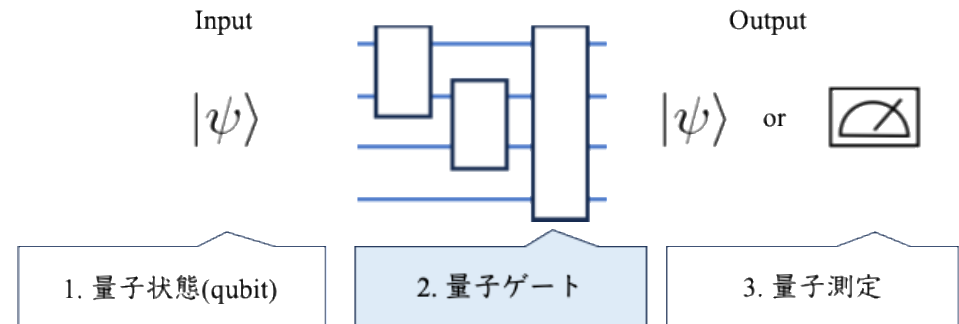
- 1 量子ビットゲート (Single-qubit gate)

例1: Pauli gate

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Bloch 球上の x, y, z 軸まわり π 回転 $\rightarrow$

$$X |0\rangle = |1\rangle, \quad X |+\rangle = |+\rangle$$



I-1. 量子計算の基礎

▶ 量子ゲート

- 1 量子ビットゲート (Single-qubit gate)

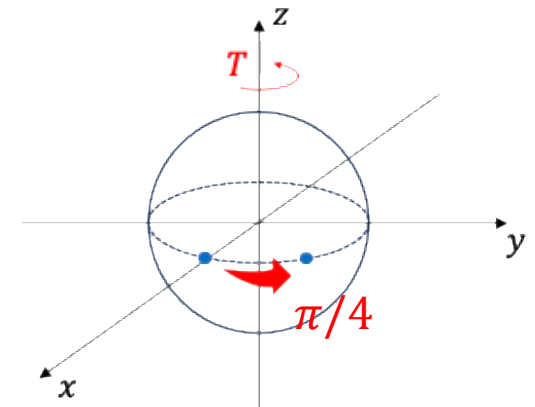
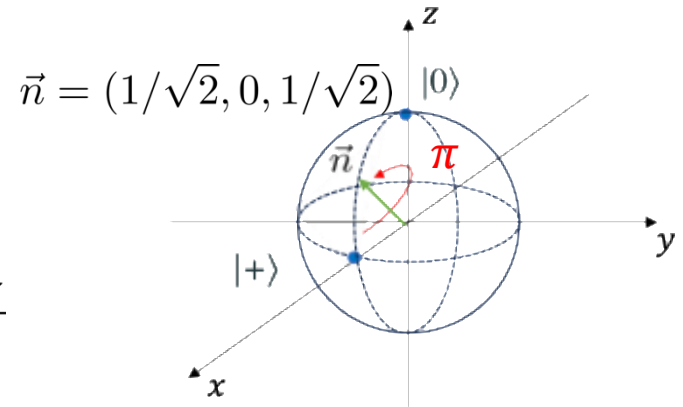
例2: Hadamard gate

$$\text{Had} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \sim e^{i(\pi/2) \frac{X+Z}{\sqrt{2}}}$$

$$\text{Had} |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle, \quad \text{Had} |1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle \quad \text{X, Z 基底の変換}$$

例3: T gate (non-Clifford gate, 後述)

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \sim e^{i(\pi/8)Z}$$



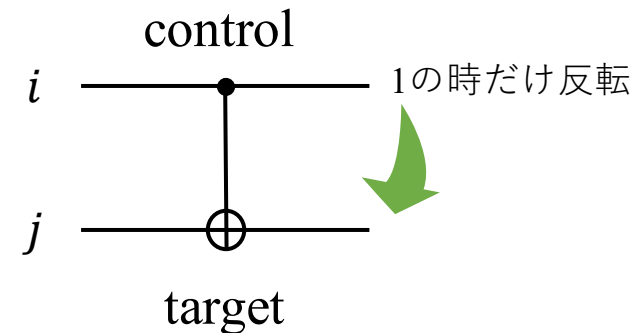
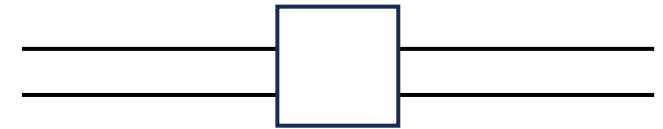
I-1. 量子計算の基礎

▶ 量子ゲート

- 2量子ビットゲート (Two-qubit gate)

例: CNOT gate

$$\begin{aligned}\text{CNOT}_{ij} &= |0\rangle\langle 0|_i \otimes I_j + |1\rangle\langle 1|_i \otimes X_j \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}\end{aligned}$$



状態をエンタングル

$$\text{CNOT}(\text{Had} \otimes I) |00\rangle = \frac{1}{\sqrt{2}} \text{CNOT}(|0\rangle + |1\rangle) |0\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

I-1. 量子計算の基礎

▶ 量子ゲート

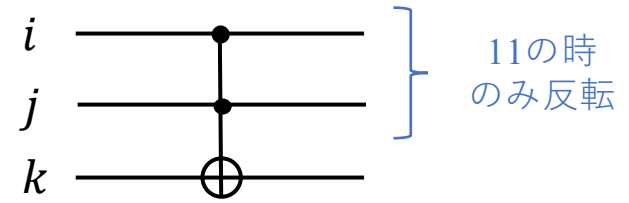
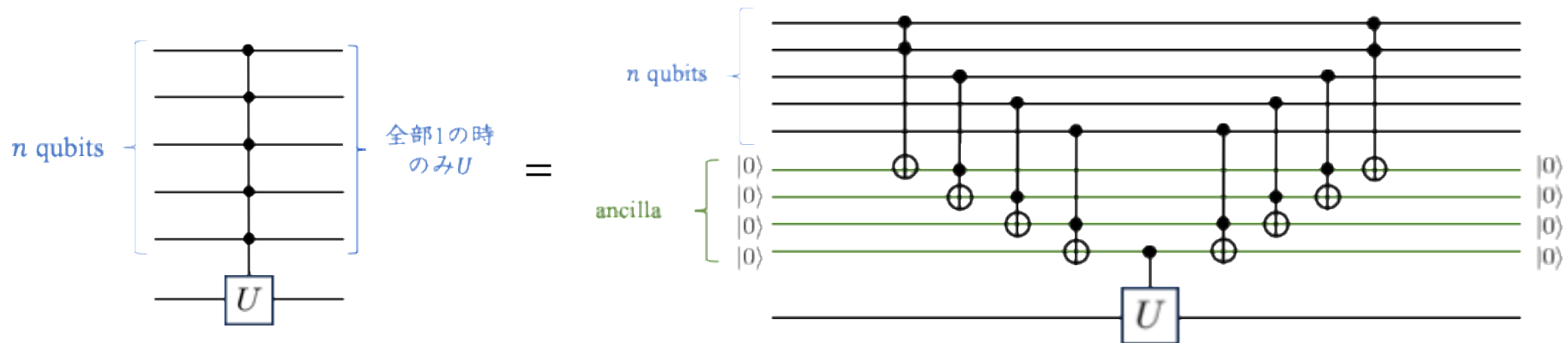
- 多量子ビットゲート

例: Toffoli gate (non-Clifford gate, 後述)

$$\text{Toffoli}_{ijk} = (I_{ij} - |11\rangle\langle 11|_{ij}) \otimes I_k + |11\rangle\langle 11|_{ij} \otimes X_k$$

例: 一般の多重制御ゲート

$$C^n[U] = \{I_n - (|1\rangle\langle 1|)^{\otimes n}\} \otimes I + (|1\rangle\langle 1|)^{\otimes n} \otimes U$$



I-1. 量子計算の基礎

▶ 量子測定

Z 基底での射影測定 (で十分)

$$P_0 = |0\rangle\langle 0|, \quad P_1 = |1\rangle\langle 1|$$

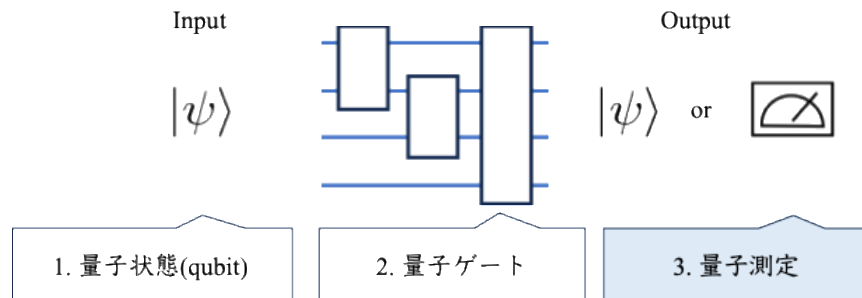
量子状態 $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ に対して、

[測定結果 $x = 0, 1$ を得る確率]

$$p_x = \|P_x |\psi\rangle\|^2 = \begin{cases} |\alpha|^2 & (x = 0) \\ |\beta|^2 & (x = 1) \end{cases}$$

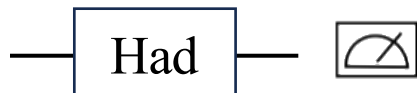
[測定後の量子状態]

$$|\psi'_x\rangle = \frac{1}{\sqrt{p_x}} P_x |\psi\rangle \propto \begin{cases} |0\rangle & (x = 0) \\ |1\rangle & (x = 1) \end{cases}$$



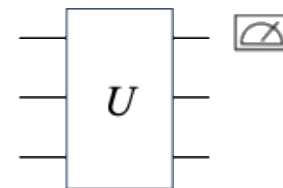
その他の基底

基底変換



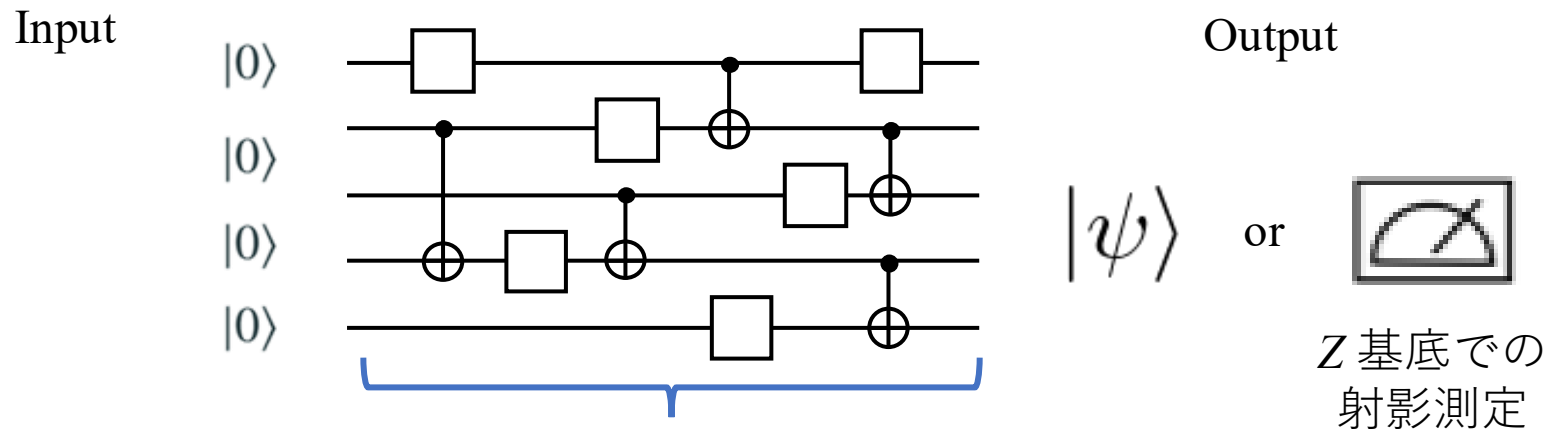
POVM測定

補助系の射影測定



I-1. 量子アルゴリズムと計算コスト

▶ 量子アルゴリズムとコスト

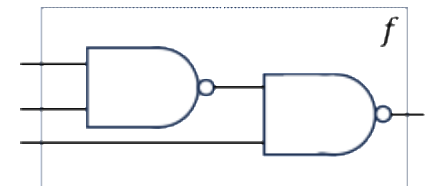


量子アルゴリズムのコスト = “基本ゲート” の個数

“ユニバーサルゲートセット”: 任意の unitary U を表現できる

古典計算では

NAND gate で任意の関数 $f: \{0,1\}^N \rightarrow \{0,1\}^M$ を表現可能



I-1.量子アルゴリズムと計算コスト

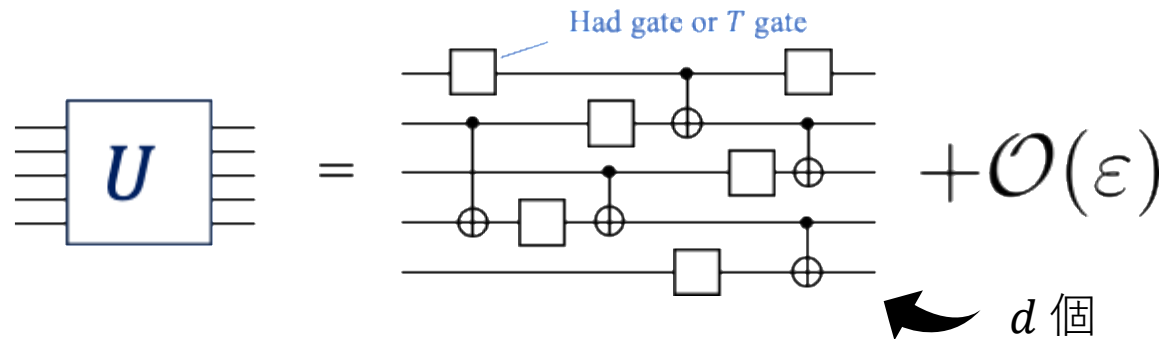
▶ ユニバーサルゲートセット

その他の universal gate set {Had, Toffoli} など

定理. ゲートの組 $G = \{\text{Had}, T, \text{CNOT}\}$ は次の意味で universal :

任意の n -qubit unitary U , 精度 $\varepsilon > 0$ に対して

$$\|U - (u_d)_{x_d} \cdots (u_2)_{x_2} (u_1)_{x_1}\| \leq \varepsilon, \quad u_{d'} \in G \quad \text{とできる。}$$



$d = d(n, \varepsilon)$: 量子アルゴリズムのコスト (基本ゲート数)

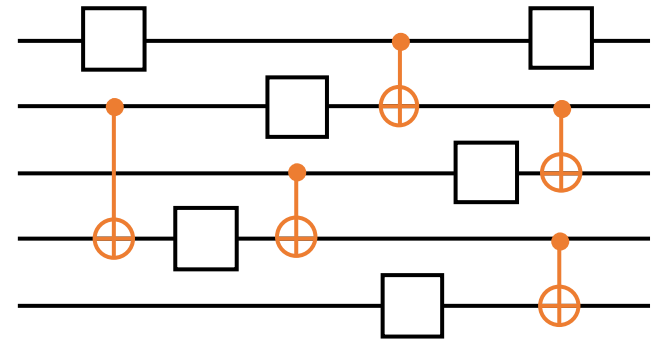
I-1.量子アルゴリズムと計算コスト

▶ ユニバーサルゲートセット

定理

ゲートの組 $G = \{\text{Had}, T, \text{CNOT}\}$ は universal

2 qubit 間の couple



I-1.量子アルゴリズムと計算コスト

▶ ユニバーサルゲートセット

定理

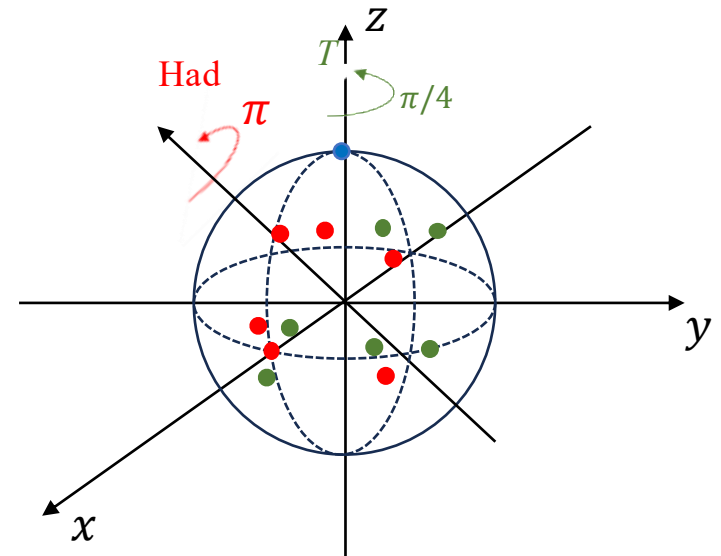
ゲートの組 $G = \{\text{Had}, T, \text{CNOT}\}$ は universal

任意の 1 qubit 回転を作る

Solovay-Kitaev の定理

任意の 1 qubit unitary u は $\text{polylog}(1/\varepsilon)$ 個の Had, T gate で誤差 ε 以内で実装可能。

$$\begin{aligned} \|u - u_d \cdots u_1\| &\leq \varepsilon, & u_{d'} &\in \{\text{Had}, T\} \\ d &= \text{polylog}(1/\varepsilon) \end{aligned}$$



回転の合成で稠密に球面を埋め尽くす

I-1.量子アルゴリズムと計算コスト

▶ Non-Clifford gates

定理

ゲートの組 $G = \{\text{Had}, T, \text{CNOT}\}$ は universal

Non-Clifford gate (Pauli 群を保たない $\exists P : \text{Pauli s.t. } T^\dagger P T : \text{not Pauli}$)

“有用な量子計算には必須のリソース”

その他に Toffoli など

Clifford gate (Had, CNOT, etc.) のみの回路は古典で効率よく計算可能

Gottesman-Knill の定理

その一方で

FTQCで実装コストが高い

魔法状態蒸留, 魔法状態栽培, etc.

多くのコスト評価:

Non-Clifford (T , Toffoli) ゲート数

I-1.量子アルゴリズムと計算コスト

▶ Non-Clifford gates

定理

ゲートの組 $G = \{\text{Had}, T, \text{CNOT}\}$ は universal

Non-Clifford gate (Pauli 群を保たない $\exists P : \text{Pauli s.t. } T^\dagger P T : \text{not Pauli}$)

その他に Toffoli など

“有用な量子計算には必須のリソース”

Clifford gate (Had, CNOT, etc.) のみの回路は古典で効率よく計算可能

Gottesman-Knill の定理

その一方で

(リソース理論)

招待講演者 (8/7):
高木 隆司 先生

FTQCで実装コストが高い

魔法状態蒸留, 魔法状態栽培, etc.

(FTQC)

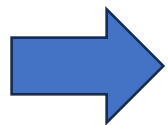
招待講演者 (8/7):
吉岡 信行 先生

I-1. 次節に行く前に: 計算コスト

▶ 計算コスト

ユニバーサルゲートセット $G = \{\text{Had}, T, \text{CNOT}\}$ の量子ゲート数 $d = d(n, \varepsilon)$

(実際には重要だが...)



G にまで分解するのは大変 & 計算コストが G の選択に依存

以降の議論

任意の $O(1)$ -qubit gate を利用可能な状況でのゲート数

Fact (Solovay-Kitaev + α)

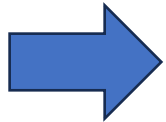
任意の 定数-qubit unitary u は universal gate set G 中の
 $\text{polylog}(1/\varepsilon)$ 個のゲートで誤差 ε 以内で実装可能。

I-1. 次節に行く前に: 計算コスト

▶ 計算コスト

ユニバーサルゲートセット $G = \{\text{Had}, T, \text{CNOT}\}$ の量子ゲート数 $d = d(n, \epsilon)$

(実際には重要だが...)



G にまで分解するのは大変 & 計算コストが G の選択に依存

以降の議論

任意の $\mathcal{O}(1)$ -qubit gate を利用可能な状況でのゲート数

ユニバーサル
ゲートセット G
による回路深さ,
量子ゲート数

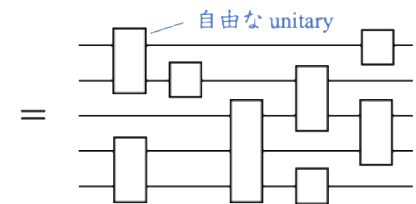
$\leq$

自由な
 $\mathcal{O}(1)$ -qubit gate
による回路深さ,
量子ゲート数 N_g

$\times \text{polylog}(N_g/\epsilon)$

✓ 計算コストは本質的に同じ & 議論が簡単

n -qubit gate



Outline

I. 導入

I-1. 量子計算の基礎

I-2. 量子多体系と Hamiltonian シミュレーション

II. 積公式アルゴリズム

III. Post-Trotter アルゴリズム

IV. 最近の進展と関連する問題

I-2. 種々の量子アルゴリズム

量子計算機では効率よく解けて古典では難しい問題を探することが重要

Grover 探索 2乗加速 L. K. Grover, STOC (1996)

$\{1, 2, \dots, N\}$ から条件を満たすものを探索

Shor の素因数分解 指数加速 P. W. Shor, FOCS (1994)

自然数 N を割る素因数を探索 $15 = 3 \times 5$

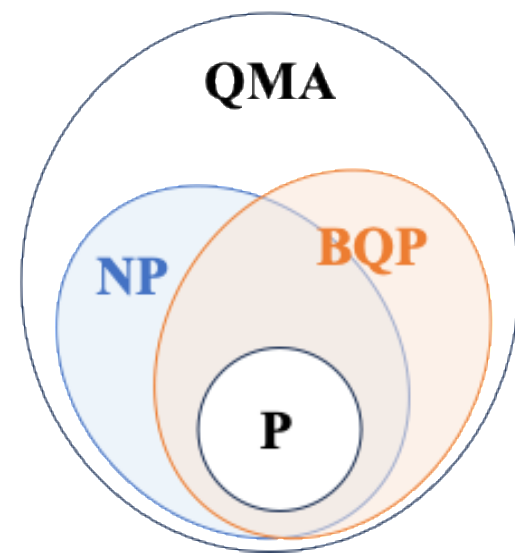
線形方程式問題 (HHL) 指数加速

$A|x\rangle = |b\rangle$ の解 を得る

A. W. Harrow, A. Hassidim, and S. Lloyd, PRL 103, 150502 (2009).

量子優位性があって
実用的な応用もある問題は？

量子多体系のシミュレート(?)



I-2. 量子多体系のシミュレーション

▶ モチベーション

量子力学に従う物質の性質を予言・説明したい！



固体物性, 化学分子, etc.

応答現象, 相転移, 化学反応, 触媒, etc.

"Nature isn't classical, and if you want to make a simulation of nature, you'd better make it quantum mechanical."

1981(MIT) "1st Symposium on Physics and Computation"



R. Feynman
(1982)

<https://www.nobelprize.org/prizes/physics/1965/feynman/facts/>

- ・ 巨大な状態ベクトルを自然に表現して、メモリを指数削減
- ・ 巨大な行列 (Hamiltonian) に関する演算も高速化 (?)

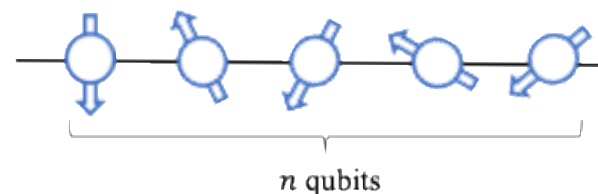
I-2. 量子多体系の記述

▶ 量子多体系の Hamiltonian

→ 格子 $\Lambda = \{1, 2, \dots, n\}$ 上のスピン $S = 1/2$ の系 (n -qubit系)

指数的に大きな自由度?

$$H = \sum_{s=1}^{4^n} c_s P_s, \quad P_s: n\text{-qubit Pauli}, \quad c_s \in \mathbb{R}$$



それ以外の系も基本的に n -qubit系にマップして考える

フェルミオン系 ... Jordan-Wigner 変換, Bravyi-Kitaev 変換など

ボソン系(∞ 次元) ... 粒子数保存, Lieb-Robinson 限界など

に基づき有限次元で打ち切り

“物理的に自然” なクラス: Local Hamiltonian (局所ハミルトニアン)

I-2. 量子多体系の記述

▶ 量子多体系のクラス

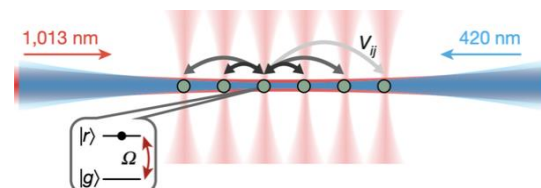
H が k 体以下の相互作用しか持たないとき、 k -local であるという。

例1: Heisenberg 模型

$$H = J \sum_{i=1}^n (X_i X_{i+1} + Y_i Y_{i+1} + Z_i Z_{i+1}) \quad k = 2$$

例2: PXP 模型 Rydberg 原子系で実現

$$H = \sum_i P_{i-1} X_i P_{i+1}, \quad P_i = \frac{1 + Z_i}{2} \quad k = 3$$



H. Bernien, et al., Nature 551, 579–584 (2017)

局所ハミルトニアン = locality k が高々定数の k -local ハミルトニアン

自由度: ${}_n C_k \in \mathcal{O}(n^k)$ ← エルミート行列の 4^n より指数的に小さい

I-2. 量子多体系の記述

▶ 相互作用のレンジ

有限距離相互作用 相互作用の届く距離が定数

例1: Heisenberg 模型

$$H = J \sum_{i=1}^n (X_i X_{i+1} + Y_i Y_{i+1} + Z_i Z_{i+1})$$

→ 物性物理で考える単純化された模型の多く

最近接相互作用, etc.

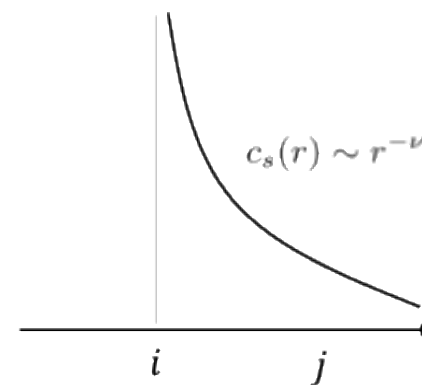


自由度: $\mathcal{O}(n)$

長距離相互作用 相互作用が距離について冪減衰

$$c_s(r) \sim r^{-\nu} \quad \text{自由度: } \mathcal{O}(n^k)$$

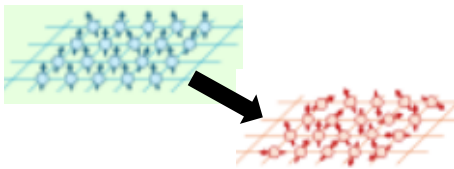
→ dipole int. ($\nu = 3$), Coulomb int. ($\nu = 1$), 量子化学



I-2. 量子多体系の問題

▶ 量子多体系の問題

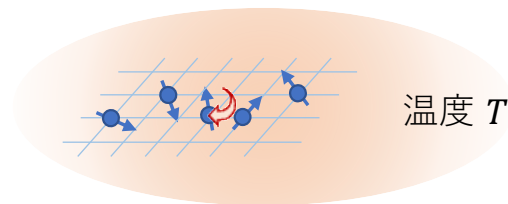
ダイナミクス



$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle$$

→ 応答など動的現象

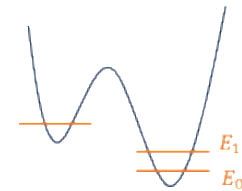
熱平衡状態



$$\rho_\beta = \frac{e^{-\beta H}}{Z_\beta}$$

→ 磁性などの相転移

固有値/固有状態



$$H |E_n\rangle = E_n |E_n\rangle$$

→ 極低温物理・量子化学

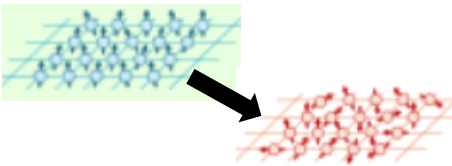
量子多体系を自然に表現する量子計算機で高速に解けたら嬉しい！

I-2. 量子多体系のシミュレーション

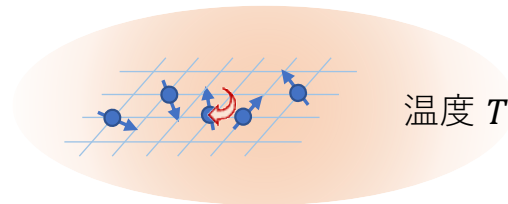
▶ 量子多体系の問題と量子加速

計算対象が量子多体系である $\neq$ 量子計算機で効率よく計算できる

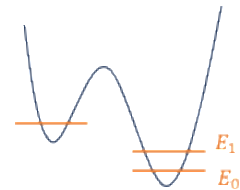
ダイナミクス



熱平衡状態



基底状態(エネルギー)



量子計算機でも一般には指数時間かかると予想

Gibbs sampling

Local Hamiltonian Problem

“QMA完全問題”

固有状態の“ヒント”があれば、量子計算機で指数加速
(量子位相推定アルゴリズム)

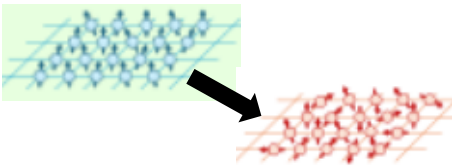
C. Cade, et al, ICALP 2023, pp. 32:1-32.19, 2023

I-2. 量子多体系のシミュレーション

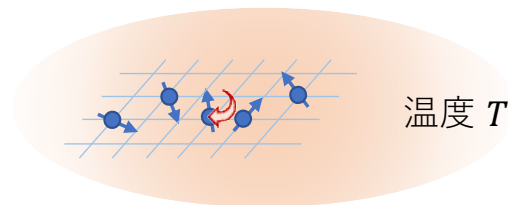
▶ 量子多体系の問題と量子加速

計算対象が量子多体系である $\neq$ 量子計算機で効率よく計算できる

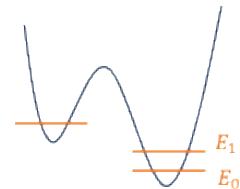
ダイナミクス



熱平衡状態



基底状態(エネルギー)



量子計算機でも一般には指数時間かかると予想

Gibbs sampling

Local Hamiltonian Problem

→

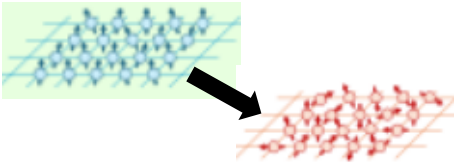
招待講演者 (8/5):
桑原 知剛 先生

招待講演者 (8/5):
高橋 惇 先生

I-2. 量子多体系のシミュレーション

▶ 量子多体系の問題と量子加速

ダイナミクス



Schrodinger 方程式の解

$$|\psi(t)\rangle = U(t) |\psi\rangle$$

時間発展演算子 $U(t)$: 時間非依存の場合 [$H(t) = H$]

$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle$$

$$U(t) = e^{-iHt} = \sum_{n=0}^{\infty} \frac{(-iHt)^n}{n!}$$

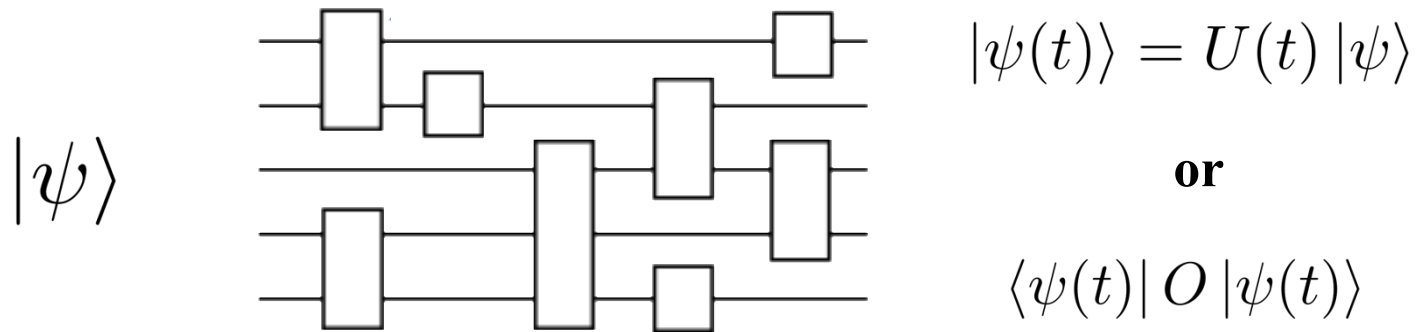
一般の local Hamiltonian に対して、

量子計算による指数加速が存在すると強く期待

“Hamiltonian シミュレーション”

I-2. 量子多体系のシミュレーション

▶ Hamiltonian シミュレーション: 問題の定義



系サイズ n , 時刻 t に対する誤差 $\varepsilon \in (0,1)$ 以内のシミュレーション

状態準備: $\| |\psi'\rangle - U(t) |\psi\rangle \| \leq \varepsilon$ となる状態 $|\psi'\rangle$ を出力

期待値計算: $|O'(t) - \langle\psi(t)| O |\psi(t)\rangle| \leq \varepsilon$ となる値 $O'(t)$ を出力

Poly($n, t, 1/\varepsilon$)-time で量子ダイナミクスをシミュレートできるか？

I-2. 量子多体系のシミュレーション

▶ Hamiltonian シミュレーション

一般の local Hamiltonian に対して、

量子計算による指数加速が存在すると強く期待

多項式時間の量子アルゴリズムの存在

→ 次節以降 (積公式, Post-Trotter, etc.) で構成

古典計算機では指数時間かかると強く期待

I-2. 量子多体系のシミュレーション

▶ Hamiltonian シミュレーション

一般の local Hamiltonian に対して、

量子計算による指数加速が存在すると強く期待

多項式時間の量子アルゴリズムの存在

→ 次節以降 (積公式, Post-Trotter, etc.) で構成

古典計算機では指数時間かかると強く期待

量子アルゴリズム $\approx$ 量子多体系のダイナミクス

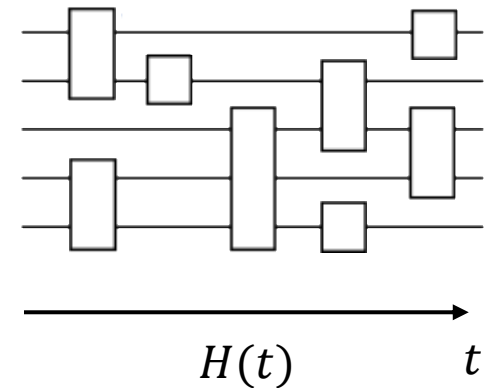
A. M. Childs, “Quantum Information in Continuous Time” MIT (2004)

仮に量子ダイナミクスを古典計算機で効率よく計算できると、

量子計算機を古典計算で効率よく再現できる

← 多分そうではないと信じられている ($P \neq BQP$)

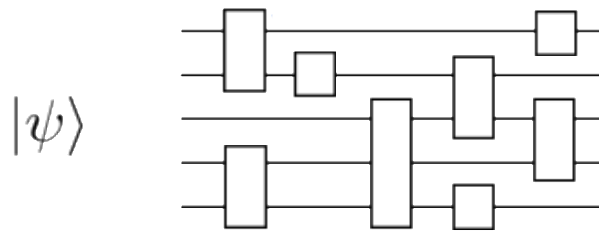
$\approx$ **BQP 完全な問題**
量子計算機で効率よく解ける
問題の中で最も難しい



I. ここまでのまとめ

▶ Hamiltonian シミュレーション

Local Hamiltonian H の下での量子ダイナミクスシミュレーション



$$|\psi(t)\rangle = U(t) |\psi\rangle$$

or

$$\langle\psi(t)| O |\psi(t)\rangle$$

$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle$$

量子計算による指数加速 & 物性物理・量子化学などへの応用

以降の話:

$\text{Poly}(n, t, 1/\varepsilon)$ -time の効率的な量子アルゴリズム(積公式, Post-Trotter)

特に、その中で最も高速な量子アルゴリズムは何か？

Outline

I. 導入

II. 積公式アルゴリズム

II-1. 積公式と実装方法

II-2. アルゴリズムの誤差と計算複雑性

II-3. 高次積公式

III. Post-Trotter アルゴリズム

IV. 最近の進展と関連する問題

II-1. 積公式アルゴリズム

▶ 積公式アルゴリズムとは

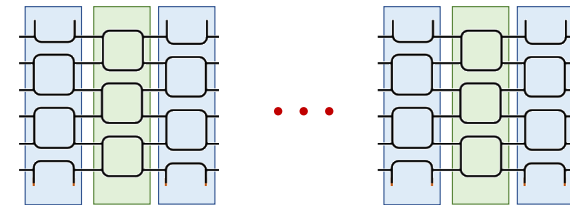
- Hamiltonian シミュレーションで最も歴史の長い量子アルゴリズム。

S. Lloyd, Science (1997)

Trotterization と呼ばれる

- “積公式”に基づいて、量子状態 $|\psi(t)\rangle = e^{-iHt} |\psi\rangle$ を実現

- 量子回路が単純で
現在の実機でもしばしば実装。



- 単純さ
- 他分野(非平衡物性, etc.)との関連
- 良い計算コスト(後述)
- 多くの派生手法

で今なお標準的手法

II-1. 積公式

以降は時間非依存系の時間発展 e^{-iHt} を考える

▶ 積公式とは 行列に対する近似的な指数法則

H. F. Trotter, Proc. Amer. Math. Soc. **10**,
545 (1959)

行列 H_1, H_2 に対して時間 τ が十分小さいならば ($\tau \rightarrow 0$),

$$e^{-iH_2\tau} e^{-iH_1\tau} = e^{-i(H_1+H_2)\tau} + \mathcal{O}(\tau^2)$$

注: 行列 A, B が $A = B + \mathcal{O}(f(\tau)) \Leftrightarrow \|A - B\| \in \mathcal{O}(f(\tau))$

$H = H_1 + H_2$ と分解される時、

1次の積公式
(Lie-Trotter公式)

$$T_1(\tau) = e^{-iH_2\tau} e^{-iH_1\tau} = e^{-iH\tau} + \mathcal{O}(\tau^2) \quad \tau \text{ の 1 次まで近似}$$

II-1. 積公式

▶ 積公式

定理 (Lie-Trotter公式).

H. F. Trotter, Proc. Amer. Math. Soc. **10**, 545 (1959)

$$\tau \rightarrow 0 \text{ で } e^{-iH_2\tau} e^{-iH_1\tau} = e^{-i(H_1+H_2)\tau} + \mathcal{O}(\tau^2)$$

Proof.--- 左辺を Taylor 展開する.

$$\begin{aligned} e^{-iH_2\tau} e^{-iH_1\tau} &= [1 - iH_2\tau + \mathcal{O}(\tau^2)][1 - iH_1\tau + \mathcal{O}(\tau^2)] \\ &= 1 - i(H_1 + H_2)\tau + \mathcal{O}(\tau^2) \\ &= e^{-i(H_1+H_2)\tau} + \mathcal{O}(\tau^2) \end{aligned}$$

II-1. 積公式による量子アルゴリズム

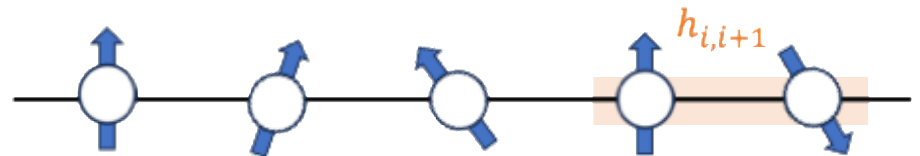
▶ セットアップ

S. Lloyd, Science (1997)

1次元格子上的の最近接2体相互作用の模型 (2-local Hamiltonian)

$$H = \sum_{i=1}^{n-1} h_{i,i+1}, \quad h_{i,i+1}: i, i+1\text{th-qubits上のPauli op.}$$

例1. 横磁場 Ising 模型:



$$h_{i,i+1} = JZ_iZ_{i+1} + hX_i$$

例2. Heisenberg 模型: $h_{i,i+1} = J(X_iX_{i+1} + Y_iY_{i+1} + Z_iZ_{i+1})$

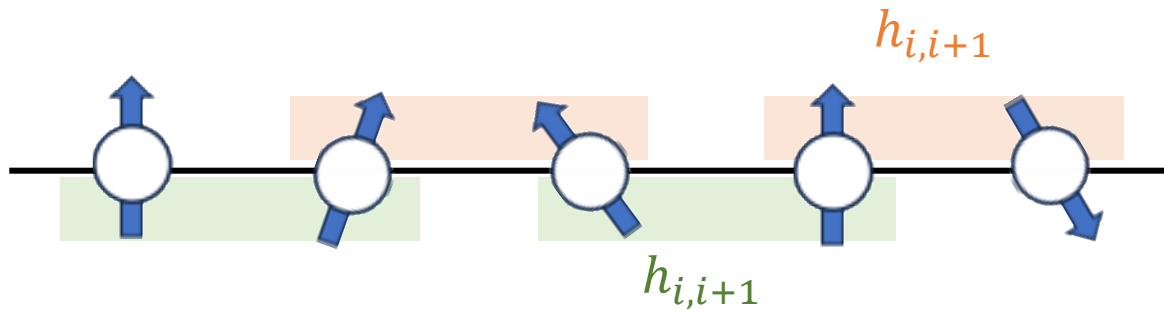
1 site あたりのエネルギー
(Extensiveness)

$$g = \max_i (\|h_{i-1,i}\| + \|h_{i,i+1}\|) \in \mathcal{O}(1)$$

II-1. 積公式による量子アルゴリズム

▶ 分解の構成 十分短い時間 τ の時間発展 $e^{-iH\tau}$ を考える

積公式: $e^{-iH\tau} \simeq e^{-iH_2\tau} e^{-iH_1\tau}$ ($H = H_1 + H_2, \tau \rightarrow 0$)
 $\mathcal{O}(1)$ -qubit ゲートで実装できるように



$$H = H_1 + H_2, \quad H_1 = \sum_{i:\text{odd}} h_{i,i+1}, \quad H_2 = \sum_{i:\text{even}} h_{i,i+1}.$$

互いに可換 ↑ 互いに可換

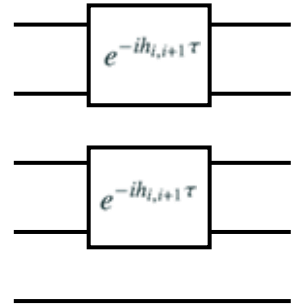
II-1. 積公式による量子アルゴリズム

▶ 量子ゲートの構成

十分短い時間 τ 限定

$$H_1 = \sum_{i:\text{odd}} h_{i,i+1}$$

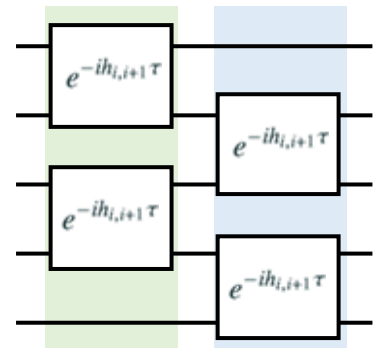
$$\begin{aligned} e^{-iH_1\tau} &= e^{-i \sum_{i:\text{odd}} h_{i,i+1} \tau} \\ &= \prod_{i:\text{odd}} e^{-ih_{i,i+1}\tau} \end{aligned}$$



十分短い時間 τ の時間発展

$$e^{-iH\tau} \simeq \left(\prod_{i:\text{even}} e^{-ih_{i,i+1}\tau} \right) \left(\prod_{i:\text{odd}} e^{-ih_{i,i+1}\tau} \right)$$

$e^{-iH_2\tau} \qquad e^{-iH_1\tau}$



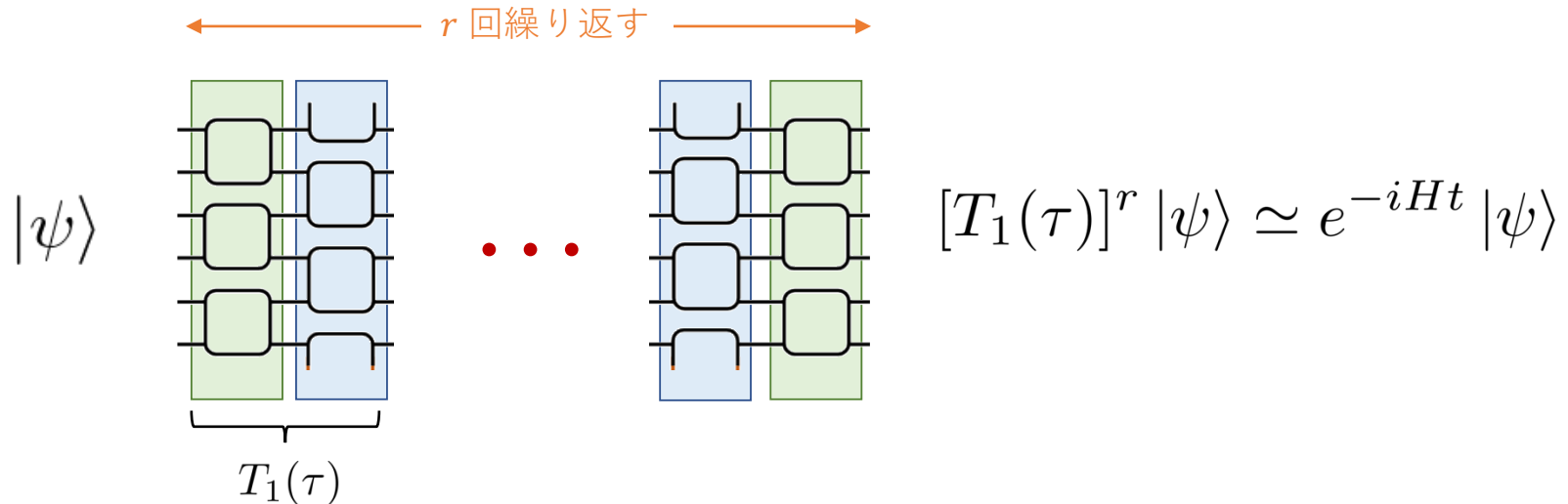
十分短い時間 τ に対しては

2-qubit ゲート数: $n - 1$ ゲート深さ: 2 で実装可能

II-1. 積公式による量子アルゴリズム

▶ 量子アルゴリズムの構成 十分長い時間 t の時間発展 e^{-iHt}

時間 t を r 分割 [$\tau = t/r$: 十分小] $\rightarrow e^{-iH\tau} \cong T_1(\tau)$ の量子回路を r 回

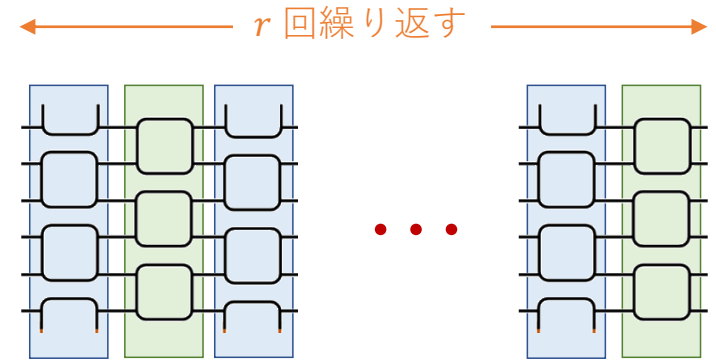


ゲート深さ: $2r$ 2-qubit ゲート数: $r \times (n - 1) \in \mathcal{O}(rn)$

II-2. 計算コストの決定

▶ 計算コストの決定

ゲート深さ: $2r$ 2-qubit ゲート数: rn



繰り返し数 (Trotter数) r が必要

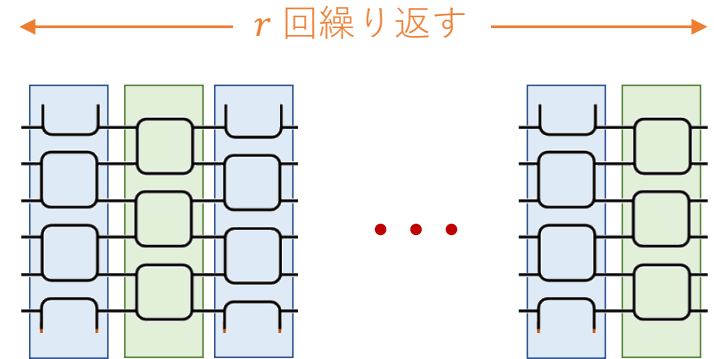
→ 出力が常に $e^{-iHt} |\psi\rangle$ を誤差 ε 以内で近似するよう設定

$$\sup_{|\psi\rangle; \langle\psi|\psi\rangle=1} \left(\| e^{-iHt} |\psi\rangle - \underbrace{[T_1(t/r)]^r}_{\text{出力}} |\psi\rangle \| \right) \leq \varepsilon$$

II-2. 計算コストの決定

▶ 計算コストの決定

ゲート深さ: $2r$ 2-qubit ゲート数: rn



繰り返し数 (Trotter数) r が必要

→ 出力が常に $e^{-iHt} |\psi\rangle$ を誤差 ε 以内で近似するよう設定

$$\|e^{-iHt} - [T_1(t/r)]^r\| \leq \varepsilon$$

量子アルゴリズムの計算コストを決定する

≈ 量子アルゴリズムの誤差を決定する

II-2. アルゴリズムの誤差評価

▶ コスト決定の準備

$$\|e^{-iHt} - [T_1(t/r)]^r\| \leq \varepsilon \text{ を満たす } r \text{ は?}$$

$$U = e^{-iHt/r},$$

$$V = T_1(t/r)$$

$$U, V : \text{unitary} \text{ に対して } \|U^r - V^r\| \leq r\|U - V\|$$

$$\|e^{-iHt} - [T_1(t/r)]^r\| \leq r \times \|e^{-iHt/r} - T_1(t/r)\|$$

積公式 1回分の誤差

$$r \times \|e^{-iHt/r} - T_1(t/r)\| \leq \varepsilon \text{ を満たす } r \text{ を見つければ OK}$$

II-2. アルゴリズムの誤差評価

▶ 積公式の誤差

$$\|e^{-iH\tau} - T_1(\tau)\| \leq \underbrace{(\tau \text{ によらない係数})}_{\text{系サイズ } n \text{ には依存する}} \times \tau^2 \in \mathcal{O}(\tau^2)$$

$$e^{-i(H_1+H_2)\tau}$$

$$e^{-iH_2\tau}e^{-iH_1\tau}$$

系サイズ n には依存する

ナイーブな評価

$$\|e^{-iH\tau} - T_1(\tau)\| \in \mathcal{O}((\|H_1\|\tau)^2, (\|H_2\|\tau)^2) = \mathcal{O}((gn\tau)^2)$$

$$\|H_1\| = \sum_{i:\text{odd}} \|h_{i,i+1}\| \leq \frac{n}{2} \times g$$

1 site あたり
のエネルギー $g \in \mathcal{O}(1)$

積公式の誤差は系サイズ n の 2 乗に比例する(?)

II-2. アルゴリズムの誤差評価

▶ 積公式の誤差

ナイーブな評価

積公式の誤差は系サイズ n の 2 乗に比例する(?)

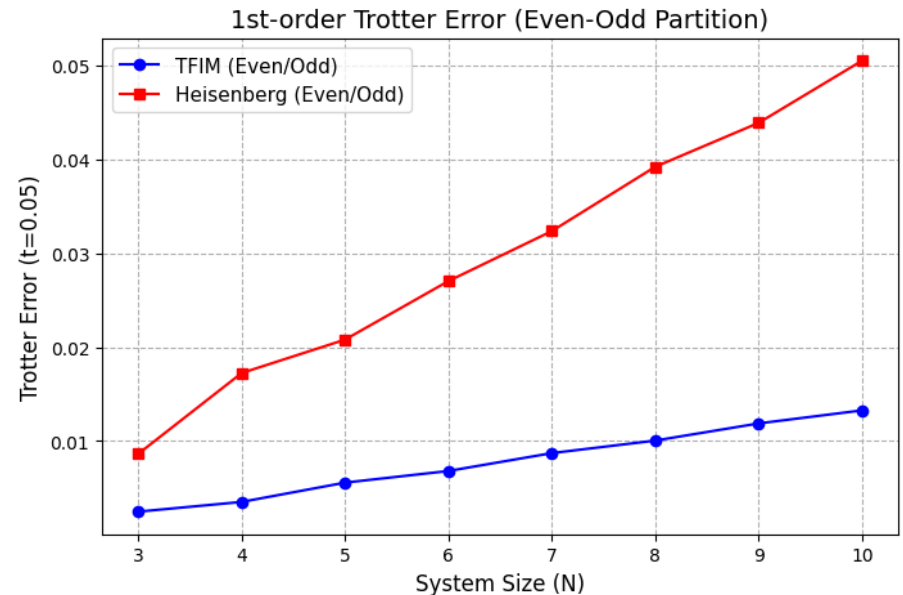
数値計算してみると →

$$\|e^{-iH\tau} - T_1(\tau)\|$$

Heisenberg 模型

横磁場 Ising 模型

誤差は系サイズ n に比例



← 誤差の交換子スケーリング & Hamiltonian H の局所性 による普遍的性質

II-2. アルゴリズムの誤差評価

▶ 普遍的な誤差の n 依存性

$$H = H_1 + H_2, \quad H_1 = \sum_{i:\text{odd}} h_{i,i+1}, \quad H_2 = \sum_{i:\text{even}} h_{i,i+1}.$$

定理1. (誤差の交換子スケーリング)

$$\|e^{-iH\tau} - T_1(\tau)\| \leq \frac{\tau^2}{2} \|[H_1, H_2]\|$$

定理2. (局所性によるスケーリング改善)

H が 2-local, 最近接相互作用 Hamiltonian のとき、

$$\|[H_1, H_2]\| \leq 2g^2 n \in \mathcal{O}(n) \quad g : 1 \text{ site 当たりエネルギー}$$

II-2. アルゴリズムの誤差評価

▶ 交換子スケーリング

定理1. (誤差の交換子スケーリング)

$$\|e^{-iH\tau} - T_1(\tau)\| \leq \frac{\tau^2}{2} \|[H_1, H_2]\|$$

Proof.---

$$\|e^{-iH\tau} - T_1(\tau)\| = \|-e^{-iH\tau}(e^{iH\tau}T_1(\tau) - 1)\| = \|\underbrace{e^{iH\tau}T_1(\tau)}_{f(\tau)} - \underbrace{1}_{f(0)}\|$$

$$f(\tau) - f(0) = \int_0^\tau d\tau' \frac{d}{d\tau'} f(\tau')$$

II-2. アルゴリズムの誤差評価

▶ 交換子スケーリング

定理1. (誤差の交換子スケーリング)

Proof.---

$$\begin{aligned}\|e^{-iH\tau} - T_1(\tau)\| &= \|-e^{-iH\tau}(e^{iH\tau}T_1(\tau) - 1)\| = \|e^{iH\tau}T_1(\tau) - 1\| \\&= \left\| \int_0^\tau d\tau' \frac{d}{d\tau'} \left[e^{iH\tau'} T_1(\tau') \right] \right\| \\&= \left\| \int_0^\tau d\tau' \frac{d}{d\tau'} \left(\underbrace{e^{i(H_1+H_2)\tau'}}_{i(H_1+H_2)} \underbrace{e^{-iH_2\tau'}}_{-iH_2} \underbrace{e^{-iH_1\tau'}}_{-iH_1} \right) \right\|\end{aligned}$$

愚直に微分

II-2. アルゴリズムの誤差評価

▶ 交換子スケーリング

定理1. (誤差の交換子スケーリング)

Proof.---

$$\|e^{-iH\tau} - T_1(\tau)\|$$

$$= \left\| \int_0^\tau d\tau' e^{i(H_1+H_2)\tau'} e^{-iH_2\tau'} \left(e^{iH_2\tau'} iH_1 e^{-iH_2\tau'} - iH_1 \right) e^{-iH_1\tau'} \right\|$$

$$\leq \int_0^\tau d\tau' \left\| \underbrace{e^{iH_2\tau'} H_1 e^{-iH_2\tau'}}_{f(\tau')} - \underbrace{H_1}_{f(0)} \right\|$$

$$f(\tau) - f(0) = \int_0^\tau d\tau' \frac{d}{d\tau'} f(\tau')$$

II-2. アルゴリズムの誤差評価

▶ 交換子スケーリング

定理1. (誤差の交換子スケーリング)

Proof.---

$$\begin{aligned}\|e^{-iH\tau} - T_1(\tau)\| &\leq \int_0^\tau d\tau' \left\| \int_0^{\tau'} d\tau'' \frac{d}{d\tau''} \left(\underbrace{e^{iH_2\tau''}}_{iH_2} H_1 \underbrace{e^{-iH_2\tau''}}_{-iH_2} \right) \right\| \\ &\leq \int_0^\tau d\tau' \int_0^{\tau'} d\tau'' \| [H_2, H_1] \| = \frac{\tau^2}{2} \| [H_1, H_2] \| \quad \square\end{aligned}$$

“ H_1 と H_2 が可換なら誤差がゼロになり指数法則が成立”

という事実を正確に反映する誤差

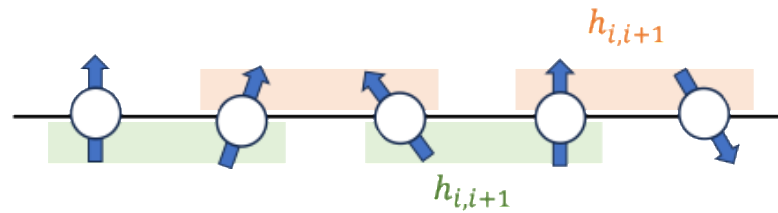
II-2. アルゴリズムの誤差評価

▶ 誤差と局所性

定理1.

$$\|e^{-iH\tau} - T_1(\tau)\| \leq \frac{\tau^2}{2} \|[H_1, H_2]\|$$

$$H = H_1 + H_2, \quad H_1 = \sum_{i:\text{odd}} h_{i,i+1}, \quad H_2 = \sum_{i:\text{even}} h_{i,i+1}.$$



定理2. (局所性によるスケーリング改善)

H が 2-local, 最近接相互作用 Hamiltonian のとき、

$$\|[H_1, H_2]\| \leq 2g^2n \in \mathcal{O}(n) \quad g : 1 \text{ site 当たりエネルギー}$$

II-2. アルゴリズムの誤差評価

▶ 誤差と局所性

定理2.

$$\|[H_1, H_2]\| \leq 2g^2n \in \mathcal{O}(n)$$

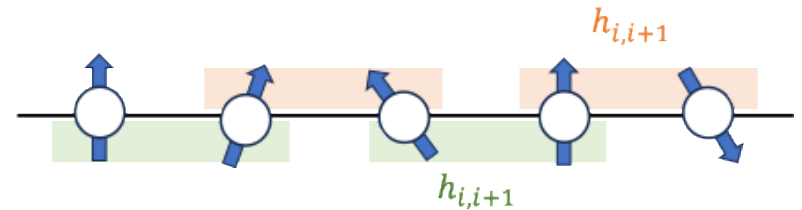
Proof.---

$$\|[H_1, H_2]\| = \left\| \sum_{i:\text{odd}} \sum_{j:\text{even}} [h_{i,i+1}, h_{j,j+1}] \right\|$$

$\overbrace{\hspace{10em}}^{O(n^2) \text{ terms}}$

$$= \left\| \sum_{i:\text{odd}} \sum_{j=i\pm 1} [h_{i,i+1}, h_{j,j+1}] \right\|$$

$$\leq \sum_{i:\text{odd}} 2 \underbrace{\|h_{i,i+1}\|}_{\leq g} (\underbrace{\|h_{i-1,i}\|}_{\leq g} + \underbrace{\|h_{i+1,i+2}\|}_{\leq g}) \leq 2g^2n \in \mathcal{O}(n) \quad \square$$



“本来現れる $\mathcal{O}(n^2)$ 個の項が locality, range により $\mathcal{O}(n)$ にまで減少”

II-2. アルゴリズムの誤差評価

▶ 誤差

定理1 (交換子スケーリング).

$$\|e^{-iH\tau} - T_1(\tau)\| \leq \frac{\tau^2}{2} \|[H_1, H_2]\|$$

定理2. (1d, 2-local, 最近接)

$$\|[H_1, H_2]\| \leq 2g^2n$$

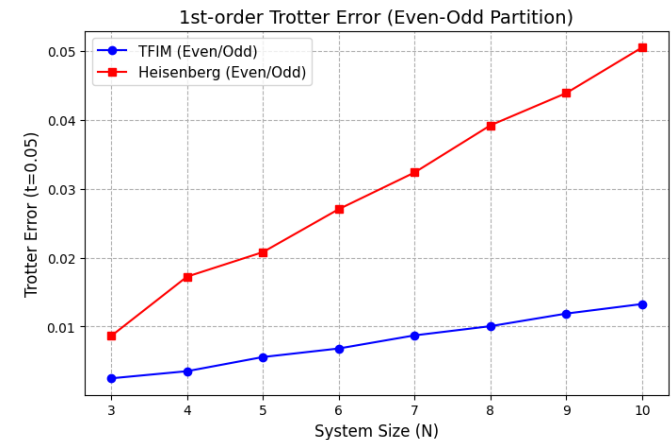
$$\|e^{-iH\tau} - T_1(\tau)\| \leq g^2n\tau^2 \in \mathcal{O}(n\tau^2)$$

まとめ: 積公式の誤差は locality, range により系サイズ n に比例

当初の解析 $= \mathcal{O}((gn\tau)^2)$

誤差は系サイズ n の 2 乗に比例する (?)
← 過大評価

正確な結果



II-2. アルゴリズムの誤差評価

▶ 誤差

定理1 (交換子スケーリング).

$$\|e^{-iH\tau} - T_1(\tau)\| \leq \frac{\tau^2}{2} \|[H_1, H_2]\|$$

定理2. (1d, 2-local, 最近接)

$$\|[H_1, H_2]\| \leq 2g^2n$$

$$\|e^{-iH\tau} - T_1(\tau)\| \leq g^2n\tau^2 \in \mathcal{O}(n\tau^2)$$

まとめ: 積公式の誤差は locality, range により系サイズ n に比例

→ 誤差の減少は一般の $d \in \mathcal{O}(1)$ 次元, $k \in \mathcal{O}(1)$ -local な系で存在

$$\|e^{-iH\tau} - T_1(\tau)\| \leq kg^2n\tau^2 \in \mathcal{O}(g^2n\tau^2)$$

有限距離相互作用系

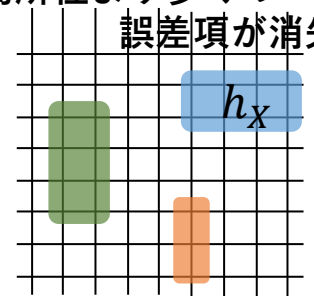
$$g \in \mathcal{O}(1)$$

1 site あたりのエネルギー

長距離相互作用系

$$g = \int r^{-\nu} d^d r \in \mathcal{O}(n^{1-\nu/d})$$

局所性より多くの
誤差項が消失



II-2. アルゴリズムの複雑性

▶ 計算コストの決定

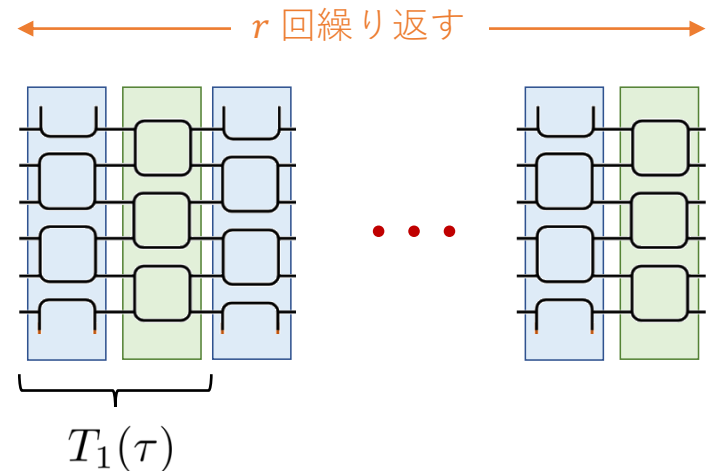
※ 有限距離相互作用系

ゲート深さ: $2r$ 2-qubit ゲート数: rn

Trotter 数 r の決定 ← 精度保証

$$r \times \left\| e^{-iHt/r} - T_1(t/r) \right\| \leq \varepsilon \quad \text{を満たす } r$$

積公式 1 回分の誤差

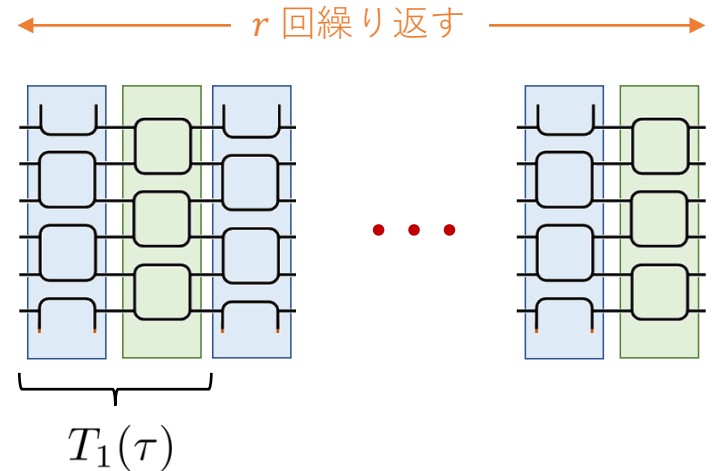


II-2. アルゴリズムの複雑性

▶ 計算コストの決定

※ 有限距離
相互作用系

ゲート深さ: $2r$ 2-qubit ゲート数: rn



Trotter 数 r の決定 ← 精度保証

$$r \times \underbrace{g^2 n \left(\frac{t}{r} \right)^2}_{\text{積公式の誤差上限(定理1,2)}} \leq \varepsilon \quad \text{を満たす } r$$

積公式の誤差上限(定理1,2)

$$\Rightarrow r = \left\lceil \frac{g^2 n t^2}{\varepsilon} \right\rceil \in \mathcal{O}\left(\frac{n t^2}{\varepsilon}\right)$$

$$\begin{array}{ll} \text{ゲート深さ:} & \mathcal{O}\left(\frac{n t^2}{\varepsilon}\right) \\ \text{2-qubit ゲート数:} & \mathcal{O}\left(\frac{(n t)^2}{\varepsilon}\right) \end{array}$$

Hamiltonian シミュレーションは量子計算で効率よく実行可能

Outline

I. 導入

II. 積公式アルゴリズム

II-1. 積公式と実装方法

II-2. アルゴリズムの誤差と計算複雑性

II-3. 高次積公式

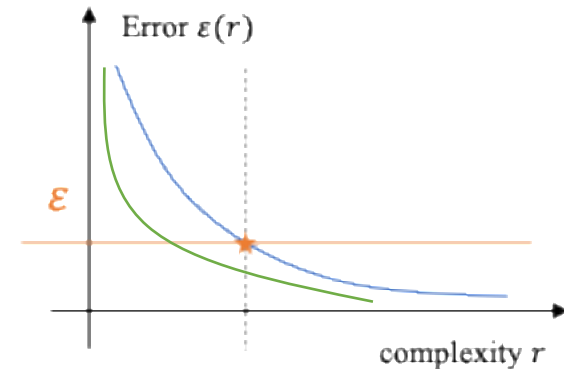
III. Post-Trotter アルゴリズム

IV. 最近の進展と関連する問題

II-3. 高次の積公式

計算コストの決定：

$$r \times \left\| e^{-iHt/r} - T_1(t/r) \right\| \leq \varepsilon$$



“計算コストは、量子アルゴリズムの誤差によって決まる”

← 近似精度の改善 = 計算コストの改善

高次の積公式 $T_p(\tau) = e^{-iH\tau} + \mathcal{O}(\tau^{p+1}), \quad p \in \mathbb{N}$

II-3. 高次の積公式

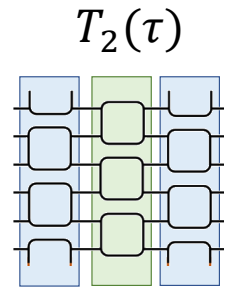
▶ 高次の積公式

$$T_p(\tau) = e^{-iH\tau} + \mathcal{O}(\tau^{p+1}), \quad p \in \mathbb{N}$$

2次の積公式 (Suzuki公式)

M. Suzuki, Commun.Math. Phys. **51**, 183 (1976).

$$T_2(\tau) = e^{-iH_1\tau/2} e^{-iH_2\tau} e^{-iH_1\tau/2} = e^{-iH\tau} + \mathcal{O}(\tau^3)$$



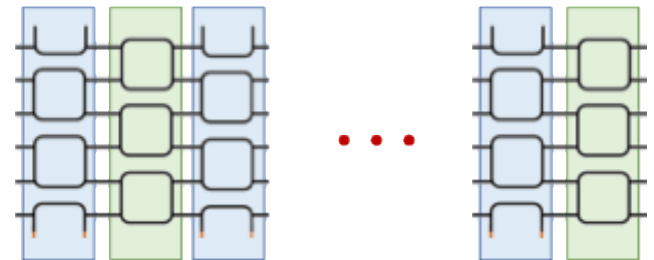
p次の積公式 (Suzuki公式)

M. Suzuki, J. Math. Phys. **32**, 400–407 (1991)

$$T_{2q}(\tau) = \{T_{2q-2}(\alpha_q\tau)\}^2 T_{2q-2}((1 - 4\alpha_q)\tau) \{T_{2q-2}(\alpha_q\tau)\}^2, \quad \alpha_q = \frac{1}{4 - 4^{\frac{1}{2q-1}}}$$

但し、1回分の深さが $5^{p/2}$ で増えるので

一般には定数程度の p しか使わない



その他にも Forest-Ruth-Suzuki 公式など

II-3. 高次の積公式

▶ 高次積公式の誤差 1次 $\|e^{-iH\tau} - T_1(\tau)\| \leq \frac{\tau^2}{2} \| [H_1, H_2] \| \in \mathcal{O}(n\tau^2)$

高次積公式でも locality, range により誤差が本質的に減少

A. M. Childs, et al, PRX (2021)

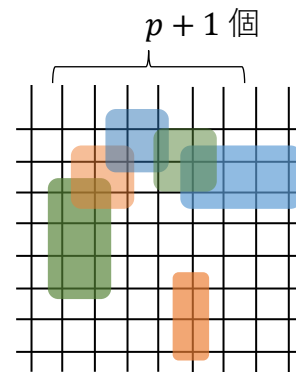
定理 (誤差の交換子スケーリング & 局所性).

$$\|e^{-iH\tau} - T_p(\tau)\| \leq C_p \sum_{\gamma_0, \dots, \gamma_p} \underbrace{\| [H_{\gamma_p}, [\dots, [H_{\gamma_1}, H_{\gamma_0}]] \|}_{p+1 \text{ 重の多重交換子}} \tau^{p+1}$$

更に H が k -local ならば

$$\|e^{-iH\tau} - T_p(\tau)\| \leq C_p (2kg)^p n \tau^{p+1} \in \mathcal{O}(n(g\tau)^{p+1})$$

有限距離相互作用なら n に比例



II-3. 高次の積公式

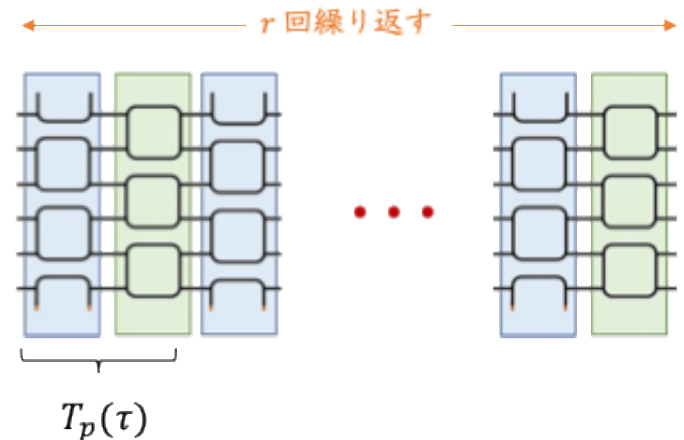
▶ Hamiltonian シミュレーション

有限距離相互作用

$$e^{-iHt} \simeq \{T_p(\tau)\}^r, \quad \tau = t/r$$

繰り返し数 r の決定 ← 精度保証

$$r \times \underbrace{\left\| e^{-iHt/r} - T_p(t/r) \right\|}_{\text{積公式 1 回分の誤差}} \leq \varepsilon \in \mathcal{O}(n(t/r)^{p+1})$$



高次 p では近似精度が良くなることで計算コストも改善

回路深さ $\mathcal{O}(r) = \mathcal{O}\left(t \left(\frac{nt}{\varepsilon}\right)^{\frac{1}{p}}\right)$ ゲート数 $\mathcal{O}(rn) = \mathcal{O}\left(nt \left(\frac{nt}{\varepsilon}\right)^{\frac{1}{p}}\right)$

II-3. 高次の積公式

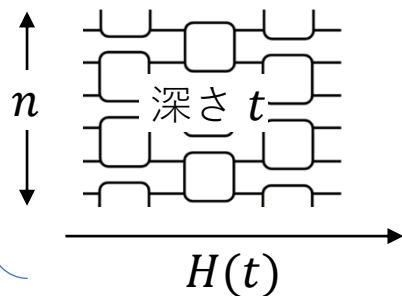
▶ 積公式と理論下限

定理. (量子ゲート数の下限)

J. Haah, et al., SIAM (2021)

任意の有限距離相互作用系の時間発展を再現するアルゴリズムは
量子ゲート数を少なくとも $\tilde{\Omega}(nt)$ だけ必要とする

Proof (直感的な説明). --- 量子回路 $\approx$ 量子ダイナミクス



時間 t の時間発展で
深さ t の量子回路
を再現可能

アルゴリズムはゲート数 nt
の量子回路も包含

高次積公式は？

ゲート数: $(nt)^{1+\frac{1}{p}} \varepsilon^{-\frac{1}{p}}$

理論下限に近い n, t 依存性を達成

II. まとめ: 積公式アルゴリズム

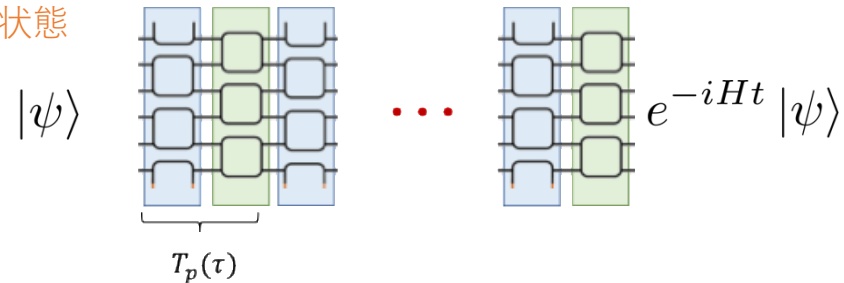
▶ 最近の進展

先ほどの解析: 最悪計算時間

Worst-case error

$$\sup_{|\psi\rangle; \langle\psi|\psi\rangle=1} (\|e^{-iH\tau} |\psi\rangle - T_p(\tau) |\psi\rangle\|) \in \mathcal{O}(n\tau^{p+1}) \rightarrow \text{計算したい状態の誤差を過大評価}$$

任意の初期状態



特別な初期状態のクラスでは誤差の縮小 → 高速化

- 高エンタングル状態 ($S_\rho \sim \log d$)

$$(\text{誤差}) \in \mathcal{O}(\sqrt{n}\tau^{p+1})$$

Q. Zhao, et al., Nature Physics 21, 1338 (2025)

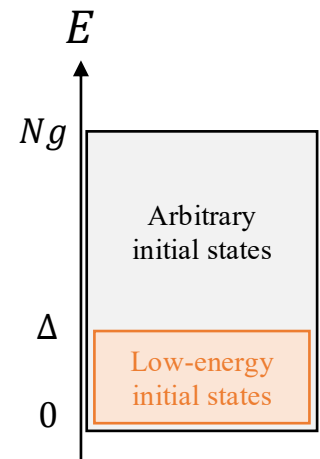
- 低エネルギー状態 (エネルギー $\leq \Delta$)

$$(\text{誤差}) \in \mathcal{O}((\Delta + g \log n)\tau^{p+1})$$

K. Mizuta and T. Kuwahara, PRL 135, 130602(2025)

(最大で) n に関して
更なる 2乗加速

(最大で) n に関して
更なる 指数加速



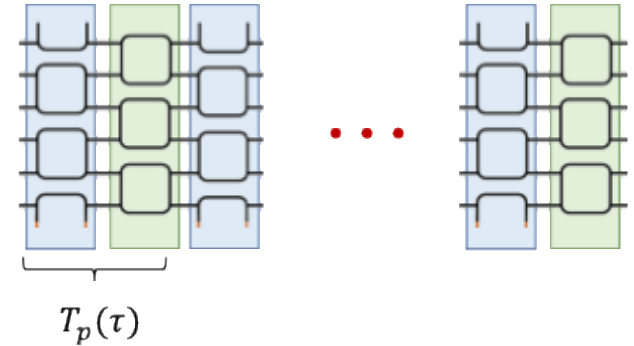
局所性以外にも特定のクラスの物理的性質により高速化の可能性

II. まとめ: 積公式アルゴリズム

▶ まとめ

$$T_p(\tau) = e^{-iH\tau} + \mathcal{O}(\tau^{p+1})$$

$$T_1(\tau) = e^{-iH_2\tau}e^{-iH_1\tau}$$



- Hamiltonian シミュレーションが多項式時間で
実行できることを示した最初の量子アルゴリズム

- 計算コストの系サイズ n , 時刻 t 依存性は理論下限に近い

← “誤差の交換子スケーリング” & Hamiltonian の局所性

- 多くの派生手法や近年でも大きな改善

ランダムコンパイル

E. Campbell, PRL 123, 070503 (2019)

積公式は最も古い手法でありながら今なお標準的で将来有望な量子アルゴリズム

Outline

I. 導入

II. 積公式アルゴリズム

III. Post-Trotter アルゴリズム

III-1. ユニタリ線型結合

III-2. 量子特異値変換/量子固有値変換

IV. 最近の進展と関連する問題

III. Post-Trotter アルゴリズム

▶ 積公式 (前節) (1990s~)

回路深さ $\mathcal{O}\left(t \left(\frac{nt}{\varepsilon}\right)^{\frac{1}{p}}\right)$ ゲート数 $\mathcal{O}\left(nt \left(\frac{nt}{\varepsilon}\right)^{\frac{1}{p}}\right)$

→ 多項式時間. n 依存性は良いが誤差 ε に関しては悪い

Central question: 最も高速な量子アルゴリズムは？

理論下限 $\tilde{\Omega}(nt)$

✓ 実用的な量子計算のため ✓ 量子計算機の性能を限界まで引き出せる？

▶ Post-Trotter アルゴリズム (2010s~)

・ ユニタリ線形結合法 D. W. Berry, et al., PRL114, 090502 (2015). → これから

・ 量子特異値変換 A. Gilyén, et al., STOC (2019). → ラフに紹介

コストが ε に関して指数的に改善 & 時間 t についても理論上ベスト

III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合とは Linear Combination of Unitaries (LCU)

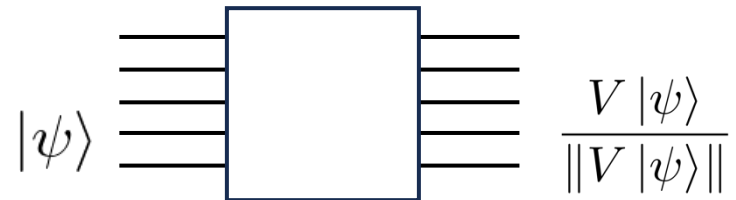
→ 読んで字の如く、ユニタリ演算子の線型結合で書ける演算子

$$V = \sum_{j=1}^J c_j U_j, \quad c_j > 0, \quad U_j : \text{unitary}$$

量子アルゴリズムとして次の操作を指す.

状態 $|\psi\rangle$ が与えられた時

$$V |\psi\rangle = \sum_{j=1}^J c_j U_j |\psi\rangle$$



を出力する

III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合とは Linear Combination of Unitaries (LCU)

例. 量子多体系の local Hamiltonian

$$H = \sum_{j=1}^{\text{poly}(n)} c_j P_j = \sum_{j=1}^{\text{poly}(n)} \underbrace{|c_j|}_{c_j} [\underbrace{\text{sign}(c_j) P_j}_{U_j}]$$

Locality により $J = \text{poly}(n)$ 個の線形結合, U_j は $\mathcal{O}(1)$ -qubit gate

例2. その他いろいろ

$$\cdot H^2 = \sum_{j,j'} c_j c_{j'} U_j U_{j'}$$

$$\cdot \text{時間発展演算子 } e^{-iHt} = \sum_{q=0}^{\infty} \frac{(-iHt)^q}{q!} = \sum_{q=0}^{\infty} \sum_{j_1, \dots, j_q} \frac{(-it)^q}{q!} \underbrace{c_{j_1} \dots c_{j_q}}_{c_j} \underbrace{U_{j_1} \dots U_{j_q}}_{U_j}$$

ただし、個数無限

$$V \cong e^{-iHt} \text{ に対して } V|\psi\rangle = \sum_{j=1}^J c_j U_j |\psi\rangle \text{ ができれば Hamiltonian シミュレーション}$$

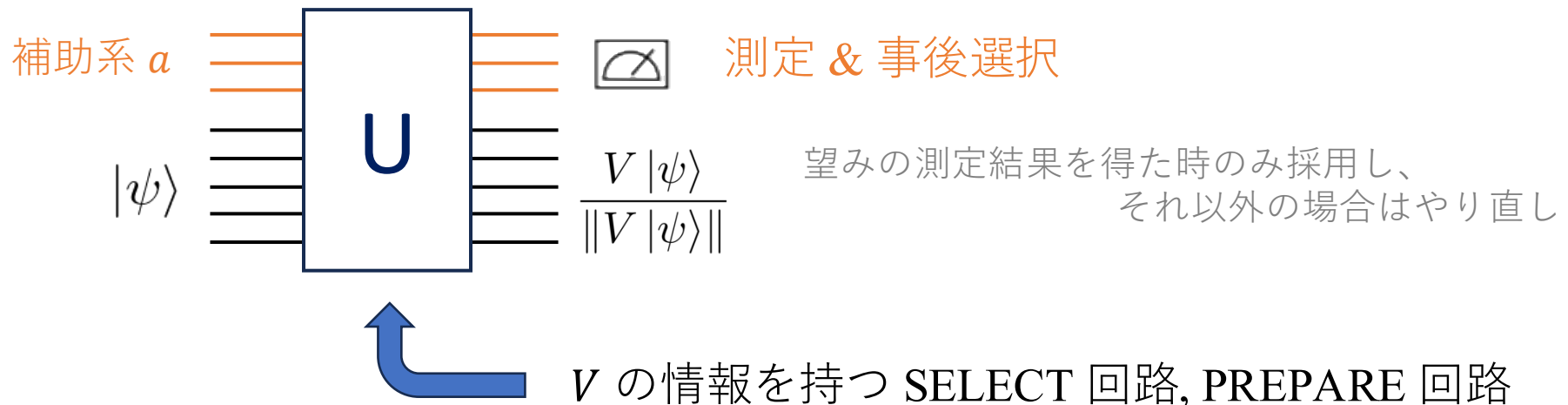
III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装

$$V|\psi\rangle = \sum_{j=1}^J c_j U_j |\psi\rangle \quad \text{をどうやって出力する?}$$

一般に非ユニタリ

“補助系の用意 & 補助系への射影測定で非ユニタリな演算を実行”



III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装

$$V = \sum_{j=1}^J c_j U_j, \quad c_j > 0, \quad U_j : \text{unitary}$$

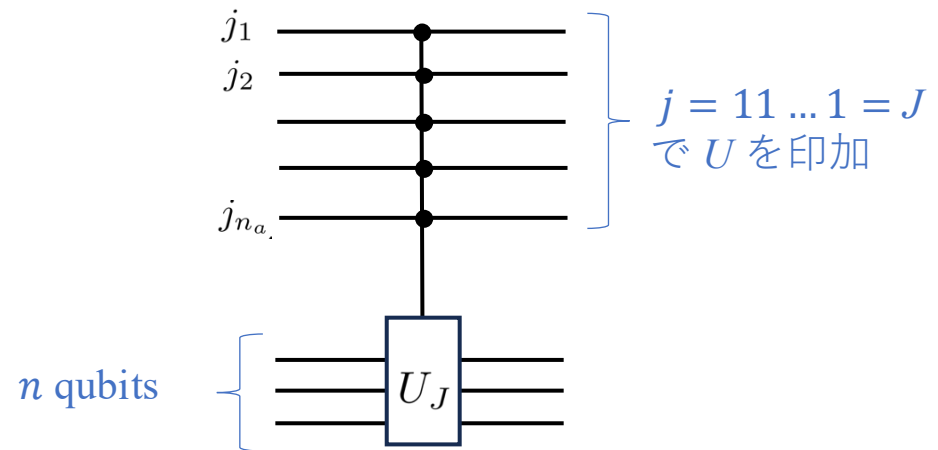
J 個の自由度をもつ補助系 a を用意し次の量子回路を準備

SELECT 回路 “ユニタリ部分の情報を持つ”

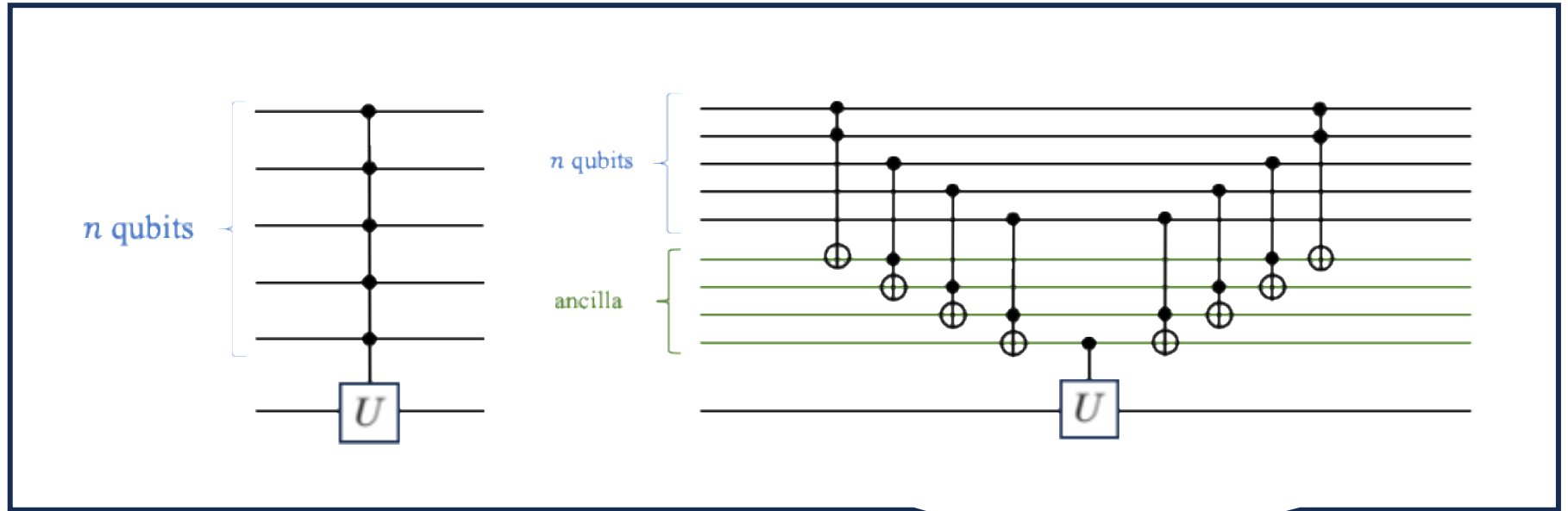
$$S_V = \sum_{j=1}^J |j\rangle \langle j|_a \otimes U_j \quad \rightarrow \text{多重制御ゲート}$$

状態 j の時 U_j

$$|j\rangle_a = \underbrace{|j_1 j_2 \cdots j_{n_a}\rangle}_{\lceil \log_2 J \rceil \text{ qubits}}, \quad j_d = 0, 1$$

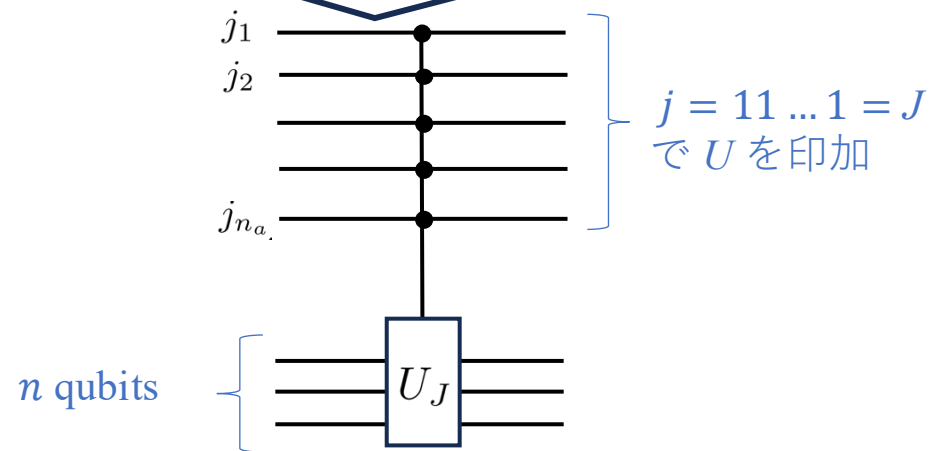


III-1. ユニタリ線形結合 (LCU)



状態 j の時 U_j

$$|j\rangle_a = |\underbrace{j_1 j_2 \cdots j_{n_a}}_{\lceil \log_2 J \rceil \text{ qubits}}\rangle, \quad j_d = 0, 1$$



III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装

$$V = \sum_{j=1}^J c_j U_j, \quad c_j > 0, \quad U_j : \text{unitary}$$

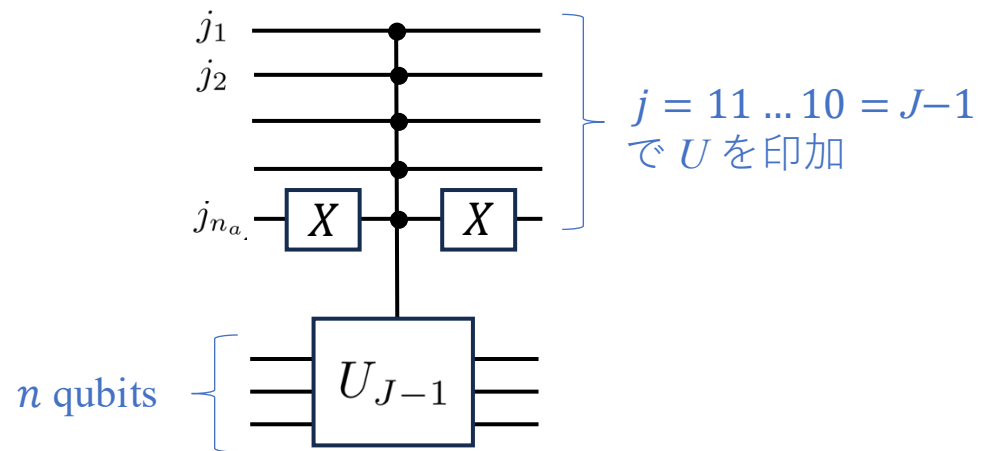
J 個の自由度をもつ補助系 a を用意し次の量子回路を準備

SELECT 回路 “ユニタリ部分の情報を持つ”

$$S_V = \sum_{j=1}^J |j\rangle \langle j|_a \otimes U_j \rightarrow \text{多重制御ゲート}$$

状態 j の時 U_j

$$|j\rangle_a = \left| \underbrace{j_1 j_2 \cdots j_{n_a}}_{\lceil \log_2 J \rceil \text{ qubits}} \right\rangle, \quad j_d = 0, 1$$



III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装

$$V = \sum_{j=1}^J c_j U_j, \quad c_j > 0, \quad U_j : \text{unitary}$$

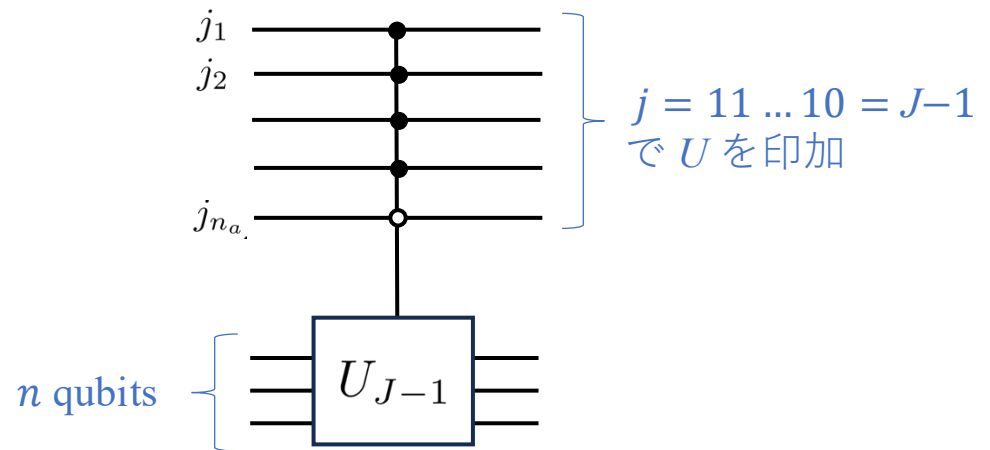
J 個の自由度をもつ補助系 a を用意し次の量子回路を準備

SELECT 回路 “ユニタリ部分の情報を持つ”

$$S_V = \sum_{j=1}^J |j\rangle \langle j|_a \otimes U_j \quad \rightarrow \text{多重制御ゲート}$$

状態 j の時 U_j

$$|j\rangle_a = \left| \underbrace{j_1 j_2 \cdots j_{n_a}}_{\lceil \log_2 J \rceil \text{ qubits}} \right\rangle, \quad j_d = 0, 1$$



III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装

$$V = \sum_{j=1}^J c_j U_j, \quad c_j > 0, \quad U_j : \text{unitary}$$

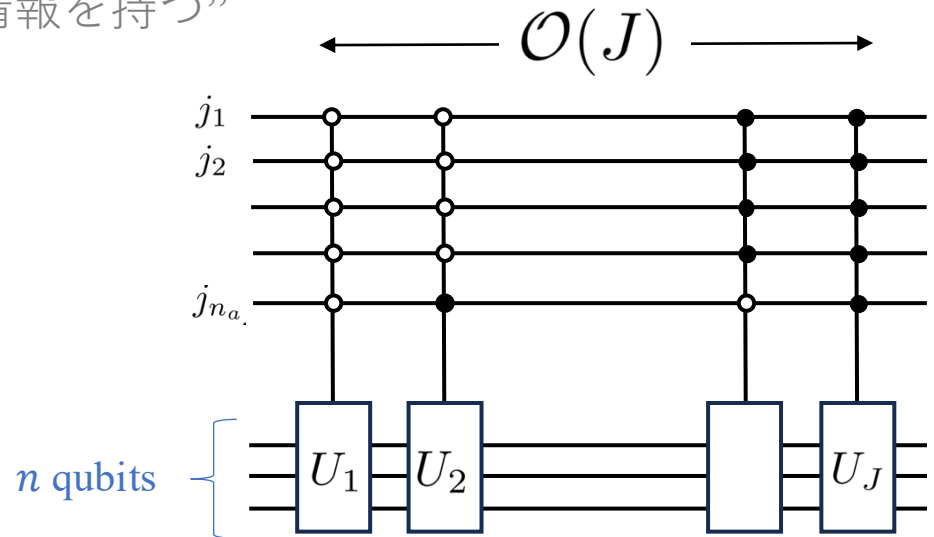
J 個の自由度をもつ補助系 a を用意し次の量子回路を準備

SELECT 回路 “ユニタリ部分の情報を持つ”

$$S_V = \sum_{j=1}^J |j\rangle \langle j|_a \otimes U_j$$

状態 j の時 U_j

$$|j\rangle_a = | \underbrace{j_1 j_2 \cdots j_{n_a}}_{\lceil \log_2 J \rceil \text{ qubits}} \rangle, \quad j_d = 0, 1$$



量子ゲート数: $\mathcal{O}(J) \times [\text{Cost of } U_j]$

III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装

$$V = \sum_{j=1}^J c_j U_j, \quad c_j > 0, \quad U_j : \text{unitary}$$

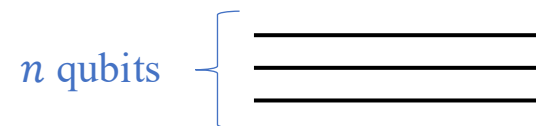
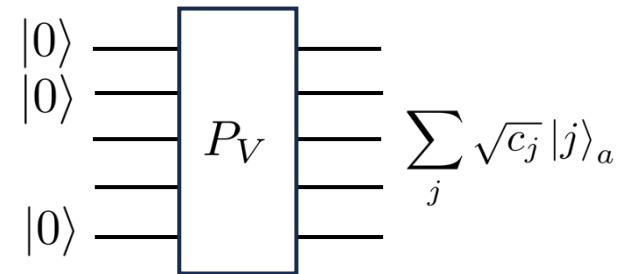
J 個の自由度をもつ補助系 a を用意し次の量子回路を準備

PREPARE 回路 “係数部分の情報を持つ”

P_V : 補助系上の unitary s.t.

$$P_V |0\rangle_a = \sum_{j=1}^J \sqrt{\frac{c_j}{\|c\|_1}} |j\rangle_a, \quad \|c\|_1 = \sum_{j=1}^J |c_j|$$

量子ゲート数: $\mathcal{O}(J)$

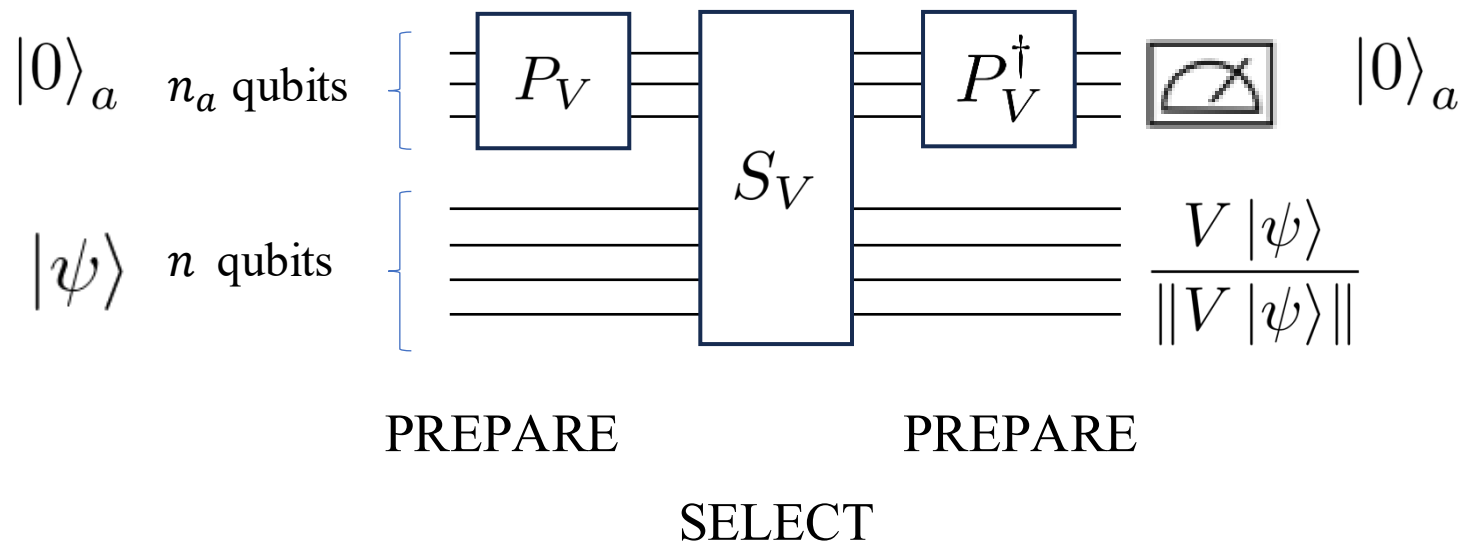


III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装

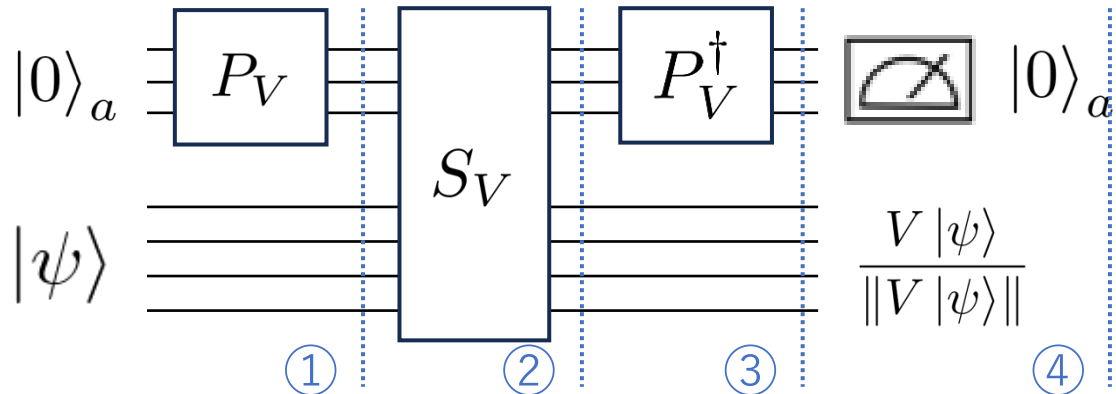
$$V |\psi\rangle = \sum_{j=1}^J c_j U_j |\psi\rangle$$

次の量子回路 & 測定で確率的に変換 $|\psi\rangle \rightarrow \frac{V |\psi\rangle}{\|V |\psi\rangle\|}$ を実行できる



III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装

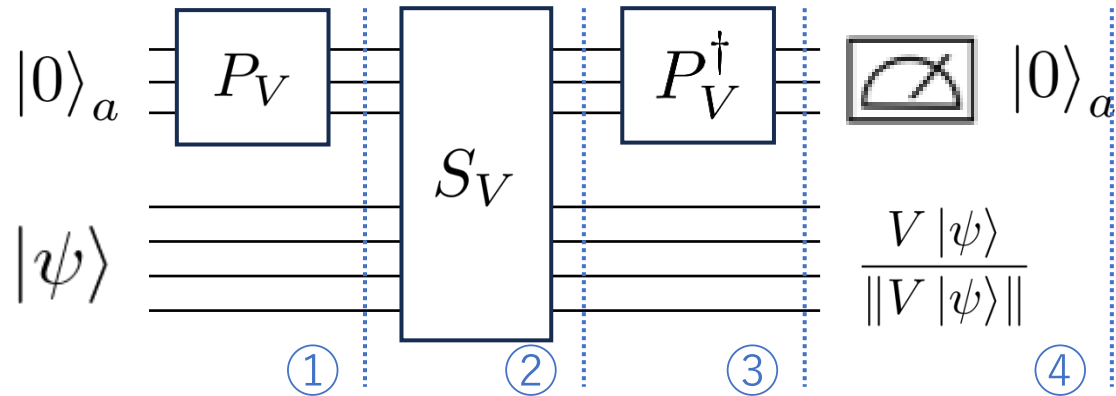


$$\textcircled{1} \quad P_V |0\rangle_a \otimes |\psi\rangle = \sum_{j=1}^J \sqrt{\frac{c_j}{\|c\|_1}} |j\rangle_a \otimes |\psi\rangle$$

$$\textcircled{2} \quad = \left(\underbrace{\sum_{j'} |j'\rangle \langle j'|}_\text{SELECT} \otimes U_{j'} \right) \sum_{j=1}^J \sqrt{\frac{c_j}{\|c\|_1}} |j\rangle_a \otimes |\psi\rangle = \sum_{j=1}^J \sqrt{\frac{c_j}{\|c\|_1}} |j\rangle_a \otimes U_j |\psi\rangle$$

III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装



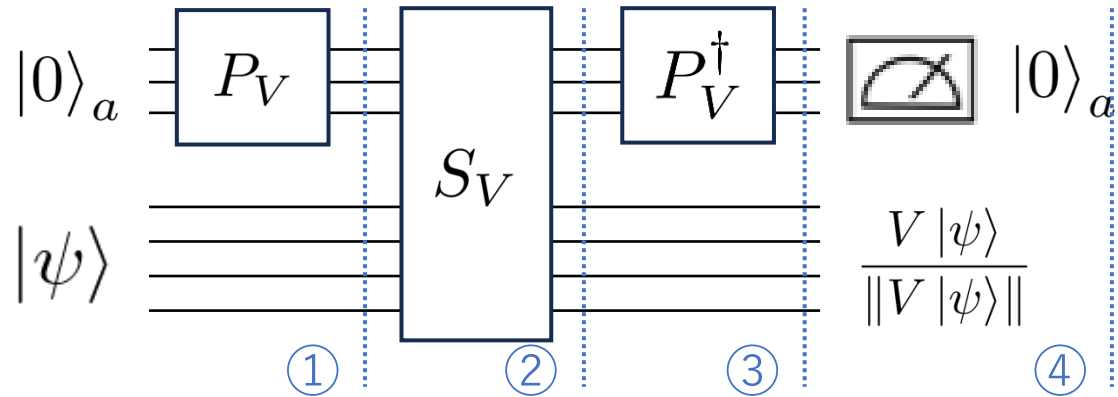
$$\textcircled{3} \quad P_V = \left(\sum_{j=1}^J \sqrt{\frac{c_j}{\|c\|_1}} |j\rangle_a \right) \langle 0|_a + \left(\cdots \right) \langle 1|_a + \left(\cdots \right) \langle 2|_a + \cdots$$

と書けるので

$$P_V^\dagger = |0\rangle_a \left(\sum_{j=1}^J \sqrt{\frac{c_j}{\|c\|_1}} \langle j|_a \right) + |1\rangle_a \left(\cdots \right)^\dagger + |2\rangle_a \left(\cdots \right)^\dagger + \cdots$$

III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装

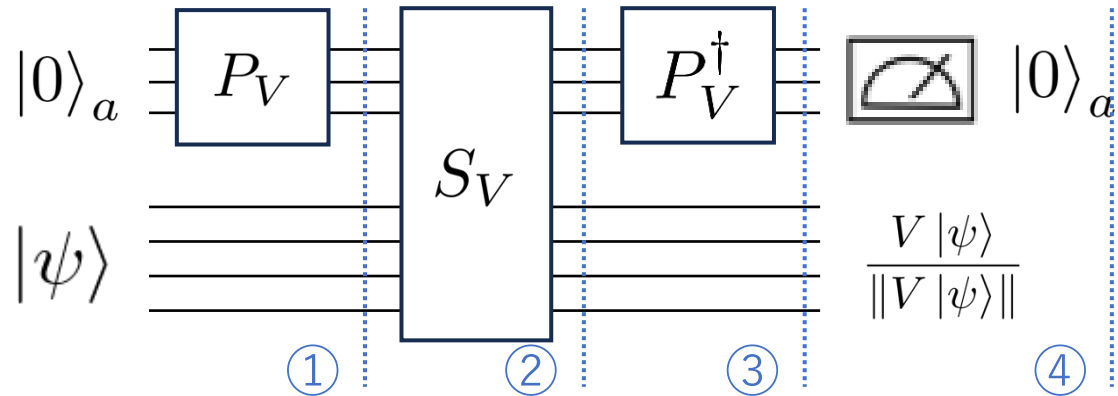


③

$$\begin{aligned}
 &= |0\rangle_a \left(\sum_{j'=1}^J \sqrt{\frac{c_{j'}}{\|c\|_1}} \langle j'|_a \right) \sum_{j=1}^J \sqrt{\frac{c_j}{\|c\|_1}} |j\rangle_a \otimes U_j |\psi\rangle \\
 &\quad + |1\rangle_a \left(\cdots \right) + |2\rangle_a \left(\cdots \right) + \cdots
 \end{aligned}$$

III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装



$$\textcircled{3} \quad = \frac{1}{\|c\|_1} |0\rangle_a \otimes \sum_{j=1}^J c_j U_j |\psi\rangle + |1\rangle_a \left(\cdots \right) + |2\rangle_a \left(\cdots \right) + \cdots$$

V (under the sum) 消したい (under the terms $|1\rangle_a$ and $|2\rangle_a$)

④ 補助系を射影測定し、 $|0\rangle_a$ が得られた場合を事後選択

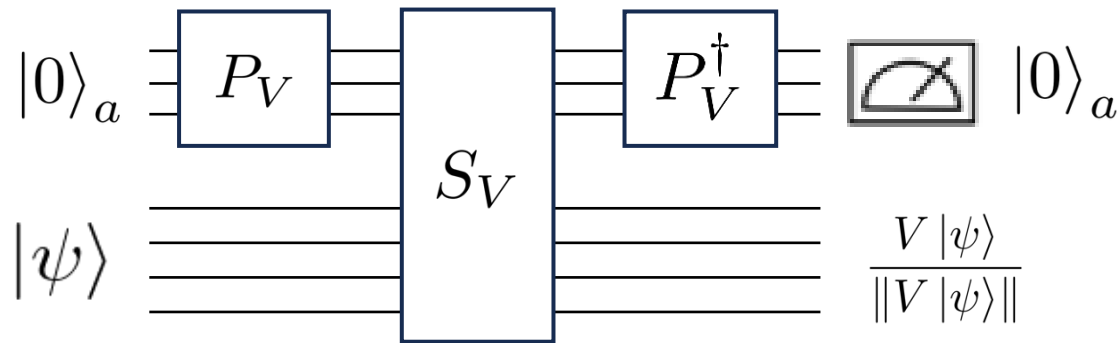
$$\text{測定後: } (|0\rangle \langle 0|_a \otimes I) \left(\textcircled{3}'\text{s state} \right) = |0\rangle_a \otimes \frac{V|\psi\rangle}{\|V|\psi\rangle\|}$$

III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装

$$\text{測定後: } (|0\rangle \langle 0|_a \otimes I) \left(\textcircled{3}'\text{s state} \right) = \frac{V |\psi\rangle}{\|V |\psi\rangle\|}$$

$$\text{測定成功確率: } p_0 = \left\| (|0\rangle \langle 0|_a \otimes I) \left(\textcircled{3}'\text{s state} \right) \right\|^2 = \frac{\|V |\psi\rangle\|^2}{(\|c\|_1)^2}$$



$$\text{必要なゲート数: } \left[\text{PREPARE } (P_V), \text{ SELECT } (S_V) \text{ のゲート数} \right] \times \frac{1}{p_0}$$

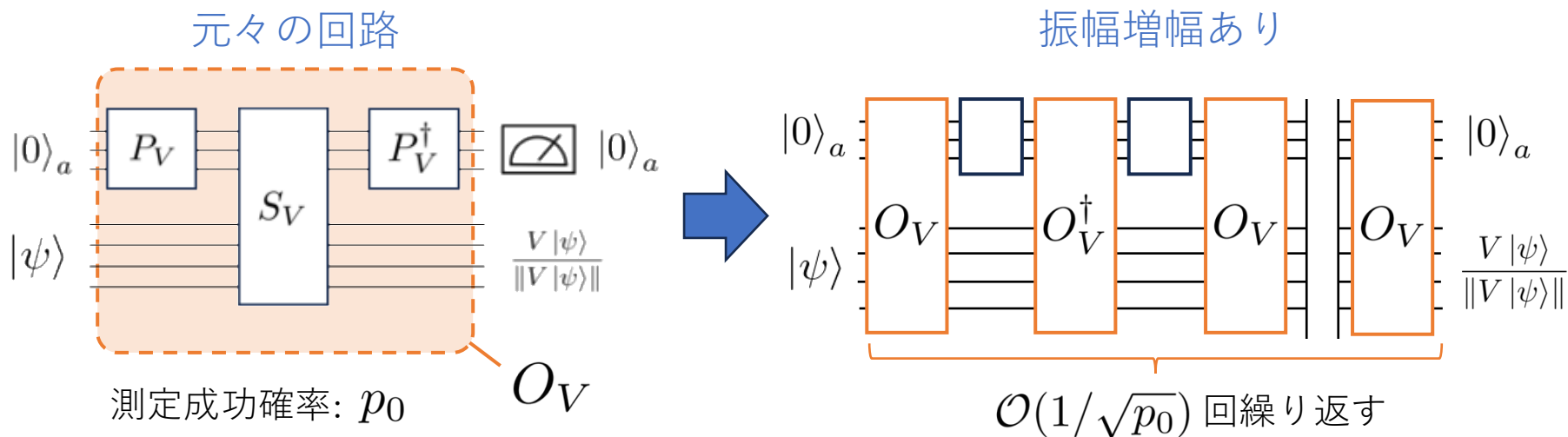
測定成功に必要な回数

III-1. ユニタリ線形結合 (LCU)

▶ ユニタリ線型結合の実装(量子振幅増幅)

“量子振幅増幅” Quantum amplitude amplification G. Brassard, et al., arXiv:quant-ph/0005055 (2000)

量子回路を深くする代わりに成功確率を 1 に増幅

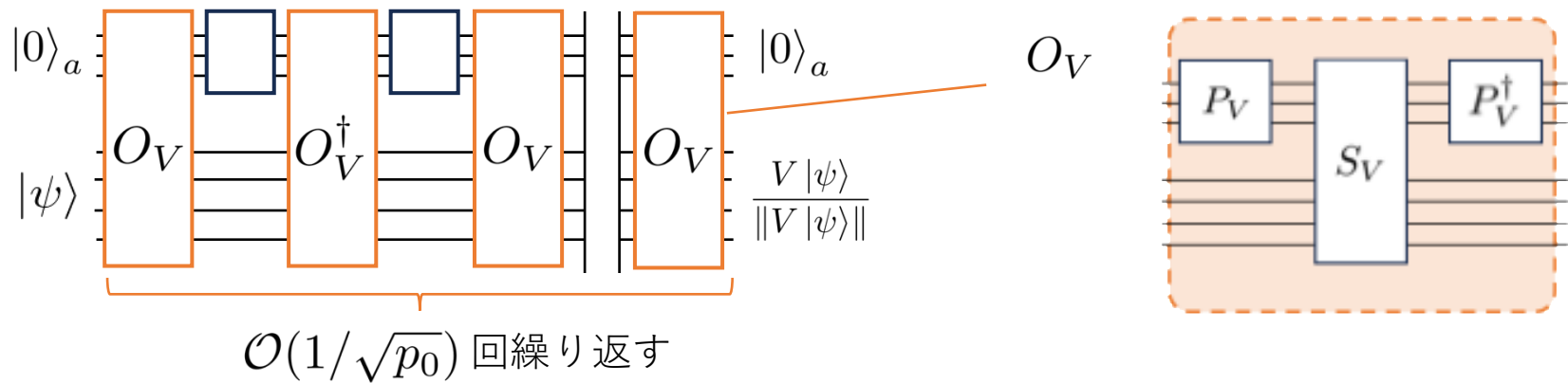


ゲート数: $\left[\text{PREPARE } (P_V), \text{ SELECT } (S_V) \text{ のゲート数} \right] \times \frac{1}{\sqrt{p_0}}$ 2乗加速 (Grover探索)

III-1. ユニタリ線形結合 (LCU)

▶ まとめ: LCU の実装

$$\text{演算 } V|\psi\rangle = \sum_{j=1}^J c_j U_j |\psi\rangle \text{ を実行}$$



$$\text{ゲート数: } \underbrace{\mathcal{O}(J)}_{\text{PREPARE}} + \underbrace{J \times [\text{Cost of } U_j]}_{\text{SELECT}} \times \frac{\|c\|_1}{\|V|\psi\rangle\|} \times \frac{1}{\sqrt{p_0}}$$

個数 J と係数和 $\|c\|_1$ が大きくなければ効率よく実装可能

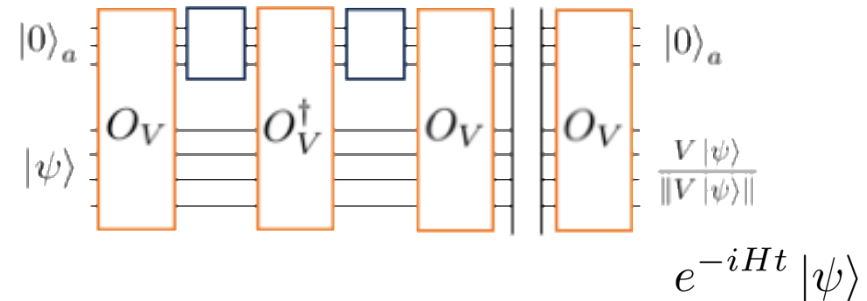
III-1. LCU による Hamiltonian シミュレーション

やりたい事

$$V = \sum_{j=1}^J c_j U_j \simeq e^{-iHt} \text{ として}$$

Hamiltonian シミュレーション

$$|\psi\rangle \rightarrow e^{-iHt} |\psi\rangle$$



▶ セットアップ k -local で有限距離相互作用の Hamiltonian

$$H = \sum_{j=1}^{\mathcal{O}(n)} c_j U_j, \quad \alpha \equiv \|c\|_1 = \sum_{j=1}^{\mathcal{O}(n)} |c_j| \leq gn \in \mathcal{O}(n)$$

g : 1 site 当たりエネルギー

例: 横磁場 Ising 模型 $\{U_j\} = \{Z_j, X_j X_{j+1}\}$

例: Heisenberg 模型 $\{U_j\} = \{X_j X_{j+1}, Y_j Y_{j+1}, Z_j Z_{j+1}\}$

III-1. LCU による Hamiltonian シミュレーション

▶ Hamiltonian の SELECT 回路, PREPARE 回路

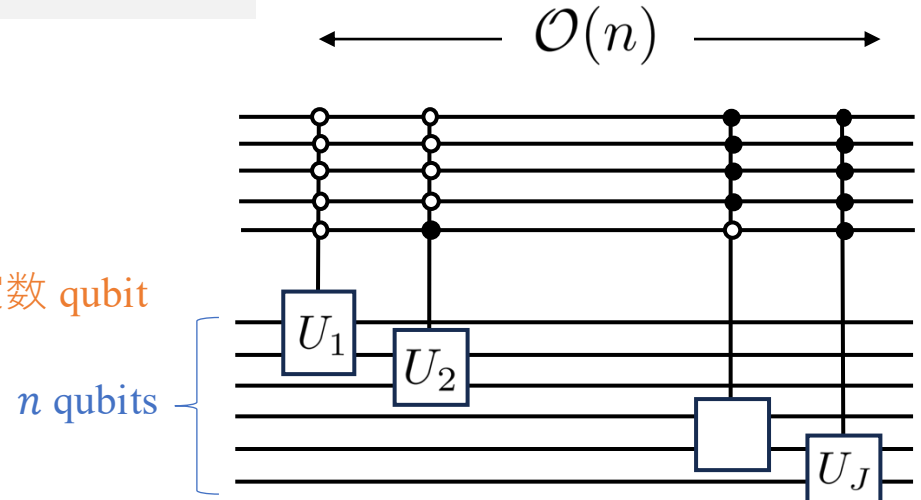
R. Babbush, et al., PRX 8, 041015 (2018)

SELECT回路

$$S_H = \sum_{j=1}^{\mathcal{O}(n)} |j\rangle \langle j|_a \otimes U_j$$

定数 qubit

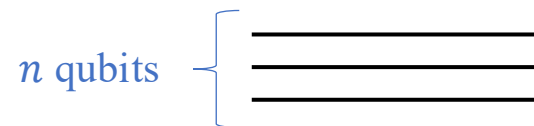
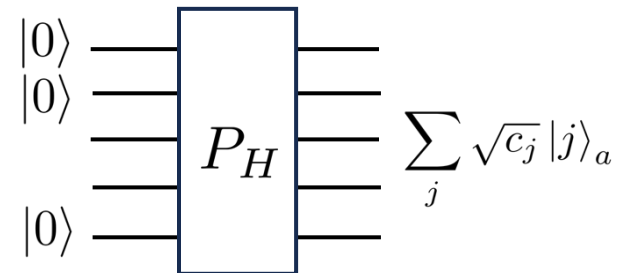
→ $\mathcal{O}(n)$ 個の $\mathcal{O}(1)$ -qubit gate



PREPARE回路

$$P_H |0\rangle_a = \sum_{j=1}^{\mathcal{O}(n)} \sqrt{\frac{c_j}{\alpha}} |j\rangle_a$$

→ $\mathcal{O}(n)$ 個の $\mathcal{O}(1)$ -qubit gate



III-1. LCU による Hamiltonian シミュレーション

▶ Hamiltonian simulation の実行

$$e^{-iHt} = \sum_{q=0}^{\infty} \frac{(-iHt)^q}{q!} = \sum_{q=0}^{\infty} \sum_{j_1, \dots, j_q} \underbrace{\frac{(-it)^q}{q!} c_{j_1} \cdots c_{j_q}}_{c'_j} \underbrace{U_{j_1} \cdots U_{j_q}}_{U'_j}$$

として、 $V|\psi\rangle = \sum_j c'_j U'_j |\psi\rangle$ ができれば ok

問題点1 係数和が指数的に大きい $\alpha \equiv \sum_{j=1}^{\mathcal{O}(n)} |c_j| \leq gn$ g : 1 site 当たりエネルギー

$$\|c'\|_1 = \sum_{q=0}^{\infty} \sum_{j_1, \dots, j_q} \frac{t^q}{q!} |c_{j_1} \cdots c_{j_q}| = e^{\alpha t} \rightarrow \text{ゲート数} \propto \|c'\|_1$$

問題点2 線型結合の個数が無限

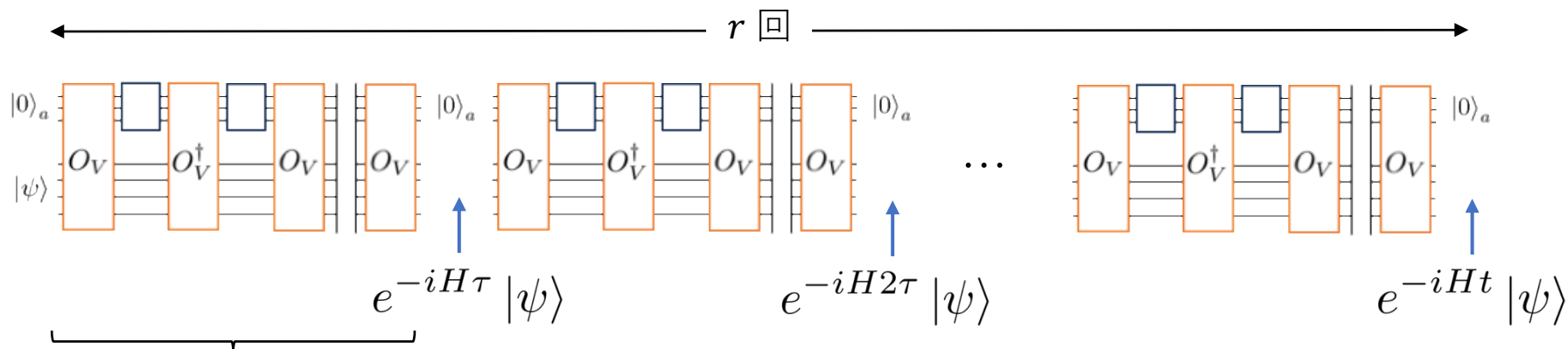
次数 q が無限に続く $\rightarrow$ ゲート数 $\propto$ 個数 J

III-1. LCU による Hamiltonian シミュレーション

▶ Hamiltonian simulation の実行

問題点1 係数和が指数的に大きい $\|c'\|_1 = \sum_{q=0}^{\infty} \sum_{j_1, \dots, j_q} \frac{t^q}{q!} |c_{j_1} \cdots c_{j_q}| = e^{\alpha t}$

→ 時間 t を r 分割して $\tau = t/r$ の時間発展を繰り返す



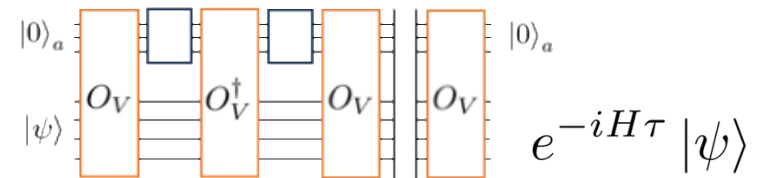
係数和 $\|c'\|_1 = e^{\alpha\tau} = e^{\alpha t/r}$ が定数となるよう設定: $r = \alpha t$

→ 実装コスト: $r = \alpha t \in \mathcal{O}(gnt)$ に比例するように

$$\alpha \equiv \sum_{j=1}^{\mathcal{O}(n)} |c_j| \leq gn$$

III-1. LCU による Hamiltonian シミュレーション

▶ Hamiltonian simulation の実行



問題点2 線型結合の個数が無限

$$e^{-iH\tau} = \sum_{q=0}^{\infty} \frac{(-iH\tau)^q}{q!} \rightarrow \text{有限の打ち切り次数 } Q \text{ (十分大きい) を導入}$$

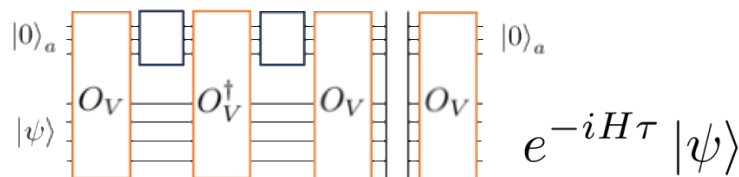
$$V = \sum_{q=0}^Q \frac{(-iH\tau)^q}{q!} = \sum_{q=0}^Q \sum_{j_1, \dots, j_q} \frac{(-i\tau)^q}{q!} \underbrace{c_{j_1} \cdots c_{j_q} U_{j_1} \cdots U_{j_q}}_{\mathcal{O}(n)}$$

$$H = \sum_{j=1}^{\mathcal{O}(n)} c_j U_j \text{ の情報を } Q \text{ 回}$$

$$\begin{aligned} V &\simeq e^{-iH\tau} \\ \text{の実装コスト: } &\left[\text{PREPARE } (P_V), \text{ SELECT } (S_V) \text{ のゲート数} \right] \times \frac{\|c'\|_1}{\|V|\psi\rangle\|} \\ &\frac{1}{\sqrt{p_0}} \in \mathcal{O}(1) \end{aligned}$$

III-1. LCU による Hamiltonian シミュレーション

▶ Hamiltonian simulation の実行



問題点2 線型結合の個数が無限

$$e^{-iH\tau} = \sum_{q=0}^{\infty} \frac{(-iH\tau)^q}{q!} \rightarrow \text{有限の打ち切り次数 } Q \text{ (十分大きい) を導入}$$

$$V = \sum_{q=0}^Q \frac{(-iH\tau)^q}{q!} = \sum_{q=0}^Q \sum_{j_1, \dots, j_q} \frac{(-i\tau)^q}{q!} \underbrace{c_{j_1} \cdots c_{j_q} U_{j_1} \cdots U_{j_q}}_{\text{H の情報を } Q \text{ 回}}$$

$$H = \sum_{j=1}^{\mathcal{O}(n)} c_j U_j \text{ の情報を } Q \text{ 回}$$

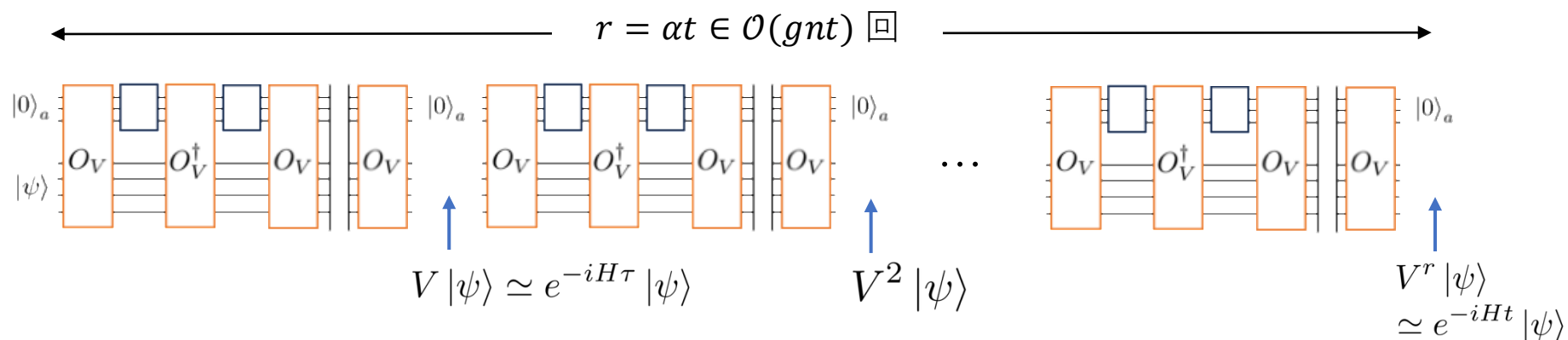
$$V \simeq e^{-iH\tau}$$

の実装コスト: $\left[\text{PREPARE } (P_H), \text{ SELECT } (S_H) \text{ のゲート数} \right] \times Q \in \mathcal{O}(nQ)$

III-1. LCU による Hamiltonian シミュレーション

▶ Hamiltonian simulation の実行

アルゴリズムと計算コスト



量子ゲート数: $r \times \mathcal{O}(nQ) = \mathcal{O}(gn^2tQ)$

計算コストに現れる次数 Q はどう決定？

← 精度保証 ε : $\sup_{|\psi\rangle} (\|e^{-iHt}|\psi\rangle - V^r|\psi\rangle\|) \leq \varepsilon, \quad V = \sum_{q=0}^Q \frac{(-iH\tau)^q}{q!} \simeq e^{-iH\tau}$

III-1. LCU による Hamiltonian シミュレーション

▶ 計算コストの決定

“計算コストの決定” $\approx$ “量子アルゴリズムの誤差の決定”

$$r \times \left\| e^{-iH\tau} - \sum_{q=0}^Q \frac{(-iH\tau)^q}{q!} \right\| \leq \text{Const.} \times \varepsilon \quad \text{を満たす次数 } Q$$

1回分の V による誤差

誤差上限の評価

$$\begin{aligned} \left\| e^{-iH\tau} - \sum_{q=0}^Q \frac{(-iH\tau)^q}{q!} \right\| &\leq \sum_{q=Q+1}^{\infty} \frac{(\|H\|\tau)^q}{q!} \leq \sum_{q=Q+1}^{\infty} \frac{(\alpha t/r)^q}{q!} \\ &= \sum_{q=Q+1}^{\infty} \frac{1}{q!} \leq \frac{1}{Q!} \end{aligned}$$

$\|H\| \leq \sum_j |c_j| = \alpha$
 $r = \alpha t$
相互作用の局所性使わず

III-1. LCU による Hamiltonian シミュレーション

▶ 計算コストの決定

“計算コストの決定” $\approx$ “量子アルゴリズムの誤差の決定”

$$gnt \times \frac{1}{Q!} \leq \text{Const.} \times \varepsilon \quad \text{を満たす次数 } Q \quad \Rightarrow \quad Q \in \mathcal{O}\left(\frac{\log(gnt/\varepsilon)}{\log \log(gnt/\varepsilon)}\right)$$

最終的な量子ゲート数 (有限距離相互作用系)

$$\mathcal{O}(gn^2tQ) = \mathcal{O}\left(n^2t \frac{\log(nt/\varepsilon)}{\log \log(nt/\varepsilon)}\right)$$

“Post-Trotter アルゴリズム”

時間 t に関して線形 $\rightarrow$ 理論上ベスト / 精度 ε に対して対数的

III-1. 積公式との比較とまとめ

実装する近似式

近似誤差

ゲート数

ユニタリ
線型結合

Post-Trotter

D. W. Berry, et al.,
PRL. (2015).

多項式近似

$$e^{-iH\tau} \simeq \sum_{q=0}^Q \frac{(-iH\tau)^q}{q!} \quad \mathcal{O}\left(nt \times \frac{1}{Q!}\right) \quad \mathcal{O}\left(n^2 t \frac{\log(nt/\varepsilon)}{\log \log(nt/\varepsilon)}\right)$$

t 依存性が良い(理論下限) / ε 依存性が指数的に良い

← 多項式近似の誤差が指数的に高速な減衰

積公式
(前節)

A. M. Childs, et al,
PRX (2021)

$$e^{-iH\tau} \simeq e^{-iH_2\tau} e^{-iH_1\tau} \quad \mathcal{O}\left(\frac{nt^{p+1}}{r^p}\right) \quad \mathcal{O}\left((nt)^{1+\frac{1}{p}} \varepsilon^{-\frac{1}{p}}\right)$$

n 依存性が良い (理論下限に近い)

← 量子多体系の局所性を反映する交換子スケーリング

結論:

どちらが良い? → 時と場合による (n, t, ε 次第)

Outline

I. 導入

II. 積公式アルゴリズム

III. Post-Trotter アルゴリズム

III-1. ユニタリ線型結合

III-2. 量子特異値変換/量子固有値変換

IV. 最近の進展と関連する問題

III-2. 量子特異値変換 / 量子固有値変換

► Post-Trotter アルゴリズム

ユニタリ
線型結合

D. W. Berry, et al.,
PRL. (2015).

実装する近似式

$$e^{-iH\tau} \simeq \sum_{q=0}^Q \frac{(-iH\tau)^q}{q!}$$

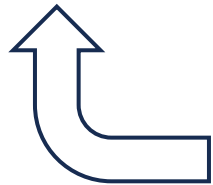
$\tau = t/r$ を r 回

使う量子回路

SELECT (S_H)

PREPARE ($P_H, P_H^\dagger$)

量子特異値変換
/ 量子固有値変換



$$e^{-iHt} \simeq \sum_{q=0}^Q \frac{(-iHt)^q}{q!}$$

ブロック埋め込み

O_H

一般の多項式変換 $|\psi\rangle \rightarrow f(H) |\psi\rangle$ を

を統一的な手法でかけるための量子アルゴリズム

H が一般の行列 $\rightarrow$ 量子特異値変換, H がエルミート行列 $\rightarrow$ 量子固有値変換

III-2. 量子特異値変換 / 量子固有値変換

▶ ブロック埋め込み 量子特異値変換を構成する基本回路(oracle)

定義

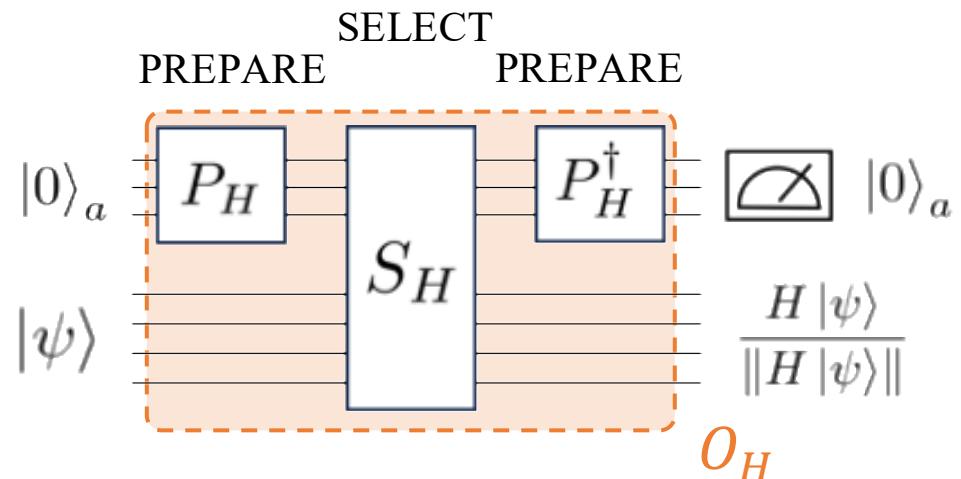
行列 H をブロック成分
にもつユニタリ行列

$$O_H = \begin{pmatrix} H/\alpha & * \\ * & * \end{pmatrix} \begin{matrix} |0\rangle_a \otimes I \\ \langle 0|_a \otimes I \end{matrix}$$

$$H = \sum_j c_j U_j \text{ (LCU) の場合}$$

$$|\psi\rangle \rightarrow \sum_j c_j U_j |\psi\rangle$$

を実装する量子回路と一緒



その他の場合: 疎行列 H , 純粋化が既知な密度行列 ρ G. H. Low, et al., Quantum 3, 163 (2019).

III-2. 量子特異値変換 / 量子固有値変換

▶ 量子固有値変換 H : エルミート行列

定理.

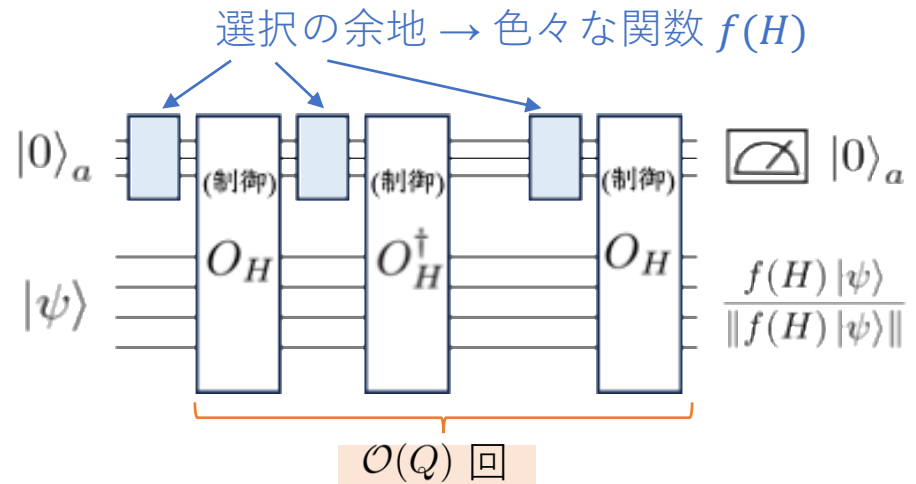
A. Gilyén, et al., STOC (2019)

(制御)ブロック埋め込み $O_H, O_H^\dagger$ を
 $\mathcal{O}(Q)$ 回使う量子回路を考える。

任意の Q 次多項式 $f(x)$ s.t. $|f(x)| \leq 1, \forall x \in [-1, 1]$ に対して次が可能。

$$|\psi\rangle \rightarrow \frac{f(H) |\psi\rangle}{\|f(H) |\psi\rangle\|}$$

成功確率: $p_0 \propto \|f(H) |\psi\rangle\|^2$



なぜ量子固有値変換とよぶ?

$$f(H) = \sum_i f(\lambda_i) |\lambda_i\rangle \langle \lambda_i|$$

H の固有値・固有ベクトル

III-2. 量子特異値変換 / 量子固有値変換

► Hamiltonian シミュレーション

多項式を選択

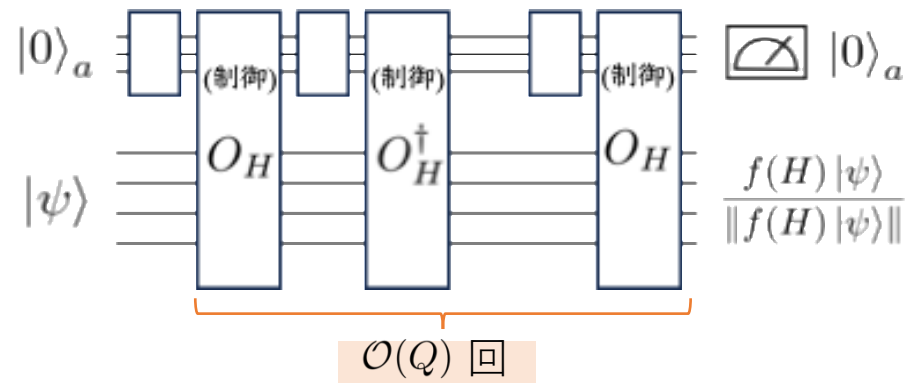
$$f(H) = \sum_{q=0}^Q \frac{(-iHt)^q}{q!} \simeq e^{-iHt}$$

→ 誤差 $< \varepsilon$ の保証に基づき次数 Q を決定: $Q \in \mathcal{O}\left(\underbrace{\alpha t}_{\mathcal{O}(nt)} + \frac{\log(1/\varepsilon)}{\log \log(1/\varepsilon)}\right)$

量子ゲート数

$$\left[\text{ブロック埋め込み } O_H, O_H^\dagger \text{ のゲート数} \right] \times Q \in \mathcal{O}(n[nt + \log(1/\varepsilon)])$$

: PREPARE, SELECT と同じ $\mathcal{O}(n)$



III-2. 量子特異値変換 / 量子固有値変換

▶ アルゴリズム比較

ゲート数

積公式

$$\mathcal{O}((nt)^{1+\frac{1}{p}} \varepsilon^{-\frac{1}{p}})$$

n 依存性が良い

A. M. Childs, et al, PRX (2021)

ユニタリ 線型結合

$$\mathcal{O}\left(n^2 t \frac{\log(nt/\varepsilon)}{\log \log(nt/\varepsilon)}\right)$$

D. W. Berry, et al., PRL. (2015).

Post-Trotter アルゴリズム
 t 依存性が理論上ベスト
 ε 依存性が指数的に良い

量子特異値変換

$$\mathcal{O}(n[nt + \log(1/\varepsilon)])$$

A. Gilyén, et al., STOC (2019)

Black-box H に対しては optimal
(但し、局所性を反映しない)

量子特異値変換:

少ない計算コスト

ユニタリ線型結合:

高い拡張性 (時間依存系, 量子開放系, etc.)

G. H. Low, et al., arXiv:1805.00675

III-3. 余談: 量子アルゴリズムの大統一

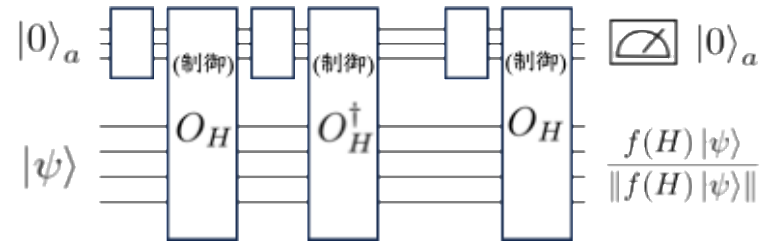
▶ 量子アルゴリズムの大統一

“Grand unification of quantum algorithms”

J. M. Martyn, et al., PRX Quantum 2, 040203 (2021)

$$|\psi\rangle \rightarrow f(H) |\psi\rangle$$

$f(H)$: 任意の Q 次多項式関数



様々な多項式関数 $f(H)$ の選択によって

様々な量子アルゴリズムのタスクを統一的に実行可能

例0: Hamiltonian シミュレーション 指数加速

多項式

$$f(H) = \sum_{q=0}^Q \frac{(-iHt)^q}{q!} \simeq e^{-iHt}$$

計算コスト
($\propto$ 次数 Q)

$$Q \in \mathcal{O}(\alpha t + \log(1/\varepsilon))$$

ブロック埋め込み O_H の呼び出し回数

III-3. 余談: 量子アルゴリズムの大統一

▶ 様々なアルゴリズムの実行

例1: Gibbs サンプリング

$$\rho_\beta = \frac{e^{-\beta H}}{Z_\beta} \quad \text{熱平衡状態の準備}$$

: Bell pair $|\Psi_+\rangle^{\otimes n}$ に $|\Psi_+\rangle^{\otimes n} \rightarrow (e^{-\beta(H+I)/2} \otimes I) |\Psi_+\rangle^{\otimes n}$ で実行

多項式

Jacobi-Anger 展開

$$f(H) = e^{-\beta} \left(I_0(\beta) + 2 \sum_{q=0}^Q (-1)^q I_q(\beta) T_q(H) \right) \simeq e^{-\beta(H+I)}$$

計算コスト
($\propto$ 次数 Q)

$$Q \in \tilde{O} \left(\beta \alpha \sqrt{\frac{2^n}{Z_\beta}} \right)$$

III-3. 余談: 量子アルゴリズムの大統一

▶ 様々なアルゴリズムの実行

例2: 線型方程式問題 指数加速

連立方程式 $A|x\rangle = |b\rangle$ の解を準備

$$|b\rangle \rightarrow |x\rangle = A^{-1} |b\rangle$$

多項式 $f(A) \simeq A^{-1}$

計算コスト
($\propto$ 次数 Q) $Q \in \mathcal{O}(\kappa \log(\kappa/\varepsilon))$
 $\kappa : A$ の条件数

例3: Grover 探索 2 乗加速

多項式 $f(H) = T_Q(H)$
Chebyshev 多項式

例4: 量子位相推定 (固有値計算) 指数加速

多項式 $f(H) \simeq \text{sign}(H)$
符号関数

量子アルゴリズム
の大統一

量子アルゴリズムの種類 = 多項式関数 $f(H)$ の選択

計算コスト = 多項式関数 $f(H)$ の次数 Q

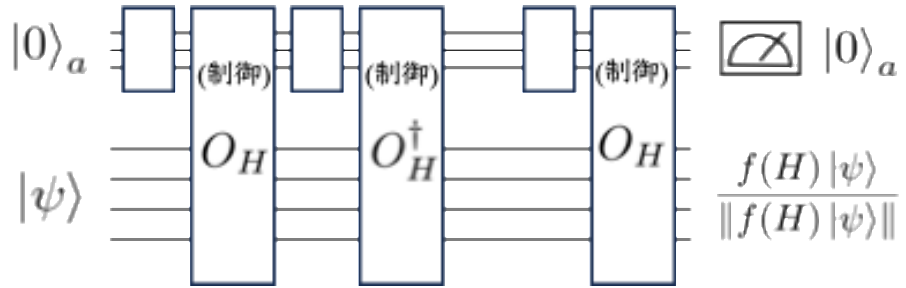
III. まとめ: Post-Trotter アルゴリズム

▶ まとめ

ユニタリ線形結合 / 量子特異値変換

時間発展演算子の多項式近似

$$f(H) = \sum_{q=0}^Q \frac{(-iHt)^q}{q!} \simeq e^{-iHt} \quad \text{を実装} \quad \text{ゲート数} \quad \tilde{O}(n^2t)$$



- ・ 積公式アルゴリズムより指数的に良い ε 依存性 ← 多項式近似の早い収束
- ・ Hamiltonian シミュレーション以外にも様々なタスク(大統一)

“Trotter (積公式) に置き換わると考えられた標準的な量子アルゴリズム”

- ・ 高精度な計算が得意
- ・ 高い汎用性
- ・ 現時点での実装例は少ない (補助系, 長距離ゲート)

Outline

I. 導入

II. 積公式アルゴリズム

III. Post-Trotter アルゴリズム

IV. 最近の進展と関連する問題

IV. 最近の進展

▶ 標準的な量子アルゴリズム

ゲート数

積公式

A. M. Childs, et al, PRX (2021)

$$\mathcal{O}((nt)^{1+\frac{1}{p}} \varepsilon^{-\frac{1}{p}})$$

n 依存性が良い

局所性を反映する
交換子スケーリング

Post-Trotter

D. W. Berry, et al., PRL. (2015).

A. Gilyén, et al., STOC (2019)

$$\mathcal{O}(n[nt + \log(1/\varepsilon)])$$

t 依存性が理論上ベスト
 ε 依存性が指数的に良い

多項式近似の誤差の指数減衰

Goal

理論上可能な最速の量子アルゴリズムは？

n, t, ε 依存性の全てが良い

ゲート数
の理論下限

$$\tilde{\Omega}(nt)$$

J. Haah, et al., SIAM (2021)

→ **Key:** “量子多体系の局所性を反映” & “急速な近似誤差の収束”

IV. 最近の進展

▶ Haah-Hastings-Kothari-Low (HHKL) アルゴリズム

J. Haah, et al., SIAM (2021)

Lieb-Robinson 限界による e^{-iHt} の分解

局所性

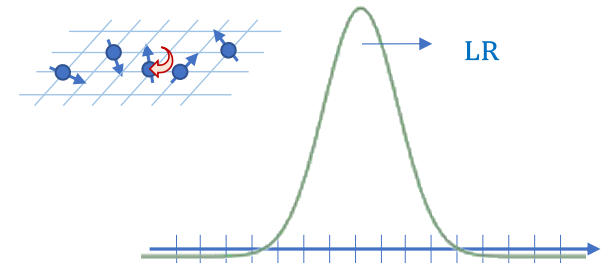
+ サブルーチンとして量子特異値変換

量子ゲート数

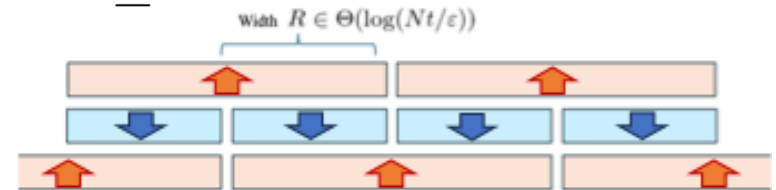
(有限距離相互作用系)

$$nt \times \text{polylog}(nt/\varepsilon)$$

Near-optimal (理論下限 nt と $\log$ の 差)



$$e^{-iHt} \simeq$$



有限距離相互作用系では理論下限のゲート数を達成

ゲート深さを小さくするには大量の補助 qubit 必要、長距離相互作用系では効率悪い

IV. 最近の進展

▶ 複数積公式アルゴリズム

G. H. Low, et al., arXiv:1907.11679 (2019)

積公式アルゴリズムのユニタリ線型結合

Hamiltonianの局所性

多項式近似

$$M_{pJ}(\tau) = \sum_{j=1}^J c_j [T_p(\tau/k_j)]^{k_j} = e^{-iH\tau} + \mathcal{O}(\tau^{p+J})$$

量子ゲート数

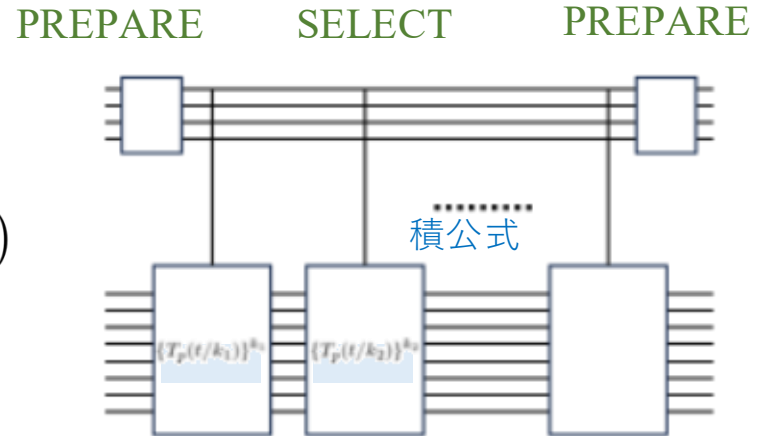
K. Mizuta, Quantum 10, 1974 (2026), AQIS2026

(有限距離相互作用系) 理論下限に近い

$$n^{1+\frac{1}{p+1}} t \times \text{polylog}(nt/\varepsilon)$$

(長距離相互作用系) 有効な計算手法

$$n^{k+\frac{1}{p+1}+1-\frac{\nu}{d}} t \times \text{polylog}(nt/\varepsilon)$$



積公式のような良い系サイズ n 依存性

/ Post-Trotter のような良い時刻 t , 誤差 ε 依存性

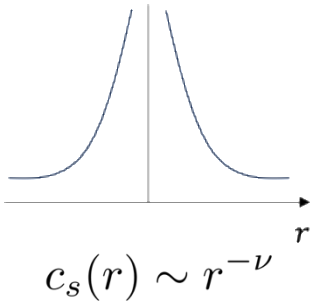
長距離相互作用系でも効率良い

IV. 最近の進展

有限距離相互作用系では n, t, ε について同時に良い
量子計算手法はある程度確立 (HHKL, 複数積公式)。

(多くの ongoing な課題)

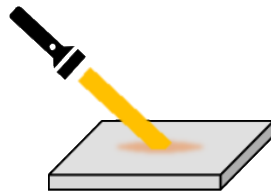
長距離
相互作用系



M. C. Tran, et al,
PRX 9, 031006 (2019).

G. H. Low, et al.,
PRX 4, 020323 (2023).

時間依存系



$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle$$

K. Mizuta, et al.,
Quantum (2023), QIP 2023

K. Mizuta, et al.,
arXiv:2410.14243, QIP 2025

量子開放系

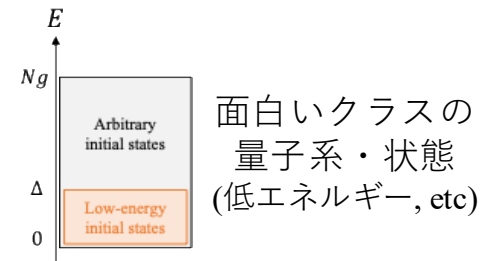
$$\rho(t) = e^{\mathcal{L}t} \rho$$

$$\mathcal{L}\rho = -i[H, \rho] + \sum_{m=1}^M (2L_m \rho L_m^\dagger - \{L_m^\dagger L_m, \rho\})$$

M. Kliesch, et al.,
PRL. **107**, 120501 (2011).

R. Cleve, C. Wang,
arXiv:1612.09512 (2016).

“物理”を反映
した高速化



K. Mizuta, T. Kuwahara,
PRL 135, 130602 (2025)

R. King, et al.,
PRL **136**, 110601 (2026)

(忘れてはいけない課題) スケーリングだけでなく実時間でどこまで高速化？

IV. 関連する話題 (雑多に)

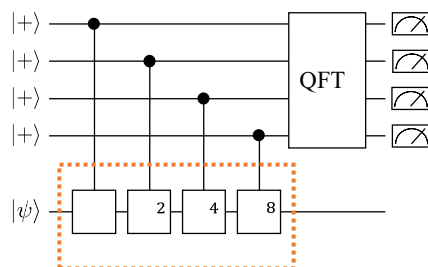
▶ 固有値計算・固有状態準備

$$H |E_n\rangle = E_n |E_n\rangle$$

量子位相推定

サブルーチンとして

$$U = e^{-iH}$$

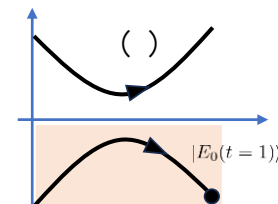


断熱量子状態準備

終状態として

$$H(t) = (1 - t/T)H_0 + (t/T)H_1$$

T. Albash, et al., Rev. Mod. Phys. **90**, 015002 (2018)



▶ 熱平衡状態シミュレーション

虚時間発展

$$e^{-\beta(H_1+H_2)} = e^{-\beta H_2} e^{-\beta H_1} + \mathcal{O}(\beta^2)$$

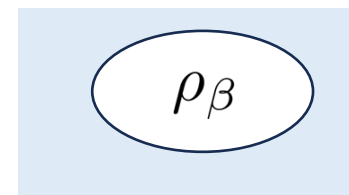
積公式

→ Markov chain Monte Carlo (MCMC)

散逸状態準備 開放系の定常状態として

$$\lim_{t \rightarrow \infty} e^{\mathcal{L}_\beta t} \rho = \rho_\beta$$

L. Lin –
APL Computational Physics, 2025



IV. 関連する話題 (雑多に)

▶ その他にも

線形微分方程式

$$\frac{d}{dt} |\psi(t)\rangle = A(t) |\psi(t)\rangle$$

Hamiltonian シミュレーション線型結合

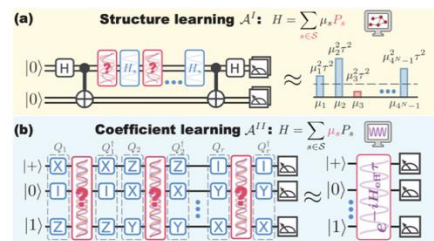
D. An, J.-P. Liu, L. Lin, PRL **131**, 150603 (2023).

Hamiltonian 学習

$$H(\{c_i\}) = \sum_i c_i P_i$$

ダイナミクスから推定

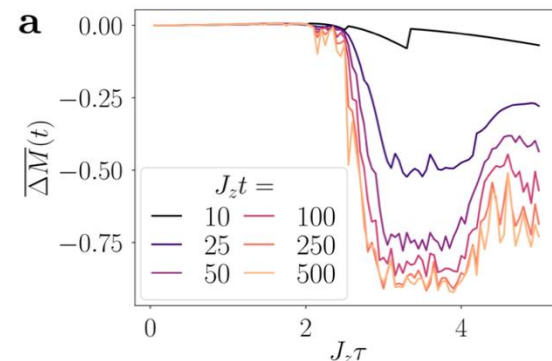
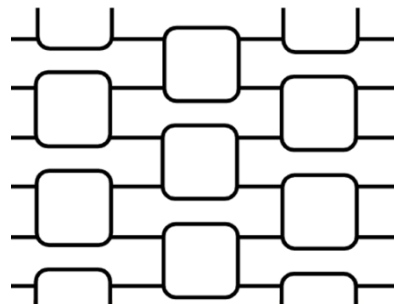
A. Dutkiewicz, et al., Quantum **8**, 1537 (2024).



非平衡現象

積公式タイプの量子回路
における熱化・相転移など
← 積公式の誤差

T. N. Ikeda, et al., PRL **133**, 030401 (2024).



L. M. Sieberer, et al., Npj Quantum Inf. (2019).

IV. 最後に

▶ Hamiltonian シミュレーション

- ✓ 量子ダイナミクスの計算: 古典と比較した指数加速
- ✓ 量子アルゴリズム: 局所性などの物理を使って最善のコストを目指す

量子多体系をシミュレートする
(古典/量子)アルゴリズム

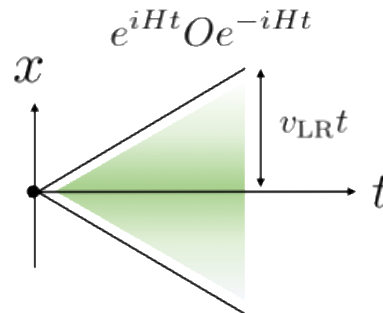
||

“量子多体系の普遍的性質を
的確に表現・理解すること”

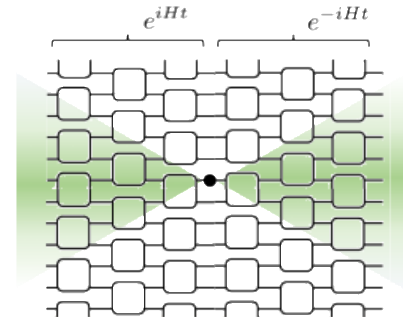
例. Lieb-Robinson 限界

M. C. Tran, et al, PRX (2019) A. M. Childs, et al, PRX (2021)

“量子多体系
の情報伝播”



“時間発展の
ゲート分解”



Appendix:

量子特異値変換

(元々 III の代わりに話す予定でしたが時間の都合で変更したので、ここで供養します)

III-1. 量子特異値変換とは？

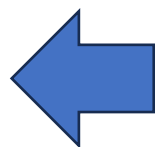
III-2. 定式化 (量子信号処理, ブロック埋め込み, 量子ビット化)

III-3. アルゴリズムと計算複雑性

III-1. 量子特異値変換とは

▶ 量子特異値変換とは

量子特異値変換: 一般の(有限次元)行列 A に対する多項式変換 $f(A)$



拡張

量子固有値変換: 行列 A がエルミート ($= H$) の時の量子特異値変換

以降では、簡単のため量子固有値変換について議論する

III-1. 量子特異値変換とは

▶ (復習) 行列多項式

2^n 次元エルミート行列 $H = \sum_{i=1}^{2^n} \lambda_i |\lambda_i\rangle \langle \lambda_i|$, 多項式関数 $f(x)$ に対して、

$$f(H) = \sum_{i=1}^{2^n} f(\lambda_i) |\lambda_i\rangle \langle \lambda_i|$$

例: 指数関数 $f(x) = e^{-ixt} = \sum_{n=0}^{\infty} \frac{(-ixt)^n}{n!}$

$$f(H) = e^{-iHt} = \sum_j \left(\sum_{n=0}^{\infty} \frac{(-i\lambda_j t)^n}{n!} \right) |\lambda_j\rangle \langle \lambda_j| = \sum_{n=0}^{\infty} \frac{(-iHt)^n}{n!}$$

時間発展演算子

III-1. 量子特異値変換とは

► [定義] 量子固有値変換とは

定義 (量子固有値変換).

2^n 次元エルミート行列 $H = \sum_{i=1}^{2^n} \lambda_i |\lambda_i\rangle \langle \lambda_i|$, 多項式関数 $f(x)$ に対して、

$$|\psi\rangle \rightarrow \frac{f(H) |\psi\rangle}{\|f(H) |\psi\rangle\|}$$

とする変換を量子固有値変換と呼ぶ。
(quantum eigenvalue transformation, QET)



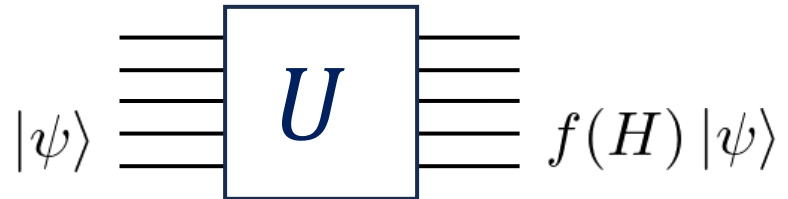
III-1. 量子特異値変換とは

► [定義] 量子固有値変換とは

定義 (量子固有値変換).

2^n 次元エルミート行列 $H = \sum_{i=1}^{2^n} \lambda_i |\lambda_i\rangle \langle \lambda_i|$, 多項式関数 $f(x)$ に対して、

$$|\psi\rangle \rightarrow \frac{f(H) |\psi\rangle}{\|f(H) |\psi\rangle\|}$$



仮に指数関数 $f(x) = e^{-ixt}$ の量子固有値変換ができると...?

$$|\psi\rangle \rightarrow \frac{e^{-iHt} |\psi\rangle}{\|e^{-iHt} |\psi\rangle\|} = e^{-iHt} |\psi\rangle \quad \text{“Hamiltonianシミュレーション達成”}$$

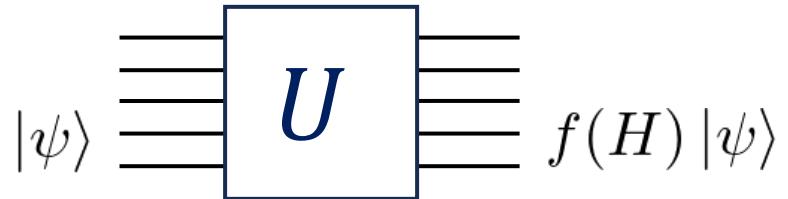
III-1. 量子特異値変換とは

▶ [定義] 量子固有値変換とは

定義 (量子固有値変換).

2^n 次元エルミート行列 $H = \sum_{i=1}^{2^n} \lambda_i |\lambda_i\rangle \langle \lambda_i|$, 多項式関数 $f(x)$ に対して、

$$|\psi\rangle \rightarrow \frac{f(H) |\psi\rangle}{\|f(H) |\psi\rangle\|}$$



実際には...

- ・ 変換が可能な関数 $f(x)$ のクラスは何？
- ・ どのような量子回路で関数 $f(x)$ の変換が可能？
- ・ その変換にかかる量子アルゴリズムのコストはどれくらい？

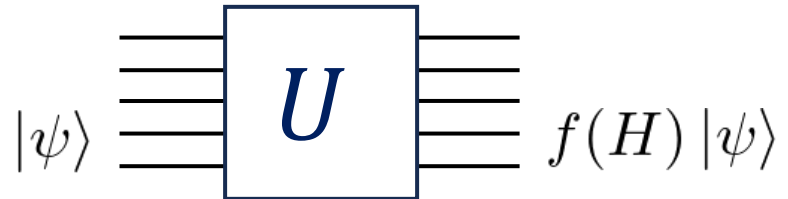
III-1. 量子特異値変換とは

▶ [定義] 量子固有値変換とは

定義 (量子固有値変換).

2^n 次元エルミート行列 $H = \sum_{i=1}^{2^n} \lambda_i |\lambda_i\rangle \langle \lambda_i|$, 多項式関数 $f(x)$ に対して、

$$|\psi\rangle \rightarrow \frac{f(H) |\psi\rangle}{\|f(H) |\psi\rangle\|}$$



まとめ (ネタバレ)

- ・ 変換可能な関数 $f(x)$: 一般の(規格化された)有限次数 d の多項式
- ・ 量子回路の構成: III-2 → ブロック埋め込み, 量子ビット化, 量子信号処理
- ・ 計算コスト: III-3 → 多項式の次数 d に比例

Appendix:

量子特異値変換

(元々 III の代わりに話す予定でしたが時間の都合で変更したので、ここで供養します)

III-1. 量子特異値変換とは？

III-2. 定式化 (量子信号処理, ブロック埋め込み, 量子ビット化)

III-3. アルゴリズムと計算複雑性

III-2. 量子固有値変換の定式化

▶ 定式化の道具

1. 量子信号処理 (Quantum Signal Processing, QSP)

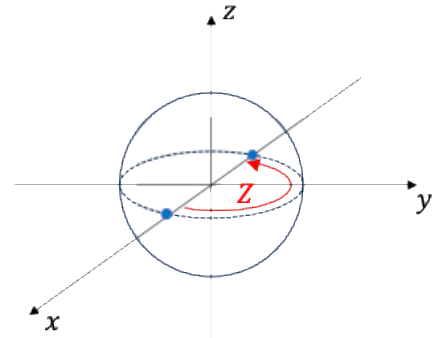
1 qubit 上の Pauli X, Z 回転を合成して、
多項式関数 $f(x)$ の信号を生成する量子制御

2. ブロック埋め込み (Block-encoding)

行列 H の情報をユニタリー回路 O_H に埋め込み

3. 量子ビット化 (qubitization)

ブロック埋め込み O_H を使って 1qubit 系の QSP を再現



III-2.1. 量子信号処理

▶ 唐突ですが 1 qubit 上の回転

Pauli X 回転 x (or θ): 正確には制御できないシグナル

$x = \cos \theta$ ($\theta \in [0, \pi]$) として

$$W(x) \equiv e^{i\theta X} = \cos \theta I + i \sin \theta X = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix}$$

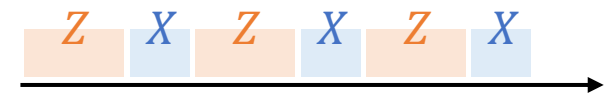
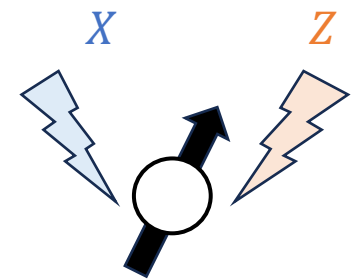
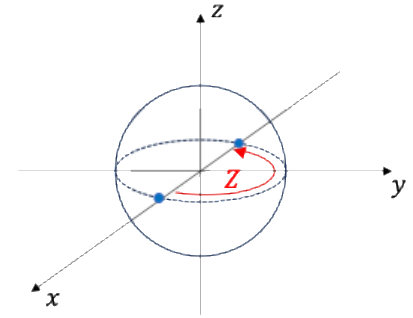
Pauli Z 回転

ϕ : 正確に制御可能なパラメータ

$$e^{i\phi Z} = \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix}$$

量子信号処理とは...

Pauli X 回転と制御可能な Pauli Z 回転
を合成して所望の回転を得る操作



III-2.1. 量子信号処理

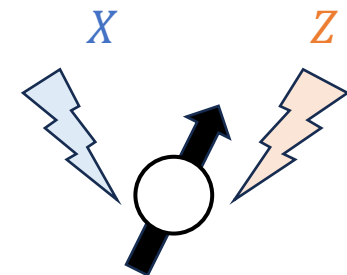
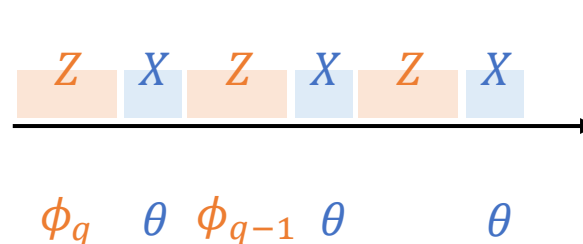
▶ 量子信号処理

Pauli X 回転と制御可能な Pauli Z 回転の合成

定義.

$x = \cos \theta \in [-1, 1]$, $\vec{\phi}_q = (\phi_0, \dots, \phi_q) \in \mathbb{R}^{q+1}$ 次数 q の量子信号処理を次で定義。

$$\begin{aligned} R[\vec{\phi}_q](x) &= \underbrace{e^{i\phi_0 Z}}_{\text{orange}} \underbrace{e^{i\theta X}}_{\text{blue}} \underbrace{e^{i\phi_1 Z}}_{\text{orange}} \underbrace{e^{i\theta X}}_{\text{blue}} \dots \underbrace{e^{i\phi_{q-1} Z}}_{\text{orange}} \underbrace{e^{i\theta X}}_{\text{blue}} \underbrace{e^{i\phi_q Z}}_{\text{orange}} \\ &= \underbrace{e^{i\phi_0 Z}}_{\text{orange}} \prod_{i=1}^q \left[\underbrace{W(x)}_{\text{blue}} \underbrace{e^{i\phi_i Z}}_{\text{orange}} \right] \end{aligned}$$



III-2.1. 量子信号処理

▶ 量子信号処理

パラメータ $\vec{\phi}_q = (\phi_0, \dots, \phi_q) \in \mathbb{R}^{q+1}$ を制御すれば、
シグナル x を多項式変換した回転を実現できる！

Fact.

$$R[\vec{\phi}_q](x) = e^{i\phi_0 Z} \prod_{i=1}^q [W(x)e^{i\phi_i Z}] = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

の形にかける。 $P(x), Q(x)$: 多項式

→ 証明は一旦置いといて、具体例で考えてみる

III-2.1. 量子信号処理 (QSP)

▶ QSPでできること

事実.

$$R[\vec{\phi}_q](x) = e^{i\phi_0 Z} \prod_{i=1}^q [W(x)e^{i\phi_i Z}] = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

例1. 次数 $q = 1$ の場合

$$\begin{aligned} R[(\phi_0, \phi_1)](x) &= \underbrace{\begin{pmatrix} e^{i\phi_0} & 0 \\ 0 & e^{-i\phi_0} \end{pmatrix}}_Z \underbrace{\begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix}}_X \underbrace{\begin{pmatrix} e^{i\phi_1} & 0 \\ 0 & e^{-i\phi_1} \end{pmatrix}}_Z \\ &\quad \begin{matrix} P(x) : 1\text{次関数} & Q(x) : 0\text{次関数} \end{matrix} \\ &= \begin{pmatrix} \underline{e^{i(\phi_0+\phi_1)}x} & \underline{ie^{i(\phi_0-\phi_1)}\sqrt{1-x^2}} \\ ie^{-i(\phi_0-\phi_1)}\sqrt{1-x^2} & \underline{e^{-i(\phi_0+\phi_1)}x} \end{pmatrix} \\ &\quad \begin{matrix} & P^*(x) \end{matrix} \end{aligned}$$

III-2.1. 量子信号処理 (QSP)

▶ QSPでできること

事実.

$$R[\vec{\phi}_q](x) = e^{i\phi_0 Z} \prod_{i=1}^q [W(x)e^{i\phi_i Z}] = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

例2. 次数 $q = 2$ の場合 (簡単のため $\phi_0 = \phi_2 = 0$)

$$\begin{aligned} R[(0, \phi_1, 0)](x) &= \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} \begin{pmatrix} e^{i\phi_1} & 0 \\ 0 & e^{-i\phi_1} \end{pmatrix} \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} \\ &\quad \text{\textcolor{blue}{} } P(x) : 2\text{次関数} \\ &= \begin{pmatrix} \frac{(2 \cos \phi_1)x^2 - e^{-i\phi_1}}{i(2 \cos \phi_1)x\sqrt{1-x^2}} & i(2 \cos \phi_1)x\sqrt{1-x^2} \\ i(2 \cos \phi_1)x\sqrt{1-x^2} & (2 \cos \phi_1)x^2 - e^{i\phi_1} \end{pmatrix} \end{aligned}$$

パラメータ ϕ_1 を tune すれば色々な 2次関数 $P(x)$ を実現できる！

III-2.1. 量子信号処理 (QSP)

▶ QSPでできること

事実.

$$R[\vec{\phi}_q](x) = e^{i\phi_0 Z} \prod_{i=1}^q [W(x)e^{i\phi_i Z}] = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

例3. 一般の次数 q の場合 (簡単のため全ての $\phi = 0$)

$$\begin{aligned} R[\vec{\phi}_q = \vec{0}](x) &= [W(x)]^q = e^{iq\theta X} = \begin{pmatrix} \cos q\theta & i \sin q\theta \\ i \sin q\theta & \cos q\theta \end{pmatrix} \\ &= \begin{pmatrix} \frac{T_q(x)}{iU_{q-1}(x)\sqrt{1-x^2}} & iU_{q-1}(x)\sqrt{1-x^2} \\ iU_{q-1}(x)\sqrt{1-x^2} & T_q(x) \end{pmatrix} \end{aligned}$$

$P(x)$: 次数 q の Chebyshev 多項式

一般にどのような関数 $P(x)$ を実現できる？

III-2.1. 量子信号処理 (QSP)

▶ QSPでできること

定理.

$$R[\vec{\phi}_q](x) = e^{i\phi_0 Z} \prod_{i=1}^q [W(x)e^{i\phi_i Z}] = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

次数 q の QSP が実現可能な関数 $P(x), Q(x)$ の必要十分条件は、
次の条件 (i)-(iii) を全て満たすこと。

(i) $P(x)$ は次数 q 以下の多項式関数 (ii) $P(x)$: 偶関数 (q : 偶数), 奇関数 (q : 奇数)

(iii) $|P(x)|^2 + (1-x^2)|Q(x)|^2 = 1, \forall x \in [-1,1]$ (unitarity)

広い多項式関数 $P(x)$
を実現可能:

4つの次数 q のQSP で実現可能な $P_1(x), \dots, P_4(x)$ の和で
任意の (規格化された) q 次多項式を表現可能

III-2.1. 量子信号処理 (QSP)

▶ 余談: QSP の元々の応用

→ 核磁気共鳴 (NMR) の制御

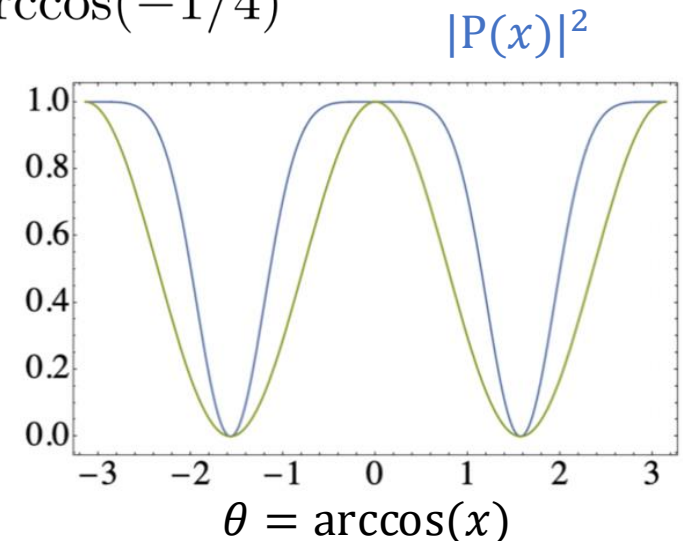
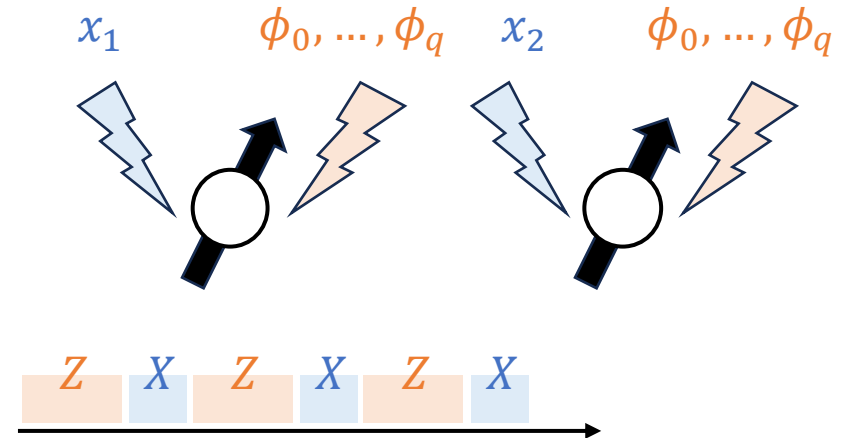
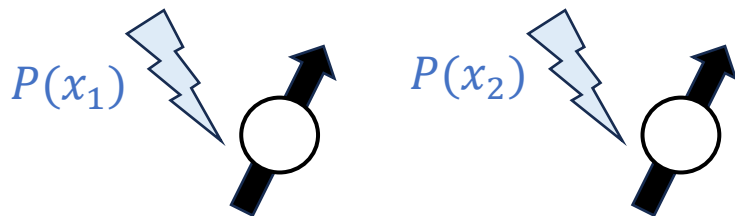
例. BB1 pulse

S. Wimperis, J. Magn. Reson., Ser. B 109, 221. (1994)

$$\vec{\phi} = (\pi/2, -\chi, 2\chi, 0, -2\chi, \chi), \quad \chi = (1/2) \arccos(-1/4)$$

↓

$$|P(x)|^2 = \frac{1}{8}x^2(3x^8 - 15x^6 + 35x^4 - 45x^2 + 30)$$



V.M. Bastidas, et al., Phys. Rev. B 109, 014306 (2024)

III-2. 量子固有値変換の定式化

▶ 量子固有値変換

$$|\psi\rangle \rightarrow f(H) |\psi\rangle, \quad f(H) = \sum_{i=1}^{2^n} f(\lambda_i) |\lambda_i\rangle \langle \lambda_i|$$

1. 量子信号処理

1 qubit 上の Pauli X, Z 回転を合成し、多項式関数を実現

$$R[\vec{\phi}_q](x) = e^{i\phi_0 Z} \prod_{i=1}^q [W(x)e^{i\phi_i Z}] = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

$W(x)$: 対角成分にシグナル x を持つ

$W(x)$ を q 回呼び出し

$$W(x) = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix}$$

広いクラスの次数 q の多項式関数を実現可能

III-2. 量子固有値変換の定式化

▶ 量子固有値変換

$$|\psi\rangle \rightarrow f(H) |\psi\rangle, \quad f(H) = \sum_{i=1}^{2^n} f(\lambda_i) |\lambda_i\rangle \langle \lambda_i|$$

2. ブロック埋め込み

H の情報を埋め込む
ユニタリ行列(量子回路)

$$O_H = \left(\begin{array}{cc} H & * \\ * & * \end{array} \right) \left. \vphantom{\begin{array}{cc} H & * \\ * & * \end{array}} \right\} 2^n \text{ 次元}$$

$\underbrace{\hspace{10em}}_{2^n \text{ 次元}}$

3. 量子ビット化 (qubitization)

↑ を 1 qubit 上の回転として扱う

$$O_H \sim \bigoplus_{i=1}^{2^n} \left(\begin{array}{cc} \lambda_i & \sqrt{1 - (\lambda_i)^2} \\ \sqrt{1 - (\lambda_i)^2} & -\lambda_i \end{array} \right)$$

シグナル x



量子信号処理

$$\left(\begin{array}{cc} f(\lambda_i) & * \\ * & * \end{array} \right)$$

多項式関数！

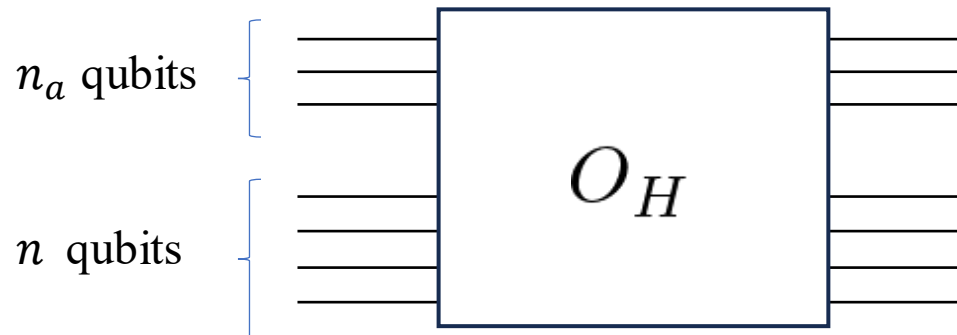
III-2.2. ブロック埋め込み (block-encoding)

▶ ブロック埋め込みとは H の情報を埋め込むユニタリ行列(量子回路)

定義. n_a qubits の補助系 a を用意する。

$$O_H = \begin{pmatrix} H/\alpha & * \\ * & * \end{pmatrix} \begin{matrix} |0\rangle_a \otimes I \\ \langle 0|_a \otimes I \end{matrix}$$

のブロック行列で書ける量子回路(ユニタリ行列)。



III-2.2. ブロック埋め込み (block-encoding)

▶ ブロック埋め込みとは H の情報を埋め込むユニタリ行列(量子回路)

定義.

n_a -qubit の補助系 a を用意する。ある補助系 a の状態 $|0\rangle_a$ があって、

$$(\langle 0|_a \otimes I) O_H (|0\rangle_a \otimes I) = \frac{H}{\alpha}, \quad \exists \alpha > 0$$

を満たす $n_a + n$ qubit 上のunitary行列 O_H をのブロック埋め込みという。

(注1) ブロック埋め込み O_H が存在するための必要十分条件は

(注2) $H \rightarrow H/\alpha$ と規格化すれば $\alpha \rightarrow 1$ とできる。

以降では規格化する。その場合の H の固有値 $\lambda_i \in [-1, 1]$

III-2.3. 量子ビット化 (qubitization)

▶ 量子ビット化とは...

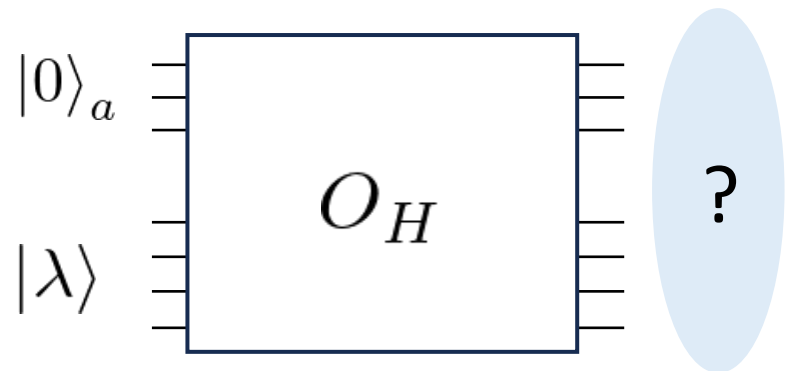
ブロック埋め込み O_H を 1 qubit 回転として表現 $O_H = \left(\begin{array}{cc} H & * \\ * & * \end{array} \right) \left. \vphantom{\begin{array}{cc} H & * \\ * & * \end{array}} \right\} 2^n \text{ 次元}$

▶ 各固有状態への作用

補助系 $|0\rangle_a$ と H の固有状態 $|\lambda\rangle$ ($|\lambda| \leq 1$) に

ブロック埋め込み O_H をかけてみる

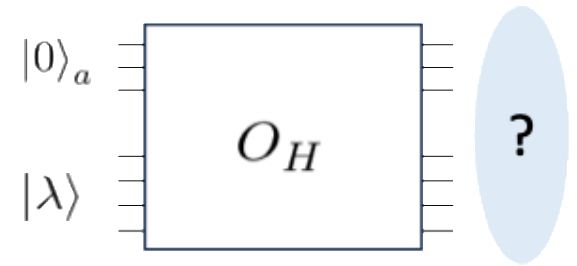
$$O_H(|0\rangle_a \otimes |\lambda\rangle)$$



III-2.3. 量子ビット化 (qubitization)

▶ 各固有状態への作用

補助系 $|0\rangle_a$ と H の固有状態 $|\lambda\rangle$ ($|\lambda| \leq 1$) に
ブロック埋め込み O_H をかけてみる

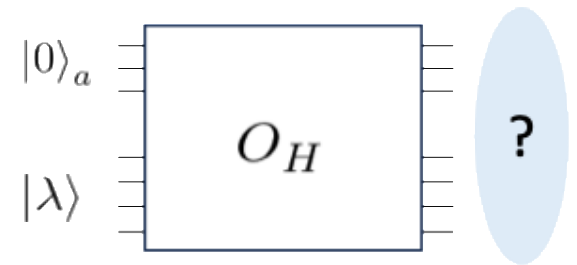


$$\begin{aligned} & \underline{O_H(|0\rangle_a \otimes |\lambda\rangle)} \\ &= \underbrace{(|0\rangle \langle 0|_a \otimes I)}_{\text{互いに直交}} \underline{O_H(|0\rangle_a |\lambda\rangle)} + \underbrace{(1 - |0\rangle \langle 0|_a \otimes I)}_{\text{足して 1}} \underline{O_H(|0\rangle_a |\lambda\rangle)} \end{aligned}$$

III-2.3. 量子ビット化 (qubitization)

▶ 各固有状態への作用

補助系 $|0\rangle_a$ と H の固有状態 $|\lambda\rangle$ ($|\lambda| \leq 1$) に
ブロック埋め込み O_H をかけてみる



$$\begin{aligned} O_H(|0\rangle_a \otimes |\lambda\rangle) &= (|0\rangle \langle 0|_a \otimes I) O_H(|0\rangle_a |\lambda\rangle) + \underbrace{(1 - |0\rangle \langle 0|_a \otimes I) O_H(|0\rangle_a |\lambda\rangle)}_{\text{互いに直交}} \\ &= H \end{aligned}$$

ブロック埋め込み.

$$(\langle 0|_a \otimes I) O_H(|0\rangle_a \otimes I) = \frac{H}{\alpha}$$

規格化して $\alpha = 1$

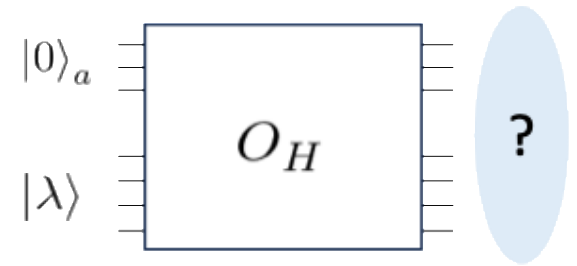
$\sqrt{1 - \lambda^2} |\Psi_\lambda^\perp\rangle$ とおく。

よく分からない $n_a + n$ qubit 状態

III-2.3. 量子ビット化 (qubitization)

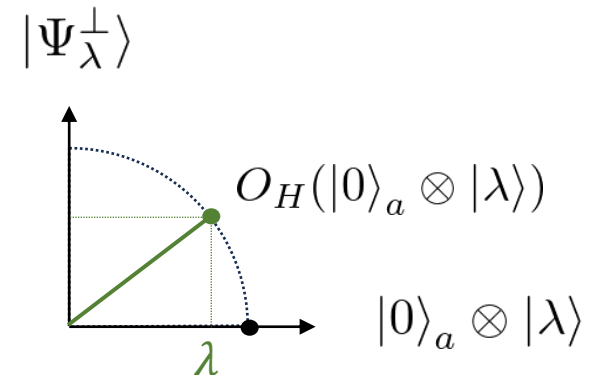
▶ 各固有状態への作用

補助系 $|0\rangle_a$ と H の固有状態 $|\lambda\rangle$ ($|\lambda| \leq 1$) に
 ブロック埋め込み O_H をかけてみる



$$\begin{aligned}
 & O_H(|0\rangle_a \otimes |\lambda\rangle) \\
 &= (|0\rangle \langle 0|_a \otimes I) O_H(|0\rangle_a |\lambda\rangle) + (1 - |0\rangle \langle 0|_a \otimes I) O_H(|0\rangle_a |\lambda\rangle) \\
 &= H \\
 &= \lambda \underbrace{|0\rangle_a \otimes |\lambda\rangle}_{\text{2次元 正規直交系}} + \sqrt{1 - \lambda^2} \underbrace{|\Psi_\lambda^\perp\rangle}_{\text{互いに直交}}
 \end{aligned}$$

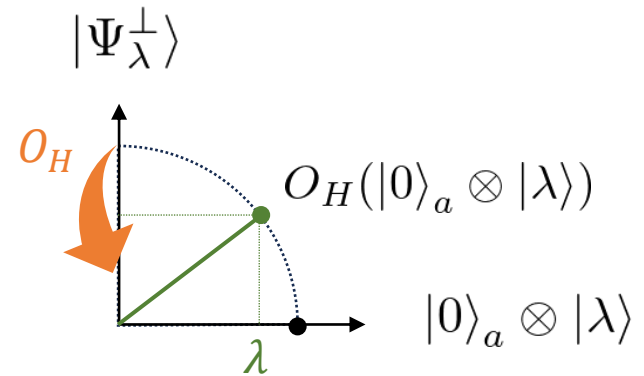

The diagram shows a 2D Hilbert space with axes labeled $|0\rangle_a \otimes |\lambda\rangle$ and $|\Psi_\lambda^\perp\rangle$. A green vector represents the state $O_H(|0\rangle_a \otimes |\lambda\rangle)$. This vector is decomposed into a component along the horizontal axis (labeled λ) and a component along the vertical axis (labeled $\sqrt{1 - \lambda^2}$). A dashed quarter-circle arc connects the tip of the vector to the axes, indicating its norm is 1. The horizontal axis is labeled "2次元 正規直交系" (2D orthonormal basis) and the vertical axis is labeled "互いに直交" (mutually orthogonal).



III-2.3. 量子ビット化 (qubitization)

▶ 各固有状態への作用

$$\begin{aligned} O_H(|0\rangle_a \otimes |\lambda\rangle) \\ = \lambda \underbrace{|0\rangle_a \otimes |\lambda\rangle}_{\text{2次元 正規直交系}} + \sqrt{1 - \lambda^2} \underbrace{|\Psi_\lambda^\perp\rangle}_{\text{2次元 正規直交系}} \end{aligned}$$



2次元 正規直交系 $\mathcal{S}_\lambda^\Psi = \{|0\rangle_a |\lambda\rangle, |\Psi_\lambda^\perp\rangle\}$ 上の

実効的な 1 qubit 回転っぽい？

→ まだ不十分 $O_H |\Psi_\lambda^\perp\rangle$ が $\mathcal{S}_\lambda^\Psi = \{|0\rangle_a |\lambda\rangle, |\Psi_\lambda^\perp\rangle\}$

で閉じていない

III-2.3. 量子ビット化 (qubitization)

▶ 各固有状態への作用

$$\begin{aligned} O_H(|0\rangle_a \otimes |\lambda\rangle) &= \lambda |0\rangle_a \otimes |\lambda\rangle + \sqrt{1 - \lambda^2} |\Psi_\lambda^\perp\rangle \\ O_H |\Phi_\lambda^\perp\rangle &= \sqrt{1 - \lambda^2} |0\rangle_a \otimes |\lambda\rangle - \lambda |\Psi_\lambda^\perp\rangle \end{aligned}$$

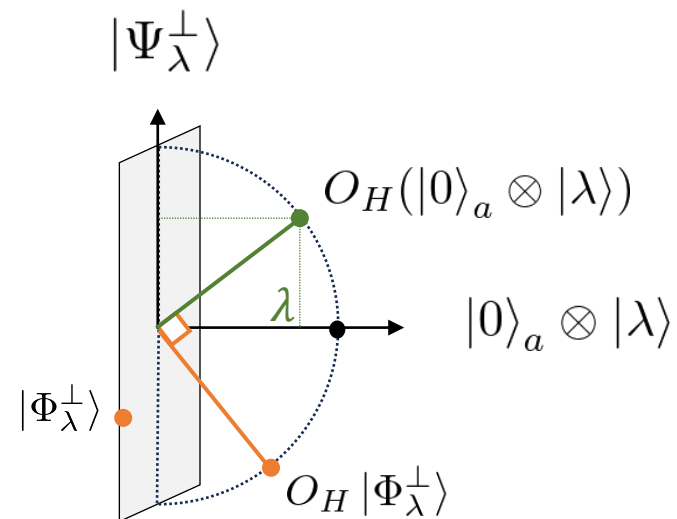
正規直交系

新しい2次元 正規直交系の構成

$$\mathcal{S}_\lambda^\Phi = \{|0\rangle_a |\lambda\rangle, |\Phi_\lambda^\perp\rangle\}$$

$$|\Phi_\lambda^\perp\rangle \equiv O_H^\dagger \left(\sqrt{1 - \lambda^2} |0\rangle_a \otimes |\lambda\rangle - \lambda |\Psi_\lambda^\perp\rangle \right)$$

よく分からない $n_a + n$ qubit 状態 を定義



III-2.3. 量子ビット化 (qubitization)

▶ 各固有状態への作用

$$O_H(|0\rangle_a \otimes |\lambda\rangle) = \lambda |0\rangle_a \otimes |\lambda\rangle + \sqrt{1 - \lambda^2} |\Psi_\lambda^\perp\rangle$$

$$O_H |\Phi_\lambda^\perp\rangle = \sqrt{1 - \lambda^2} |0\rangle_a \otimes |\lambda\rangle - \lambda |\Psi_\lambda^\perp\rangle$$

正規直交系

$$\mathcal{S}_\lambda^\Phi = \{|0\rangle_a |\lambda\rangle, |\Phi_\lambda^\perp\rangle\}$$

正規直交系

$$\mathcal{S}_\lambda^\Psi = \{|0\rangle_a |\lambda\rangle, |\Psi_\lambda^\perp\rangle\}$$

“ブロック埋め込みは各固有状態に対する 1-qubit 回転”

$$O_H = \left[\bigoplus_{i=1}^{2^n} \begin{pmatrix} \lambda_i & \sqrt{1 - (\lambda_i)^2} \\ \sqrt{1 - (\lambda_i)^2} & -\lambda_i \end{pmatrix} \right] \begin{matrix} \boxed{\mathcal{S}_\lambda^\Phi} \\ \underbrace{\oplus \left[\dots \right]}_{\text{それ以外の空間}} \end{matrix}$$

$\boxed{\mathcal{S}_\lambda^\Psi}$

III-2.3. 量子ビット化 (qubitization)

▶ 各固有状態への作用

$$O_H(|0\rangle_a \otimes |\lambda\rangle)$$

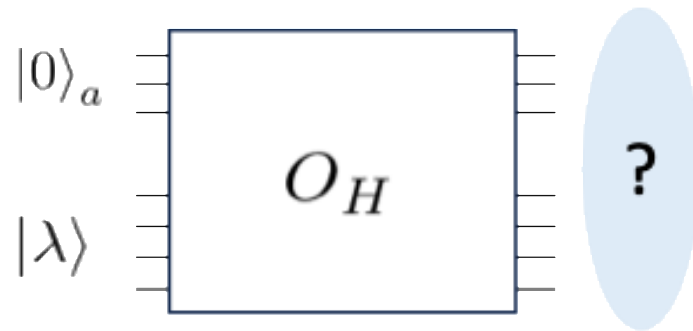
$$O_H|\Phi_\lambda^\perp\rangle$$

正規直交基底

$$\mathcal{S}_\lambda^\Phi = \{ |0\rangle_a |\lambda\rangle, |\Phi_\lambda^\perp\rangle \}$$

“ブロック埋め込み”

以降は無視



$$O_H = \left[\bigoplus_{i=1}^{2^n} \begin{pmatrix} \lambda_i & \sqrt{1 - (\lambda_i)^2} \\ \sqrt{1 - (\lambda_i)^2} & -\lambda_i \end{pmatrix} \right] \underbrace{\mathcal{S}_\lambda^\Phi}_{\mathcal{S}_\lambda^\Psi} \oplus \underbrace{\left[\dots \right]}_{\text{それ以外の空間}}$$

III-2.3. 量子ビット化 (qubitization)

▶ 各固有状態への作用

$$O_H \sim \bigoplus_{i=1}^{2^n} \begin{pmatrix} \lambda_i & \sqrt{1 - (\lambda_i)^2} \\ \sqrt{1 - (\lambda_i)^2} & -\lambda_i \end{pmatrix} \quad \mathcal{S}_\lambda^\Phi \rightarrow \mathcal{S}_\lambda^\Psi$$

$$= \bigoplus_{i=1}^{2^n} \left[e^{i(\pi/4)Z} W(\lambda_i) e^{i(\pi/4)Z} \right] \quad \mathcal{S}_\lambda^\Phi \rightarrow \mathcal{S}_\lambda^\Psi$$

Pauli X 回転

$$W(x) = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix}$$

ブロック埋め込みは実効的な 1-qubit 回転、ただし基底を変更する
(量子ビット化, qubitization)

III-2. 定式化まとめ

1. 量子信号処理 1 qubit 上の Pauli X, Z 回転を合成し、多項式関数を実現

$$R[\vec{\phi}_q](x) = e^{i\phi_0 Z} \prod_{i=1}^q [W(x)e^{i\phi_i Z}] = \begin{pmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{pmatrix}$$

2. ブロック埋め込み & 3. 量子ビット化

$$O_H = \begin{pmatrix} H & * \\ * & * \end{pmatrix} \quad \text{Hamiltonian } H \text{ の情報を埋め込み}$$

$$\sim \bigoplus_{i=1}^{2^n} \left[e^{i(\pi/4)Z} \underbrace{W(\lambda_i)}_{\text{Pauli } X \text{ 回転}} e^{i(\pi/4)Z} \right] \quad \begin{array}{l} \text{1 qubit 回転} \\ \text{(但し、基底を変換)} \end{array}$$

Pauli Z 回転に相当するゲートがあれば $|\psi\rangle \rightarrow f(H)|\psi\rangle$ が可能！

Appendix:

量子特異値変換

(元々 III の代わりに話す予定でしたが時間の都合で変更したので、ここで供養します)

III-1. 量子特異値変換とは？

III-2. 定式化 (量子信号処理, ブロック埋め込み, 量子ビット化)

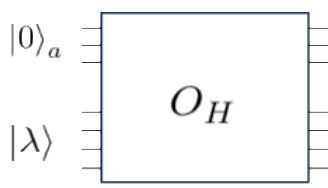
III-3. アルゴリズムと計算複雑性

III-3. 量子固有値変換

▶ アルゴリズム 量子信号処理を再現するために以下のものを用意

その1. ブロック埋め込み (Pauli X 回転)

$$O_H \sim \bigoplus_{i=1}^{2^n} \begin{pmatrix} \lambda_i & \sqrt{1 - (\lambda_i)^2} \\ \sqrt{1 - (\lambda_i)^2} & -\lambda_i \end{pmatrix} \quad \begin{array}{c} \boxed{S_\lambda^\Phi} \rightarrow \boxed{S_\lambda^\Psi} \end{array}$$

$$\sim \bigoplus_{i=1}^{2^n} \left[e^{i(\pi/4)Z} \underbrace{W(\lambda_i)}_{\text{Pauli X 回転}} e^{i(\pi/4)Z} \right] \quad \begin{array}{c} \boxed{S_\lambda^\Phi} \rightarrow \boxed{S_\lambda^\Psi} \end{array}$$


その2. ブロック埋め込み, 共役 (Pauli X 回転)

$$O_H^\dagger \sim \bigoplus_{i=1}^{2^n} \begin{pmatrix} \lambda_i & \sqrt{1 - (\lambda_i)^2} \\ \sqrt{1 - (\lambda_i)^2} & -\lambda_i \end{pmatrix} \quad \begin{array}{c} \boxed{S_\lambda^\Psi} \rightarrow \boxed{S_\lambda^\Phi} \end{array}$$

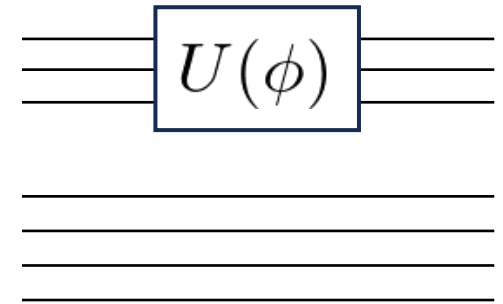
$$\sim \bigoplus_{i=1}^{2^n} \left[e^{i(\pi/4)Z} \underbrace{W(\lambda_i)}_{\text{Pauli X 回転}} e^{i(\pi/4)Z} \right] \quad \begin{array}{c} \boxed{S_\lambda^\Psi} \rightarrow \boxed{S_\lambda^\Phi} \end{array}$$

III-3. 量子固有値変換

▶ アルゴリズム 量子信号処理を再現するために以下のものを用意

その3. 補助系 a 上のゲート (Pauli Z 回転)

$$U(\phi) = e^{i\phi(2|0\rangle_a\langle 0|_a - 1)} \otimes I$$



$\mathcal{S}_\lambda^\Phi = \{|0\rangle_a |\lambda\rangle, |\Phi_\lambda^\perp\rangle\}$ に対する作用

$$\begin{cases} U(\phi) |0\rangle_a |\lambda\rangle = e^{i\phi} |0\rangle_a |\lambda\rangle \\ U(\phi) |\Phi_\lambda^\perp\rangle = e^{-i\phi} |\Phi_\lambda^\perp\rangle \end{cases}$$

$$U(\phi) \sim \bigoplus_{i=1}^{2^n} \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix} e^{i\phi Z}$$



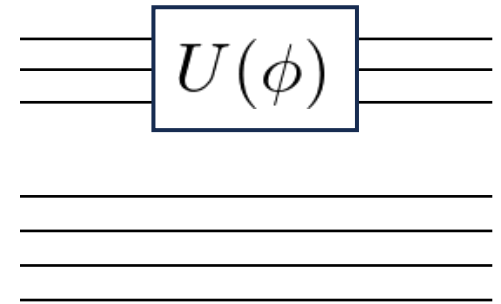
$U(\phi)$ は実効的な 1 qubit 上の Pauli Z 回転 (基底は変更なし)

III-3. 量子固有値変換

▶ アルゴリズム 量子信号処理を再現するために以下のものを用意

その3. 補助系 a 上のゲート (Pauli Z 回転)

$$U(\phi) = e^{i\phi(2|0\rangle_a\langle 0|_a - 1)} \otimes I$$



$$\mathcal{S}_\lambda^\Phi$$

$$\mathcal{S}_\lambda^\Psi$$

に対する作用は同じ

$$U(\phi) \sim \bigoplus_{i=1}^{2^n} \left[e^{i\phi Z} \right] \quad \boxed{\mathcal{S}_\lambda^\Phi} \rightarrow \boxed{\mathcal{S}_\lambda^\Phi} \quad U(\phi) \sim \bigoplus_{i=1}^{2^n} \left[e^{i\phi Z} \right] \quad \boxed{\mathcal{S}_\lambda^\Psi} \rightarrow \boxed{\mathcal{S}_\lambda^\Psi}$$

$U(\phi)$ は実効的な 1 qubit 上の Pauli Z 回転 (基底は変更なし)

III-3. 量子固有値変換

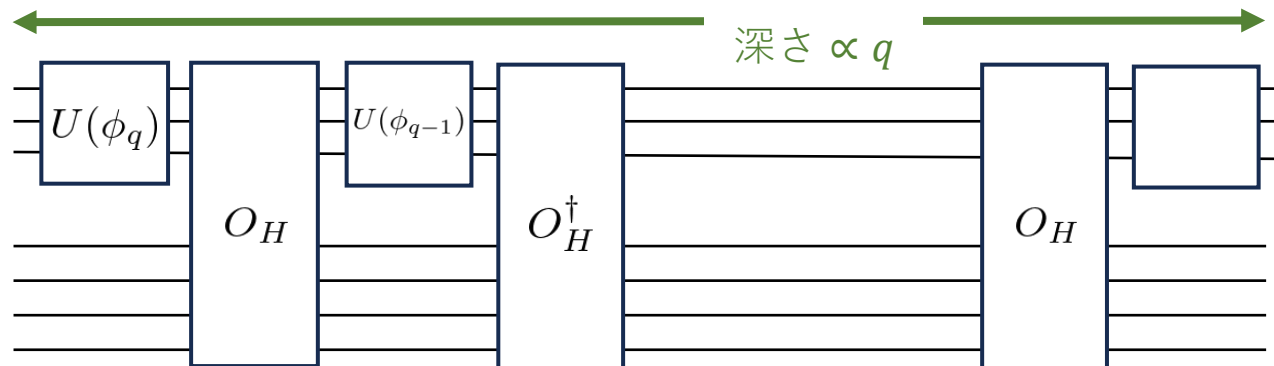
▶ アルゴリズム 量子信号処理を再現する

定義. (量子固有値変換)

次数 q の量子固有値変換を次で定める。

$$V[\vec{\phi}_q] = \underbrace{U(\phi_0)}_{\text{Pauli X 回転 (+補正)}} \underbrace{O_H}_{\text{Pauli Z 回転}} \underbrace{U(\phi_1)}_{\text{Pauli X 回転 (+補正)}} \cdots \underbrace{O_H}_{\text{Pauli Z 回転}} \underbrace{U(\phi_{q-2})}_{\text{Pauli X 回転 (+補正)}} \underbrace{O_H^\dagger}_{\text{Pauli Z 回転}} \underbrace{U(\phi_{q-1})}_{\text{Pauli X 回転 (+補正)}} \underbrace{O_H}_{\text{Pauli Z 回転}} \underbrace{U(\phi_q)}_{\text{Pauli X 回転 (+補正)}}$$

$\vec{\phi}_q = (\phi_0, \dots, \phi_q) \in \mathbb{R}^{q+1}$ は、制御可能なパラメータ。



III-3. 量子固有値変換

▶ アルゴリズム 量子信号処理を再現する

定義. (量子固有値変換)

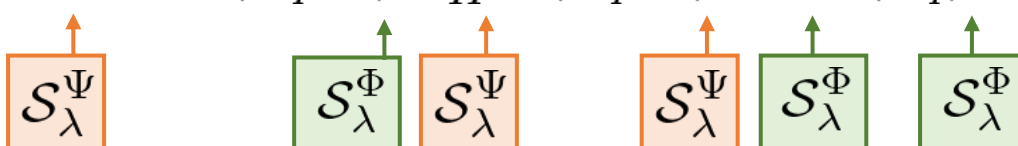
次数 q の量子固有値変換を次で定める。

$$V[\vec{\phi}_q] = \underbrace{U(\phi_0)}_{\text{Pauli X 回転 (+補正)}} \underbrace{O_H}_{\text{Pauli Z 回転}} \underbrace{U(\phi_1)}_{\text{Pauli X 回転 (+補正)}} \cdots \underbrace{O_H}_{\text{Pauli Z 回転}} \underbrace{U(\phi_{q-2})}_{\text{Pauli X 回転 (+補正)}} \underbrace{O_H^\dagger}_{\text{Pauli Z 回転}} \underbrace{U(\phi_{q-1})}_{\text{Pauli X 回転 (+補正)}} \underbrace{O_H}_{\text{Pauli Z 回転}} \underbrace{U(\phi_q)}_{\text{Pauli X 回転 (+補正)}}$$

$\vec{\phi}_q = (\phi_0, \dots, \phi_q) \in \mathbb{R}^{q+1}$ は、制御可能なパラメータ。

なぜ O_H と $O_H^\dagger$ が交互？ → 1 qubit 系の基底変換を整合させるため

$$U(\phi_0) O_H U(\phi_1) \cdots O_H U(\phi_{q-2}) O_H^\dagger U(\phi_{q-1}) O_H U(\phi_q)$$



III-3. 量子固有値変換

▶ アルゴリズムの行列成分

量子信号処理を再現する

$$\begin{aligned}
 V[\vec{\phi}_q] &= U(\phi_0)O_HU(\phi_1)\cdots O_HU(\phi_{q-2})O_H^\dagger U(\phi_{q-1})O_HU(\phi_q) \\
 &\sim \bigoplus_{i=1}^{2^n} \left[\begin{array}{c} \text{orange} \quad \text{blue} \quad \text{orange} \quad \text{blue} \quad \text{orange} \\ e^{i\phi_0 Z} \quad e^{i(\pi/4)Z}W(\lambda_i)e^{i(\pi/4)Z} \quad e^{i\phi_1 Z} \\ \vdots \quad \vdots \quad \vdots \\ e^{i(\pi/4)Z}W(\lambda_i)e^{i(\pi/4)Z} \quad e^{i\phi_q Z} \end{array} \right] \\
 &= \bigoplus_{i=1}^{2^n} \left[e^{i\phi'_0 Z} \prod_{j=1}^q \left[W(\lambda_i)e^{i\phi'_j Z} \right] \right] \\
 &= \bigoplus_{i=1}^{2^n} \left(\begin{array}{cc} P(\lambda_i) & * \\ * & * \end{array} \right)
 \end{aligned}$$

次数 q の量子信号処理

次数 q の多項式関数

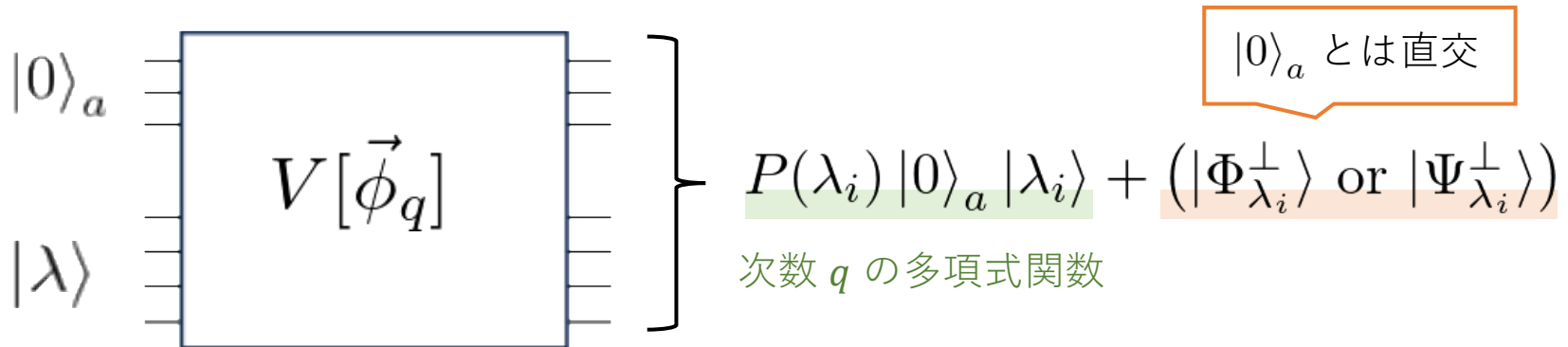
III-3. 量子固有値変換

▶ アルゴリズムの各固有状態への作用

$$V[\vec{\phi}_q] \sim \bigoplus_{i=1}^{2^n} \left(\begin{array}{cc} P(\lambda_i) & * \\ * & * \end{array} \right) - |0\rangle_a |\lambda_i\rangle$$

| $0\rangle_a |\lambda_i\rangle$

次数 q の多項式関数



III-3. 量子固有値変換

▶ アルゴリズムの一般の状態への作用

1. 補助系 $|0\rangle_a$ と一般の n -qubit 状態 $|\psi\rangle = \sum_i c_i |\lambda_i\rangle$ に
次数 q の量子固有値変換 $V[\vec{\phi}_q]$ をかける

$$|\psi\rangle = \sum_i c_i |\lambda_i\rangle$$
$$V[\vec{\phi}_q]$$
$$|0\rangle_a \otimes \sum_i c_i P(\lambda_i) |\lambda_i\rangle$$

次数 q の多項式関数

$$+ \sum_i c_i (|\Phi_{\lambda_i}^\perp\rangle \text{ or } |\Psi_{\lambda_i}^\perp\rangle)$$

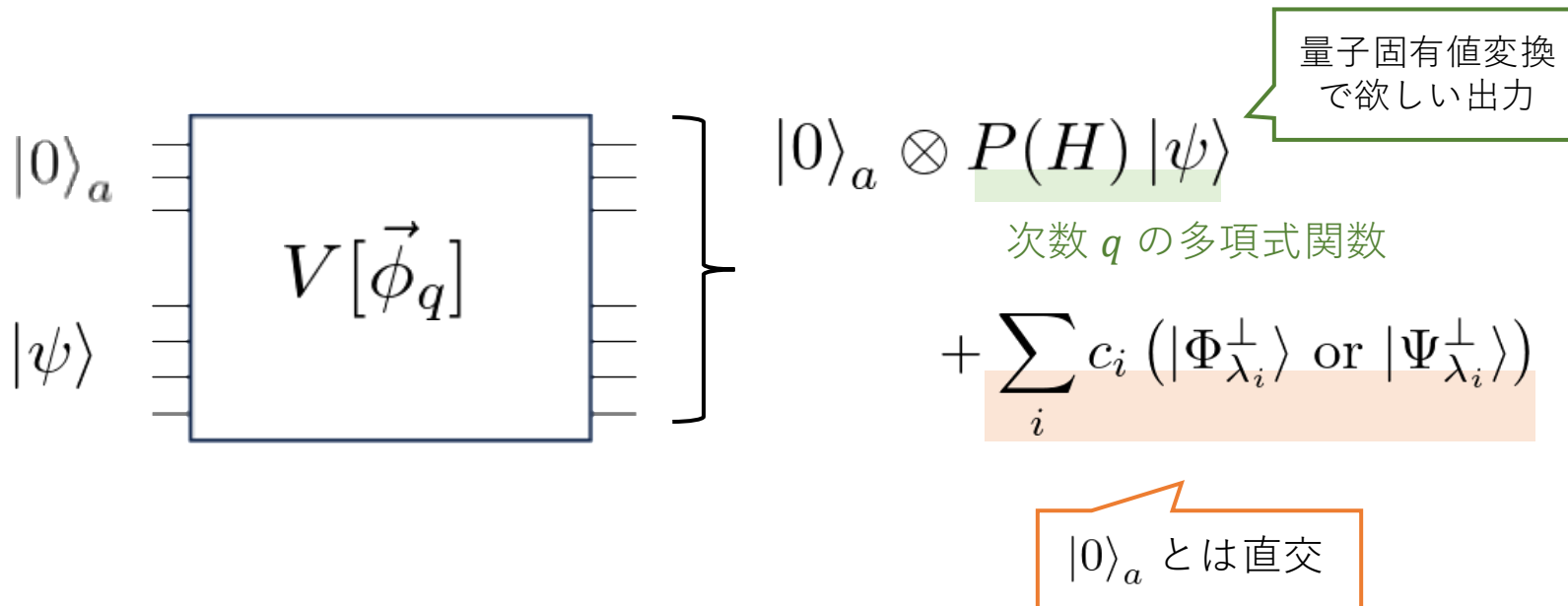
$|0\rangle_a$ とは直交

III-3. 量子固有値変換

▶ アルゴリズムの一般の状態への作用

1. 補助系 $|0\rangle_a$ と一般の n -qubit 状態 $|\psi\rangle = \sum_i c_i |\lambda_i\rangle$ に

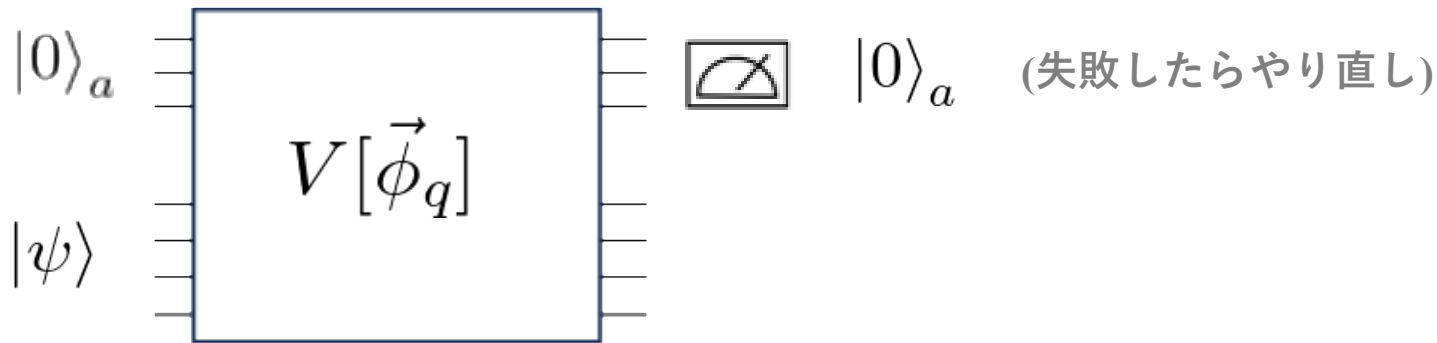
次数 q の量子固有値変換 $V[\vec{\phi}_q]$ をかける



III-3. 量子固有値変換

▶ アルゴリズムの一般の状態への作用

1. 状態 $|0\rangle_a \otimes |\psi\rangle$ に次数 q の量子固有値変換 $V[\vec{\phi}_q]$ をかける
2. 補助系 a を射影測定。 $|0\rangle_a$ の状態が得られた場合を事後選択。

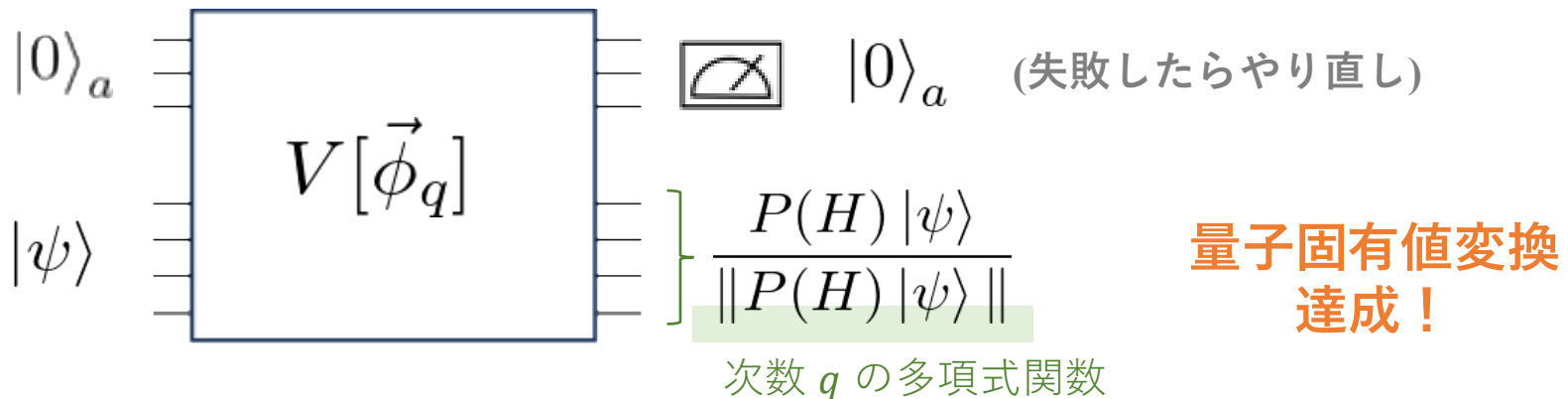


$$\text{測定後の状態} \propto \underbrace{\left(|0\rangle_a \langle 0|_a \otimes I \right)}_{\text{射影}} \left(\underbrace{|0\rangle_a \otimes P(H) |\psi\rangle}_{\text{成功}} + \sum_i c_i \left(\underbrace{|\Phi_{\lambda_i}^\perp\rangle \text{ or } |\Psi_{\lambda_i}^\perp\rangle}_{\text{失敗}} \right) \right)$$

III-3. 量子固有値変換

▶ アルゴリズムの一般の状態への作用

1. 状態 $|0\rangle_a \otimes |\psi\rangle$ に次数 q の量子固有値変換 $V[\vec{\phi}_q]$ をかける
2. 補助系 a を射影測定。 $|0\rangle_a$ の状態が得られた場合を事後選択。



測定後の状態 $\propto |0\rangle_a \otimes P(H) |\psi\rangle$ 測定の成功確率: $p_0 = \|P(H) |\psi\rangle\|^2$

III-3. 量子固有値変換

▶ ここまでのまとめ

ブロック埋め込み O_H と補助系の事後選択で

$$|\psi\rangle \rightarrow P(H) |\psi\rangle$$

が実装可能。

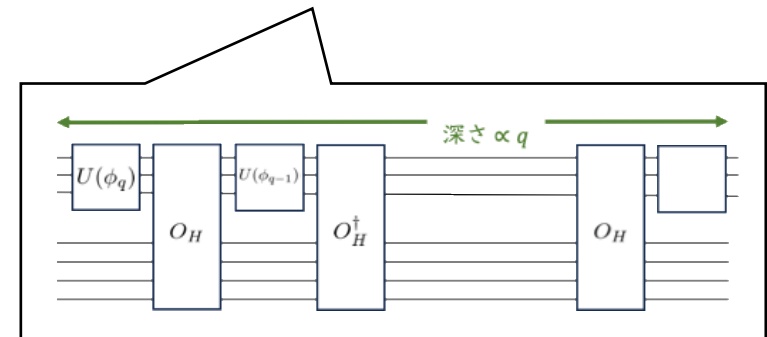
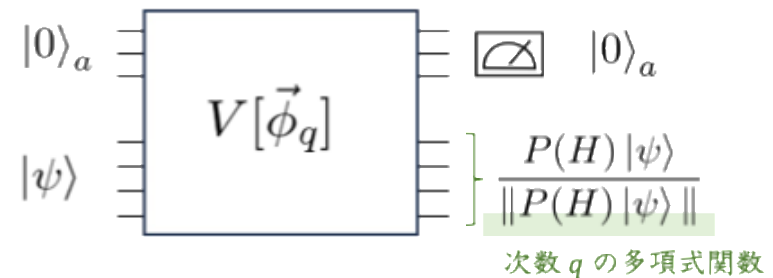
操作の成功確率: $p_0 = \|P(H) |\psi\rangle\|^2$

実装に必要なコスト:

q 回分のブロック埋め込み $O_H, O_H^\dagger$

実装可能な関数 $P(x)$:

量子信号処理で実現可能な q 次多項式



4個使えば
任意の q 次多項式関数を表現可能

III-3. 量子固有値変換

▶ 一般の関数 $H \rightarrow H/\alpha$ としていた規格化も元に戻す

定理. (一般の関数の量子固有値変換)

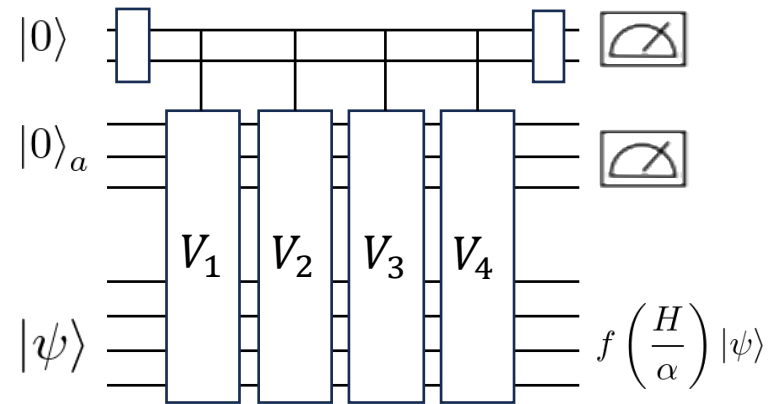
任意の規格化された q 次多項式 $f(x)$ について

$$|\psi\rangle \rightarrow f(H/\alpha) |\psi\rangle \quad \text{が可能。}$$

操作の成功確率: 定数 $\times \|f(H/\alpha) |\psi\rangle\|^2$

実装に必要なコスト:

$\mathcal{O}(q)$ 回分の(制御)ブロック埋め込み $\mathcal{O}_H, \mathcal{O}_H^\dagger$



量子計算のコスト(クエリ複雑性) q = 多項式関数 $f(H)$ の次数 q

III-3. Hamiltonian シミュレーション

▶ 時間発展の実装

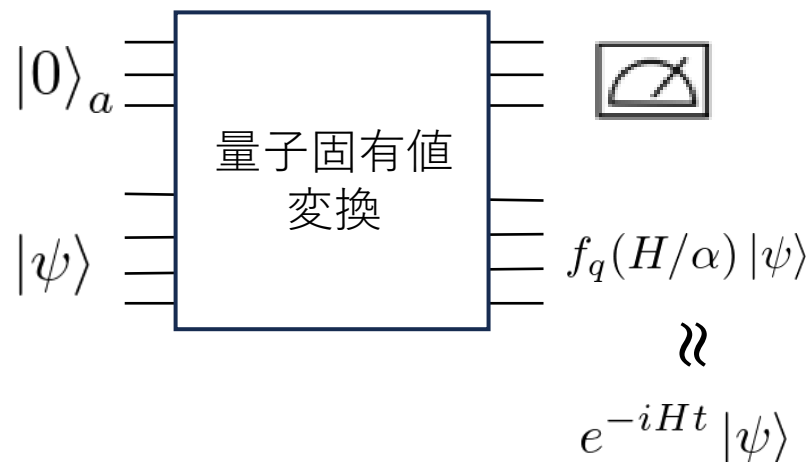
$$f(H/\alpha) = e^{-iHt} = e^{-i(H/\alpha) \cdot \alpha t}$$

無限級数

の多項式近似を実装

$$f_q(x) = \sum_{n=0}^q \frac{(-i\alpha t)^n}{n!} x^n \simeq e^{-ix \cdot \alpha t}$$

次数 q の多項式近似



成功確率: $\propto \|f_q(H/\alpha) |\psi\rangle\|^2 \simeq \|e^{-iHt} |\psi\rangle\|^2 = 1$ 定数回
繰り返せば OK

量子ゲート数: (クエリ複雑性 q) $\times$ (ブロック埋め込み O_H のゲート数)

III-3. Hamiltonian シミュレーション

▶ 計算コストの決定

量子ゲート数: (クエリ複雑性 q) $\times$ (ブロック埋め込み O_H のゲート数)

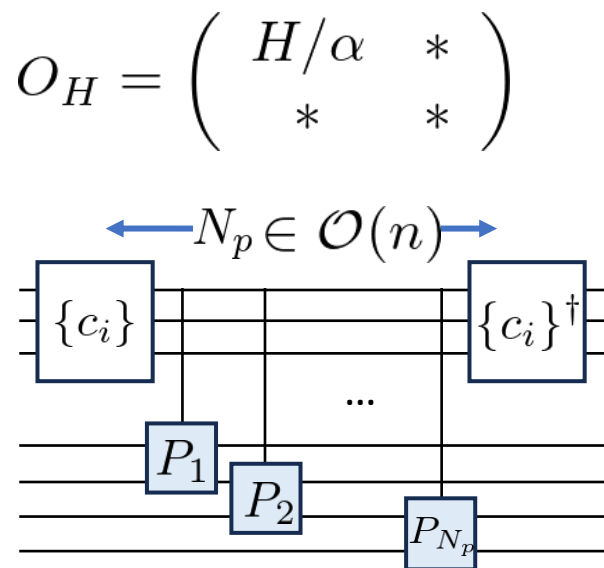
ブロック埋め込み O_H の構成

$$H = \sum_{i=1}^{N_p} c_i P_i \quad \text{の情報を埋め込む行列}$$

有限距離相互作用の local Hamiltonian の時

$$\alpha = \sum_{i=1}^{N_p} |c_i| \in \mathcal{O}(n)$$

補助 qubit 数: $n_a \in \mathcal{O}(\log n)$ 量子ゲート数: $\in \mathcal{O}(n)$



III-3. Hamiltonian シミュレーション

▶ 計算コストの決定

量子ゲート数: (クエリ複雑性 q) $\times$ (ブロック埋め込み O_H のゲート数)

← 精度保証: $\|f_q(H/\alpha) |\psi\rangle - e^{-iHt} |\psi\rangle\| \leq \varepsilon$ で決定 積公式と同様

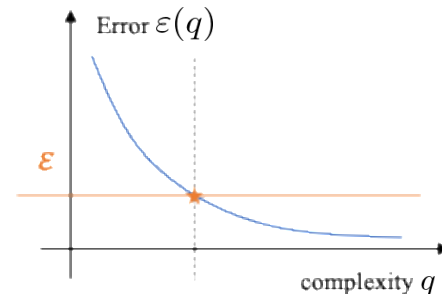
$\Leftrightarrow$ q 次多項式近似の誤差が ε 以下

$$\|f_q(x) - e^{-ix \cdot (\alpha t)}\| = \left\| \sum_{n=0}^q \frac{(-i\alpha t)^n}{n!} x^n - e^{-ix \cdot (\alpha t)} \right\| \leq \varepsilon, \quad \forall x \in [-1, 1]$$

クエリ複雑性 q

$$q \in \mathcal{O} \left(\alpha t + \frac{\log(1/\varepsilon)}{\log \log(1/\varepsilon)} \right) \subset \mathcal{O}(nt + \log(1/\varepsilon))$$

Optimal scaling



III-3. Hamiltonian シミュレーション

▶ 計算コストの決定

量子ゲート数: [クエリ数 $\mathcal{O}(gnt + \log(1/\varepsilon))$] $\times$ [O_H のゲート数 $\mathcal{O}(n)$]

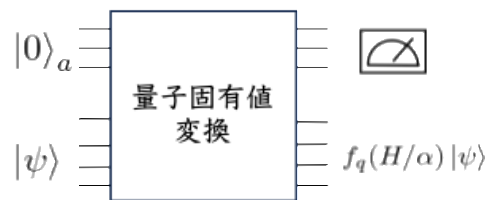
$$\in \mathcal{O}(n(nt + \log(1/\varepsilon)))$$

本セクションの
メインの結果

- ・ ブロック埋め込み O_H に関して query-optimal な量子アルゴリズム
一般の (量子多体系などの構造を仮定しない) H に対しては理論上最速

- ・ 積公式のゲート数: $\mathcal{O}\left(nt \left(\frac{nt}{\varepsilon}\right)^{\frac{1}{p}}\right)$ と比較して
 ε 依存性を指数的に改善

“Post-Trotter アルゴリズム”



III-3. 余談 (むしろこっちが重要)

▶ 量子アルゴリズムの大統一

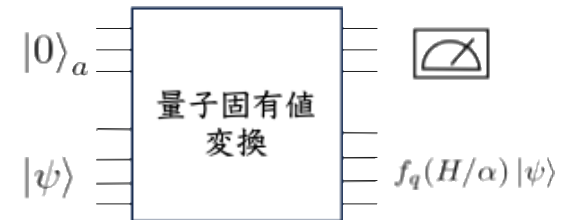
Hamiltonian シミュレーション:

量子固有値変換で

q 次多項式

$$f_q(H/\alpha) \simeq e^{-iHt}$$

計算コスト
($\propto$ 次数 q)



$$q \in \mathcal{O}(\alpha t + \log(1/\varepsilon))$$

Fact: 多項式関数 $f_q(x)$ を変えた量子固有値変換で
既存の量子アルゴリズムのほとんどを統一的に実行可能

例1: Gibbs サンプリング

$$\rho_\beta = \frac{e^{-\beta H}}{Z_\beta} \quad \text{熱平衡状態準備}$$

← Bell 状態に $e^{-\beta H/2}$ をかける量子固有値変換

多項式近似

$$f_q(x) \simeq e^{-\beta\alpha(1+x)/2}$$

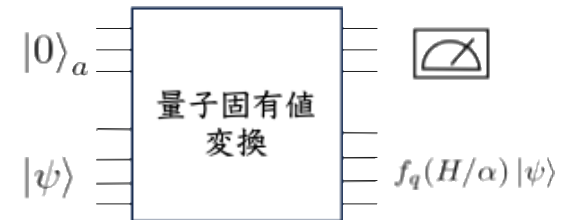
クエリ数

$$q \in \tilde{\mathcal{O}}\left(\beta\alpha\sqrt{\frac{2^n}{Z_\beta}}\right)$$

III-3. 余談 (むしろこっちが重要)

▶ 量子アルゴリズムの大統一

多項式関数 $f_q(x)$ を変えた量子固有値変換で
既存の量子アルゴリズムを統一的に実行可能



例2: 線形方程式問題

連立方程式 $A|x\rangle = |b\rangle$ の解を準備

$$|b\rangle \rightarrow |x\rangle = A^{-1} |b\rangle$$

多項式近似 $f_q(x) \simeq \frac{1}{x}$ 逆数

クエリ数 $q \in \mathcal{O}(\kappa \log(\kappa/\epsilon))$
 $\kappa : A$ の条件数

例3: Grover 探索

$f_q(x) = T_q(x)$ Chebyshev 多項式

例4: 固有値計算

(Shorの素因数分解含む) $f_q(x) \simeq \text{sgn}(x)$ 符号関数

量子アルゴリズムの種類 = 多項式関数 $f_q(x)$ の選択

量子アルゴリズムのコスト = 多項式関数 $f_q(x)$ の次数 q

III. 量子固有値変換 & 量子特異値変換

▶ まとめ

ブロック埋め込み O_H を q 回使って

$$|\psi\rangle \rightarrow f(H/\alpha) |\psi\rangle$$

$[f(x): q \text{ 次多項式関数}]$ の変換を実装

Hamiltonian シミュレーション

$$q \text{ 次多項式} \quad f_q(H/\alpha) \simeq e^{-iHt} \quad \text{量子ゲート数} \in \mathcal{O}(n(nt + \log(1/\varepsilon)))$$

量子アルゴリズムの大統一

その他の様々な量子アルゴリズムも

q 次多項式の量子固有値変換 & 計算コストが次数 q に比例、で統一的に理解

