

Utility scale quantum computing

量子ユーティリティー授業

量子エラー緩和

2025/09/24

Translated and modified by Kazumasa Umezawa

Created by Toshinari Itoko

アジェンダ

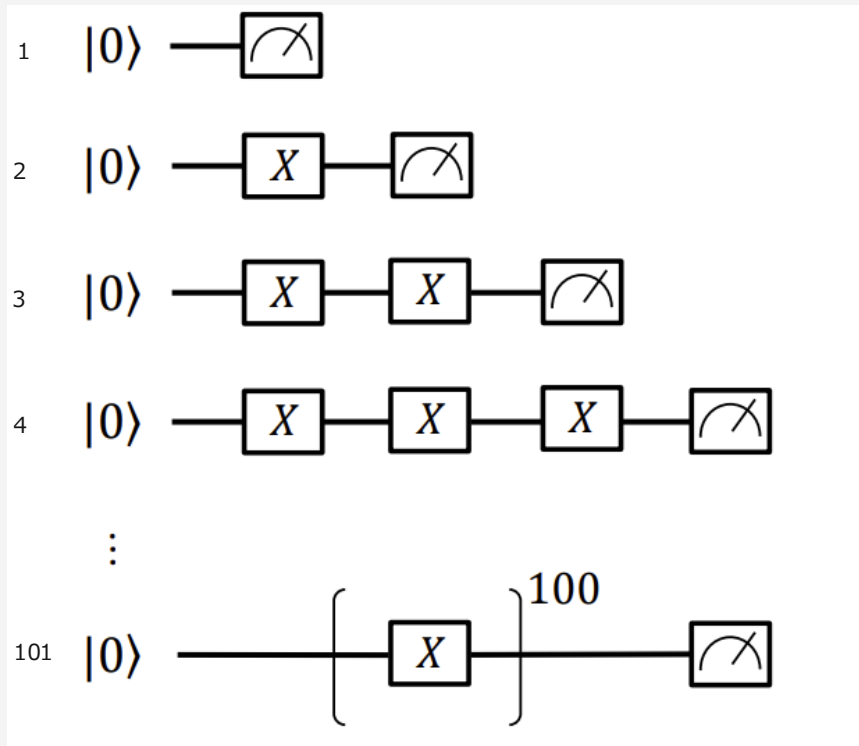
本日の目標：量子ノイズと量子エラー緩和技術の基本的な内容を一緒に勉強しましょう。

1. 量子ノイズ/エラーとは
2. エラー抑制・緩和技術
3. Qiskitにおける設定
4. まとめ

量子ノイズ/エラーとは

クイズ

以下のような101個の回路を実行した場合、期待値 $\langle Z \rangle$ はどうなるでしょうか。



各回路は400ショット

【簡単な復習】

① 計算基底状態

$$\begin{array}{c} |0\rangle \\ |1\rangle \end{array}$$

② 量子ゲート

$$X$$

bit flip

$$Z$$

phase flip

③ 測定



④ 期待値 $\langle Z \rangle$

$$\text{量子状態 } |\psi\rangle \text{ とすると、} \langle Z \rangle = \langle \psi | Z | \psi \rangle$$

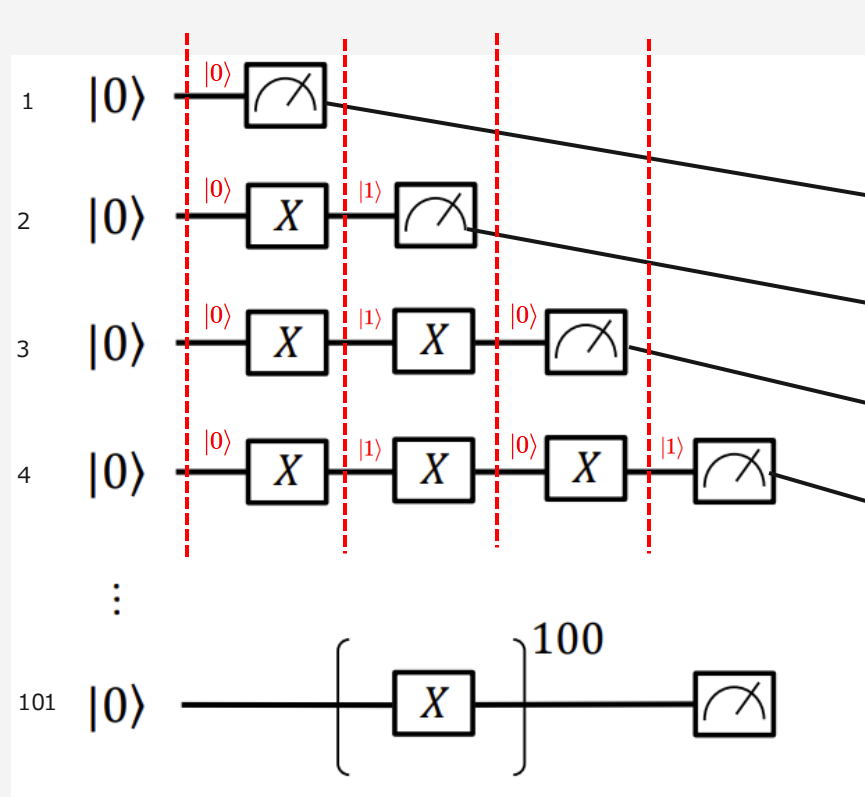
⑤ 計算

$$\begin{array}{l} X|0\rangle = |1\rangle \\ X|1\rangle = |0\rangle \end{array}$$

$$\begin{array}{l} Z|0\rangle = +1|0\rangle \\ Z|1\rangle = -1|1\rangle \end{array}$$

クイズ

以下のような101個の回路を実行した場合、期待値 $\langle Z \rangle$ はどうなるでしょうか。



各回路は400ショット

測定



$$\langle 0|Z|0\rangle = +1$$

$$\langle 1|Z|1\rangle = -1$$

$$\langle 0|Z|0\rangle = +1$$

$$\langle 1|Z|1\rangle = -1$$

$$\langle Z \rangle = (-1)^d$$

d : 回路の深さ

【簡単な復習】

$$X|0\rangle = |1\rangle$$

$$X|1\rangle = |0\rangle$$

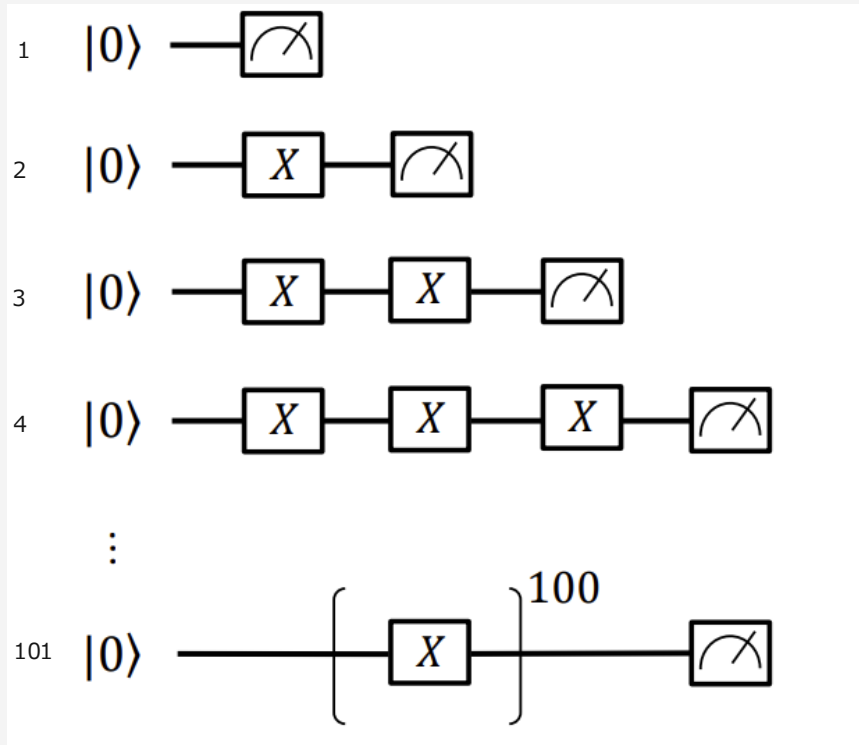
$$Z|0\rangle = +1|0\rangle$$

$$Z|1\rangle = -1|1\rangle$$

Qt

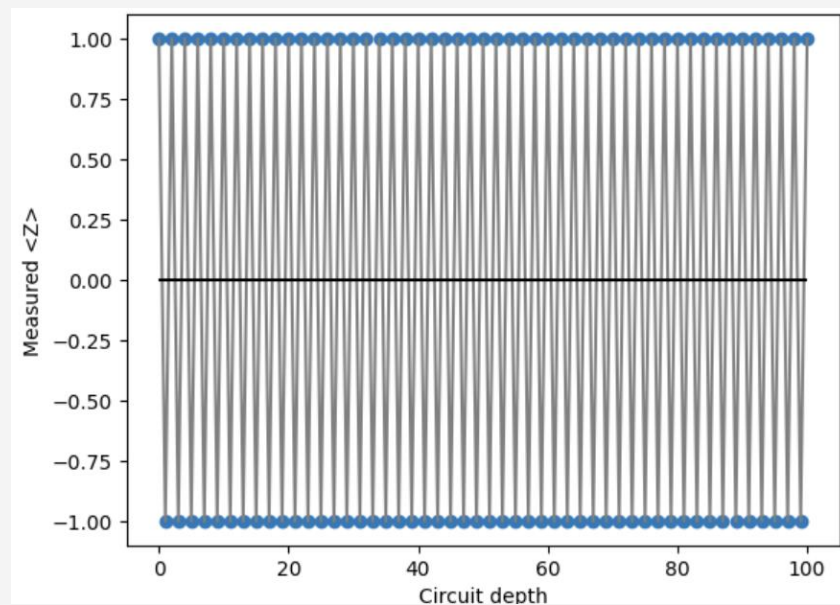
クイズ

以下のような101個の回路を実行した場合、期待値 $\langle Z \rangle$ はどうなるでしょうか。



各回路は400ショット

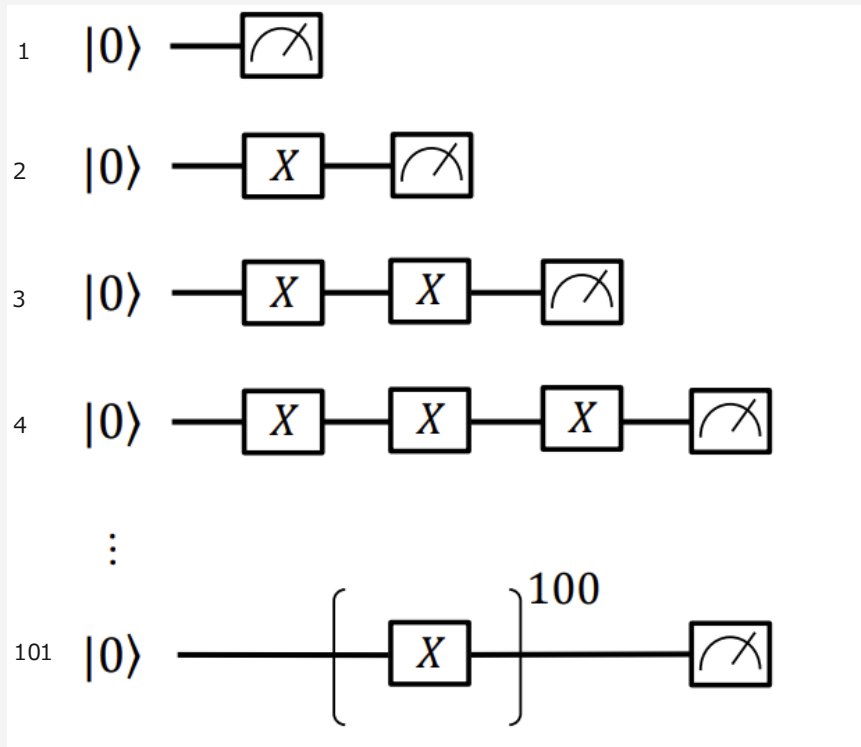
【理想的な結果】



実際にはノイズによるエラーがあるので・・・

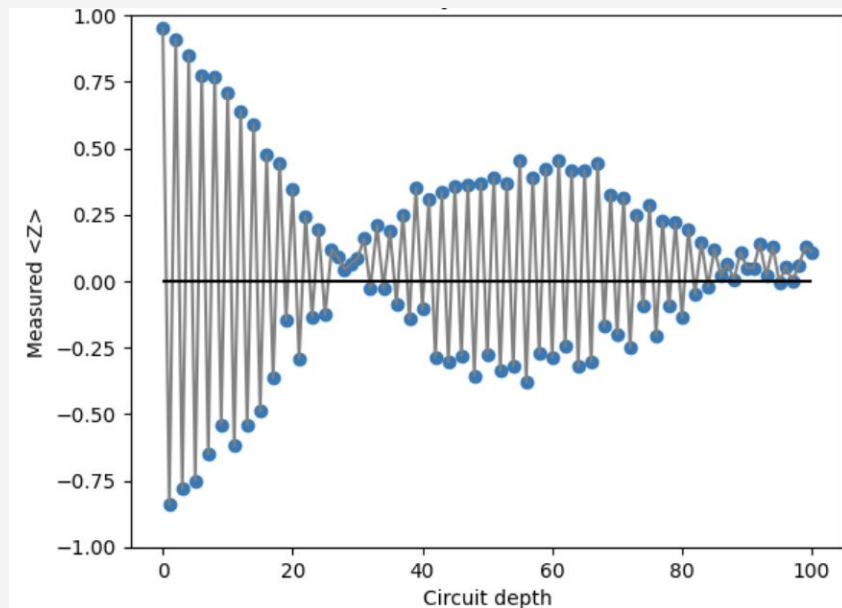
クイズ

以下のような101個の回路を実行した場合、期待値 $\langle Z \rangle$ はどうなるでしょうか。



各回路は400ショット

【ノイズによるエラーを考慮した結果】



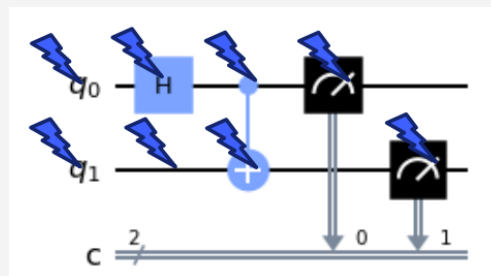
ノイズによるエラーをみていきたいと思います。

量子ノイズ

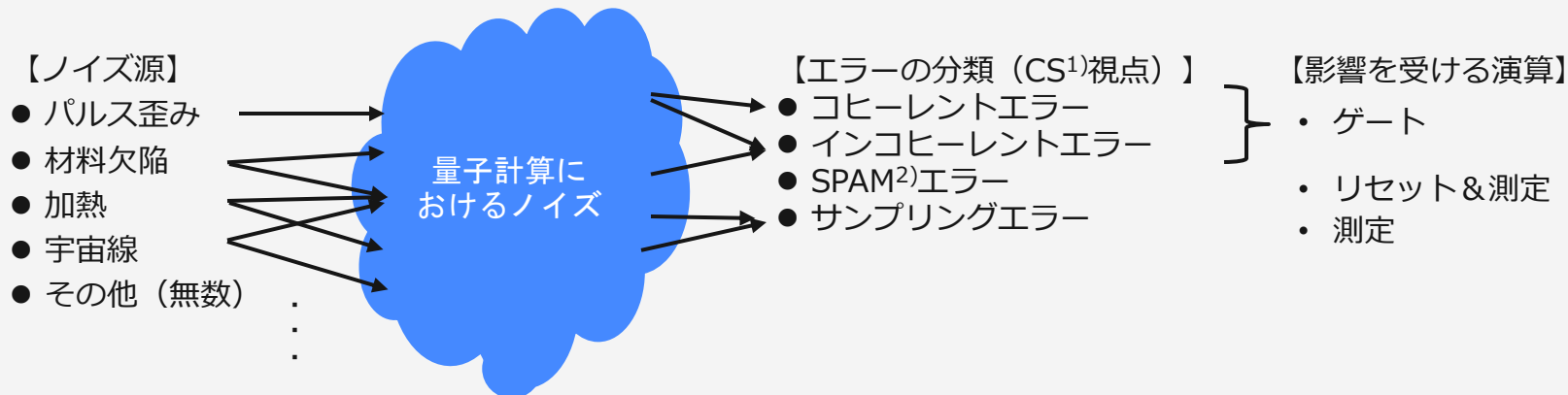
量子状態の測定結果に期待される結果を乱す、望ましくない変換全般を指します。

✓ 量子ノイズはどこからでも発生します。

- 初期化
- ゲート（アイドル状態のときも）
- 測定



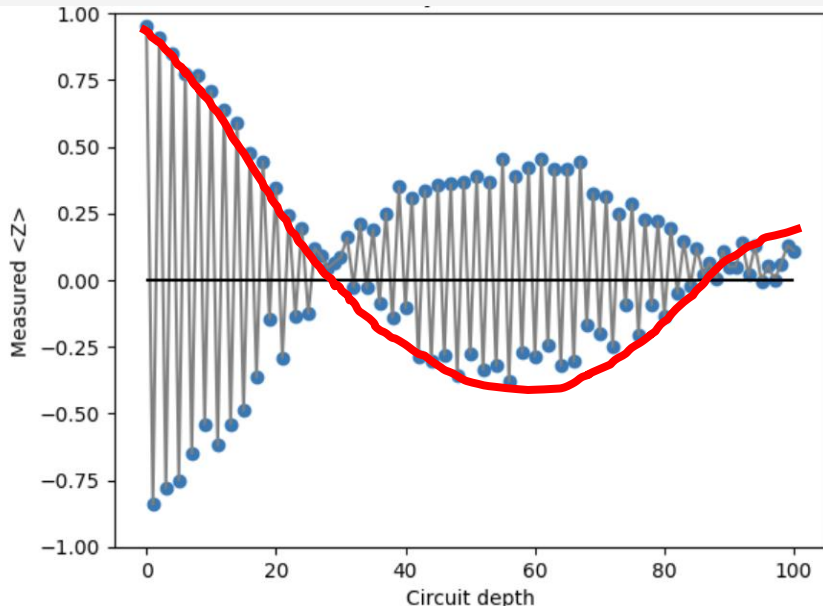
ノイズにより以下のような（計算）エラーが生じます。



コヒーレントエラー

量子状態が系統的にずれていくエラーです。

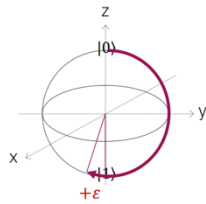
赤線の振動している部分で影響が出ています。



あるゲートが本来 n だけ回転させるべきところを、キャリブレーションの不完全さにより $n+\epsilon$ だけ回転させてしまう状況を考えます。

Miscalibration

$$R_x(\pi) \longrightarrow R_x(\pi + \epsilon)$$



Quantum Tok

【原因】

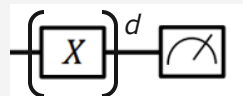
- ✓ キャリブレーションの誤り（例：パルス振幅、量子ビット周波数）
- ✓ 量子ビット間の望ましくない相互作用

【性質】

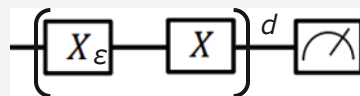
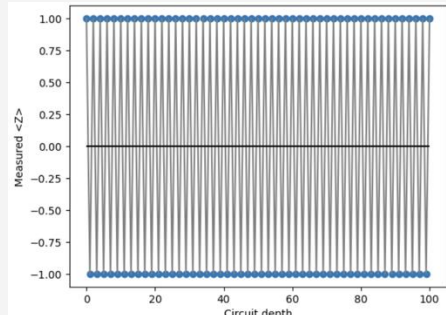
- ✓ これらのエラーはユニタリーであり、情報を破壊せず、理論的には逆転可能です。

【対策】

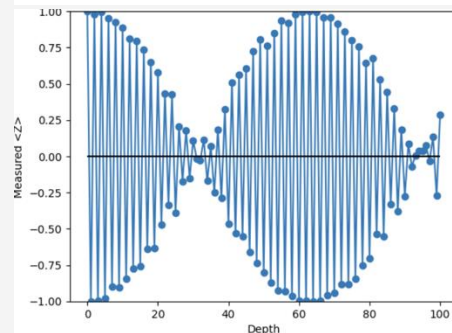
- ✓ （より良いキャリブレーション）、エラーの緩和／抑制



$$\langle \psi | Z | \psi \rangle = \cos(d\pi) = (-1)^d$$



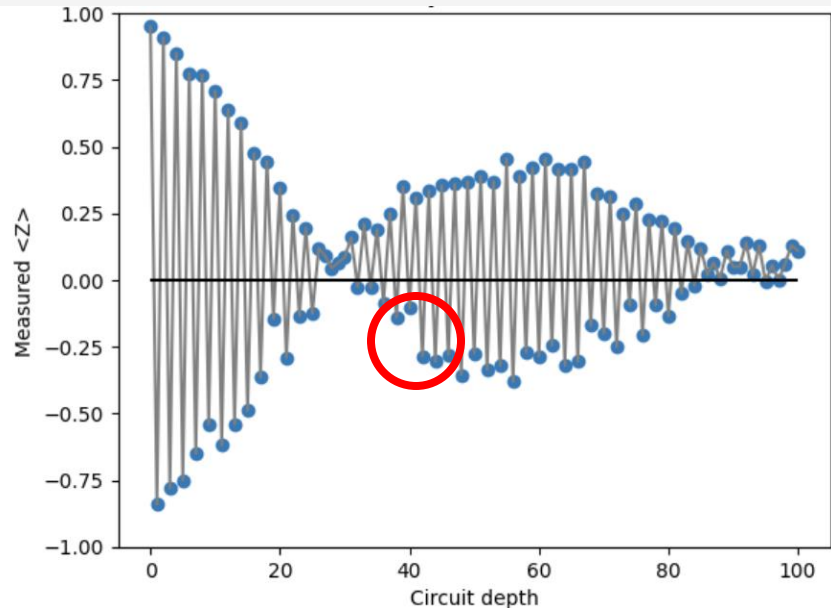
$$\langle \tilde{\psi} | Z | \tilde{\psi} \rangle = \cos(d\pi + d\epsilon)$$



サンプリングエラー（ショットエラー）

量子測定の実験的な性質や有限回の測定回数に由来するエラーです。

赤枠のようにスムーズでない部分で影響が出ています。



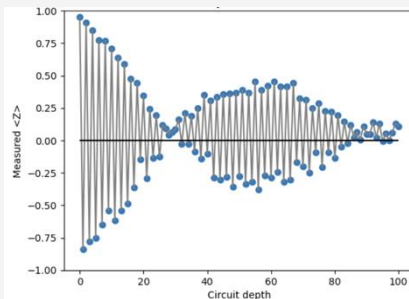
【原因】

✓ 量子物理学の核心

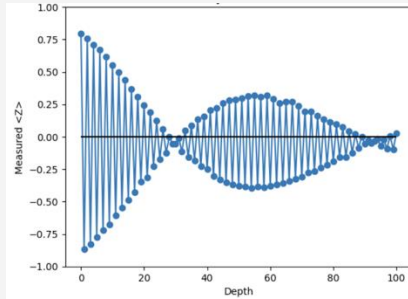
【対策】

✓ ショット数を増やす

ショット数：400



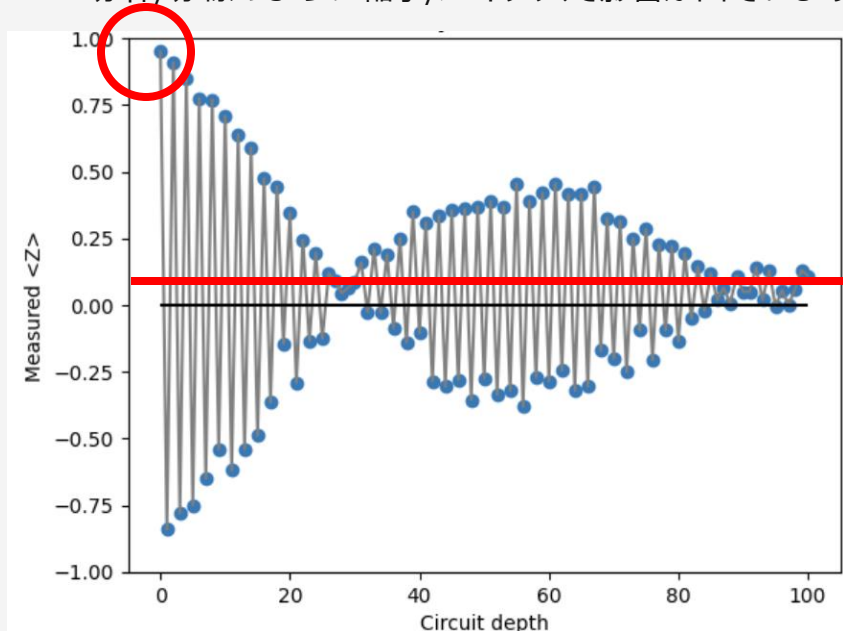
ショット数：10000



SPAMエラー

SPAMエラーとは量子ビットの準備と読み取りで生じるエラーです。

赤枠/赤線のように縮小/バイアスで影響が出ています。



【原因】

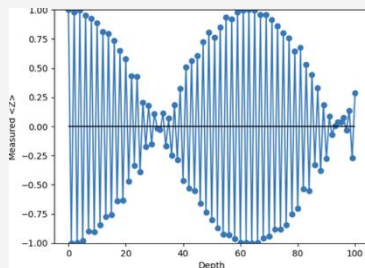
- ✓ 誤判別（量子ビット状態の読み出し）
- ✓ 不完全なリセット（以前に測定された状態）

【性質】

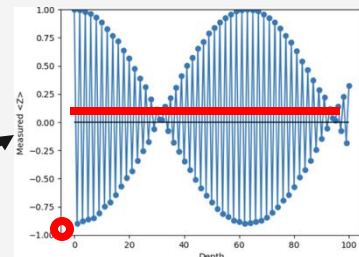
- ✓ 古典的なエラー（ビット反転エラー）

【対策】

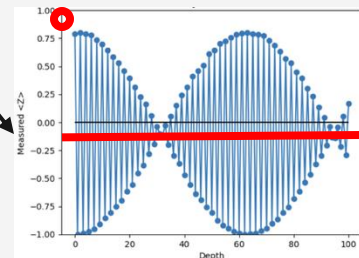
- ✓ （より良いキャリブレーション）、エラーの緩和/抑制、ロングショットインターバル



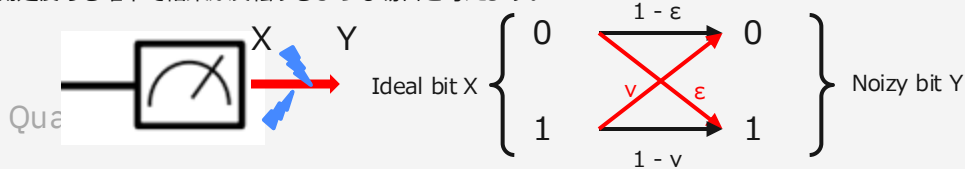
$\epsilon=0$
 $v=0.05$



$\epsilon=0.1$
 $v=0$



測定後ある確率で結果が反転するような場合を考えます。



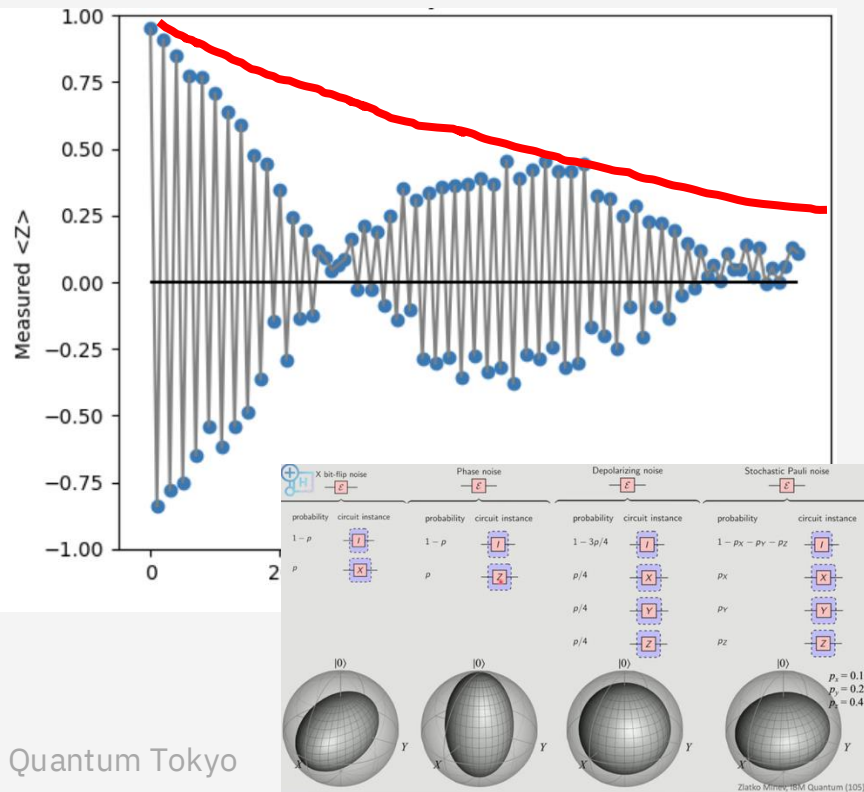
期待値は

$$\langle Z \rangle_{\text{noisy}} = a \langle Z \rangle_{\text{ideal}} + b$$

インコヒーレントエラー

量子状態がランダムに乱れるエラーです。

赤線のように減衰している部分で影響が出ています。



【原因】

- ✓ これらのエラーの起源は量子的であり、量子システムが環境と相互作用することで発生します。

【性質】

- ✓ 相互作用は混合状態を生み出し、出力にランダム性を導入してしまいます。

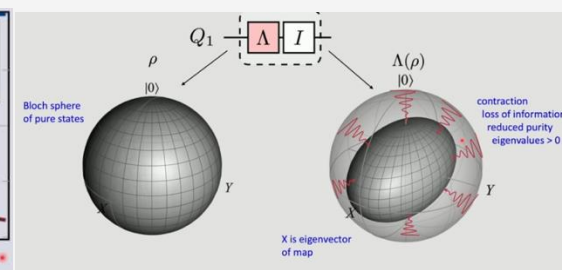
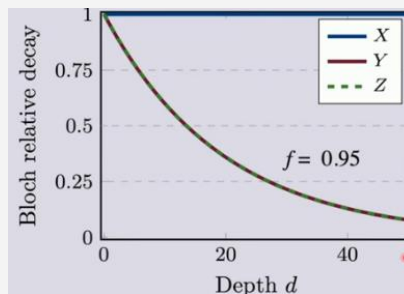
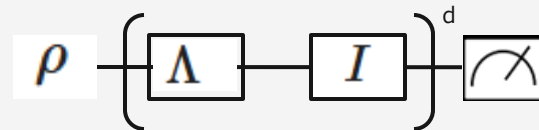
【対策】

- ✓ エラーの緩和

【直観的に】

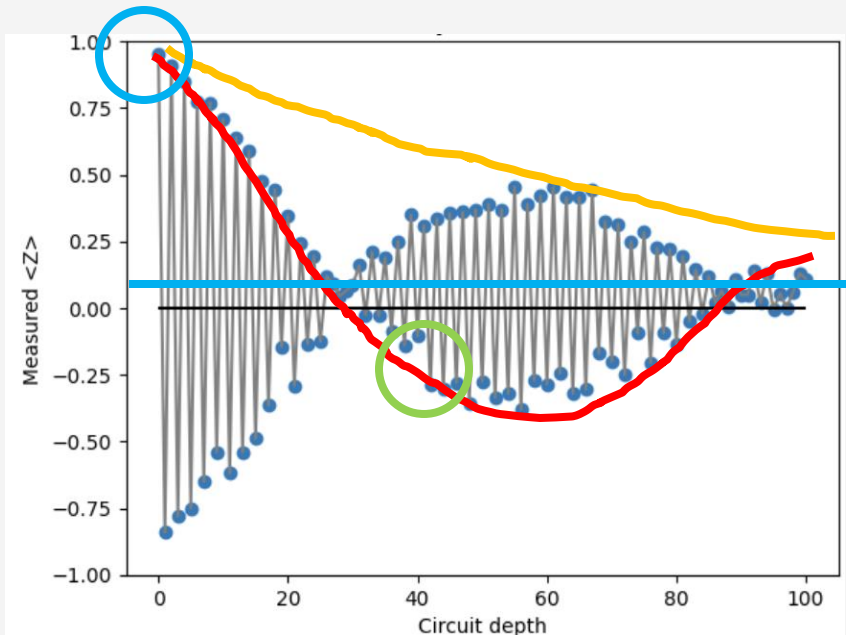
コヒーレントエラーは毎回同じ方向にずれるのに対して、インコヒーレントエラーではランダムにずれるので平均すると偏りが打ち消しあい、減衰していくイメージです。

確率 p で X が確率 $1-p$ でなにもしないようなエラーがかかる場合を考えます。







量子エラー

まとめるとクイズの結果に対して影響を与えていたエラーは以下のとおりとなります。



【量子エラー】

グラフ	エラー	説明
	コヒーレントエラー	量子状態が系統的にずれていくエラーです。
	サンプリングエラー	量子測定の確率的な性質や有限回の測定回数に由来するエラーです。
	SPAMエラー	SPAMエラーとは量子ビットの準備と読み取りで生じるエラーです。
	インコヒーレントエラー	量子状態がランダムに乱れるエラーです。

エラー抑制・緩和技術

量子エラー訂正？量子エラー抑制？量子エラー緩和？

ノイズによるエラーにどう対処すればいいのでしょうか？

【量子エラー訂正】

【考え方】

- ✓ 「エラーが起きても検知して直す」仕組みです。
- ✓ 情報を冗長に符号化します。
- ✓ シンドローム測定によりエラーを検出し、回路的に修正します。

【特徴】

- ✓ 正しく動作すれば「論理量子ビット」は長時間安定します。
- ✓ 実用化には大量の物理量子ビットが必要となります。

【例】

- ✓ Shor符号
- ✓ Surface code

【量子エラー抑制】

【考え方】

- ✓ 「そもそもエラーが起きにくくする」仕組みです。
- ✓ 量子ビットを環境から守る、ノイズを打ち消す、操作を工夫します。

【特徴】

- ✓ ハードウェアやパルス設計のレベルで工夫します。
- ✓ エラーそのものを物理的に減らすので、追加のビットは不要となります。
- ✓ ただし完全には消すことはできません。

【例】

- ✓ Dynamical Decoupling (DD)
- ✓ 最適パルス設計
- ✓ ハードウェアの冷却・絶縁

【量子エラー緩和】

【考え方】

- ✓ 「エラーは起きる前提で、計算結果に与える影響を後から補正する」仕組みです。
- ✓ 出力分布を統計的に補正したり、ノイズを外挿して「ノイズがなかったらこうだった」と推定します。

【特徴】

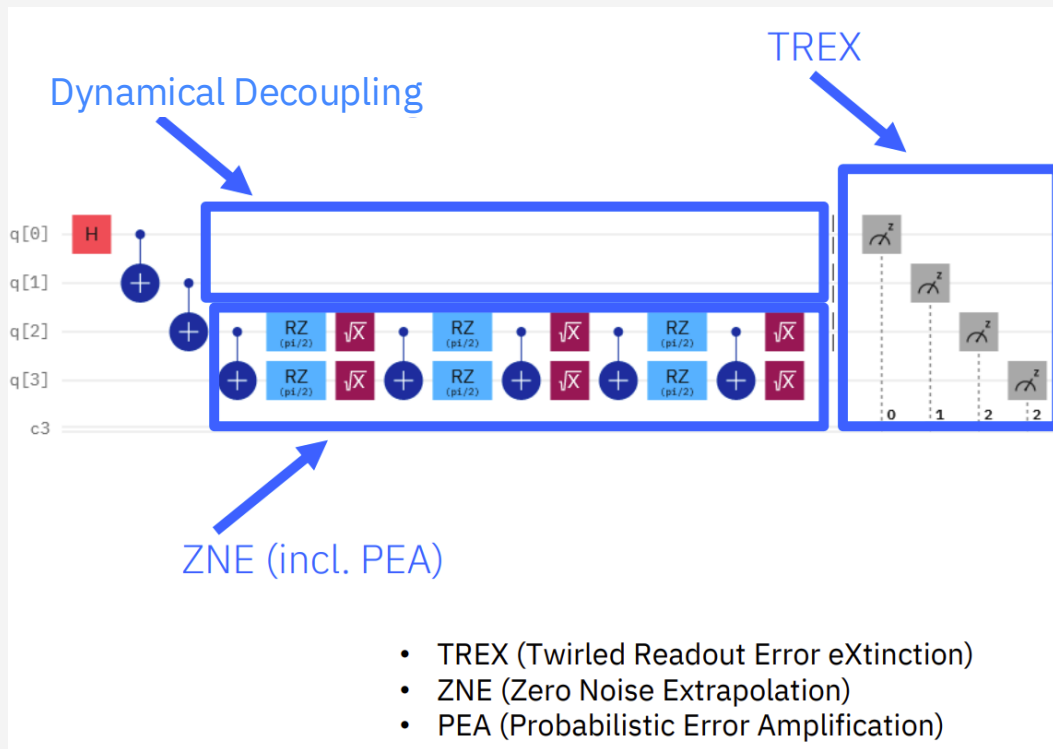
- ✓ 古典的ポストプロセス（測定データを補正）になります。
- ✓ 量子ビットの数が少なくても使えます。なのでNISQ時代で重要です。
- ✓ 完全な補正はできません。

【例】

- ✓ Zero Noise Extrapolation (ZNE)
- ✓ Twirled Readout Error eXtinction (TREX)

エラー抑制および緩和技術

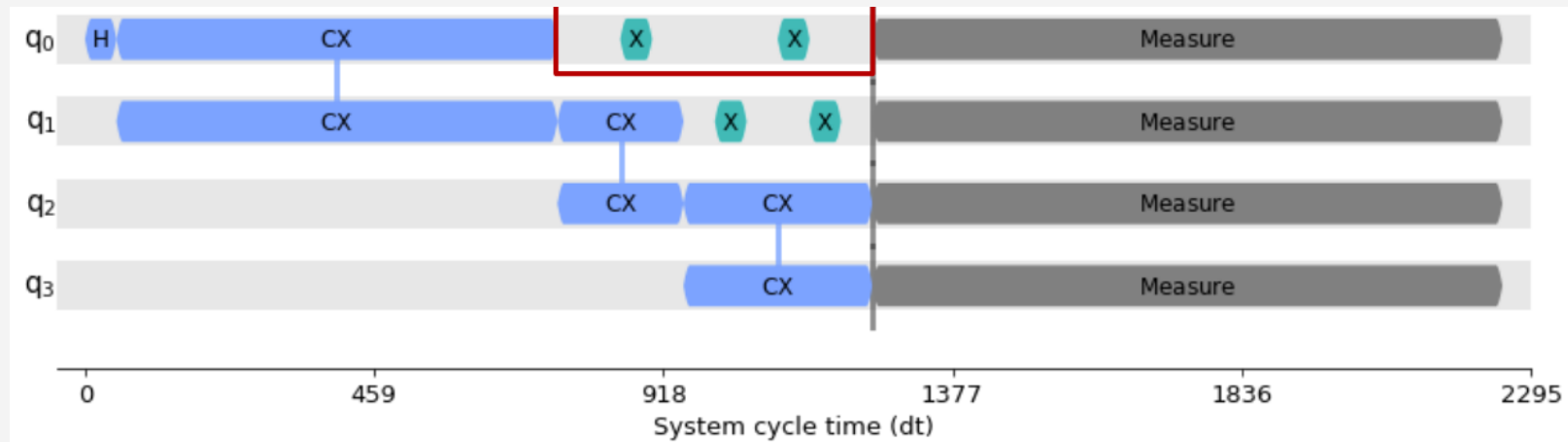
エラーの種類によって、抑制と緩和の手法も異なります。異なる種類の手法を組み合わせることも可能です。



エラー抑制 : Dynamical Decoupling (DD)

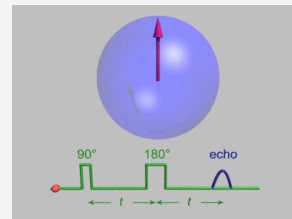
量子ビットのアイドル時間におけるエラーを効果的に抑制します。

✓ 挿入ゲートは恒等演算子になるように追加されます。例えば $X-X$ 、 $X-Y-X-Y$



【直観的な仕組み】

1. 量子ビットを放置すると環境ノイズが蓄積してズレが大きくなります。
2. 適切なタイミングでnパルス（反転操作）を入れます。
3. 反転させると1によるズレが反転後逆向きに変わるので全体で打ち消しあうことになります。



エラー緩和：Twirled Readout Error eXtinction (TREX)

ランダムなパウリ操作で測定誤差を単純化し、期待値を簡単に補正する手法です。

【実際の測定】

1. 1量子ビットを測定するとき、理想的には状態 $|0\rangle$ を測れば常に0に、状態 $|1\rangle$ を測れば常に1になります。ですが実際にはノイズにより

- $|0\rangle$ が誤って1と判定される確率 r_0
- $|1\rangle$ が誤って0と判定される確率 r_1

が存在します。

2. Z演算子の期待値を測るとき

- 理想の期待値は $\langle Z \rangle = p(0) - p(1)$
- 実際の期待値は $E = a \langle Z \rangle + b$

という形に歪みます。

a：縮み

b：バイアス



【トワリング（ランダム化）で平均化】

3. 測定する前に、わざとランダムにビットを反転（Xゲート）してみます。

- 半分の回路ではそのままにします。
- 測定半分の回路では例えばXをかけて測定します。ただし測定後に結果を反転して元に戻します。

こうすると、先ほどのバイアス項bは消すことができます。

4. 校正回路を利用してaを求めます。

5. 3で得た未知の状態を測定したときの期待値を4で得たaで割ることでノイズ補正後の期待値を得ることができます。

これよりTREXは個別に r_0 、 r_1 を知らなくても補正することができる仕組みになっています。

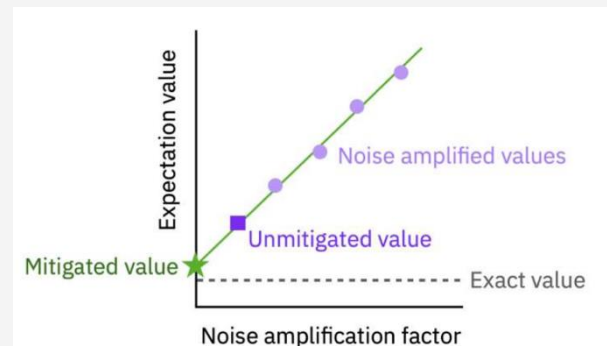


エラー緩和 : Zero Noise Extrapolation (ZNE)

わざとノイズを強めて、その傾向からノイズゼロを推測する方法です。

【ZNEの流れ】

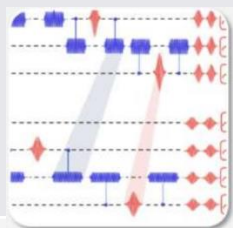
1. 本当は「ノイズのない量子計算結果」が欲しいです。
2. でも実際の量子コンピュータはノイズが入ってしまいます。
3. そこで、同じ回路を「ノイズをわざと増やした状態」で何通りか実行します。例: 通常（ノイズ倍率 = 1）、ノイズ2倍、ノイズ3倍 …
4. 結果が「ノイズ倍率」に対してどう変化するかをフィットして、ノイズ倍率 = 0 に外挿（エクストラポレート）します。



【ノイズ増幅】

パルスストレッチング

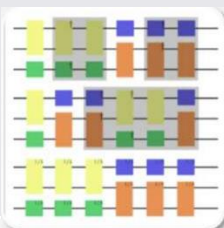
パルスの長さを伸ばして、ゲート動作を遅くします。遅ければ環境ノイズに晒される時間が長くなるので、ノイズが増えます。



Kandala et al. Nature (2019)

Gate folding

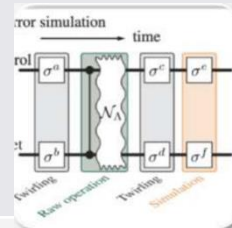
ゲート U を $UU^{\dagger}U$ に置き換えます。論理的には同じですが、実行時間が長くなりノイズが増えます。



Shultz et al. PRA (2022)

Probabilistic Error Amplification (PEA)

複雑なノイズをパウリトワリングでパウリチャネル化（IXYZのどれかが確率で入るだけ）のシンプルなノイズにして、その確率を意図的に増やすことでノイズを増幅させています。



Li & Benjamin PRX (2017)

Qiskitにおける設定

EstimatorOptions

Estimator primitiveを使用する場合、 EstimatorOptions() オブジェクトを使用してエラー緩和とエラー抑制技術のオプションを設定することができます。

Options	Sub-options	Sub-sub-options	Choices	Default
default_shots				4096
optimization_level		0 / 1		1
resilience_level		0 / 1 / 2		1
dynamical_decoupling	enable		True / False	False
	sequence_type		'XX' / 'XpXm' / 'XY4'	'XX'
	extra_slack_distribution		'middle' / 'edges'	'middle'
	scheduling_method		'asap' / 'alap'	'alap'
resilience	measure_mitigation		True / False	True
	measure_noise_learning	num_randomizations		32
		shots_per_randomization		'auto'
	zne_mitigation		True / False	False
	zne	noise_factors		(1, 3, 5)
		extrapolator	'exponential' / 'linear' / 'double_exponential' / 'polynomial_degree_(1 <= k <= 7)'	('exponential', 'linear')
twirling	enable_gates		True / False	False
	enable_measure		True / False	True
	num_randomizations			'auto'
	shots_per_randomization			'auto'
	strategy		'active' / 'active-circuit' / 'active-accum' / 'all'	'active-accum'

resilience_level	説明
resilience_level = 0	エラー緩和なし
resilience_level = 1	測定トワリング+トワリング読み出しエラー抑制 (TREX)
resilience_level = 2	resilience_level_1 + ゼロノイズ外挿 (ZNE) + ゲート・トワリング

resilience_level を変化させることで、異なるエラー緩和技術を適用します。

TREX

```
estimator.options.twirling.enable_measure = True
estimator.options.twirling.num_randomizations = "auto"
estimator.options.twirling.shots_per_randomization = "auto"
```

ZNE

```
estimator.options.resilience.zne_mitigation = True
estimator.options.resilience.zne.noise_factors = [1, 3, 5]
estimator.options.resilience.zne.extrapolator = ("exponential", "linear")
```

Dynamical decoupling

```
estimator.options.dynamical_decoupling.enable = True # Default: False
estimator.options.dynamical_decoupling.sequence_type = "XX"
```

まとめ

量子ノイズと量子エラー緩和技術の基本的な内容を一緒に勉強しました。

【量子ノイズ/エラーとは】

量子ノイズとは量子状態の測定結果として期待される結果を乱す、望ましくない変換全般を指します。

エラー	説明
コヒーレントエラー	量子状態が系統的にずれていくエラーです。
サンプリングエラー	量子測定の確率的な性質や有限回の測定回数に由来するエラーです。
SPAMエラー	量子ビットの準備と読み取りで生じるエラーです。
インコヒーレントエラー	量子状態がランダムに乱れるエラーです。

【エラー抑制・緩和技術】

エラー技術	説明
Dynamical Decoupling (DD)	量子ビットのアイドル時間におけるエラーを効果的に抑制します。
Twirled Readout Error eXtinction (TREX)	ランダムなパウリ操作で測定誤差を単純化し、期待値を簡単に補正する手法です。
Zero Noise Extrapolation (ZNE)	わざとノイズを強めて、その傾向からノイズゼロを推測する方法です。

【Qiskitにおける設定】

Estimator primitiveを使用する場合、EstimatorOptions() オブジェクトを使用してエラー緩和とエラー抑制技術のオプションを設定することができます。

最後に：もっと勉強したい方へ



Q Search



Quantum Tokyo へようこそ

学習コンテンツ

Qiskit の始め方

IBM Quantum Platform 教材 日本語訳

ユーティリティ・スケール量子コンピューティング

IBM Research Blog 日本語版

(旧) Qiskit テキストブック 日本語版

(旧) Qiskit テキストブック (Qiskit コース) 日本語版

(旧) Qiskit ドキュメント・チュートリアル 日本語版リンク集

IBM Quantum Challenge

Qiskit Global Summer School

(Qiskit 夏の学校) 資料 日本語版

Quantum Tokyo 過去イベント資料



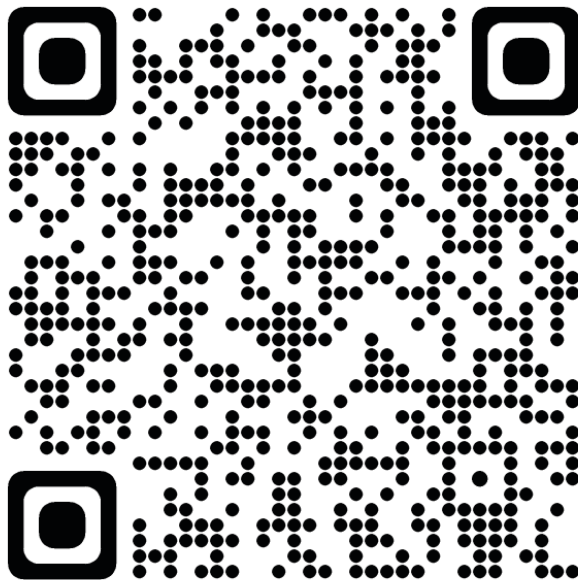
ユーティリティ・スケール量子コンピューティング

概要

このイベント・リプレイ・コースは、IBM Quantum®が東京大学と共同で開発し実施した14のLessonとLabで構成されています。このコースでは、量子コンピューティングにおける幅広い重要なトピックを網羅しつつ、実用規模（ユーティリティ・スケール）の量子計算を構築することに重点を置いています。最終的な結果として、2023年6月にNature誌の表紙を飾った論文と非常によく似た課題を扱います。

翻訳元はこちらです：[IBM Quantum Learning の Utility-scale quantum computing](https://quantum-tokyo.github.io/introduction/courses/utility-scale-quantum-computing/overview-ja.html)

- [1. はじめに](#)
- [2. 量子ビット・量子ゲート・量子回路](#)
- [3. 量子テレポーテーション](#)
- [4. グローバーのアルゴリズム](#)
- [5. 量子位相推定](#)
- [6. 量子変分アルゴリズム](#)
- [7. 量子系のシミュレーション](#)
- [8. 古典計算によるシミュレーション](#)
- [9. 量子ハードウェア](#)
- [10. 量子回路の最適化](#)
- [11. 量子エラー緩和](#)
- [12. 量子ユーティリティの実験 I](#)
- [13. 量子ユーティリティの実験 II](#)
- [14. 量子ユーティリティの実験 III](#)



<https://quantum-tokyo.github.io/introduction/courses/utility-scale-quantum-computing/overview-ja.html>

