応用

量子コンピュータの誤り訂正技術
―物理に即したトポロジカル表面符号―

徳永裕己（日本電信電話（株）NTT セキュアプラットフォーム研究所）

物理に即した誤り訂正とは

量子コンピュータの実現が難しい理由はなんであろうか？　一番大きな理由に挙げられるのがやはり誤りが起きやすいという点であろう．まず，スピンなどの量子状態を思い通りに正確に制御すること自体が難しい．そして，放っておいても原理的にだんだんと状態が壊れていってしまう（デコヒーレンスと呼ぶ）．もちろん現在我々が使っているコンピュータも放っておいて永久に壊れないというわけではないが，我々が日常的に使う時間ではほとんど安定しているので安心して使えるわけである．ところが量子コンピュータの場合には放っておいて安定している時間がせいぜい秒単位かもしれない（扱う物理状態によってはもっともっと短い）．物理を専攻としていない人にも，イメージを持ってもらうためにもう少し説明を続けよう．量子コンピュータは，量子力学が成り立つような原子，電子などの物質の最小単位を主に扱うことで実現するとされている．そして，量子ビットとして扱いたい原子などの周りにもそれらを並べるための入れ物や，制御するためのものなどさまざまなものがあり（当然それらも原子でできている），それらとも，物理の原理に則り自然に相互作用を起こしてしまう．このような余計な作用を除ききるのは非常に難しい．さらに言うと，たとえ周りに何もなくても真空と作用が起きてしまう．これらの不要な作用は誤りとなって積もっていくのである．

このような状況で，まったく物理的に誤りを起こさない量子コンピュータを作ることは可能であろうか？　この答えはほぼ間違いなく NO であろう．それでは問いを「論理的に誤りを起こさないことが可能か？」と変えるとどうだろうか．つまり，物理的に誤りは起きるのだがそれを誤り訂正しながら，論理的には正しく量子計算ができるか，という問いかけになる．この問いも量子コンピュータの研究黎明期には不可能なのではないかと思われたが，その後，答えは YES になるということが示された．しかし，そこには物理的にまだ不自然な仮定があったり，許される誤り率が 0.001% 程度と非常に小さかったりと，可能とはいえかなり悲観的なものであった．ところが，2000 年代後半になってこの見込みが大幅に改善した．なぜこのような改善が可能になったかというと，非常に物理に即した符号技術であるトポロジカル表面符号を用いた量子計算が構築されたからである．それまでの量子誤り訂正符号は既存の符号技術を量子コンピュータ用に無理やり当てはめたようなところがあった．しかし郷に入っては郷に従えで，物理の原理から来る基本制約を考慮したうえで，適した量子誤り訂正符号はできないだろうかと検討すると，有望な符号が発見された．そしてその符号を用いて誤り訂正を行いながら量子計算を行うフォールトトレラント量子計算が生み出され，以前より大幅に高い誤り率である 1% 程度が許されることが示されたのである．さてそれではその基本的な物理制約はなんであろうか？　それは粒子の相互作用は近距離のものほど強いということである．つまり遠く離れた量子ビット間で演算をするのは不自然であり，近くのものとだけ演算を行うのが物理的に自然ということになる．自然な構築法になれば操作の手間が少なくなり効率的になり，それが耐え得る誤り率にも直結するため符号としての性能も上がる

わけである．それではその符号とはどのようなものであろうか？　次章から具体的に説明を始めよう．

量子誤り訂正符号

まず量子ビットとその誤りについて述べる．物理的な基底状態 $|0\rangle=\begin{pmatrix}1\\0\end{pmatrix}$, $|1\rangle=\begin{pmatrix}0\\1\end{pmatrix}$ で構成される量子ビット $|\psi\rangle=\alpha|0\rangle+\beta|1\rangle$（ここで $|\alpha|^2+|\beta|^2=1$）に対して，誤りはパウリ演算子 $X=\begin{pmatrix}0&1\\1&0\end{pmatrix}$, $Z=\begin{pmatrix}1&0\\0&-1\end{pmatrix}$ を用いて表されるビットエラー $X|\psi\rangle=\alpha|1\rangle+\beta|0\rangle$ と位相エラー $Z|\psi\rangle=\alpha|0\rangle-\beta|1\rangle$ の2種類を考えればよい．古典的にはない位相エラーがビットエラーと双対的に存在するのが量子の特徴的な面である．そして，もちろん誤りはそれ以外にも連続的な量の誤りが起き得るのだが，量子状態の測定時に起こる射影からこの2種類またはその両方が起きた場合に誤りは離散化することができるのでこのように簡略化して考えることができる．

次に基本的な量子誤り訂正符号の構成について簡単に述べる．ここでは標準的な量子誤り訂正符号の構成法であるスタビライザー符号を念頭に説明する．量子ビットの符号語は，誤り耐性を持たすために複数の量子ビットからなる冗長化された論理基底 $|0_L\rangle$, $|1_L\rangle$ を用いて $|\psi_L\rangle=\alpha|0_L\rangle+\beta|1_L\rangle$ と書ける．この符号語は複数の量子ビットがもつれ合ったエンタングル状態になっている．この符号語に対する検査演算子 M_i が複数存在して，誤りがないときにはすべての M_i に対して $M_i|\psi_L\rangle=|\psi_L\rangle$ となり，誤りがあるときにはいくつかの M_i に対して $M_i|\psi_L\rangle=-|\psi_L\rangle$ となる．このように誤りがないときにはまったく状態を変えないスタビライザーを検査演算子として用いるためスタビライザー符号と呼ばれる．そして誤りがあるときには上記のように正負の符号の変化があり，これが測定結果に現れる．よって，測定結果から誤りがあったのかどうか，そしてどの検査演算子にたいして誤りが検出されたのか

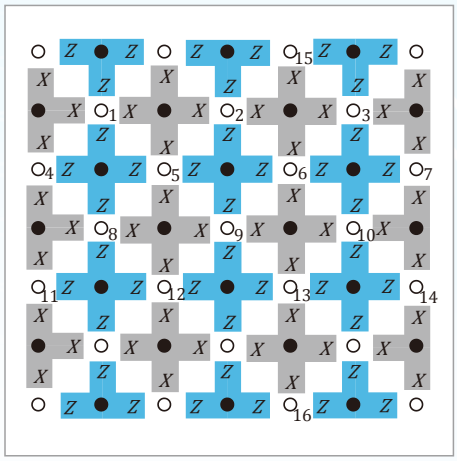

図-1　表面符号

という情報から，どこにどのような誤りがあったのかの判定が可能となるので，誤り訂正をすることができるようになる．通常スタビライザー符号においては，この検査演算子はパウリ演算子で表現される．従来の量子誤り訂正符号についての詳細な解説は文献1）をお勧めする．

トポロジカル表面符号

それではトポロジカル表面符号の構成を具体的に見ていこう（今後は単に表面符号と呼ぶ．トポロジーが関連する理由は後述する）．**図-1** に論理的な1量子ビットが複数の物理的な量子ビットを用いて冗長に符号化された符号語の構成を示す．物理的な量子ビットは2次元面に格子上に配置される．白丸の量子ビット全体で符号語が構成され，黒丸は検査演算子を測定するための補助的な量子ビットである．この図においては白丸で書かれた25個の物理的な量子ビットによって1つの論理量子ビットが構成されている．この格子のサイズを大きくする（物理的な量子ビット数を増やす）ことで符号語の冗長性が増し，誤り耐性が大きくなる．検査演算子は最近接の4つ（端は3つになる）の白丸の量子ビットに対するパウリ演算子を用いて $XXXX$ および $ZZZZ$ で表され，それぞれ位相エラー（Z エラー）およびビットエラー（X エラー）の検査演算子となる．誤りが検出される理由はパウリ演算子の非可換性 $ZX=-XZ$ から来る．たとえば**図-2** において番号5の量

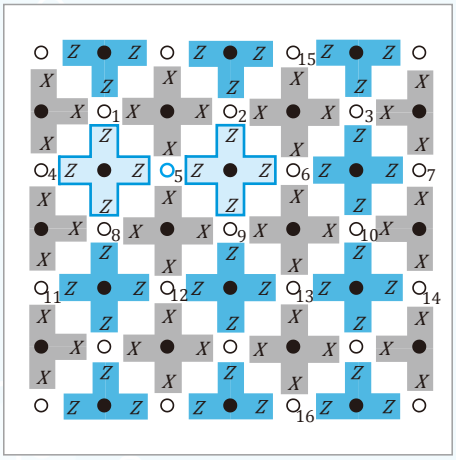

図-2
検査演算子による誤り検出

子ビット1カ所にXエラーが起きたときには，

$$Z_1 Z_4 Z_5 Z_8 (X_5|\psi_L\rangle) = -X_5 (Z_1 Z_4 Z_5 Z_8 (|\psi_L\rangle)$$
$$= -X_5 |\psi_L\rangle$$

となり符合の変化が測定において異常として検出される．同様に$Z_2 Z_5 Z_6 Z_9$の測定においても異常が検出される．そうすると，この2つの検査演算子の間にある5番の量子ビットが怪しいということになり，実際ここに誤りが起きたことを検出でき，誤りを戻してあげるためにX演算子を施すことができる．この符号が物理上重要なことはこの検査演算子が最近接の4つの量子ビットに対するものであることであり，白丸の量子ビット4つに囲まれた黒丸の量子ビットに対して，白丸と黒丸の2量子ビット間の制御演算を行った後に，黒丸の量子ビットを測定することでこの検査演算子の測定が可能であることである．符号語の準備についても似たような操作で可能である．つまり演算は隣り合った量子ビット間のみに行えれば，誤り訂正が可能な構造になっている．以前の量子誤り訂正符号では，遠く離れた量子ビット間の演算を必要とし，これは物理的な実現方法が困難であった．離れた量子ビット間の演算を近接した演算だけでシミュレートすることも可能だが，これは演算数が多くなることで量子計算に許される誤り率を大幅に落とすことになってしまっていた．

さて，以上が表面符号の誤り訂正の過程であるが，いったいこの符号はどの程度の誤りまで訂正が可能なのであろうか？　訂正が不可能となる例を見てみよう．図-1で4, 5, 6, 7番の量子ビットにすべてXエラーが起きたとしよう．このときエラーを検出するべき位置にある検査演算子$Z_1 Z_4 Z_5 Z_8$，$Z_2 Z_5 Z_6 Z_9$, $Z_3 Z_6 Z_7 Z_{10}$の測定結果からはすべて異常が検出されない．なぜならすべての測定において2カ所の量子ビットに誤りがおきていることになり，符合の逆転が2度起こり元に戻ってしまうからである．そして実はこの4, 5, 6, 7番の量子ビットにすべてXエラーが起きるという事象は，符号語のまま，元々の論理的な量子ビットに対してX演算子を施したこと，つまりビット反転を行ったことに相当する．つまり元々の論理量子ビットのデータを符号語のまま完全に変えてしまったことに相当する．よってここまでの大きな誤りはもう完全に検出できない．また別の例として，5, 6番の2カ所にXエラーが起きた場合には，検査演算子$Z_1 Z_4 Z_5 Z_8$と$Z_3 Z_6 Z_7 Z_{10}$において異常が検出されるが，このときもどこに誤りが起きたのか正確に判断できない．なぜなら4, 7番の2カ所に誤りが起きたときも同じ検査演算子で異常が検出されるからである．このとき間違った方を選び誤り訂正を施すと，4, 5, 6, 7番の量子ビットすべてにX演算子を施したことになってしまい，論理的にビット反転を行ってしまう結果となる．ちなみに，格子のサイズをもっと両側に大きく十分に符号を冗長にしてあれば5, 6番の2カ所にエラーが起きた場合のみ$Z_1 Z_4 Z_5 Z_8$と$Z_3 Z_6 Z_7 Z_{10}$において異常が検出されるため，正しく誤りが訂正できる．このように表面符号では連なった誤りに対してはその端点の検査演算子で誤りが検出される．現実には誤りがランダムに散らばって起こることが想定される．このとき検査演算子の測定結果からどのように誤りの場所を決めるのであろうか？　誤り率が小さいときは，最も少ない数の誤りが起きた場合を推定する最小重み完全マッチングアルゴリズムを用いることでほぼ完璧に誤りの位置を推定できることが知られている．また，格子のサイズを大きく符号の冗長性を高めるほうがより正確に誤り訂正が可能となる．しかし誤り率が高くなってくると，格子のサイズを大きくしても上手くいかなくなる限界がやってくる．この閾値は数値計算でしか見つかってい

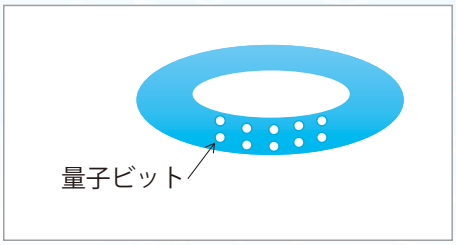

図 -3
トーラス表面上に配置された量子ビット

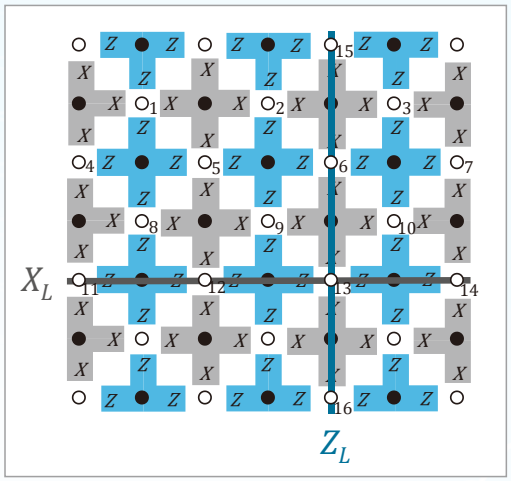

図 -4
論理演算子

ないがおよそ 10.3% 程度であることが知られている．この値は，符号の性能を見るために白丸の量子ビットに対するメモリエラーのみを考え，演算や測定によるエラーは考慮していない場合である（演算や測定にエラーが起きる量子計算の場合に耐え得る誤り率は，はじめに述べたように，1% 程度となることが知られている）．この 10.3% という値はこの種の量子誤り訂正符号の誤り率の一般的な理論限界値である量子 Gilvert-Varshamov 限界に近いことから，表面符号は物理的制約の中で可能なシンプルな構造でありながら優れた符号であることが分かる．また図 -1 のような正方格子ではビット誤り，位相誤りともに同じ許容誤り率となるが，蜂の巣形やカゴメ形などに格子の形状を変えると非対称な許容誤り率が得られることが分かっている [3]．実際の物理においてはビットエラーよりも位相エラーの方が大きい場合が多く，このような格子形状の変形により位相エラーへの耐性を高くするのも効果的である．

本章の最後に表面符号がトポロジーと関連するゆえんを書いておこう．元々，この符号は 2 次元格子の端がぐるっと回って反対側とつながっている周期境界条件で考えられていた．上下と左右の端をそれぞれつなげるとこれは**図 -3** のようにドーナツ型のトーラスになり，その表面に量子ビットがのっているという形をしていた．そして複数のトーラスに対して結び目を作っていくような操作が論理量子ビット間の制御演算に対応していた．このような 3 次元構造のものを物理的に実装するのは困難であったのだが，実は同じトポロジーを 2 次元正方格子上でシミュレートできることが示されたため物理実装も現実的に考えられるようになった．

表面符号を用いたフォールトトレラント量子計算

本章では，表面符号を用いてどのように量子計算を行っていくのかを紹介しよう．計算をしている間にもどこにでも誤りは起き得るため，符号語のまま計算を進めていき，定期的に誤り訂正を行っていく必要がある．ここでの鍵は符号語のまま汎用的な量子計算をすべて行えるかということになる．この符号語のままの計算について，実は前章でも一部を説明していたことになる．図 -1 において論理的なビット反転を符号語のまま行いたいときは，たとえば 4, 5, 6, 7 番の量子ビットにすべて X 演算子を施せばよい．端から端につながっていれば良いので 11, 12, 13, 14 番でも構わない．論理的な位相反転を行うときは，たとえば 15, 6, 13, 16 番の量子ビットすべてに Z 演算子を施せばよい．**図 -4** で示すように，これらが符号語のまま行う論理的なパウリ演算子 X_L と Z_L になる．それでは汎用的な量子計算を行うためにはどのような量子ゲートがあれば十分であろうか？　この解としては，2 量子ビット間の制御 NOT ゲート

$$\mathrm{CNOT}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$

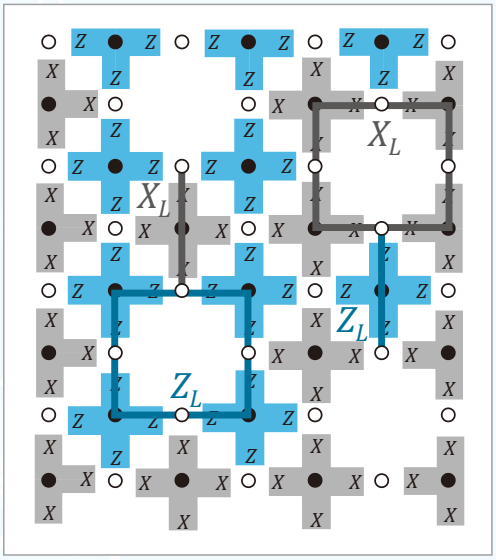

図 -5
複数の論理量子ビットの符号化

および特定の 1 量子ビットゲートである T ゲート $T=\begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}$ などがいくつかあれば十分であることが知られている．以下に制御 NOT ゲートと T ゲートをどのように符号語のまま行うかを説明する．前章では論理的な量子ビットは 1 つのみであったが，まず計算をするためには複数の論理量子ビットを用意しなければならない．ところが，図 -4 のような構造では，論理演算子が端から端まで伝わっているためこれ以上論理量子ビットを増やすことができない．これを可能とする構造を 1 つの 2 次元面内に用意することができる．**図 -5** に示すように検査演算子を一部取り除き穴が空いたような構造を作ると，この穴を端にすることができるので，論理演算子を 2 次元格子の内部に作ることができる．穴をぐるっと一周する演算子と 2 つの穴をつなぐ演算子が論理演算子となる（X_L と Z_L のどちらがそれぞれにあてはまるかで，図 -5 のように 2 通りのタイプがある）．トポロジー的には同じなのでこれで 2 次元格子内に複数の論理量子ビットを作ることができるようになる．符号の冗長性を大きくするにはこの穴のサイズと穴の間隔を大きくする．

次に重要なことは，これらの符号化された論理的な 2 量子ビット間で符号化したまま論理的な制御演算を行うことである．詳細は文献 2）などに譲るが，図 -5 のような符号語の構造は最近接の量子ビット

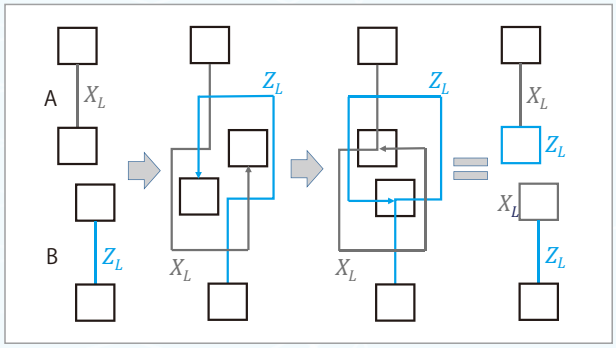

図 -6　制御 NOT ゲート

に対する演算および測定を用いて，動かすことが可能である．すると**図 -6** のようにしてお互いに結び目を作るような移動を行うことができる．これが実は符号語のまま行う論理的な制御 NOT ゲートに相当する．その理由を説明しよう．A 側を制御ビット，B 側を標的ビットとした制御 NOT ゲートは以下のように演算子を変換する．

$$I_A X_B \rightarrow I_A X_B,\ Z_A I_B \rightarrow Z_A I_B, \quad (1)$$

$$X_A I_B \rightarrow X_A X_B,\ I_A Z_B \rightarrow Z_A Z_B. \quad (2)$$

たとえば A 側が X_L 演算子でスタビライズされている論理量子ビット $(|0_L\rangle + |1_L\rangle)/\sqrt{2}$ で，B 側が Z_L 演算子でスタビライズされている論理量子ビット $|0_L\rangle$ であるときに AB 間に制御 NOT 演算を施すと，状態はエンタングル状態 $(|0_L\rangle\,|0_L\rangle + |1_L\rangle\,|1_L\rangle)/\sqrt{2}$ となり確かにスタビライザーは $X_L\,X_L$ と $Z_L\,Z_L$ に変換され，上記（2）の変換が行われている．図 -6 を見ると，結び目を作るような操作によって A 側の論理演算子 X_L が B 側にコピーされ，B 側の論理演算子 Z_L が A 側にコピーされたことになり，確かにこの操作が制御 NOT 演算の役割を果たしていることが分かる．図 -6 の最後の等号は穴を 1 周する論理演算子はサイズによらず論理的には同一であることを示している（トポロジーとしては同一である）．図 -6 からは，図 -5 の左右に記載されたような異なるタイプの論理量子ビットに対してのみしか制御 NOT ゲートが施せないように見えるかもしれないが，図 -5 の左側のタイプの 2 つの論理量子ビットに対して制御 NOT 演算をするときは，補助的に右側のタイプの論理量子ビットを介することで可

能になる．

最後に1量子ビットゲートのTゲートを符号化したまま行う方法について説明する．実はこの部分がフォールトトレラント量子計算で最も手間がかかる部分になっている．なぜなら，このゲートはスタビライザー符号の構成に基本的に用いられているパウリ演算子間の変換であるクリフォード群の要素に含まれないため符合との相性が悪く，符号語のまま演算するのが最も非効率になってしまう．しかし，汎用的な量子計算を行うためには非クリフォード群のゲートが必要なためこれを行わざるを得ない．Tゲートを行うためには，補助的な論理量子ビット

$$|A_L\rangle = \left(|0_L\rangle + e^{\frac{i\pi}{4}}|1_L\rangle\right)/\sqrt{2}$$

を用いる．これをどう作るかは後に述べるが，これがあると，**図-7(a)**のように論理的な制御NOT演算や測定を用いて，符号化したまま論理的なT_Lゲートが施せるようになる（ただし，この測定結果は確率的に2通りあり，1方の測定結果においてはさらにクリフォード群の要素の論理的1量子ビット演算を施す必要がある）．$|A_L\rangle$は最初は冗長に符号化されていない状況から作らざるを得ない．**図-7(b)**のように，冗長性のない状態になら中心のX_L演算子に対応する1量子ビットに対して直接Tゲートを行うことができる．そして，その後十分冗長な符号のサイズまで大きくする．この間は十分な誤り訂正機能を持ってないためできあがった$|A_L\rangle$は誤りが大きいものになっており，これでは使い物にならない．しかし，これを何個も集めて，制御演算や測定を行うことで誤りの小さい$|A_L\rangle$を作り出すことができることが知られており，これはマジック状態蒸留と呼ばれている．これにより，非クリフォード群のゲートも符号化したまま行えるようになり汎用的な量子計算が行えるようになる．トポロジカル表面符号を用いたフォールトトレラント量子計算のさらなる詳細については文献2）をお勧めする．研究の発展についても2）の参考文献を参照されたい．

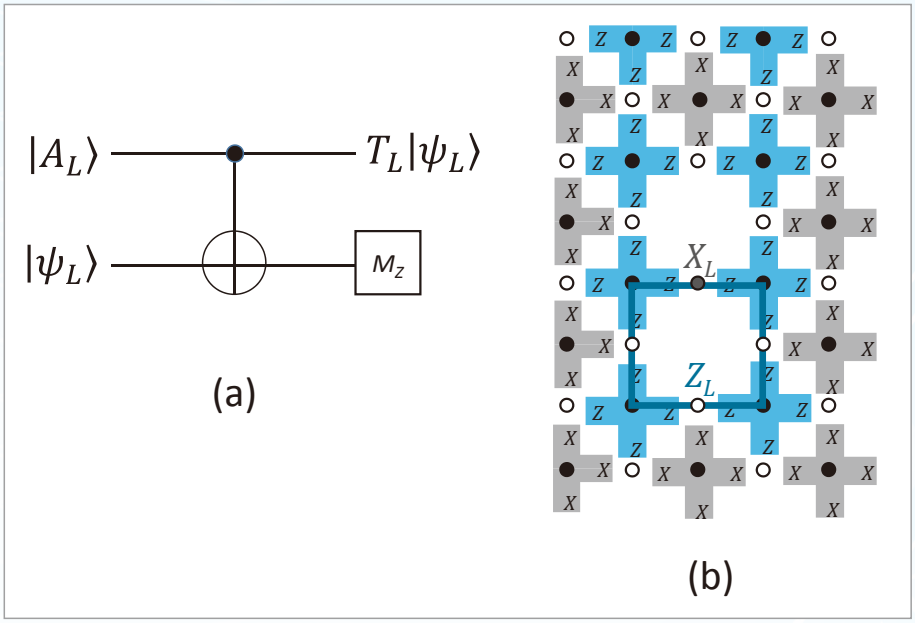

図-7 Tゲート

周辺技術と今後の展望

ここまで，表面符号を用いた基本的な量子計算の誤り訂正技術を見てきたが，最後に周辺技術について述べておこう．これまでは2次元格子上に多数の量子ビットが配置しており，隣接した量子ビットに対する制御演算はいつでも可能というモデルで話をしてきたが，状況によってはそれが難しいこともある．たとえば制御演算をいつでも行うことは難しい場合でも最初に大きな3次元状の量子もつれ状態を用意すれば後は1量子ビットの測定のみを用いて量子計算が行える測定型量子計算というモデルがあり，このモデル用の特殊な3次元量子もつれ状態を準備すればこの表面符号を実装した測定型のフォールトトレラント量子計算ができることが知られている．また，制御演算が確率的な成功（成功したかどうかの判定は分かる）の場合もある．たとえば1カ所に

並べられる量子ビット数に物理実装的に限界があり，光子を用いて離れた場所を連結するようなときにこのようなことが起きる．そのようなときでも表面符号を用いたフォールトトレラント量子計算が可能であることが分かっている[4]．

最後に誤り訂正技術の精神とは，避けられない誤りに対してリソース量を増やすことによって対処する技術といえる．実現の困難性からリソース量が増えることに難色を示す意見もあるが，誤り訂正なしでは基本的演算やメモリに求められる正確さが完璧に近くなってしまうのでこれは量子状態に対しては現実的には考えにくい．現状実装可能な量子ビット数はまだ少ないのだが，現在のコンピュータのリソース量が指数関数的に大きくなっていったように量子コンピュータのリソースも一度増えだすとぐんぐん伸びていく可能性もある．実装技術の今後のブレークスルーに期待したい．また，マジック状態蒸留などのアルゴリズムを良くすることでリソース量を減らす理論面からの改良も今後期待できる．このような効率化および実装面についての解説は次の根本氏の記事を参照されたい．

参考文献

1) Nielsen, M. A. and Chuang, I. L. : Quantum Computation and Quantum Information, Cambridge University Press (2000).
2) Fowler, A. G. et al. : Surface Code : Towards Practical Large-scale Quantum Computation, Phys. Rev. A, 86, 032324 (2012).
3) Fujii, K. and Tokunaga, Y. : Error and Loss Tolerances of Surface Codes with General Lattice Structures, Phys. Rev. A, 86, 020303 (R) (2012).
4) Fujii, K. and Tokunaga, Y. : Fault-Tolerant Topological One-Way Quantum Computation with Probabilistic Two-Qubit Gates, Phys. Rev. Lett. 105, 250503 (2010).

(2014 年 4 月 21 日受付)

徳永裕己　tokunaga.yuuki@lab.ntt.co.jp

NTT セキュアプラットフォーム研究所勤務．主任研究員（特別研究員）．量子情報処理の研究に従事．実現に向けた物理実装の面から，暗号，符号理論などの情報科学的な面まで幅広く研究を行っている．日本物理学会会員．