

誤り耐性量子コンピュータ

量子コンピュータは、量子的な重ね合わせ状態を取ることのできる素子、量子ビットを使うことで、特定の計算タスクを古典コンピュータよりも高速に実行できる計算機です。しかしながら、量子ビットの状態は、一般に精密な制御が難しく、また外部環境との相互作用によって容易に破壊されてしまう繊細なものです。現在実現している最先端の量子コンピュータであっても、一回の操作の誤り率は0.1 % 程度です。一方で、古典コンピュータの計算能力を超えると期待される、例えば量子化学計算のような多くの有用な応用先では、 10^{10} もの操作回数を必要とします。0.1 % の誤り率では、到底このような計算を実行することはできません。またハードウェアを改良して、誤り率を例えば 10^{-10} まで下げることは、技術的に非常に困難な課題です。^[1]

このような誤り率の問題を解決するのが、誤り耐性量子計算という技術です。誤り耐性量子計算では、1つの量子ビットの情報を、多数の量子ビットの上に冗長化することで、計算途中で発生する誤りを検出し、訂正する仕組みを構築します。今回は、誤り耐性量子計算の基本的な考え方を紹介します。

誤り訂正

誤り訂正は、情報を冗長化することで、誤りが発生しても元の情報を復元できるようにする技術です。最も単純な例として、反復符号があります。反復符号では、0 と 1 のビット情報をそれぞれ 000...0 と 111...1 というビット列で表すと約束します。このような約束を**符号化**と呼びます。符号化された状態であれば、もし少数のビットが何らかの原因によって反転してしまっても、そのビット列が 00...0 か 11...1 のどちらに近いかを判断することで、誤りを検出し、訂正することができます。図1にその概念図を示しました。

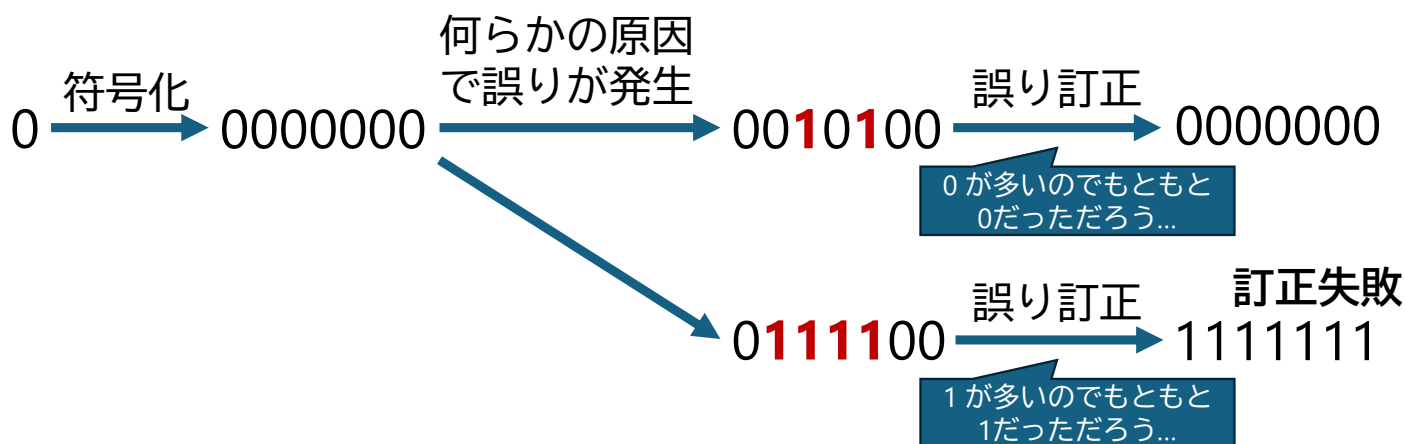


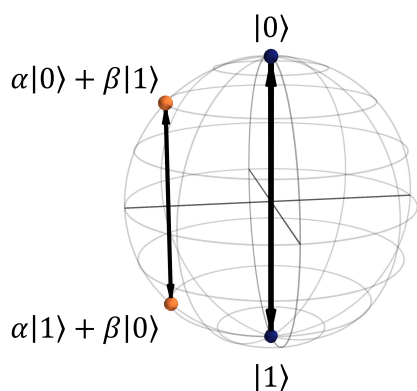
図1 反復符号による誤り訂正の概念図。0, 1 という情報をそれぞれ 0000000 と 1111111 というビット列で表すと約束すると、3ビットの誤りまでは訂正できます。

その仕組みから想像されるように、反復符号の誤り訂正能力は、冗長化の度合いに依存します。情報の保存に7ビットを用いる図1の例では、3ビットまでの誤りを訂正できますが、4ビット以上の誤りが発生すると訂正に失敗します。4ビット以上に誤りが発生する確率は、各ビットが $p \ll 1$ の確率で誤りを持つと仮定すると、おおよそ p^4 です。同じように考えると、情報の保存に9ビットを使えば、誤り訂正に失敗する確率はおおよそ p^5 となります。もし $p = 1\%$ だったとしても、7ビットを使えば 10^{-8} の確率で、9ビットを使えば 10^{-10} で、誤りを訂正できる計算です。このように、誤り訂正符号を用いることで、利用するリソース（物理ビット数）に対して指数的に誤り率を下げるができます。

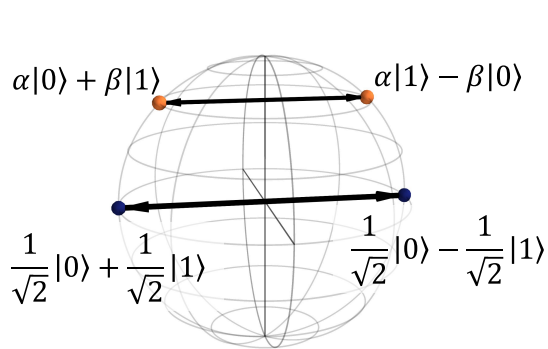
上の反復符号の議論は誤り率が $p \ll 1$ であることを仮定していました。一般に、 p があまりにも大きくなってしまうと、誤り訂正が行えなくなります。ある与えられた符号が、具体的にどの程度の誤り率まで訂正できるのかという値は、誤り訂正のしきい値と呼ばれ、その符号の構造によって決まります。しきい値以下の誤り率が達成されたとき、一般に符号を大きくしていくことによって、誤り訂正の失敗確率をいくらでも下げられることが知られています。そこで、情報伝送システムなどの設計では、物理的な誤り率を誤り訂正しきい値以下下げることが重要な課題です。

量子誤り訂正

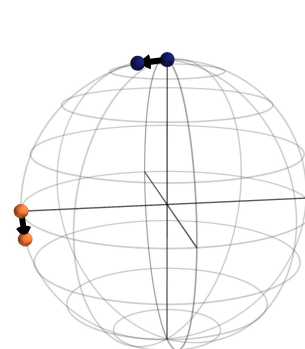
誤り訂正の考え方を量子ビットに対して応用するのが量子誤り訂正です。ただし、古典ビットの誤り訂正と大きく異なる点が2つあります。1つ目は、量子ビットの「誤り」には多種多様なものがあるという点です。古典ビットの誤りは、ビットの反転（0が1に、1が0に）という単純なものでしたが、量子ビットの誤りには、ビットの反転だけでなく、位相の反転や微妙な確率振幅のずれなど、より複雑なものが考えられます。図2に、量子ビットに発生する誤りの例を、ブロッホ球上で示しました。図2に示したように、量子ビットの誤りは、この球面上の点がある一定の規則にしたがって元の位置からずれることに対応します。このように様々な誤りを訂正可能な符号を構築することが、量子誤り訂正の難しさの一つです。



ビット反転ノイズ



位相反転ノイズ



微小回転ノイズ

図2 ブロッホ球上で示した量子ビットに発生する誤りの例。連載第1回でも登場したブロッホ球は、量子ビットの状態を表すための図で、量子ビットの1つの状態は球面上の1点に対応します。北極と南極が0, 1に対応し、それ以外の点はすべて0, 1の重ね合わせ状態を表します。量子ビットのエラーには、古典ビットでも発生するビットの反転だけでなく、位相の反転や微小な回転など、多種多様な誤りが考えられます。

2つ目は、量子的な重ね合わせ状態を保ったまま誤り訂正を行う必要があることです。その難しさを知るために、例えば量子版の反復符号を考えてみましょう。 $|0\rangle$ という情報を表すのに $|000\dots 0\rangle$ という状態を用いると約束し、 $|1\rangle$ という情報を表すのに $|111\dots 1\rangle$ という状態を用いると約束します。この約束のもとでは、 $\alpha|0\rangle + \beta|1\rangle$ という重ね合わせ状態を、 $\alpha|000\dots 0\rangle + \beta|111\dots 1\rangle$ という重ね合わせ状態によって表すことになります。

図3に、この量子誤り訂正符号の概念図を示しました。古典誤り訂正と決定的に異なる点は、データを保持している量子ビットを直接「見て」、0, 1の数を数えることができないという点です。0, 1の数を知ろうとして、符号化された状態を観測してしまうと、重ね合わせ状態が壊れてしまい、もともと保存しようとしていた情報が失われてしまいます。この情報の破壊を防ぐには、補助量子ビットを追加し、そのビットに、誤りの有無や場所という情報だけを抽出します。その後、そのビットのみを観測することで、量子重ね合わせ状態を壊さずに誤りの情報のみを抽出し、訂正を行うことができます。

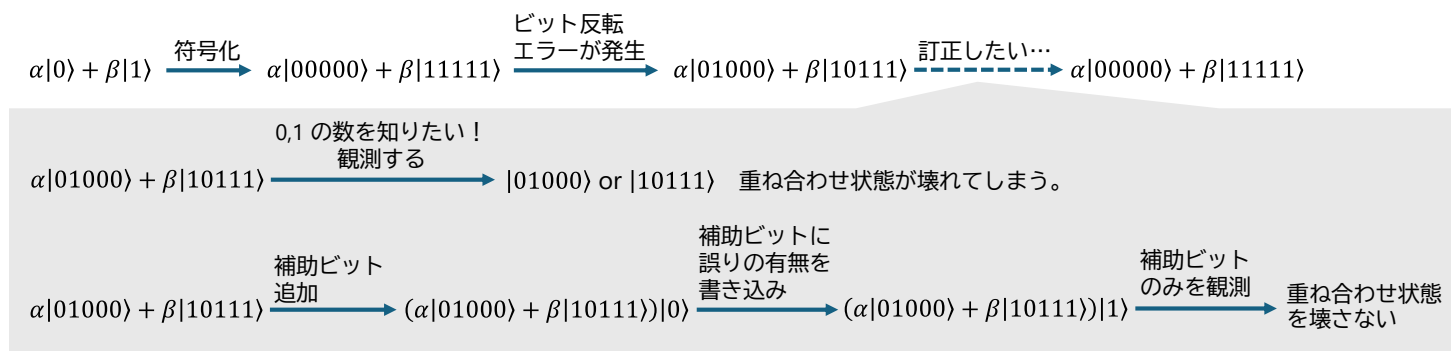


図3 量子反復符号による誤り訂正の概念図。量子ビットの重ね合わせ状態を保ったまま誤り訂正を行うためには、補助量子ビットを追加して誤りの有無や場所という情報だけを抽出する必要があります。

これらの困難を克服し、かつ現実的な物理系で実装できると考えられている代表的な量子誤り訂正符号として、表面符号があります。表面符号は、正方格子に量子ビットを並べた構造を持ち、隣接する量子ビット間のゲートのみを使って誤り訂正が行える符号化方式です。図4に、表面符号の概念図を示しました。

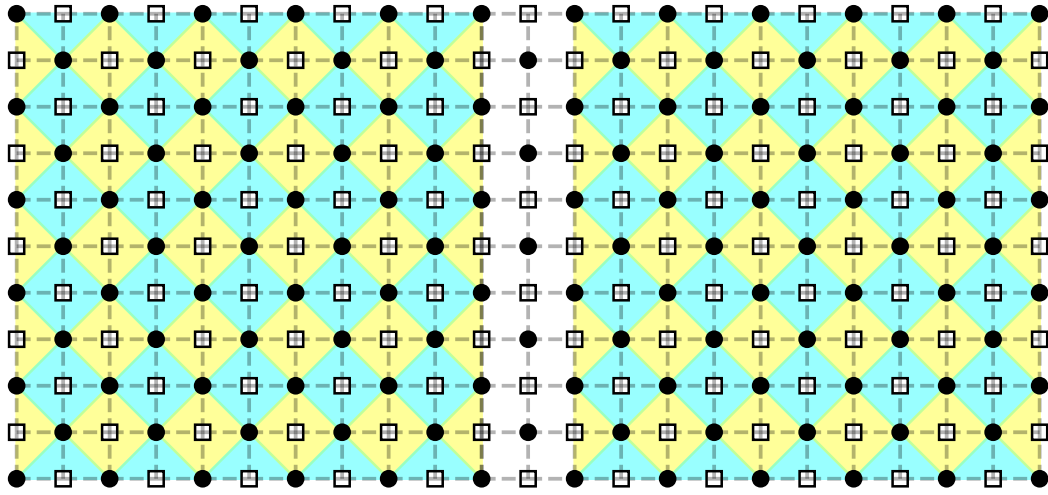


図4 表面符号の概念図。黒い点がデータ量子ビットを表し、白い点が誤り検出のための補助量子ビットを表します。黄色の塗りつぶしの中心にある補助量子ビットは、隣接するデータ量子ビットのビット反転エラーを、青色の中心にあるものは隣接ビットの位相反転エラーを抽出します。この図では左右に2個の論理量子ビットが描いています。

誤り耐性量子計算

適切な量子誤り訂正符号によって誤りを検出・訂正しながら行う量子計算のことを、誤り耐性量子計算と呼びます。どんな量子誤り訂正符号であっても誤り耐性量子計算を実現できるわけではありませ

ん。誤り耐性量子計算を実現するには、誤り訂正のために行う量子ビットへの操作や、演算のために行う量子ビットへの操作が、誤りの影響を増幅しないことが絶対条件です。特に、1 量子ビットに発生したエラーが、局所的なものにとどまるような操作を行うことが求められます。もし 1 量子ビットに発生したエラーが、全体の量子ビットに影響を及ぼすような操作を行ってしまうと、誤り訂正が不可能になってしまいます。図4に示した表面符号は、この条件を満たす量子誤り訂正符号の一つです。他にもさまざまな方式が提案されていますが、ハードウェアへの実装の容易さなどから、表面符号が最も注目されており、誤り耐性量子計算のためのロードマップも多くは表面符号を基本にしています。

例えば、量子化学計算のような応用先においては、表面符号をもとに、エラー率 0.1 % の超伝導量子ビットを 100 万個程度の集積化すれば、一定の成果が見込めると試算されています [J. Lee, et al., PRXQuantum 2, 030305 (2021)]。昨今、量子コンピュータの大規模化が飛躍的に進んでいるとはいえ、このような大規模化が実現されるためには多くの技術的ブレークスルーが必要です。しかし、最近では比較的大規模な量子誤り検出の実験的実現 [D. Bluvstein, et al., Nature 626, 58 (2024)] がされるなど、誤り耐性量子計算の実現に向けた研究はどんどん進んでいます。化学への応用も期待される誤り耐性量子計算の研究は、ハード・ソフトの両面でこれからも加速度的に進展していくことでしょう。

1. 連載第 2 回でも見たように、例えば超伝導量子ビットの制御は、電磁波のパルスを一定時間照射することで行われます。典型的には一つのゲート操作をナノ秒 (10^{-9} 秒) のオーダーの照射時間で実現しており、 10^{-10} の誤り率を達成するためには、照射時間を実に 10^{-19} 秒のオーダーで制御する必要があります。↩