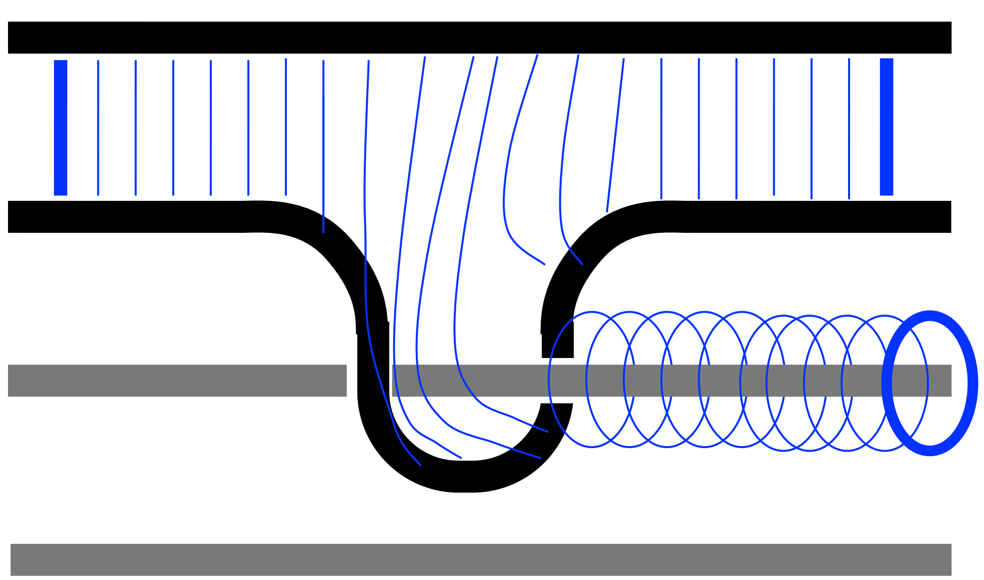
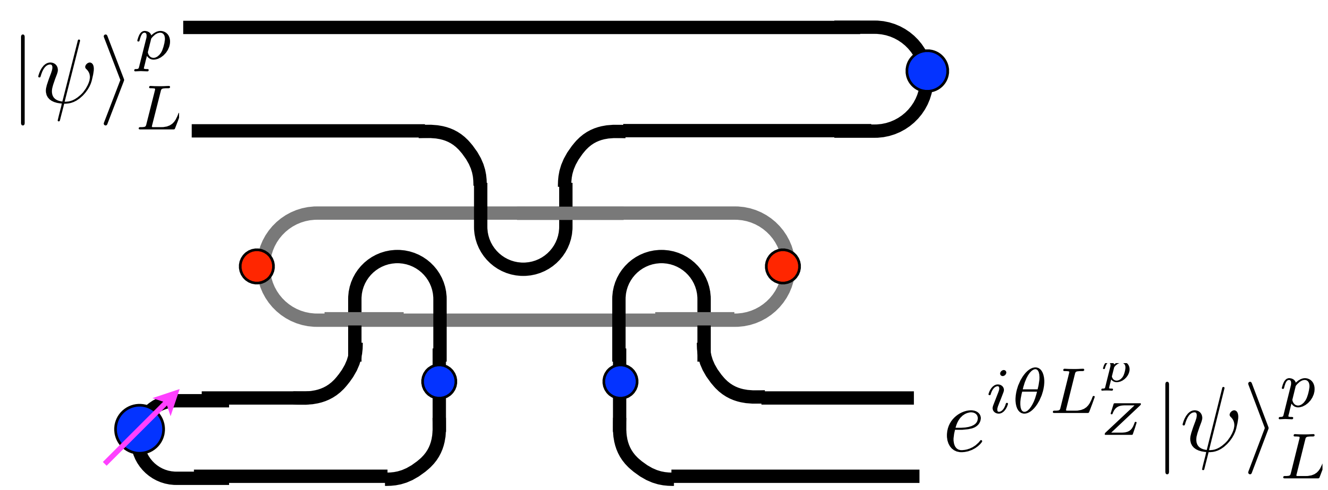
2012/11/29-30 @東京大学

量子計算超入門

藤井 啓祐

大阪大学 基礎工学研究科





目次

第一章:量子系の取り扱い

ここで勉強する内容：

Pauli行列, 基本演算, 回路図の見方

Pauli群, Stabilizer形式, Stabilizer状態,

Clifford演算，Pauli基底の測定，Gottesman-Knillの定理

non-Clifford演算，Kitaev-Solovayの定理

Qubit

◆ qubit = 量子２準位系

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

原子の内部状態，電子スピン，超伝導磁束など，直交する
２量子状態であればなんでもいい

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

ただし、 $\alpha, \beta \in \mathbb{C}$、 $|\alpha|^2 + |\beta|^2 = 1$.

Pauli行列

Pauli行列

◆ Pauli行列

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

- 反可換（anticommute）：$\{X, Z\} = XZ + ZX = -1$
- $XZ = iY$

Pauli行列

◆ Pauli行列

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

- 反可換（anticommute）：$\{X, Z\} = XZ + ZX = -1$
- $XZ = iY$

◆ qubitに対する作用

$X|0\rangle = |1\rangle \quad X|1\rangle = |0\rangle$ (bit-flip)

$Z|0\rangle = |0\rangle \quad Z|1\rangle = -|1\rangle$ (phase-flip)

$Y|0\rangle = i|1\rangle \quad Y|1\rangle = -i|0\rangle$ (bit&phase-flip + global phase)

Pauli行列

◆ Pauli行列

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

- 反可換（anticommute）：$\{X, Z\} = XZ + ZX = -1$
- $XZ = iY$

◆ qubitに対する作用

$X|0\rangle = |1\rangle \quad X|1\rangle = |0\rangle$ (bit-flip)

$Z|0\rangle = |0\rangle \quad Z|1\rangle = -|1\rangle$ (phase-flip)

$Y|0\rangle = i|1\rangle \quad Y|1\rangle = -i|0\rangle$ (bit&phase-flip + global phase)

◆ Pauli行列の固有状態（Pauli basis）

$Z \to |0\rangle, |1\rangle$ — *Z* basis

$X \to |+\rangle \equiv (|0\rangle + |1\rangle)/\sqrt{2}, |-\rangle \equiv (|0\rangle - |1\rangle)/\sqrt{2}$ — *X* basis

複合量子系（状態）

複合量子系（状態）

◆ 直積状態

- qubit a, b２つからなる合成系は $|\psi_a\rangle \otimes |\psi_b\rangle$ （テンソル積）で記述する.

複合量子系（状態）

◆ 直積状態

- qubit a, b 2 つからなる合成系は $|\psi_a\rangle \otimes |\psi_b\rangle$ （テンソル積）で記述する.
- いちいち $\otimes$ 書くのは面倒なので $|\psi_a\rangle|\psi_b\rangle$ や $|\psi_a\psi_b\rangle$ と書いたりもする.

複合量子系（状態）

◆ 直積状態

- qubit a, b 2 つからなる合成系は $|\psi_a\rangle \otimes |\psi_b\rangle$ （テンソル積）で記述する.
- いちいち $\otimes$ 書くのは面倒なので $|\psi_a\rangle|\psi_b\rangle$ や $|\psi_a\psi_b\rangle$ と書いたりもする.

- Kronecker積

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

複合量子系（状態）

◆ 直積状態

- qubit a, b２つからなる合成系は $|\psi_a\rangle \otimes |\psi_b\rangle$ （テンソル積）で記述する.
- いちいち $\otimes$ 書くのは面倒なので $|\psi_a\rangle|\psi_b\rangle$ や $|\psi_a\psi_b\rangle$ と書いたりもする.
- Kronecker積

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

- qubitが n 個ある場合も同様.
- 特に，同じ状態が n 個あるときは $|\psi\rangle^{\otimes n}$ と書いたりもする.

複合量子系（演算子）

複合量子系（演算子）

◆ 直積状態上の演算子

$$X \otimes I = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$Z \otimes Z = \begin{pmatrix} Z & 0 \\ 0 & -Z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- 演算子が n 個ある場合も同様.
- n 個同じ演算子がある場合は, $A^{\otimes n}$.

複合量子系（演算子）

◆ 直積状態上の演算子

$$X \otimes I = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$Z \otimes Z = \begin{pmatrix} Z & 0 \\ 0 & -Z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- 演算子が n 個ある場合も同様.
- n 個同じ演算子がある場合は，$A^{\otimes n}$.
- i 番目だけに演算子 A が作用する場合,

$$I \otimes I \cdots I \otimes A \otimes I \cdots I = A_i$$

と書く.

n-qubit Pauli products

n-qubit Pauli products

- *n*-qubit Pauli積:

$$\{\pm 1, \pm i\} \times \{I, X, Y, Z\}^{\otimes n} \in \mathcal{P}_n$$

Pauli行列の積はPauli行列なので，Pauli 積は群をなす

→Pauli群

n-qubit Pauli products

◆ *n*-qubit Pauli積:

$$\{\pm 1, \pm i\} \times \{I, X, Y, Z\}^{\otimes n} \in \mathcal{P}_n$$

Pauli行列の積はPauli行列なので，Pauli 積は群をなす

→Pauli群

（例）2-qubit Pauli 群:

$$\left\{ \begin{array}{c} II, IX, IY, IZ, \\ XI, XX, XY, XZ, \\ YI, YX, YY, YZ, \\ ZI, ZX, ZY, ZZ \end{array} \right\} \times \{\pm 1, \pm i\}$$

基本演算（Clifford 演算）

基本演算（Clifford 演算）

◆ Clifford 演算：

共役作用（conjugation）のもとで，Pauli products をPauli products に写すユニタリー演算子

$$\begin{array}{ccc} A & \to UAU^\dagger = & B \\ \cap & & \cap \\ \mathcal{P} & & \mathcal{P} \end{array}$$

基本演算（Clifford 演算）

◆ Clifford 演算：

共役作用（conjugation）のもとで，Pauli products をPauli products に写すユニタリー演算子

$$\begin{array}{ccc} A & \to UAU^\dagger = & B \\ \cap & & \cap \\ \mathcal{P} & & \mathcal{P} \end{array}$$

◆ Hadamard 演算：

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad HXH = Z,\ HZH = X$$

例) $H|0\rangle = |+\rangle,\ H|1\rangle = |-\rangle$ *Z*と*X* の基底の変換でよく使われる.

基本演算（Clifford 演算）

◆ Clifford 演算：

共役作用（conjugation）のもとで，Pauli products をPauli products に写すユニタリー演算子

$$\begin{array}{ccc} A & \rightarrow UAU^{\dagger} = & B \\ \cap & & \cap \\ \mathcal{P} & & \mathcal{P} \end{array}$$

◆ Hadamard 演算：

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad HXH = Z, \ HZH = X$$

例) $H|0\rangle = |+\rangle, \ H|1\rangle = |-\rangle$ *Z*と*X* の基底の変換でよく使われる.

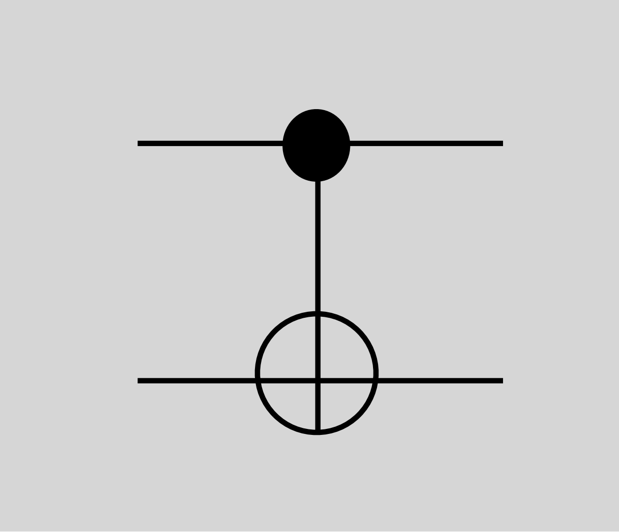
◆ Phase 演算：

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad SXS^{\dagger} = Y, \ SYS^{\dagger} = X, \ SZS = Z$$

基本演算（Clifford 演算）

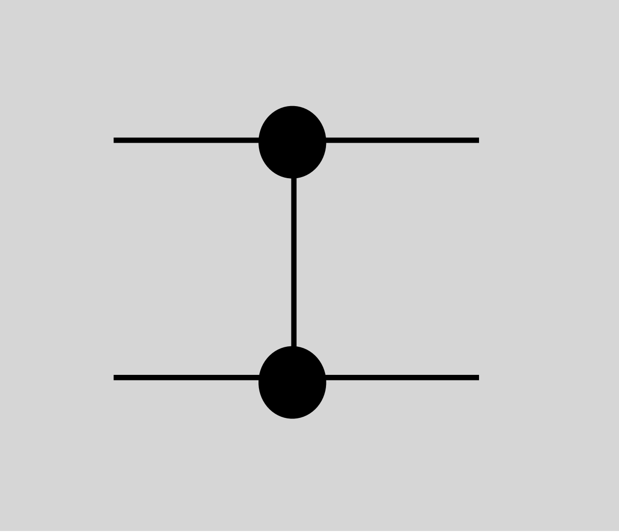
◆ CNOT 演算： $U_{\mathrm{CNOT}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes X_{\mathrm{t}}$

$$\begin{aligned} U_{\mathrm{CNOT}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= X_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\ U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CNOT}} &= I_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\ U_{\mathrm{CNOT}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\ U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes Z_{\mathrm{t}} \end{aligned}$$



◆ CZ 演算： $U_{\mathrm{CZ}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes Z_{\mathrm{t}}$

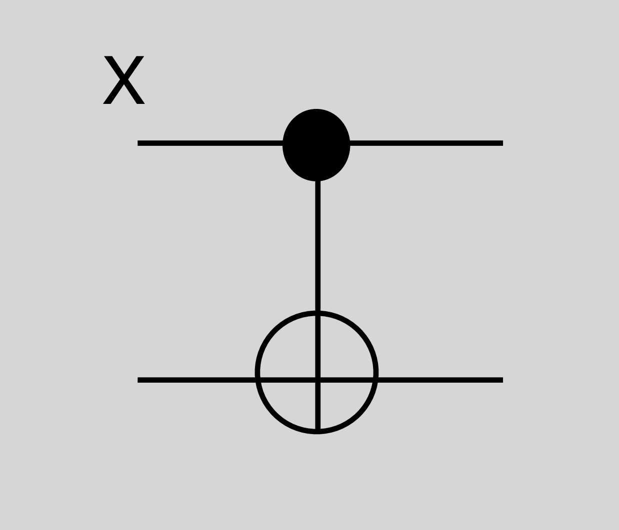
$$\begin{aligned} U_{\mathrm{CZ}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= X_{\mathrm{c}} \otimes Z_{\mathrm{t}}, \\ U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\ U_{\mathrm{CZ}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\ U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CZ}} &= I_{\mathrm{c}} \otimes Z_{\mathrm{t}} \end{aligned}$$



基本演算（Clifford 演算）

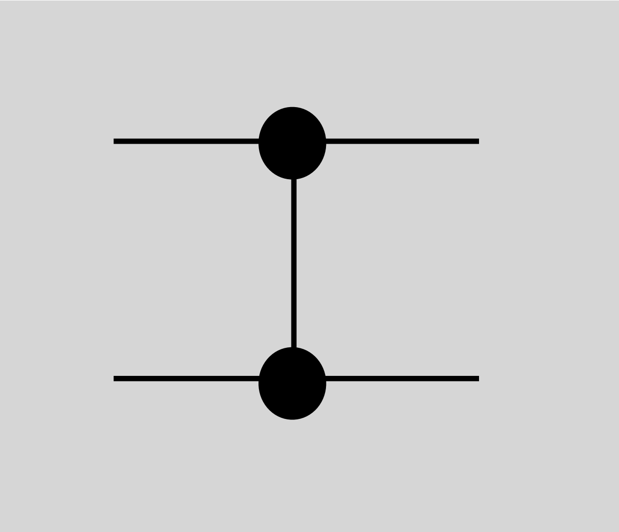
- CNOT 演算： $U_{\mathrm{CNOT}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes X_{\mathrm{t}}$

$$\begin{aligned}
U_{\mathrm{CNOT}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= X_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CNOT}} &= I_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes Z_{\mathrm{t}}
\end{aligned}$$



- CZ 演算： $U_{\mathrm{CZ}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes Z_{\mathrm{t}}$

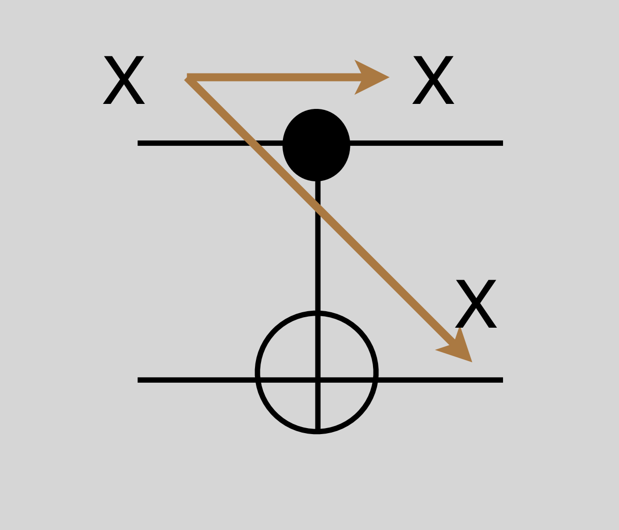
$$\begin{aligned}
U_{\mathrm{CZ}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= X_{\mathrm{c}} \otimes Z_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CZ}} &= I_{\mathrm{c}} \otimes Z_{\mathrm{t}}
\end{aligned}$$



基本演算（Clifford 演算）

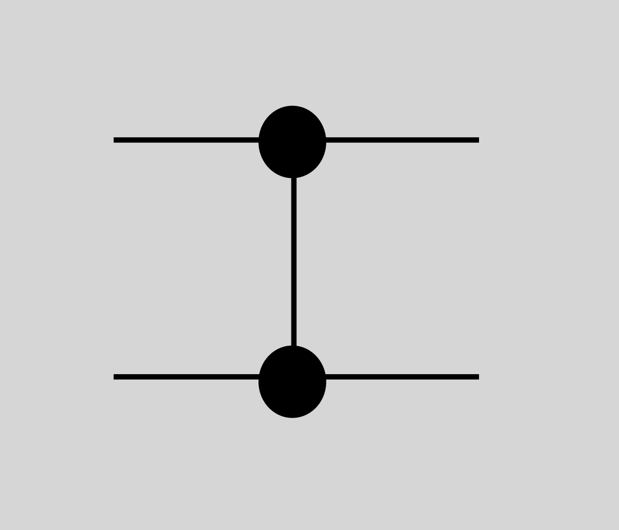
◆ CNOT 演算： $U_{\mathrm{CNOT}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes X_{\mathrm{t}}$

$$\begin{aligned}
U_{\mathrm{CNOT}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= X_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CNOT}} &= I_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes Z_{\mathrm{t}}
\end{aligned}$$



◆ CZ 演算： $U_{\mathrm{CZ}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes Z_{\mathrm{t}}$

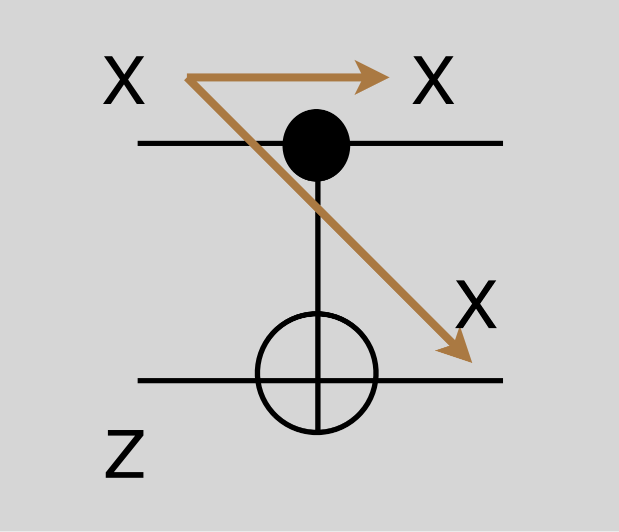
$$\begin{aligned}
U_{\mathrm{CZ}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= X_{\mathrm{c}} \otimes Z_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CZ}} &= I_{\mathrm{c}} \otimes Z_{\mathrm{t}}
\end{aligned}$$



基本演算（Clifford 演算）

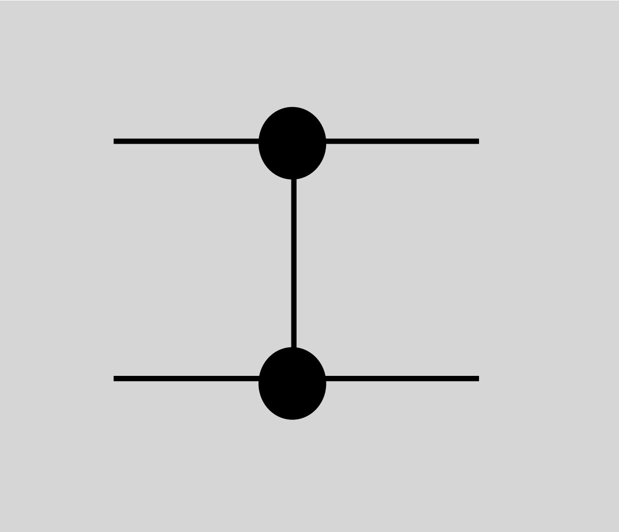
◆ CNOT 演算： $U_{\mathrm{CNOT}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes X_{\mathrm{t}}$

$$\begin{aligned} U_{\mathrm{CNOT}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= X_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\ U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CNOT}} &= I_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\ U_{\mathrm{CNOT}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\ U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes Z_{\mathrm{t}} \end{aligned}$$



◆ CZ 演算： $U_{\mathrm{CZ}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes Z_{\mathrm{t}}$

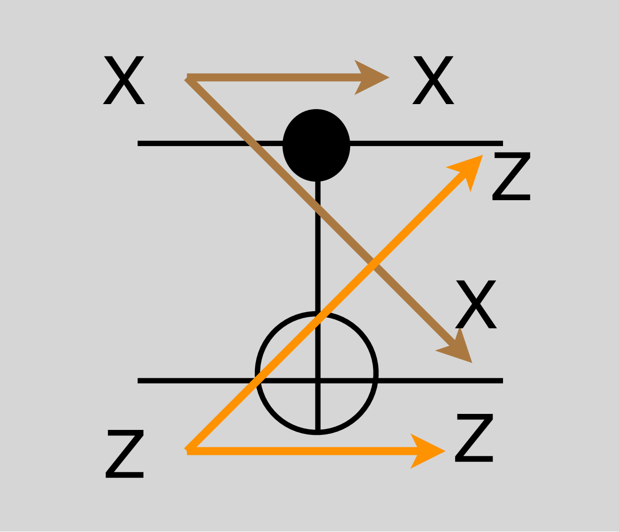
$$\begin{aligned} U_{\mathrm{CZ}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= X_{\mathrm{c}} \otimes Z_{\mathrm{t}}, \\ U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\ U_{\mathrm{CZ}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\ U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CZ}} &= I_{\mathrm{c}} \otimes Z_{\mathrm{t}} \end{aligned}$$



基本演算（Clifford 演算）

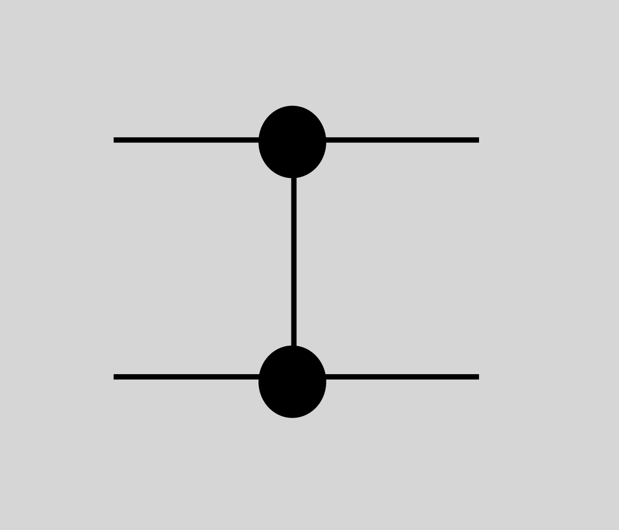
◆ CNOT 演算： $U_{\mathrm{CNOT}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes X_{\mathrm{t}}$

$$\begin{aligned} U_{\mathrm{CNOT}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= X_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\ U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CNOT}} &= I_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\ U_{\mathrm{CNOT}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\ U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes Z_{\mathrm{t}} \end{aligned}$$



◆ CZ 演算： $U_{\mathrm{CZ}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes Z_{\mathrm{t}}$

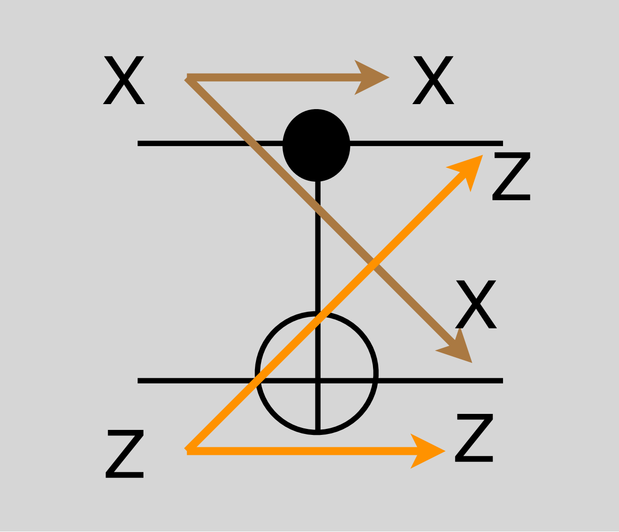
$$\begin{aligned} U_{\mathrm{CZ}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= X_{\mathrm{c}} \otimes Z_{\mathrm{t}}, \\ U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\ U_{\mathrm{CZ}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\ U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CZ}} &= I_{\mathrm{c}} \otimes Z_{\mathrm{t}} \end{aligned}$$



基本演算（Clifford 演算）

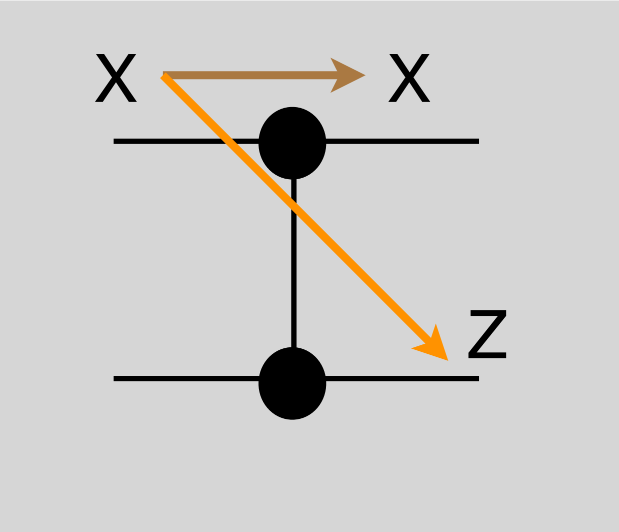
◆ CNOT 演算： $U_{\mathrm{CNOT}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes X_{\mathrm{t}}$

$$
\begin{aligned}
U_{\mathrm{CNOT}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= X_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CNOT}} &= I_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes Z_{\mathrm{t}}
\end{aligned}
$$



◆ CZ 演算： $U_{\mathrm{CZ}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes Z_{\mathrm{t}}$

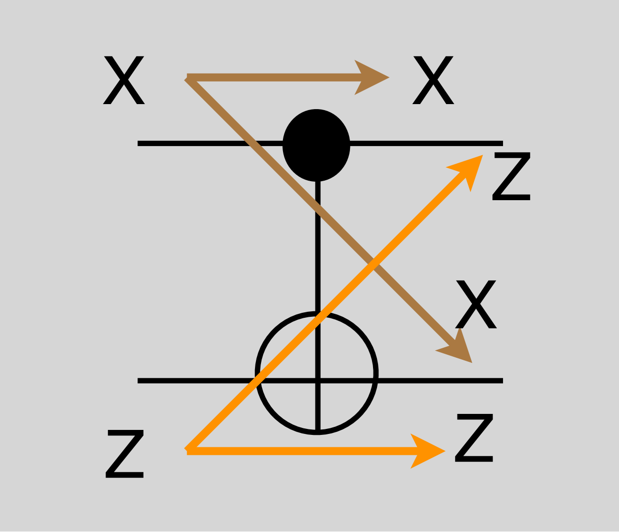
$$
\begin{aligned}
U_{\mathrm{CZ}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= X_{\mathrm{c}} \otimes Z_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CZ}} &= I_{\mathrm{c}} \otimes Z_{\mathrm{t}}
\end{aligned}
$$



基本演算（Clifford 演算）

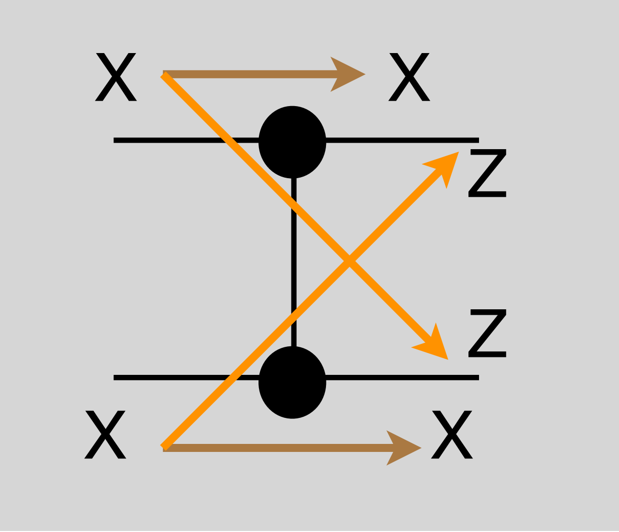
◆ CNOT 演算： $U_{\mathrm{CNOT}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes X_{\mathrm{t}}$

$$\begin{aligned}
U_{\mathrm{CNOT}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= X_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CNOT}} &= I_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes Z_{\mathrm{t}}
\end{aligned}$$



◆ CZ 演算： $U_{\mathrm{CZ}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes Z_{\mathrm{t}}$

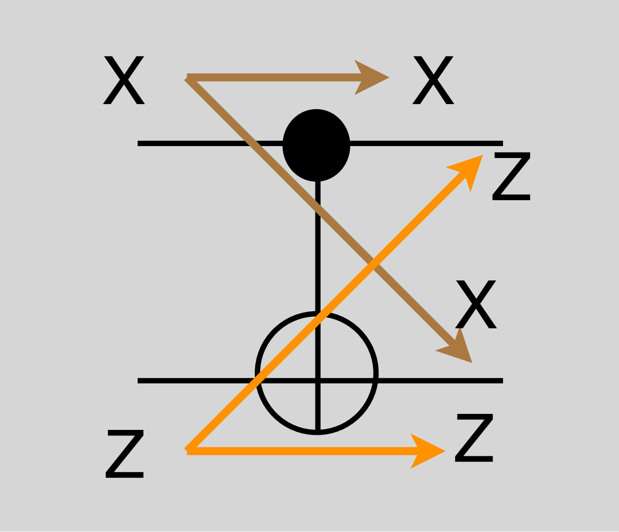
$$\begin{aligned}
U_{\mathrm{CZ}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= X_{\mathrm{c}} \otimes Z_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CZ}} &= I_{\mathrm{c}} \otimes Z_{\mathrm{t}}
\end{aligned}$$



基本演算（Clifford 演算）

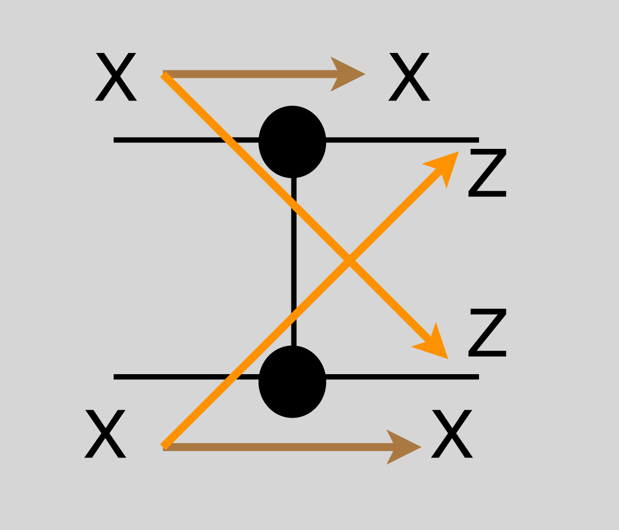
◆ CNOT 演算： $U_{\mathrm{CNOT}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes X_{\mathrm{t}}$

$$\begin{aligned}
U_{\mathrm{CNOT}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= X_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CNOT}} &= I_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\
U_{\mathrm{CNOT}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CNOT}} &= Z_{\mathrm{c}} \otimes Z_{\mathrm{t}}
\end{aligned}$$



◆ CZ 演算： $U_{\mathrm{CZ}} = |0\rangle\langle 0|_{\mathrm{c}} \otimes I_{\mathrm{t}} + |1\rangle\langle 1|_{\mathrm{c}} \otimes Z_{\mathrm{t}}$

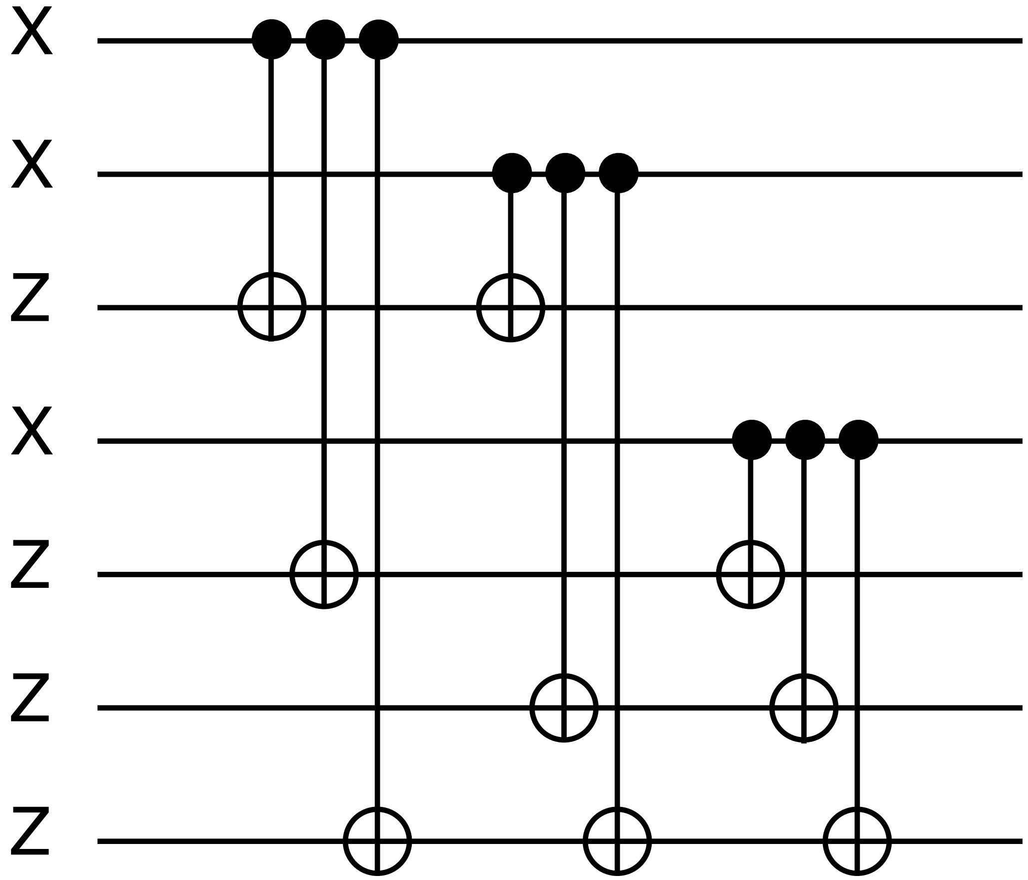
$$\begin{aligned}
U_{\mathrm{CZ}}(X_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= X_{\mathrm{c}} \otimes Z_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes X_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes X_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(Z_{\mathrm{c}} \otimes I_{\mathrm{t}})U_{\mathrm{CZ}} &= Z_{\mathrm{c}} \otimes I_{\mathrm{t}}, \\
U_{\mathrm{CZ}}(I_{\mathrm{c}} \otimes Z_{\mathrm{t}})U_{\mathrm{CZ}} &= I_{\mathrm{c}} \otimes Z_{\mathrm{t}}
\end{aligned}$$

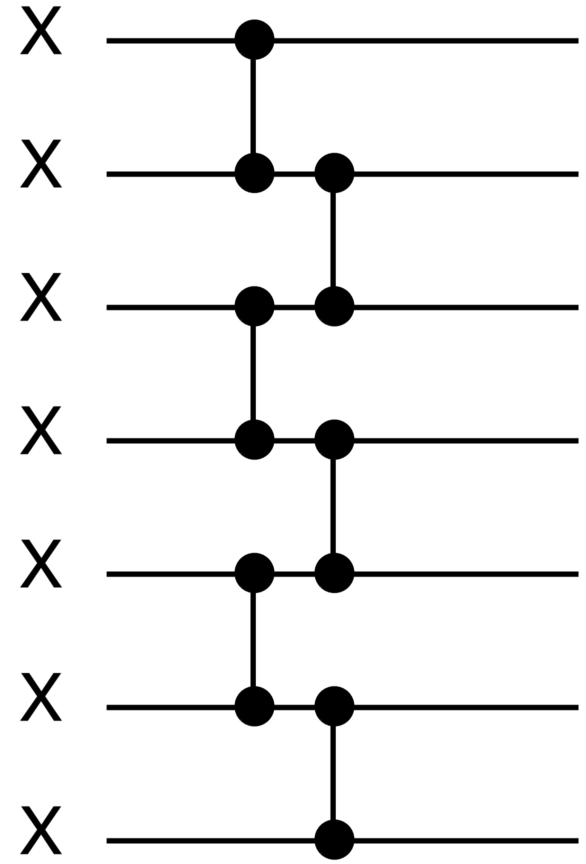


◆ 覚え方：黒丸は Z と可換，白丸は X と可換.

基本演算（Clifford 演算）

練習）以下の回路の出力側の演算子を計算せよ





Stabilizer群

- Stabilizer群：Pauli群の可換部分群

$$\mathcal{S} = \{S_i\}, S_i \in \mathcal{P}, [S_i, S_j] = 0 \text{ for all } S_i, S_j \in \mathcal{S}$$

例1) $\{IZ, XI, II, XZ\} \longleftarrow$ no overlap

例2) $\{II, XX, ZZ, -YY\} \longleftarrow$ even overlap **反可換×２=可換!**

Stabilizer群

◆ Stabilizer群：Pauli群の可換部分群

$$\mathcal{S} = \{S_i\}, S_i \in \mathcal{P}, [S_i, S_j] = 0 \text{ for all } S_i, S_j \in \mathcal{S}$$

例1) $\{IZ, XI, II, XZ\} \longleftarrow$ no overlap

例2) $\{II, XX, ZZ, -YY\} \longleftarrow$ even overlap **反可換×２=可換!**

◆ Stabilizer生成元：stabilizer群の独立な元の最大集合

（お互い他の元の積で表すことができない．）

Stabilizer群

◆ Stabilizer群：Pauli群の可換部分群

$$\mathcal{S}=\{S_i\}, S_i\in\mathcal{P}, [S_i,S_j]=0 \text{ for all } S_i,S_j\in\mathcal{S}$$

例1) $\{IZ, XI, II, XZ\} \longleftarrow$ no overlap

例2) $\{II, XX, ZZ, -YY\} \longleftarrow$ even overlap **反可換×２=可換!**

◆ Stabilizer生成元：stabilizer群の独立な元の最大集合

（お互い他の元の積で表すことができない．）

例1) $\{IZ, XI\}$ $\cancel{\{IZ, XI, XZ\}}$ 独立ではない.

例2) $\{XX, ZZ\}$

Stabilizer群

◆ Stabilizer群：Pauli群の可換部分群

$$\mathcal{S} = \{S_i\}, S_i \in \mathcal{P}, [S_i, S_j] = 0 \text{ for all } S_i, S_j \in \mathcal{S}$$

例1) $\{IZ, XI, II, XZ\} \longleftarrow$ no overlap

例2) $\{II, XX, ZZ, -YY\} \longleftarrow$ even overlap **反可換×２=可換!**

◆ Stabilizer生成元：stabilizer群の独立な元の最大集合

（お互い他の元の積で表すことができない．）

例1) $\{IZ, XI\}$ $\{\cancel{IZ, XI, XZ}\}$ 独立ではない.

例2) $\{XX, ZZ\}$

生成元 $\{\bar{S}_i\}$ から生成される Stabilizer groupを $\langle \bar{S}_i \rangle$ と書くことにする.

例3) $\langle XX, ZZ \rangle = \{II, XX, ZZ, -YY\}$

Stabilizer状態

Stabilizer状態

- Stabilizer状態：すべてのstabilizer演算子$S_i \in \mathcal{S}$に対して

$$S_i|\Psi\rangle = |\Psi\rangle$$

を満たす状態$|\Psi\rangle$.

- stabilizer群は可換群なので，同時対角化できる.
- stabilizer生成元の固有状態であれば十分.

Stabilizer状態

◆ Stabilizer状態：すべてのstabilizer演算子$S_i \in \mathcal{S}$に対して

$$S_i|\Psi\rangle = |\Psi\rangle$$

を満たす状態$|\Psi\rangle$.

- stabilizer群は可換群なので，同時対角化できる.
- stabilizer生成元の固有状態であれば十分.

例1) $\mathcal{S}_1 = \langle XX, ZZ \rangle$

例1.5) $\mathcal{S}_2 = \langle ZZ \rangle$

例2) $\mathcal{S}_3 = \langle ZZI, IZZ, XXX \rangle$

例3) $\mathcal{S}_4 = \langle XIXIXIX, IXXIIXX, IIIXXXX,$
$ZIZIZIZ, IZZIIZZ, IIIZZZZ, \rangle$

Stabilizer状態

◆ Stabilizer状態：すべてのstabilizer演算子$S_i \in \mathcal{S}$に対して

$$S_i|\Psi\rangle = |\Psi\rangle$$

を満たす状態$|\Psi\rangle$.

- stabilizer群は可換群なので，同時対角化できる.
- stabilizer生成元の固有状態であれば十分.

例1) $\mathcal{S}_1 = \langle XX, ZZ \rangle$

Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$

例1.5) $\mathcal{S}_2 = \langle ZZ \rangle$

例2) $\mathcal{S}_3 = \langle ZZI, IZZ, XXX \rangle$

例3) $\mathcal{S}_4 = \langle XIXIXIX, IXXIIXX, IIIXXXX,$
$ZIZIZIZ, IZZIIZZ, IIIZZZZ, \rangle$

Stabilizer状態

◆ Stabilizer状態：すべてのstabilizer演算子$S_i \in \mathcal{S}$に対して

$$S_i|\Psi\rangle = |\Psi\rangle$$

を満たす状態$|\Psi\rangle$.

- stabilizer群は可換群なので，同時対角化できる.
- stabilizer生成元の固有状態であれば十分.

例1) $\mathcal{S}_1 = \langle XX, ZZ \rangle$

Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$

例1.5) $\mathcal{S}_2 = \langle ZZ \rangle$

例2) $\mathcal{S}_3 = \langle ZZI, IZZ, XXX \rangle$

GHZ state $(|000\rangle + |111\rangle)/\sqrt{2}$

例3) $\mathcal{S}_4 = \langle XIXIXIX, IXXIIXX, IIIXXXX,$
$ZIZIZIZ, IZZIIZZ, IIIZZZZ, \rangle$

Stabilizer状態

◆ Stabilizer状態：すべてのstabilizer演算子$S_i \in \mathcal{S}$に対して

$$S_i|\Psi\rangle = |\Psi\rangle$$

を満たす状態$|\Psi\rangle$.

- stabilizer群は可換群なので，同時対角化できる.
- stabilizer生成元の固有状態であれば十分.

例1) $\mathcal{S}_1 = \langle XX, ZZ \rangle$

Bell state　$(|00\rangle + |11\rangle)/\sqrt{2}$

例2) $\mathcal{S}_3 = \langle ZZI, IZZ, XXX \rangle$

GHZ state　$(|000\rangle + |111\rangle)/\sqrt{2}$

例3) $\mathcal{S}_4 = \langle XIXIXIX, IXXIIXX, IIIXXXX,$
$ZIZIZIZ, IZZIIZZ, IIIZZZZ, \rangle$

例1.5) $\mathcal{S}_2 = \langle ZZ \rangle$

$\{|00\rangle, |11\rangle\}$ で張られる部分空間内の任意の状態.

Stabilizer状態

◆ Stabilizer状態：すべてのstabilizer演算子$S_i \in \mathcal{S}$に対して

$$S_i|\Psi\rangle = |\Psi\rangle$$

を満たす状態$|\Psi\rangle$.

- stabilizer群は可換群なので，同時対角化できる.
- stabilizer生成元の固有状態であれば十分.

例1) $\mathcal{S}_1 = \langle XX, ZZ \rangle$

Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$

例2) $\mathcal{S}_3 = \langle ZZI, IZZ, XXX \rangle$

GHZ state $(|000\rangle + |111\rangle)/\sqrt{2}$

例3) $\mathcal{S}_4 = \langle XIXIXIX, IXXIIXX, IIIXXXX,$
$ZIZIZIZ, IZZIIZZ, IIIZZZZ, \rangle$

こんなの難しすぎて計算できない.

例1.5) $\mathcal{S}_2 = \langle ZZ \rangle$

$\{|00\rangle, |11\rangle\}$ で張られる部分空間内の任意の状態.

Stabilizer状態

- 状態(一次元空間)が定義されるための条件：

Stabilizer状態

◆ 状態(一次元空間)が定義されるための条件：

→ stabilizer生成元の数とqubitの数が一致する場合.

(理由)それぞれの生成元に対して +1, -1の固有空間へと分割されていくので，固有状態の数は 2^(#生成元). 一方，n qubitの次元は2^n .

ZZ \ XX	+1	-1
+1	$(\|00\rangle + \|11\rangle)/\sqrt{2}$	$(\|00\rangle - \|11\rangle)/\sqrt{2}$
-1	$(\|01\rangle + \|10\rangle)/\sqrt{2}$	$(\|10\rangle - \|01\rangle)/\sqrt{2}$

Stabilizer状態

◆ 状態(一次元空間)が定義されるための条件：

→ stabilizer生成元の数とqubitの数が一致する場合.

(理由)それぞれの生成元に対して +1, -1の固有空間へと分割されていくので，固有状態の数は 2^(#生成元). 一方，n qubitの次元は2^n .

(物理屋向け）生成元を物理量（エネルギー，角運動量など），固有値の値をその値（量子数）だと思うと，状態を特定するためには，全ての量子数を指定しなければ成らない．そして，stabilizer状態は量子数がすべて+1の状態.

→ $|E=+1\rangle|l=+1\rangle$

ZZ \ XX	+1	-1
+1	$(\|00\rangle+\|11\rangle)/\sqrt{2}$	$(\|00\rangle-\|11\rangle)/\sqrt{2}$
-1	$(\|01\rangle+\|10\rangle)/\sqrt{2}$	$(\|10\rangle-\|01\rangle)/\sqrt{2}$

射影演算子

◆ 射影演算子：

固有値±1のエルミート演算子 A の固有空間への射影演算子は

$$P_{\pm} = \frac{I \pm A}{2}$$

射影演算子

◆ 射影演算子：

固有値±1のエルミート演算子 *A* の固有空間への射影演算子は

$$P_\pm = \frac{I \pm A}{2}$$

証明) $|a_\pm\rangle$ をそれぞれ±1の固有状態として，演算子 *A* は

$$A = (+1)|a_+\rangle\langle a_+| + (-1)|a_-\rangle\langle a_-|$$

と書け，一方, $I = |a_+\rangle\langle a_+| + |a_-\rangle\langle a_-|$なので，$P_\pm = |a_\pm\rangle\langle a_\pm|$ ▪

射影演算子

◆ 射影演算子：

固有値±1のエルミート演算子 *A* の固有空間への射影演算子は

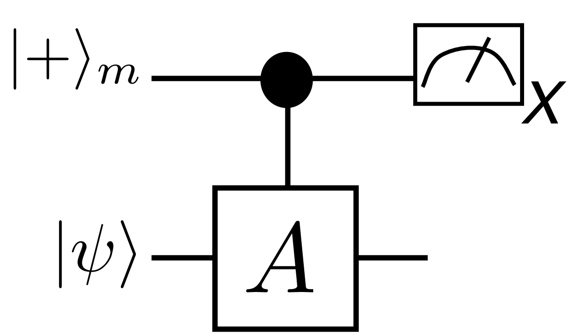
$$P_{\pm} = \frac{I \pm A}{2}$$

証明) $|a_{\pm}\rangle$ をそれぞれ±1の固有状態として，演算子 *A* は

$$A = (+1)|a_+\rangle\langle a_+| + (-1)|a_-\rangle\langle a_-|$$

と書け，一方, $I = |a_+\rangle\langle a_+| + |a_-\rangle\langle a_-|$なので，$P_{\pm} = |a_{\pm}\rangle\langle a_{\pm}|$ ▪

◆ 回路による実装：



$$|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes A$$

$$(|0\rangle_m|\psi\rangle + |1\rangle_m A|\psi\rangle)/\sqrt{2}$$

$$\Big\downarrow\ |+\rangle\langle +|$$

$$|+\rangle \underbrace{\left(\frac{I+A}{2}\right)}_{\text{Aの+1の固有状態への射影演算子}} |\psi\rangle$$

Stabilizer状態の書き下し

Stabilizer状態の書き下し

◆ stabilizer状態の書き下し方

（生成元が*X* だけの積S_i^X，*Z* だけの積 S_j^Z に分解できるとき）：

$$|\Psi\rangle = \frac{1}{\sqrt{2^{|S^X|}}} \prod_i \frac{I + S_i^X}{2} |0\rangle^{\otimes n}$$

Stabilizer状態の書き下し

◆ stabilizer状態の書き下し方

（生成元がXだけの積S_i^X，Zだけの積 S_j^Z に分解できるとき）：

$$|\Psi\rangle = \boxed{\frac{1}{\sqrt{2^{|S^X|}}}} \prod_i \frac{I + S_i^X}{2} |0\rangle^{\otimes n}$$

規格化因子

Stabilizer状態の書き下し

◆ stabilizer状態の書き下し方

（生成元がXだけの積S_i^X，Zだけの積 S_j^Z に分解できるとき）：

$$|\Psi\rangle = \boxed{\frac{1}{\sqrt{2^{|S^X|}}}} \prod_i \frac{I + S_i^X}{2} \boxed{|0\rangle^{\otimes n}}$$

規格化因子

自動的にS_j^Zの固有状態になる.

Stabilizer状態の書き下し

◆ stabilizer状態の書き下し方

（生成元がX だけの積S_i^X，Z だけの積 S_j^Z に分解できるとき）：

$$|\Psi\rangle = \underbrace{\boxed{\frac{1}{\sqrt{2^{|S^X|}}}}}_{\text{規格化因子}} \prod_i \frac{I + S_i^X}{2} \boxed{|0\rangle^{\otimes n}}$$

自動的に S_j^Z の固有状態になる.

特に，$S_i^X = \prod_k X_k^{s_k^{(i)}}$ として，$\underbrace{\text{ビット列 } s_k^{(i)}}_{\text{バイナリ表示}}$ を定義すると

$$|\Psi\rangle = \frac{1}{\sqrt{|C|}} \sum_{c_i \in C} |c_i\rangle$$

Stabilizer状態の書き下し

◆ stabilizer状態の書き下し方

（生成元がX だけの積S_i^X，Z だけの積 S_j^Z に分解できるとき）：

$$|\Psi\rangle = \boxed{\frac{1}{\sqrt{2^{|S^X|}}}} \prod_i \frac{I+S_i^X}{2} \boxed{|0\rangle^{\otimes n}}$$

規格化因子

自動的にS_j^Zの固有状態になる.

特に, $S_i^X = \prod_k X_k^{s_k^{(i)}}$として，$\underbrace{\text{ビット列 } s_k^{(i)}}_{\text{バイナリ表示}}$を定義すると

$$|\Psi\rangle = \frac{1}{\sqrt{|C|}} \sum_{c_i \in \boxed{C}} |c_i\rangle$$

GF(2)上の基底{ $s^{(i)} = (s_1^{(i)}, s_2^{(i)}, \ldots, s_n^{(i)})$ }によって張られる線形空間（古典線形符号）

Stabilizer状態の書き下し

◆ stabilizer状態の書き下し方

（生成元がX だけの積S_i^X，Z だけの積 S_j^Z に分解できるとき）：

$$|\Psi\rangle = \boxed{\frac{1}{\sqrt{2^{|S^X|}}}} \prod_i \frac{I + S_i^X}{2} \boxed{|0\rangle^{\otimes n}}$$

規格化因子

自動的にS_j^Zの固有状態になる.

特に，$S_i^X = \prod_k X_k^{s_k^{(i)}}$として，$\underbrace{\text{ビット列 } s_k^{(i)}}_{\text{バイナリ表示}}$ を定義すると

$$|\Psi\rangle = \frac{1}{\sqrt{|C|}} \sum_{c_i \in \boxed{C}} |c_i\rangle$$

GF(2)上の基底{ $s^{(i)} = (s_1^{(i)}, s_2^{(i)}, \ldots, s_n^{(i)})$ }によって張られる線形空間（古典線形符号）

上記の分解ができないときも同じ方針で作れる！

Stabilizer形式におけるユニタリ演算

◆ Clifford演算子の作用：

Clifford演算子はPauli積をPauli積に写すユニタリ演算子.

→ あるstabilizer状態から異なるstabilizer状態へ写す.

Stabilizer形式におけるユニタリ演算

◆ Clifford演算子の作用：

Clifford演算子はPauli積をPauli積に写すユニタリ演算子.

→ あるstabilizer状態から異なるstabilizer状態へ写す.

$$S_i|\Psi\rangle = |\Psi\rangle$$ ......（定義）

$$\Big\downarrow \; U\text{を作用}$$

$$US_iU^\dagger \left(U|\Psi\rangle\right) = \left(U|\Psi\rangle\right)$$

Stabilizer形式におけるユニタリ演算

◆ Clifford演算子の作用：

Clifford演算子はPauli積をPauli積に写すユニタリ演算子.

→ あるstabilizer状態から異なるstabilizer状態へ写す.

$$S_i|\Psi\rangle = |\Psi\rangle \quad \text{......（定義）}$$

$$\Big\downarrow U\text{を作用}$$

$$\boxed{US_iU^\dagger}(U|\Psi\rangle) = (U|\Psi\rangle)$$

新しい状態をstabilizeする演算子

Stabilizer形式におけるユニタリ演算

◆ Clifford演算子の作用：

Clifford演算子はPauli積をPauli積に写すユニタリ演算子.

→ あるstabilizer状態から異なるstabilizer状態へ写す.

$$S_i|\Psi\rangle = |\Psi\rangle$$ ……（定義）

$$\Big\downarrow \; U\text{を作用}$$

$$\boxed{US_iU^\dagger}(U|\Psi\rangle) = (U|\Psi\rangle)$$

新しい状態をstabilizeする演算子

Clifford演算子Uの作用は，stabilizer演算子への作用

$$S_i \to US_iU^\dagger$$

によって記述される.

Stabilizer形式におけるユニタリ演算

◆ Clifford演算子の作用（例）

Stabilizer形式におけるユニタリ演算

◆ Clifford演算子の作用（例）

- Hadamard演算:

$$H|+\rangle = |0\rangle \qquad X \to Z$$

Stabilizer形式におけるユニタリ演算

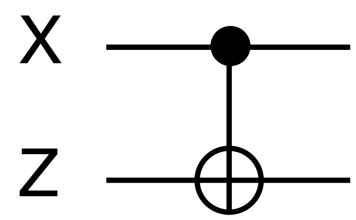
◆ Clifford演算子の作用（例）

- Hadamard演算:

$$H|+\rangle = |0\rangle \qquad X \rightarrow Z$$

- CNOT演算:

$$U_{\mathrm{CNOT}}|+\rangle_c|0\rangle_t = (|00\rangle + |11\rangle)/\sqrt{2} \qquad \begin{cases} XI \rightarrow XX \\ IZ \rightarrow ZZ \end{cases}$$



Stabilizer形式におけるユニタリ演算

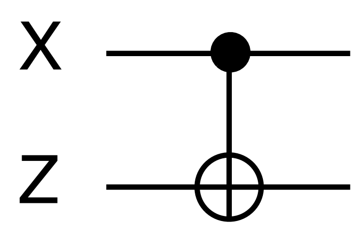
◆ Clifford演算子の作用（例）

- Hadamard演算:

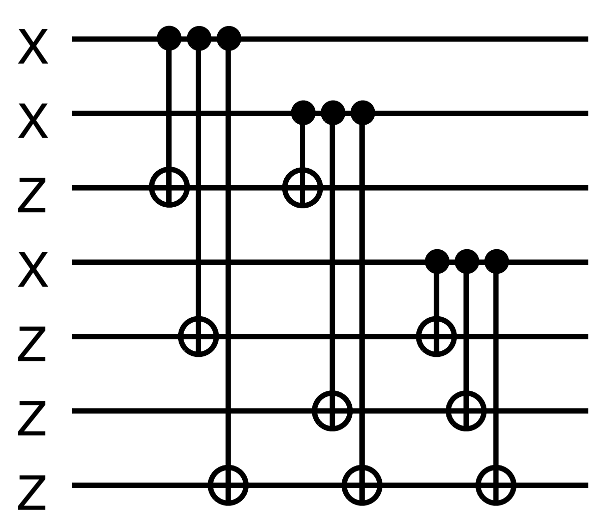
$$H|+\rangle = |0\rangle \qquad X \rightarrow Z$$

- CNOT演算:

$$U_{\mathrm{CNOT}}|+\rangle_c|0\rangle_t = (|00\rangle + |11\rangle)/\sqrt{2} \qquad \begin{cases} XI \rightarrow XX \\ IZ \rightarrow ZZ \end{cases}$$



- たくさんのCNOT演算:



Stabilizer形式におけるPauli測定

◆ Pauli積 A に関する測定：

場合１）A がスタビライザー群に含まれる.

→ 測定結果は +1 が得られ，測定後の状態は変化なし.

Stabilizer形式におけるPauli測定

- Pauli積 A に関する測定：

場合１）A がスタビライザー群に含まれる.

→ 測定結果は +1 が得られ，測定後の状態は変化なし.

場合２）A がスタビライザー群に含まれない.

→ 生成元を A と可換なものと非可換なものに分類できる.

$$\langle \underbrace{S_1, S_2, \ldots, S_k,}_{\text{非可換}} \underbrace{S_{k+1}, \ldots S_n}_{\text{可換}} \rangle$$

Stabilizer形式におけるPauli測定

◆ Pauli積 A に関する測定：

場合１）A がスタビライザー群に含まれる.

→ 測定結果は +1 が得られ，測定後の状態は変化なし.

場合２）A がスタビライザー群に含まれない.

→ 生成元を A と可換なものと非可換なものに分類できる.

$$\langle \underbrace{S_1, S_2, \ldots, S_k,}_{\text{非可換}} \underbrace{S_{k+1}, \ldots S_n}_{\text{可換}} \rangle$$

非可換なものから適当に一つ取り除き，±A を追加，可換になるように生成元を再構成する.

$$\langle \pm A, S_1 S_2, \ldots, S_1 S_k, S_{k+1}, \ldots S_n \rangle$$ ←測定後の状態の生成元

Stabilizer形式におけるPauli測定

◆ Pauli積 A に関する測定：

場合１）A がスタビライザー群に含まれる.

→ 測定結果は +1 が得られ，測定後の状態は変化なし.

場合２）A がスタビライザー群に含まれない.

→ 生成元を A と可換なものと非可換なものに分類できる.

$$\langle \underbrace{S_1, S_2, \ldots, S_k,}_{\text{非可換}} \underbrace{S_{k+1}, \ldots S_n}_{\text{可換}} \rangle$$

非可換なものから適当に一つ取り除き，$\pm A$ を追加，可換になるように生成元を再構成する.

$$\langle \underbrace{\boxed{\pm}}_{\text{測定結果}} A, S_1 S_2, \ldots, S_1 S_k, S_{k+1}, \ldots S_n \rangle$$ ←測定後の状態の生成元

Stabilizer形式におけるPauli測定

◆ Pauli積 *A* に関する測定：

場合１）*A* がスタビライザー群に含まれる.

→ 測定結果は +1 が得られ，測定後の状態は変化なし.

場合２）*A* がスタビライザー群に含まれない.

→ 生成元を *A* と可換なものと非可換なものに分類できる.

$$\langle \underbrace{S_1, S_2, \ldots, S_k,}_{\text{非可換}} \underbrace{S_{k+1}, \ldots S_n}_{\text{可換}} \rangle$$

非可換なものから適当に一つ取り除き，±*A* を追加，可換になるように生成元を再構成する.

$$\langle \underbrace{\boxed{\pm}}_{\text{測定結果}} A, \boxed{S_1 S_2}, \ldots, \boxed{S_1 S_k}, S_{k+1}, \ldots S_n \rangle$$ ←測定後の状態の生成元

測定結果　非可換×２=可換

Stabilizer形式におけるPauli測定

◆ Pauli積 *A* に関する測定：

場合１）*A* がスタビライザー群に含まれる.

→ 測定結果は +1 が得られ，測定後の状態は変化なし.

場合２）*A* がスタビライザー群に含まれない.

→ 生成元を *A* と可換なものと非可換なものに分類できる.

$$\langle \underbrace{S_1, S_2, \ldots, S_k,}_{\text{非可換}} \underbrace{S_{k+1}, \ldots S_n}_{\text{可換}} \rangle$$

非可換なものから適当に一つ取り除き，±*A* を追加，可換になるように生成元を再構成する.

$$\langle \underbrace{\boxed{\pm}}_{\text{測定結果}} A, \boxed{S_1 S_2}, \ldots, \boxed{S_1 S_k}, S_{k+1}, \ldots S_n \rangle \leftarrow \text{測定後の状態の生成元}$$

測定結果　非可換×２=可換

(具体例は後ほど，graph stateのところでやります)

Gottesman-Knillの定理

◆Gottesman-Knillの定理

入力状態がPauli基底の状態で，ユニタリー演算が全てClifford演算であり，かつ測定はPauli基底でしか行えない場合，古典コンピュータで効率よくシミュレートできる.

Gottesman-Knillの定理

◆Gottesman-Knillの定理

入力状態がPauli基底の状態で，ユニタリー演算が全てClifford演算であり，かつ測定はPauli基底でしか行えない場合，古典コンピュータで効率よくシミュレートできる．

具体的手法：

n-qubit stabilizer状態の生成元 S_i から2*n*-bitのバイナリ $s^{(i)}$ を

$$S_i = \prod_k X_k^{s_k^{(i)}} Z_k^{s_{n+k}^{(i)}}$$

のように定義する．例えば $IXXZI \to 01100|00010$ ． Clifford演算はバイナリに対するXOR演算で記述される（classical universal ですらない）．

Gottesman-Knillの定理

◆Gottesman-Knillの定理

> 入力状態がPauli基底の状態で，ユニタリー演算が全てClifford演算であり，かつ測定はPauli基底でしか行えない場合，古典コンピュータで効率よくシミュレートできる.

具体的手法：
n-qubit stabilizer状態の生成元 S_i から2*n*-bitのバイナリ $s^{(i)}$ を

$$S_i = \prod_k X_k^{s_k^{(i)}} Z_k^{s_{n+k}^{(i)}}$$

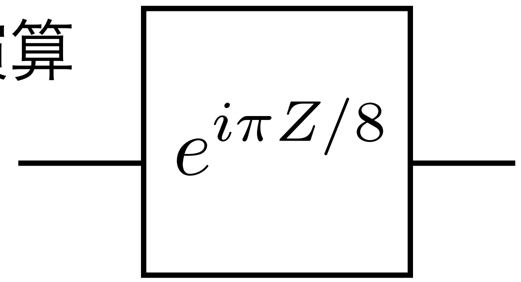
のように定義する．例えば $IXXZI \to 01100|00010$ ． Clifford演算はバイナリに対するXOR演算で記述される（classical universal ですらない）．

→Clifford演算子だけでは universalではなく，Clifford以外の演算も必要.

Kitaev-Solvayの定理

◆ Non-Clifford演算が一つでもあれば universal

- 例えば π/8 演算



（Clifford演算子ではないことを確認）

Kitaev-Solvayの定理

◆ Non-Clifford演算が一つでもあれば universal

•例えば π/8 演算

$$e^{i\pi Z/8}$$

（Clifford演算子ではないことを確認）

◆ Kitaev-Solovayの定理

ある基本演算（gate set）がSU(2)で稠密であれば,
すぐに（=polylog(1/ε)）SU(2)を覆い尽くす.

$$^\forall U \in SU(2) \text{ and } ^\forall \epsilon, \quad ^\exists S = g_1 \dots g_n, \quad \text{s.t. } d(S, U) < \epsilon$$

Kitaev-Solvayの定理

◆ Non-Clifford演算が一つでもあれば universal

•例えば π/8 演算

$$e^{i\pi Z/8}$$

（Clifford演算子ではないことを確認）

◆ Kitaev-Solovayの定理

ある基本演算（gate set）がSU(2)で稠密であれば,
すぐに（=polylog(1/ε)）SU(2)を覆い尽くす.

$$^{\forall}U \in SU(2) \text{ and } ^{\forall}\epsilon, \quad ^{\exists}S = g_1 \ldots g_n, \quad \text{s.t. } d(S,U) < \epsilon$$

•1-qubitの任意の回転は Hadamard演算とπ/8 演算で効率よく構成できる.

•1-qubitの任意の回転とCNOTがあれば任意のユニタリ演算子が構成できる.

Kitaev-Solvayの定理

Solovay-Kitaev(U,n)

if n=0, return basic approximation of U

else $U_{n\text{-}1}$=Solovay-Kitaev(U,n-1)

V,W s.t. $VWV^{\dagger}W^{\dagger}=UU^{\dagger}{}_{n\text{-}1}$

$V_{n\text{-}1}$ = Solovay-Kitaev(V,n-1)

$W_{n\text{-}1}$ = Solovay-Kitaev(V,n-1)

Return $V_{n\text{-}1}W_{n\text{-}1}V_{n\text{-}1}{}^{\dagger}W_{n\text{-}1}{}^{\dagger}U_{n\text{-}1}$

➡ $O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理

)

c approximation of U

y-Kitaev(U,n-1)

$V^{\dagger}W^{\dagger}=UU^{\dagger}{}_{n\text{-}1}$

ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n\text{-}1}{}^{\dagger}W_{n\text{-}1}{}^{\dagger}U_{n\text{-}1}$

$\Rightarrow O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理

)

c approximation of U

y-Kitaev(U,n-1)

$V^\dagger W^\dagger = UU^\dagger_{n\text{-}1}$

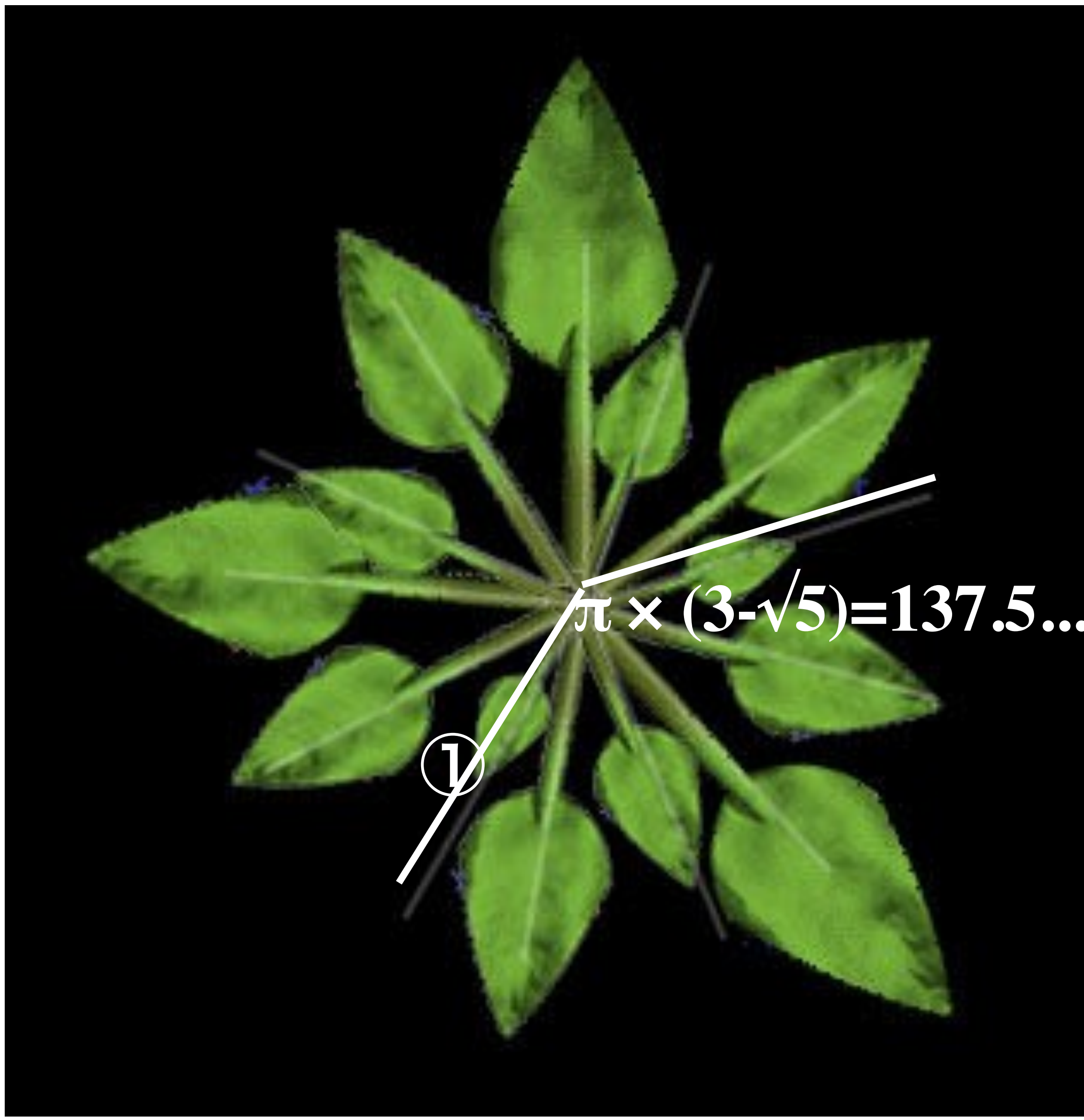
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n\text{-}1}{}^\dagger W_{n\text{-}1}{}^\dagger U_{n\text{-}1}$

$\Rightarrow O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

$V^\dagger W^\dagger = UU^\dagger_{n-1}$

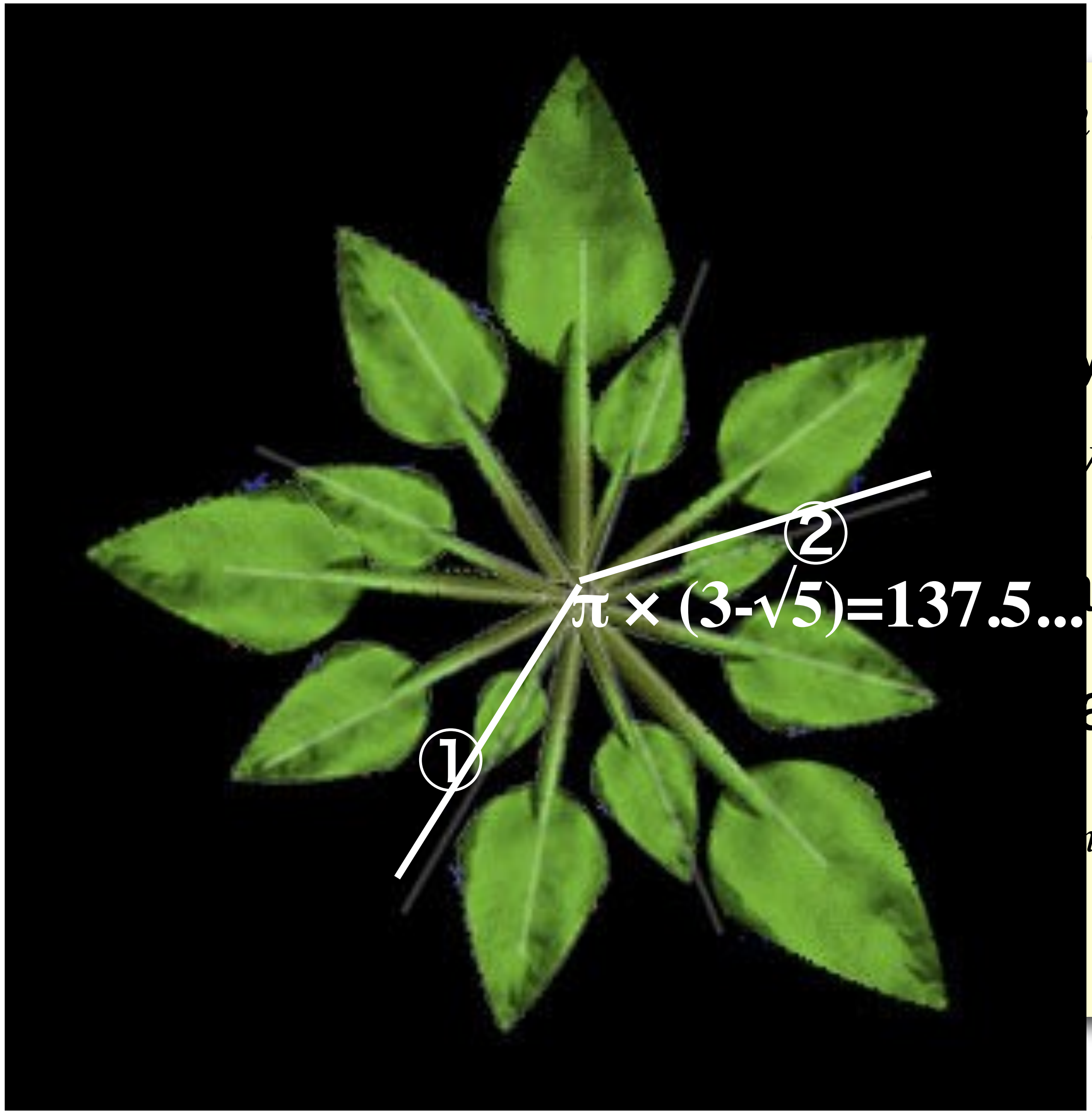
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n-1}{}^\dagger W_{n-1}{}^\dagger U_{n-1}$

→ $O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

$V^\dagger W^\dagger = UU^\dagger_{n-1}$

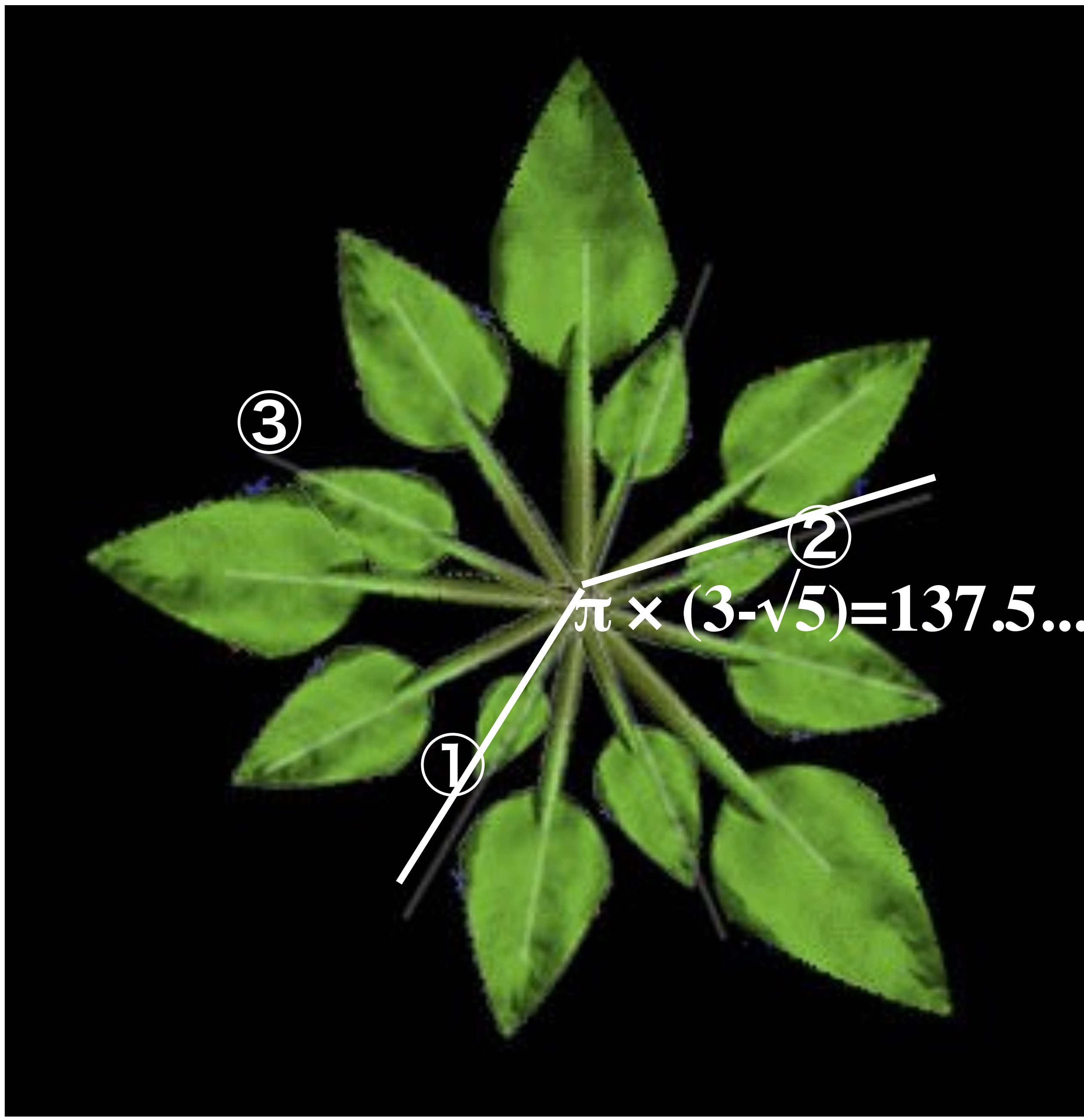
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n-1}{}^\dagger W_{n-1}{}^\dagger U_{n-1}$

$\Rightarrow O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

${}^{\dagger}W^{\dagger}=UU^{\dagger}_{n\text{-}1}$

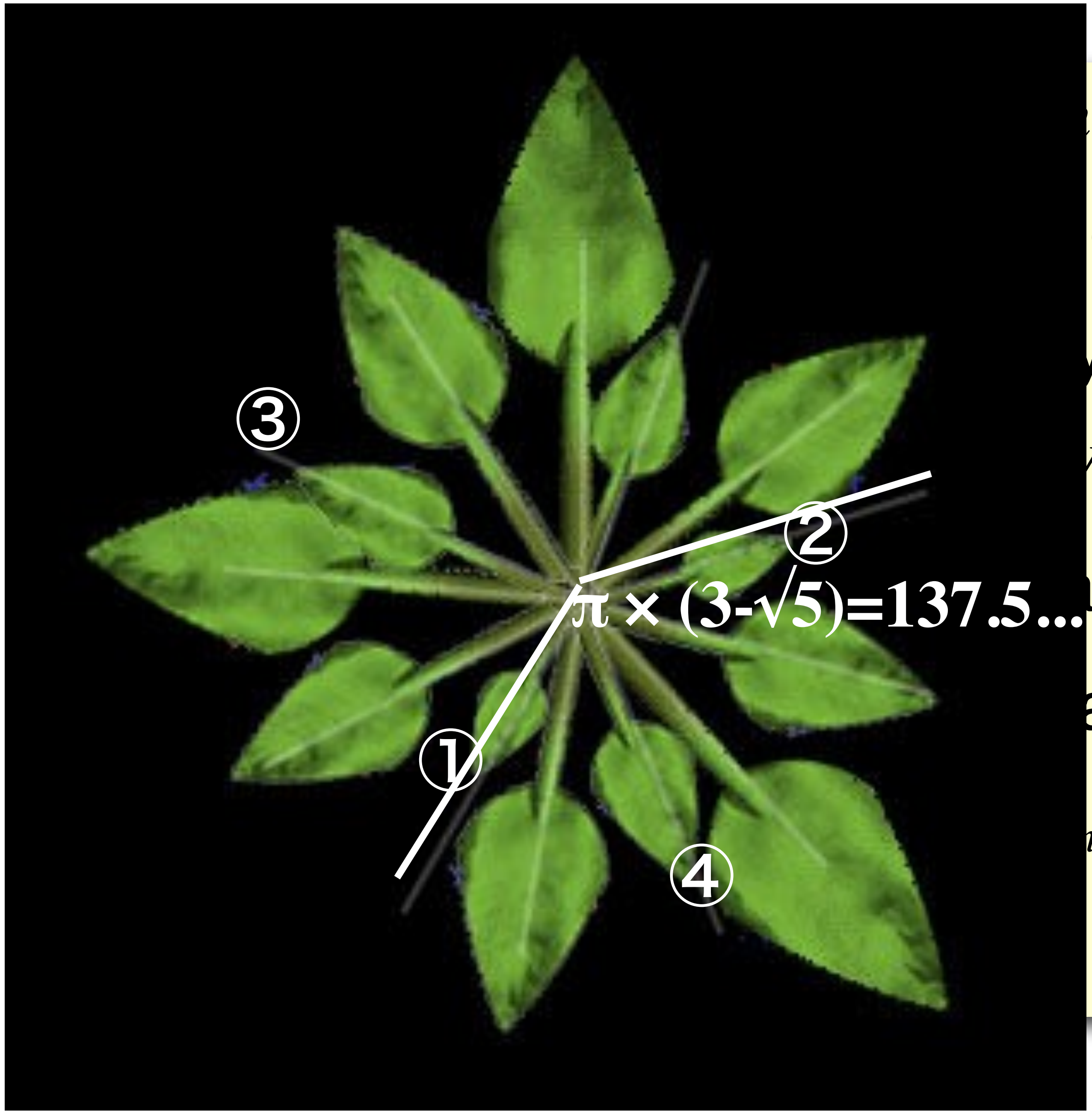
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

${}_{n\text{-}1}{}^{\dagger}W_{n\text{-}1}{}^{\dagger}U_{n\text{-}1}$

➡ $O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



…)

…c approximation of U

…y-Kitaev(U,n-1)

…$V^\dagger W^\dagger = UU^\dagger_{n-1}$

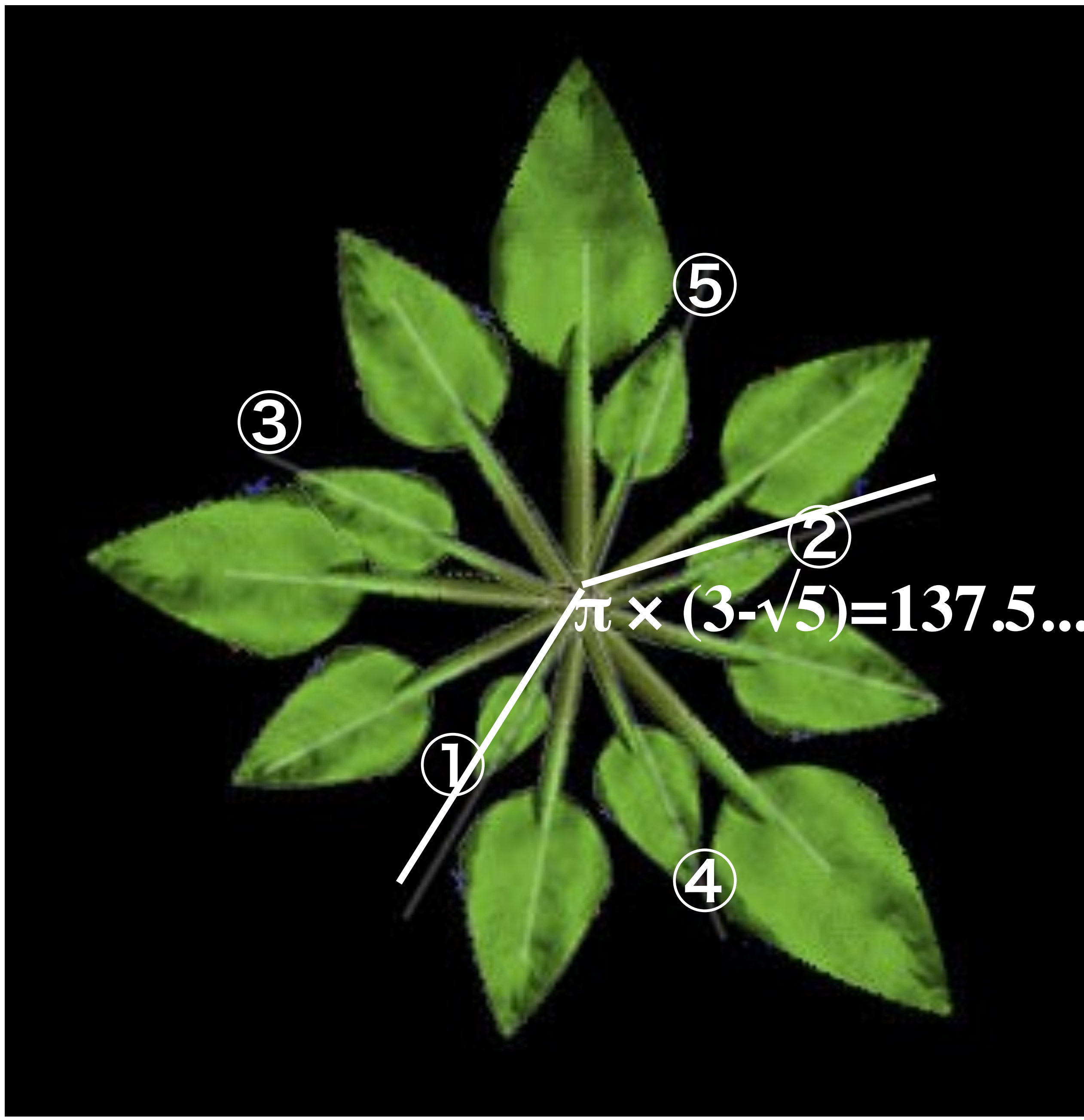
…ay-Kitaev(V,n-1)

…ay-Kitaev(V,n-1)

…$_{n-1}{}^\dagger W_{n-1}{}^\dagger U_{n-1}$

$\Rightarrow O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

$^\dagger W^\dagger = UU^\dagger_{n-1}$

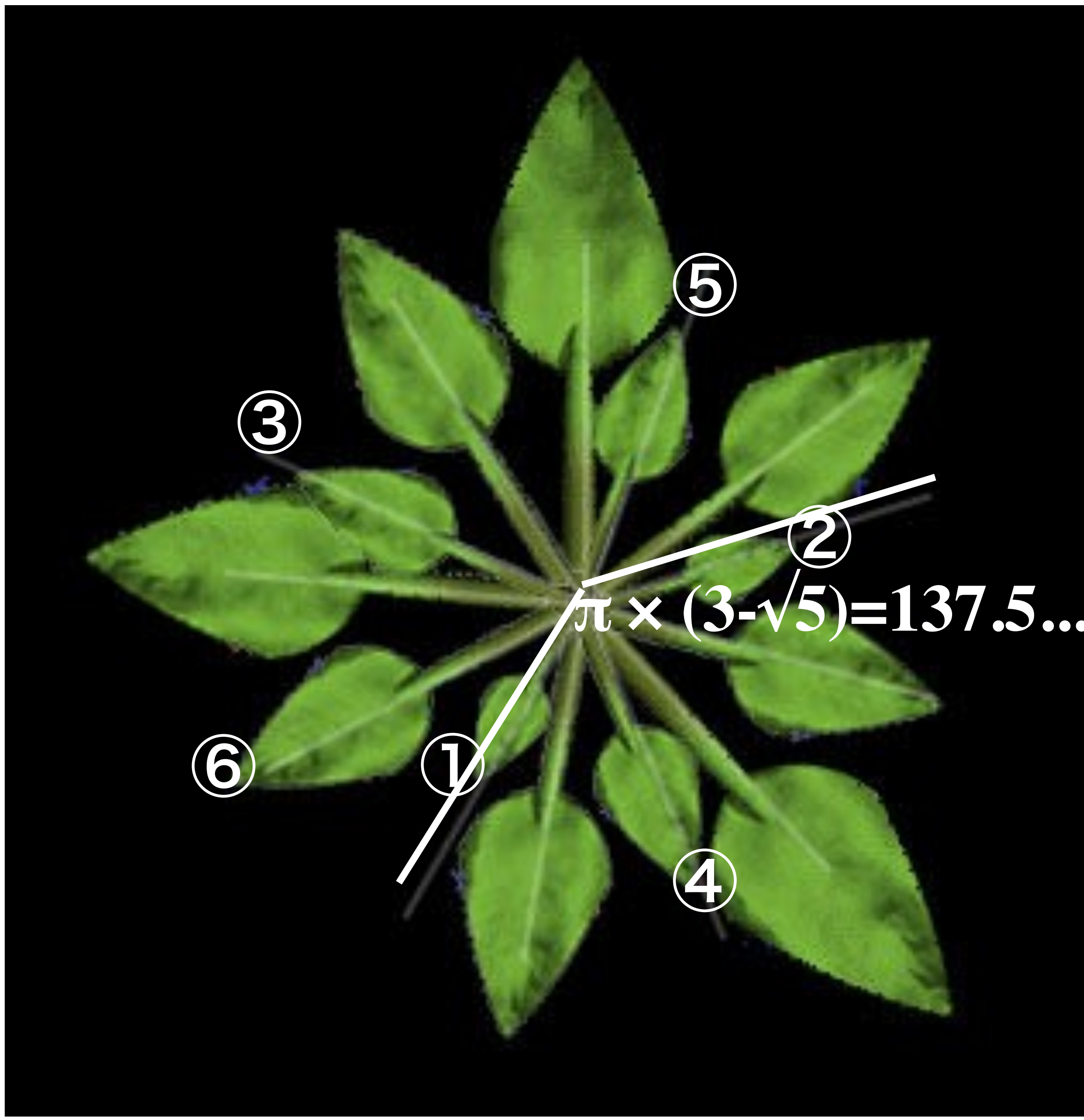
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n-1}{}^\dagger W_{n-1}{}^\dagger U_{n-1}$

$\Rightarrow O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



$)$

c approximation of U

y-Kitaev(U,n-1)

$V^{\dagger}W^{\dagger}=UU^{\dagger}_{n-1}$

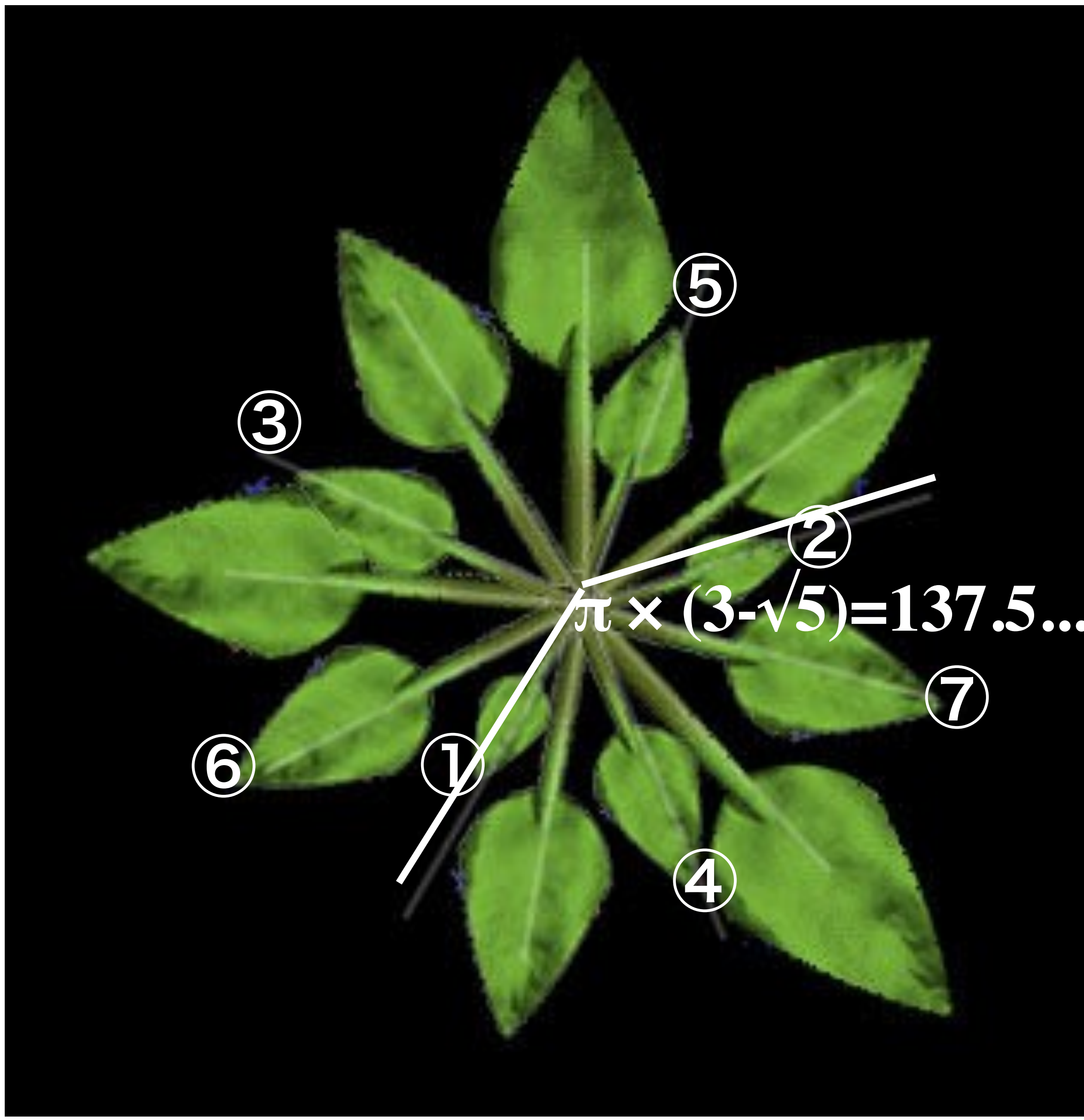
y-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n-1}{}^{\dagger}W_{n-1}{}^{\dagger}U_{n-1}$

➡ $O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

$V^\dagger W^\dagger = UU^\dagger_{n-1}$

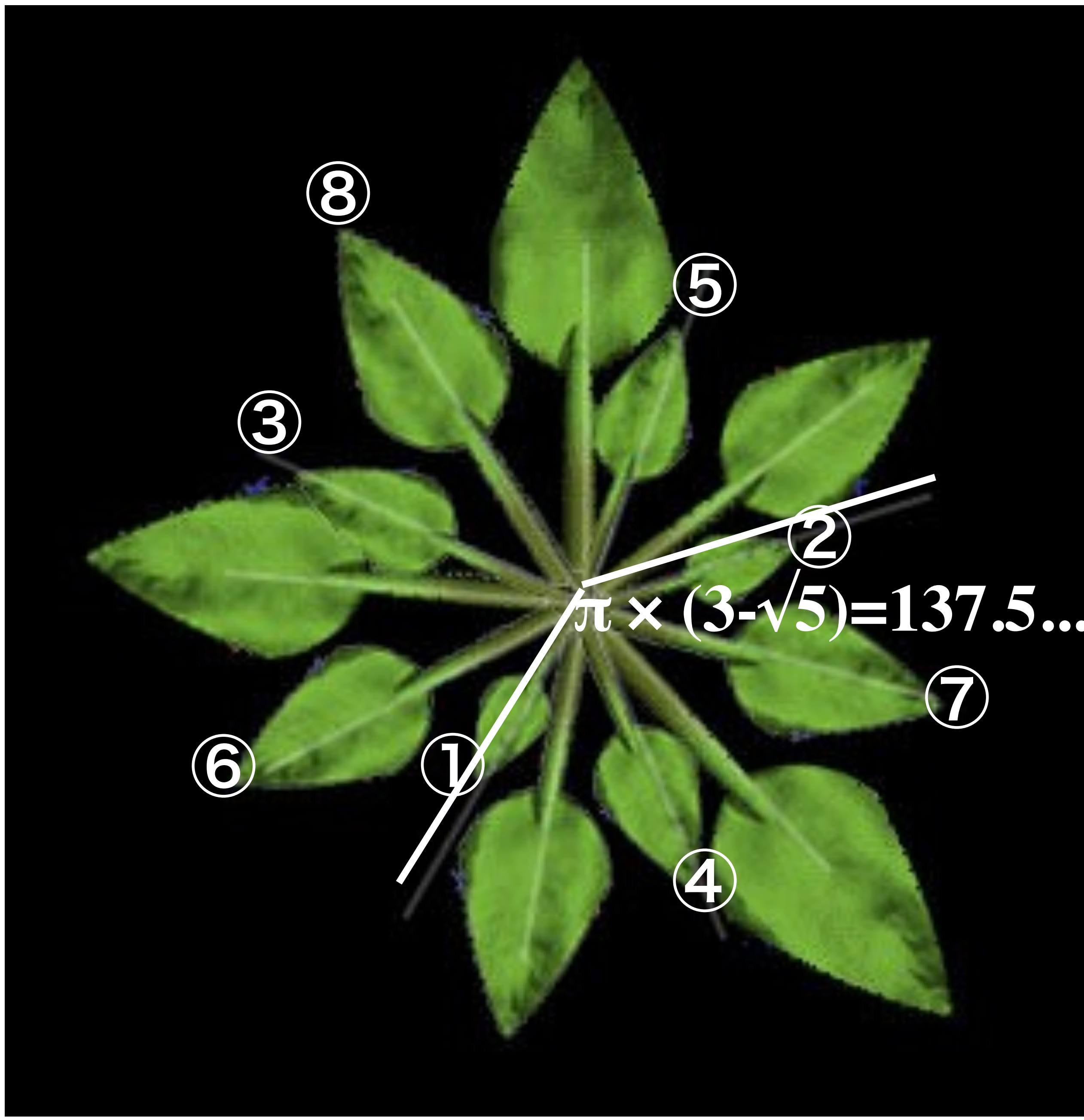
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n-1}{}^\dagger W_{n-1}{}^\dagger U_{n-1}$

$\Rightarrow O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

$V^{\dagger}W^{\dagger}=UU^{\dagger}_{n-1}$

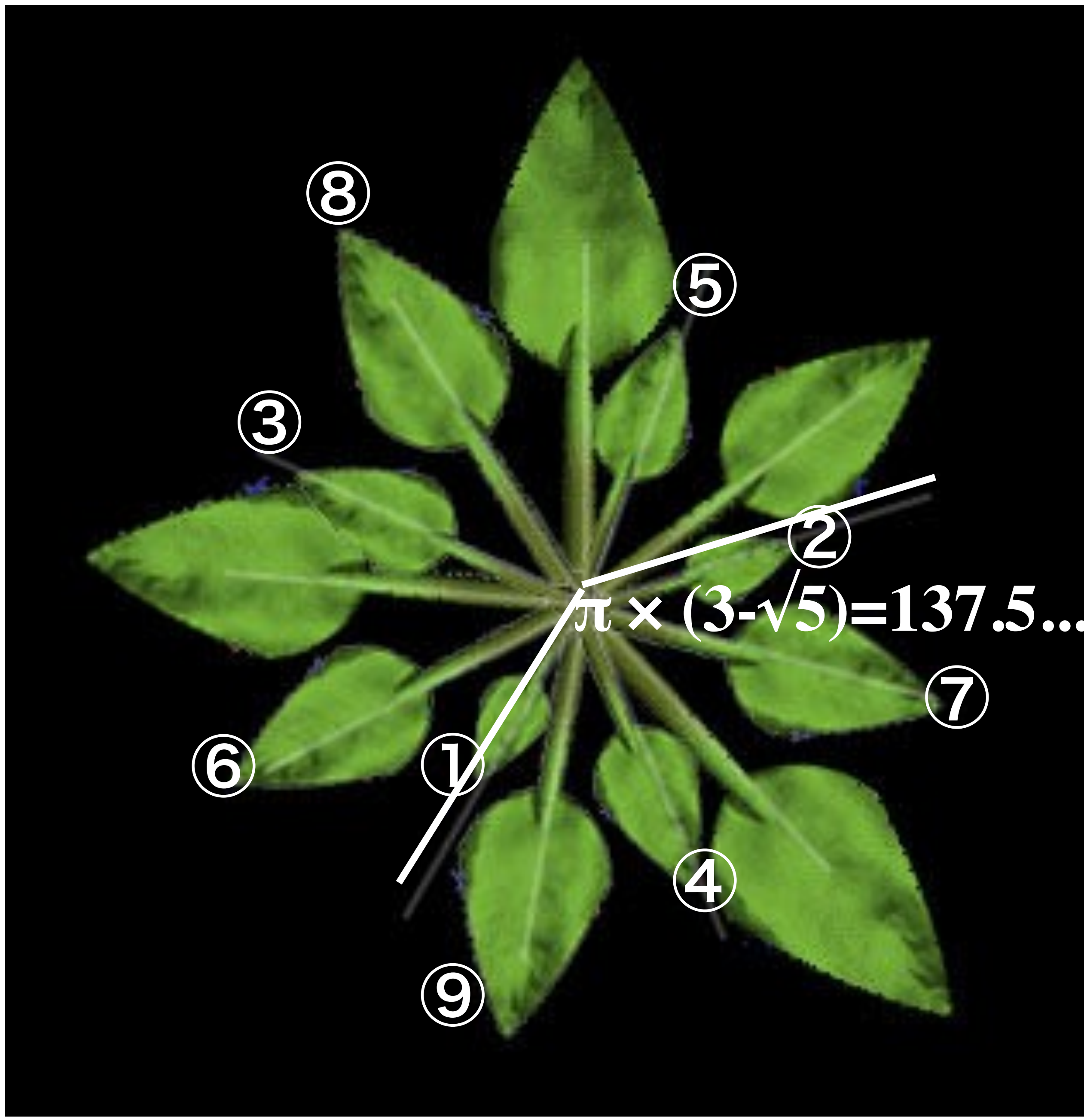
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n-1}{}^{\dagger}W_{n-1}{}^{\dagger}U_{n-1}$

$\Rightarrow O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

$V^\dagger W^\dagger = UU^\dagger_{n-1}$

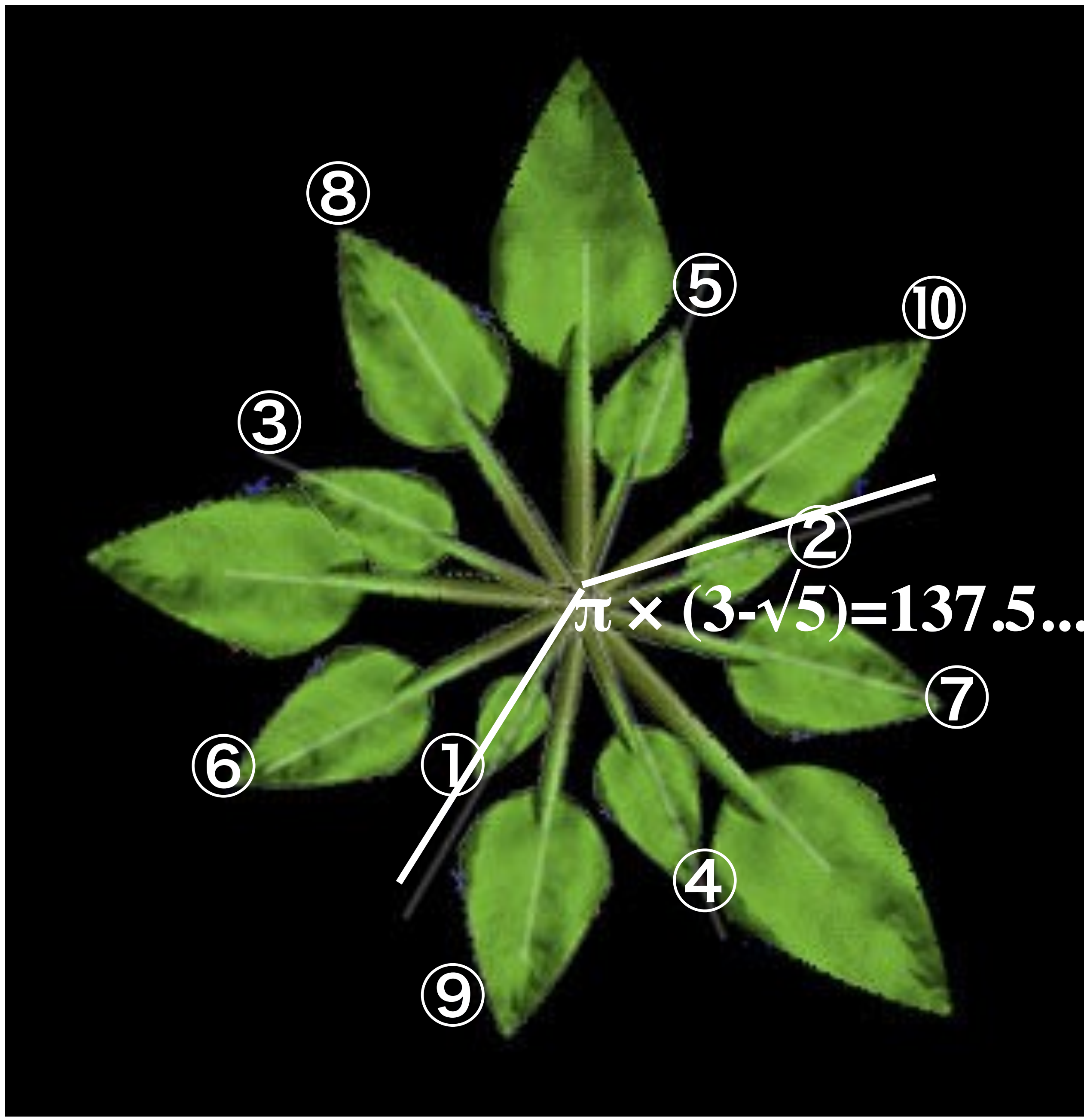
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n-1}{}^\dagger W_{n-1}{}^\dagger U_{n-1}$

$O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

$V^\dagger W^\dagger = UU^\dagger_{n-1}$

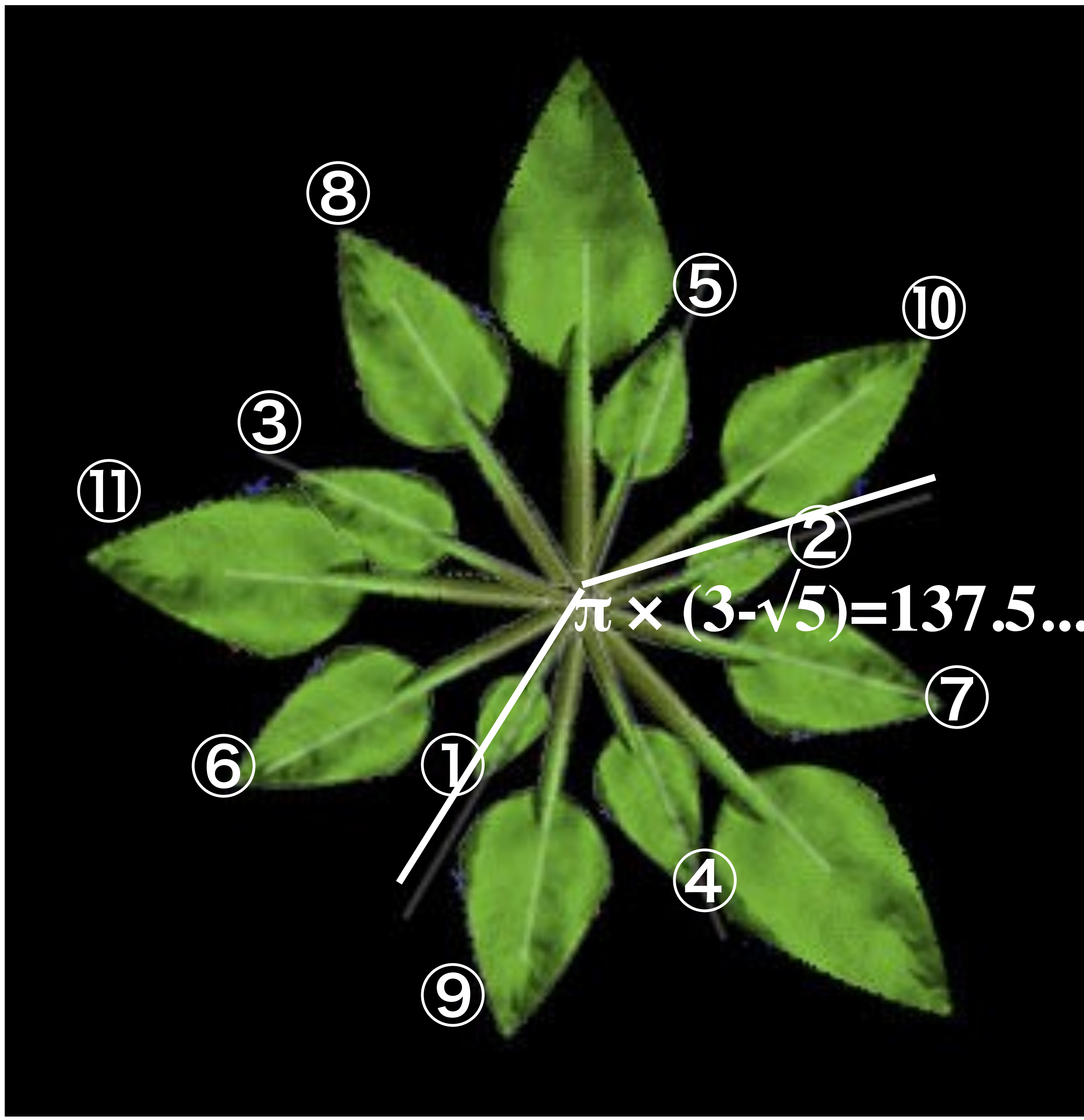
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n-1}{}^\dagger W_{n-1}{}^\dagger U_{n-1}$

➡ $O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

$V^{\dagger}W^{\dagger}=UU^{\dagger}_{n\text{-}1}$

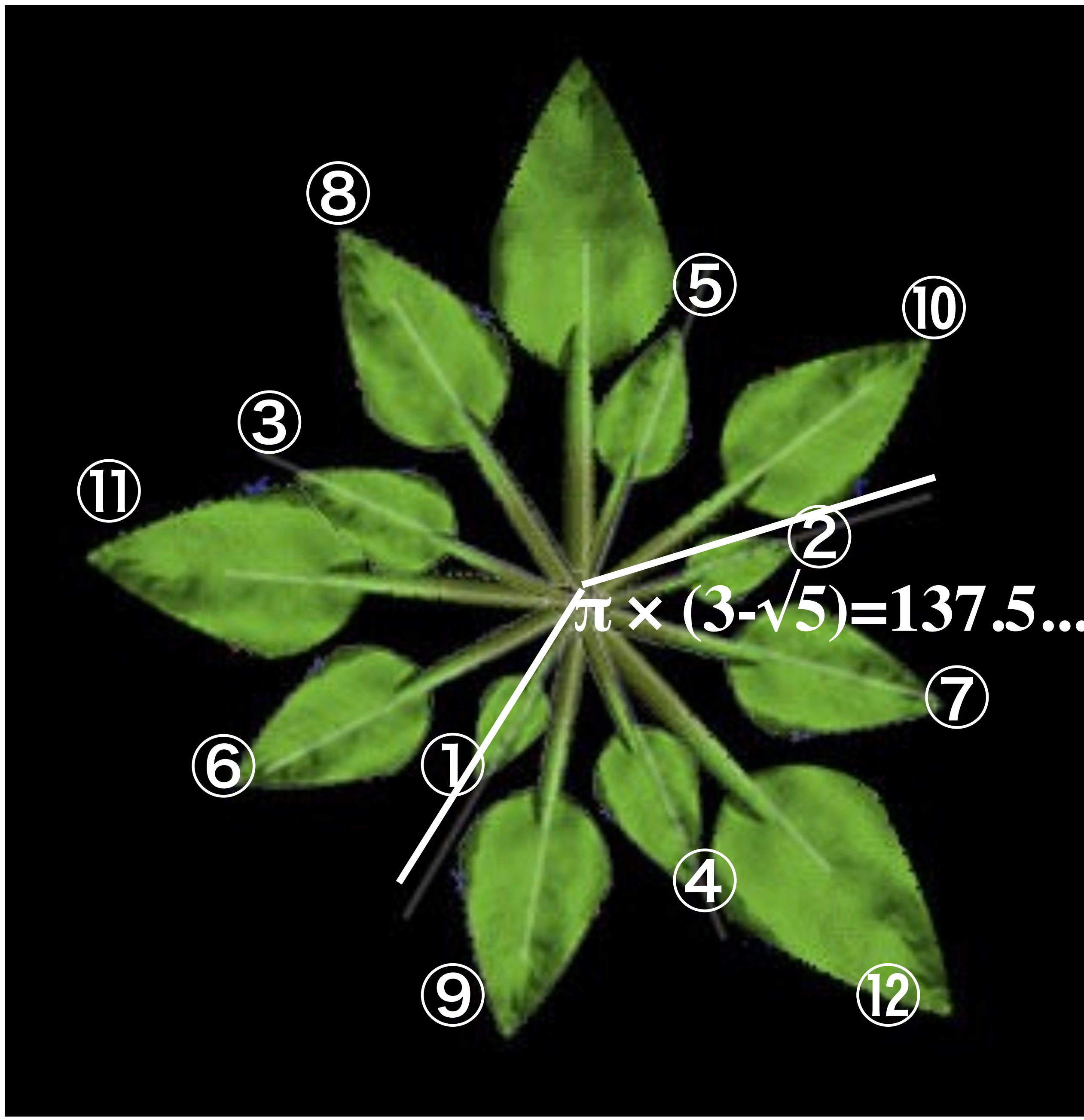
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n\text{-}1}{}^{\dagger}W_{n\text{-}1}{}^{\dagger}U_{n\text{-}1}$

$\Rightarrow O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

$V^\dagger W^\dagger = UU^\dagger_{n-1}$

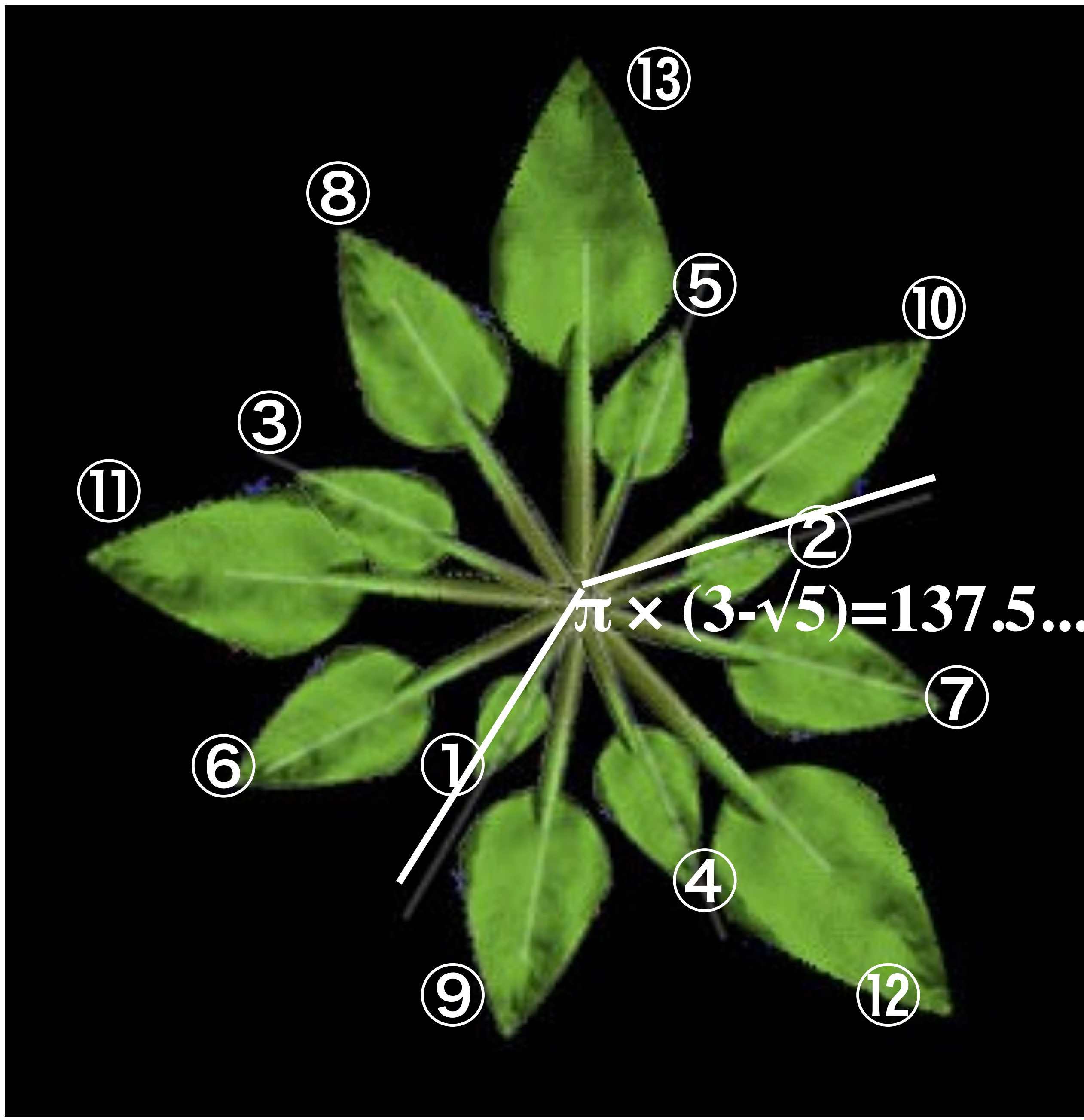
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n-1}{}^\dagger W_{n-1}{}^\dagger U_{n-1}$

$\Rightarrow O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



)

c approximation of U

y-Kitaev(U,n-1)

$V^\dagger W^\dagger = UU^\dagger_{n-1}$

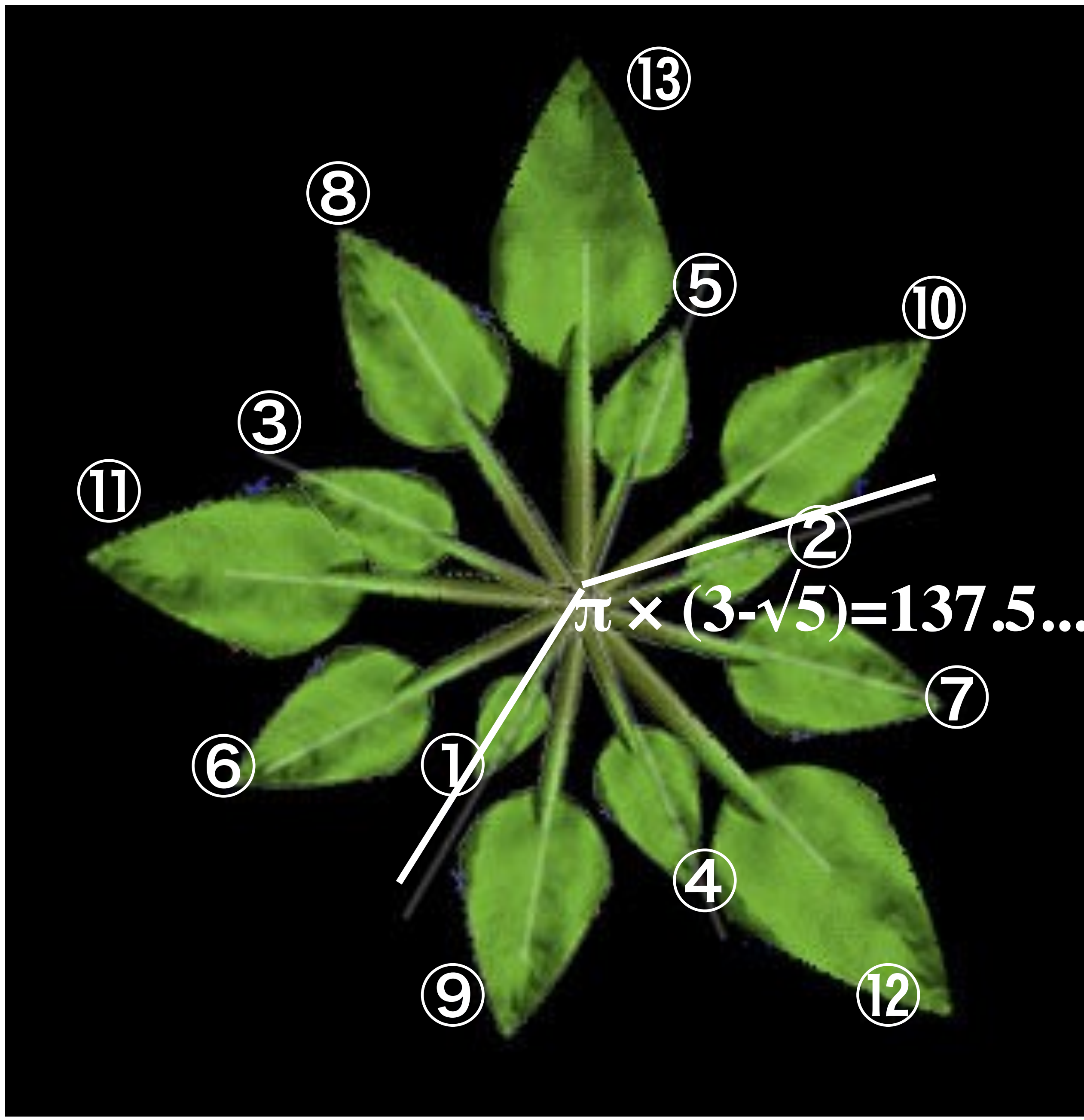
ay-Kitaev(V,n-1)

ay-Kitaev(V,n-1)

$_{n-1}{}^\dagger W_{n-1}{}^\dagger U_{n-1}$

$\Rightarrow O(\log^c(1/\epsilon))$

Kitaev-Solvayの定理



第二章:グラフ状態とMBQC

ここで勉強する内容：

第一章で勉強したツールを使ってみる.

グラフ状態，測定によるのグラフ状態の変形,

グラフ状態の性質，stabilizer状態との関係,

ついでに，第五章で必要となるMBQCの導入.

グラフ状態（cluster状態）の生成方法.

グラフ状態

- グラフ状態：グラフ G(V,E)を用いて定義される stabilizer 状態

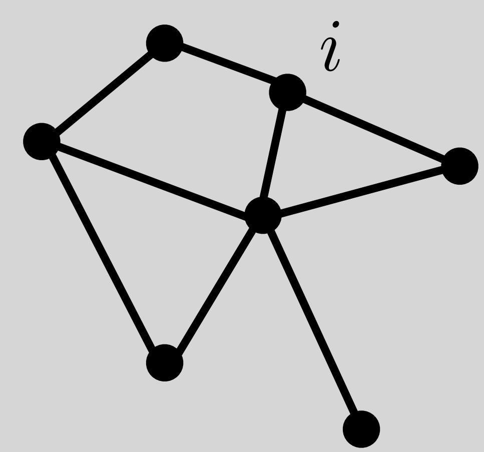
グラフ状態

◆ グラフ状態：グラフ G(V,E)を用いて定義される stabilizer 状態

stabilizer演算子は各頂点に対して

$$K_i = X_i \prod_{j \in V_i} Z_j$$

と定義される.



グラフ G(V,E)

V:頂点の集合，E:辺の集合

グラフ状態

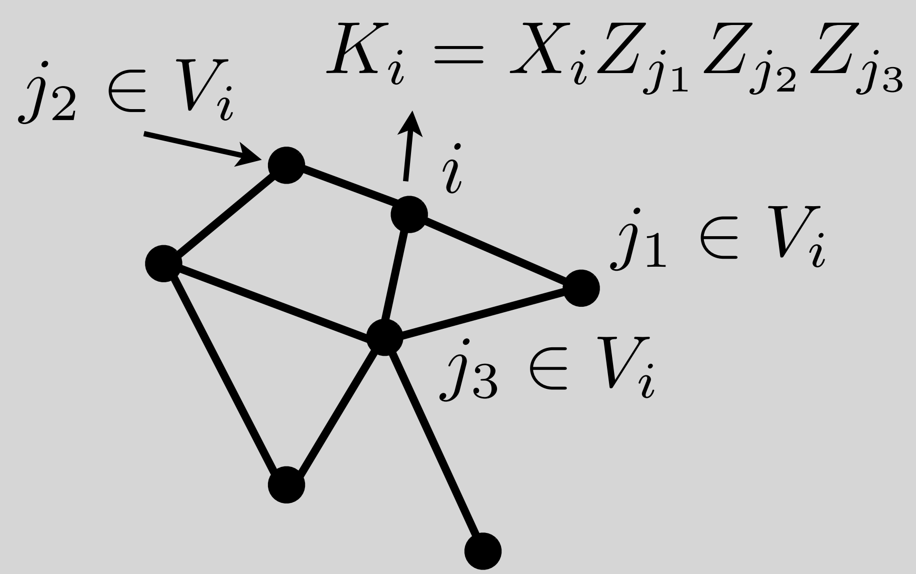
◆ グラフ状態：グラフ G(V,E)を用いて定義される stabilizer 状態

stabilizer演算子は各頂点に対して

$$K_i = X_i \prod_{j \in \boxed{V_i}} Z_j$$

点i と隣接する頂点の集合

と定義される.



グラフ G(V,E)

V:頂点の集合，E:辺の集合

Cluster状態

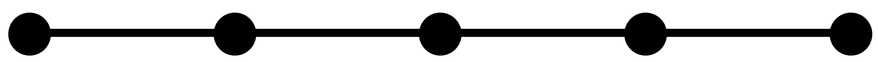
Cluster状態

特に，並進対称性のあるグラフ（直線上，正方格子，六角格子など）上で定義される場合，cluster stateと呼ばれる．

Cluster状態

特に，並進対称性のあるグラフ（直線上，正方格子，六角格子など）上で定義される場合，cluster stateと呼ばれる.

◆ 1D cluster state

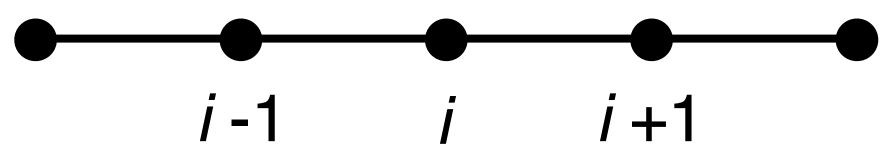


Cluster状態

特に，並進対称性のあるグラフ（直線上，正方格子，六角格子など）上で定義される場合，cluster stateと呼ばれる．

◆ 1D cluster state

$$K_i = Z_{i-1}X_iZ_{i+1}$$

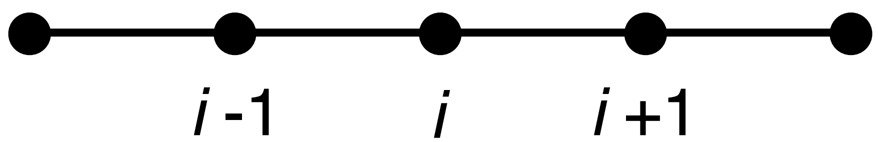


Cluster状態

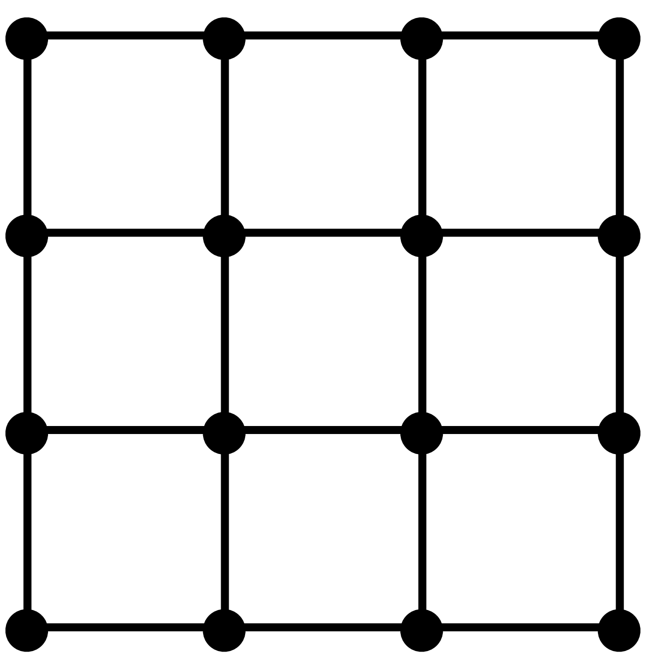
特に，並進対称性のあるグラフ（直線上，正方格子，六角格子など）上で定義される場合，cluster stateと呼ばれる．

◆ 1D cluster state

$$K_i = Z_{i-1} X_i Z_{i+1}$$



◆ 2D cluster state

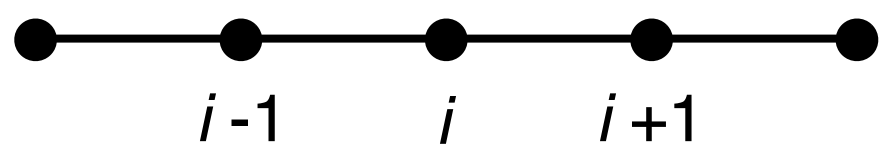


Cluster状態

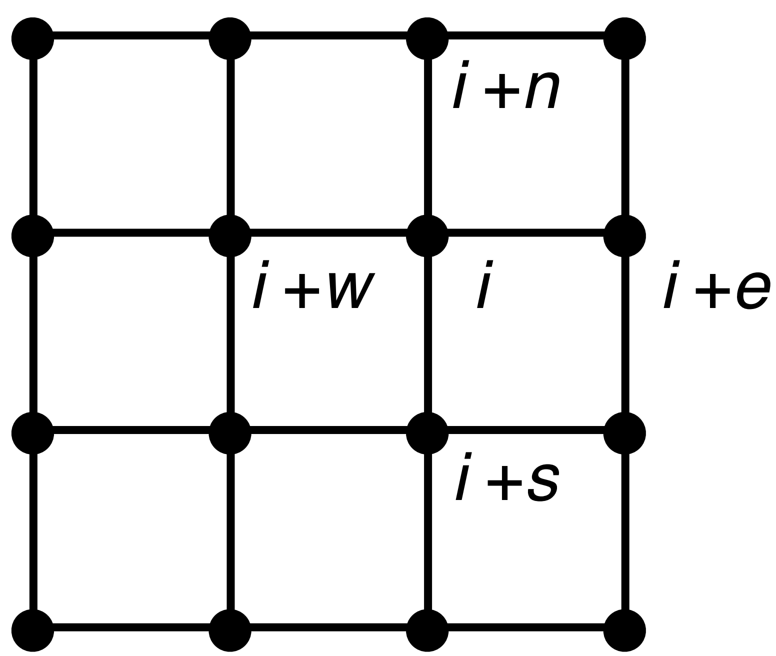
特に，並進対称性のあるグラフ（直線上，正方格子，六角格子など）上で定義される場合，cluster stateと呼ばれる．

- 1D cluster state

$$K_i = Z_{i-1} X_i Z_{i+1}$$



- 2D cluster state



$$K_i = X_i Z_{i+n} Z_{i+e} Z_{i+s} Z_{i+w}$$

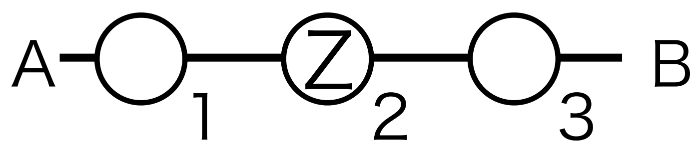
グラフ状態とPauli測定 Z

◆ Z 基底の測定

グラフ状態とPauli測定 Z

◆ Z 基底の測定

- 一直線上にあるとき



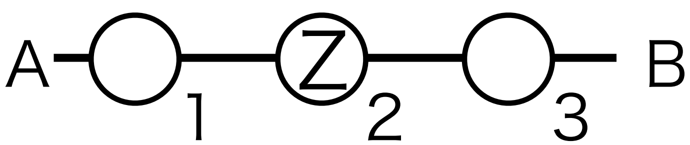
測定前の stabilizer

$$\begin{cases} S_1 &= A_z X_1 Z_2 \\ S_2 &= Z_1 X_2 Z_3 \\ S_3 &= Z_2 X_3 B_Z \end{cases}$$

グラフ状態とPauli測定 Z

◆ Z 基底の測定

- 一直線上にあるとき



測定前の stabilizer

$$\begin{cases} S_1 &= A_z X_1 Z_2 \\ S_2 &= Z_1 X_2 Z_3 \\ S_3 &= Z_2 X_3 B_Z \end{cases}$$

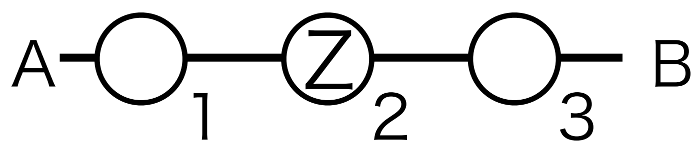
測定後の stabilizer

$$\begin{cases} S_1 = A_z X_1 (Z_2) \\ S_3 = (Z_2) X_3 B_z \end{cases}$$

グラフ状態とPauli測定 Z

◆ Z 基底の測定

- 一直線上にあるとき



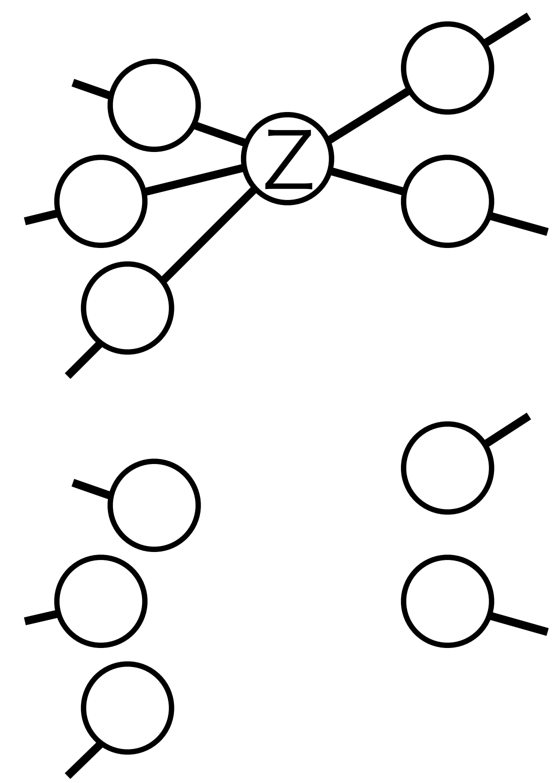
測定前の stabilizer

$$\begin{cases} S_1 &= A_z X_1 Z_2 \\ S_2 &= Z_1 X_2 Z_3 \\ S_3 &= Z_2 X_3 B_Z \end{cases}$$

測定後の stabilizer

$$\begin{cases} S_1 = A_z X_1 (Z_2) \\ S_3 = (Z_2) X_3 B_z \end{cases}$$

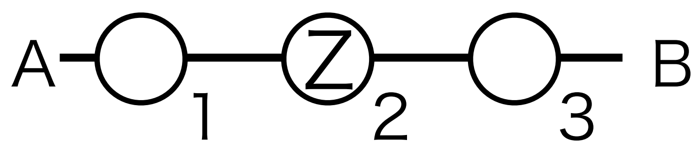
- 一般の場合



グラフ状態とPauli測定 Z

◆ Z 基底の測定

- 一直線上にあるとき



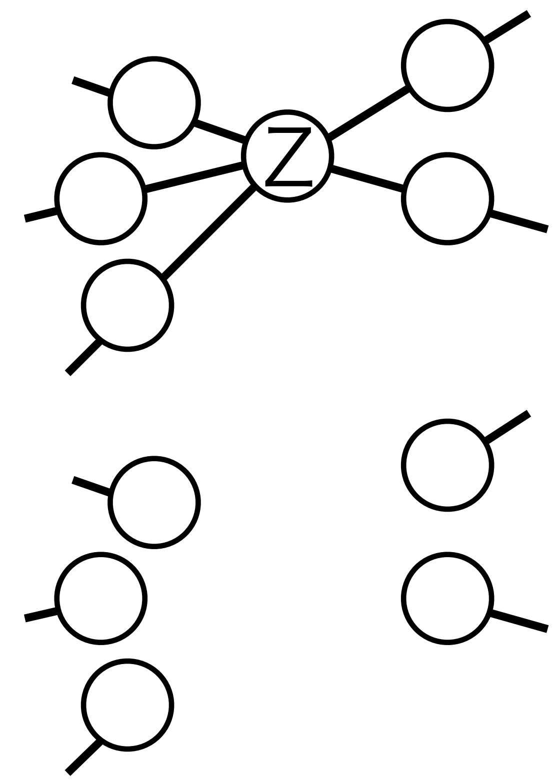
測定前の stabilizer

$$\begin{cases} S_1 &= A_z X_1 Z_2 \\ S_2 &= Z_1 X_2 Z_3 \\ S_3 &= Z_2 X_3 B_Z \end{cases}$$

測定後の stabilizer

$$\begin{cases} S_1 = A_z X_1 (Z_2) \\ S_3 = (Z_2) X_3 B_z \end{cases}$$

- 一般の場合

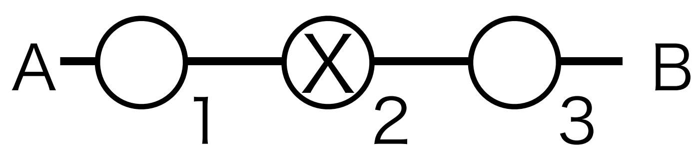


Z 測定された状態がグラフ状態から消える.

グラフ状態とPauli測定 X

◆ X 基底の測定

- 一直線上にあるとき

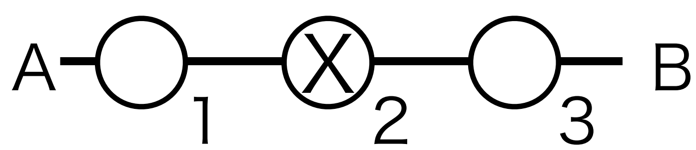


$$
\begin{aligned}
S_1 &= A_z X_1 Z_2 \\
S_2 &= \quad Z_1 X_2 Z_3 \\
S_3 &= \qquad Z_2 X_3 B_Z
\end{aligned}
$$

グラフ状態とPauli測定 X

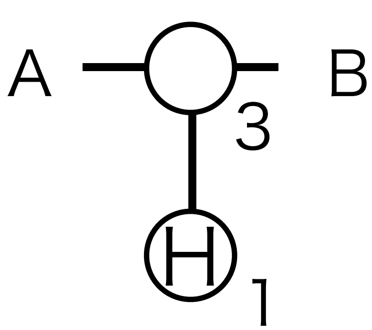
◆ X 基底の測定

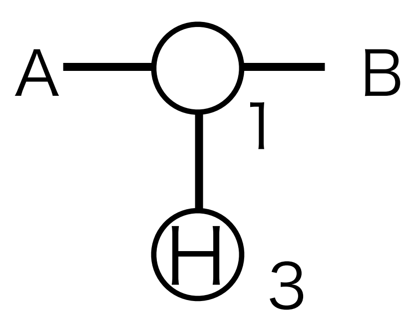
- 一直線上にあるとき



$$
\begin{aligned}
S_1 &= A_z X_1 Z_2 \\
S_2 &= \quad Z_1 X_2 Z_3 \\
S_3 &= \qquad Z_2 X_3 B_Z
\end{aligned}
$$

$$
\begin{aligned}
S_1 S_3 &= A_z X_1 X_3 B_z \\
S_2 &= Z_1 (X_2) Z_2
\end{aligned}
$$

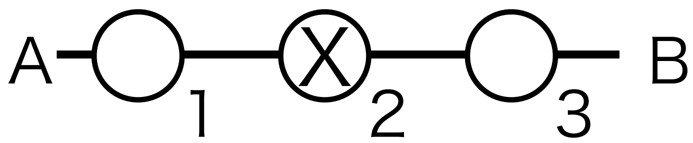




グラフ状態とPauli測定 X

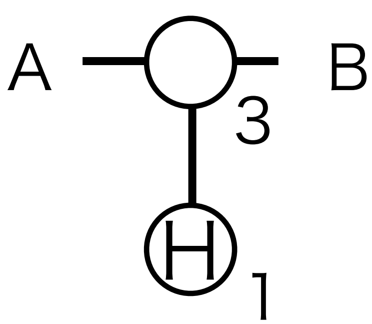
◆ X 基底の測定

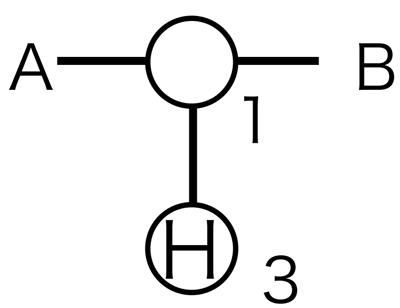
- 一直線上にあるとき

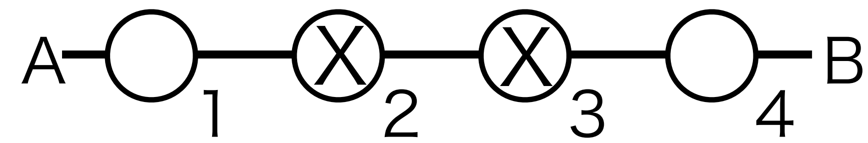


$$
\begin{aligned}
S_1 &= A_z X_1 Z_2 \\
S_2 &= Z_1 X_2 Z_3 \\
S_3 &= Z_2 X_3 B_Z
\end{aligned}
$$

$$
\begin{aligned}
S_1 S_3 &= A_z X_1 X_3 B_z \\
S_2 &= Z_1 (X_2) Z_2
\end{aligned}
$$



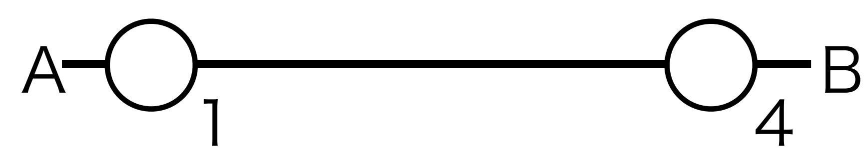




$$
\begin{aligned}
S_1 &= A_z X_1 Z_2 \\
S_2 &= Z_1 X_2 Z_3 \\
S_3 &= Z_2 X_3 Z_4 \\
S_4 &= Z_3 X_4 B_Z
\end{aligned}
$$

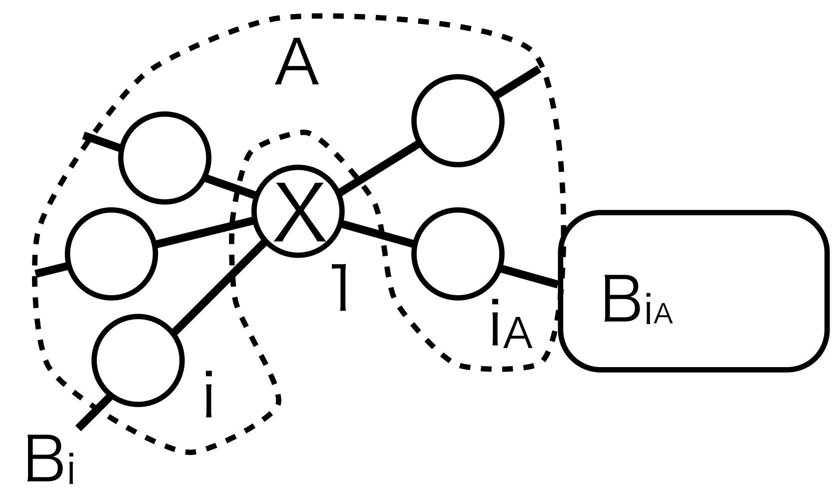
$$
S_1 S_3 = A_z X_1 (X_3) Z_4
$$

$$
S_2 S_4 = Z_1 (X_2) X_4 B_z
$$



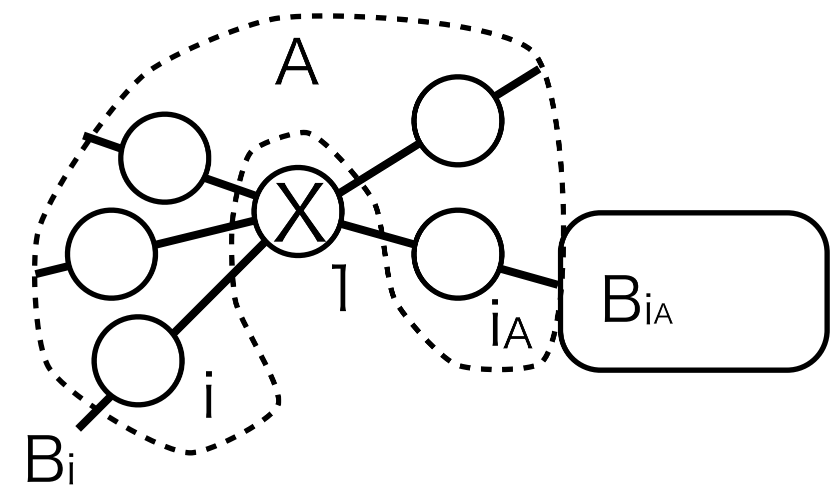
グラフ状態とPauli測定 X

- 一般の場合



グラフ状態とPauli測定 X

- 一般の場合



$$S_1 = X_1 \prod_{i \in A} Z_i$$

$$S_i = X_i Z_1 \prod_{k \in B_i} Z_k$$

$$S_1 = (X_1) \prod_{i \in A} Z_i$$

$$S_i S_{i_A} = X_i X_{i_A} \prod_{k \in B_i} Z_k \prod_{k \in B_{i_A}} Z_k$$

グラフ状態とPauli測定 X

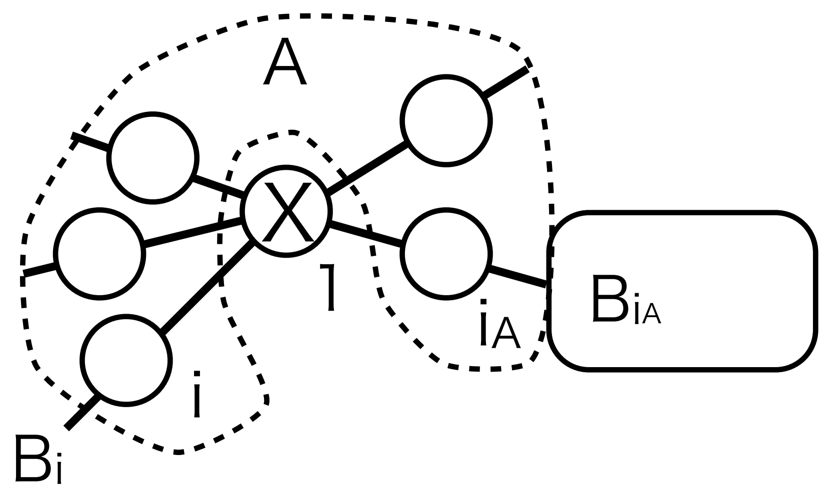
- 一般の場合

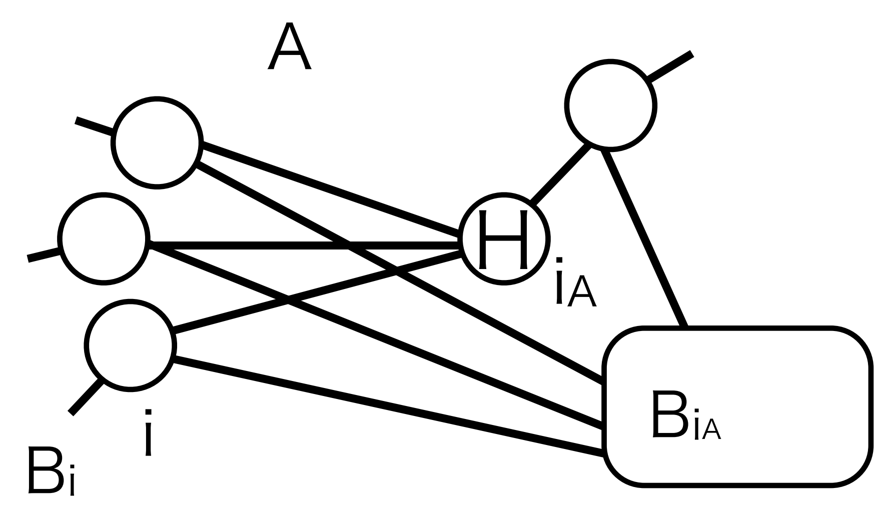
$$S_1 = X_1 \prod_{i\in A} Z_i$$

$$S_i = X_i Z_1 \prod_{k\in B_i} Z_k$$

$$S_1 = (X_1) \prod_{i\in A} Z_i$$

$$S_i S_{i_A} = X_i X_{i_A} \prod_{k\in B_i} Z_k \prod_{k\in B_{i_A}} Z_k$$

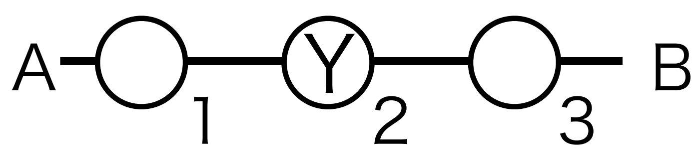




グラフ状態とPauli測定 Y

◆ Y 基底の測定

- 一直線上にあるとき

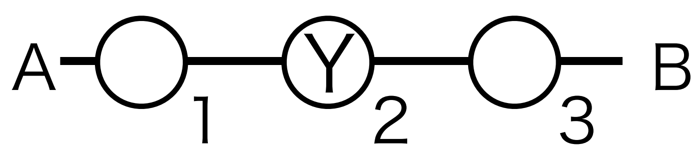


$$
\begin{aligned}
S_1 &= A_z X_1 Z_2 \\
S_2 &= Z_1 X_2 Z_3 \\
S_3 &= Z_2 X_3 B_Z
\end{aligned}
$$

グラフ状態とPauli測定 Y

◆ Y基底の測定

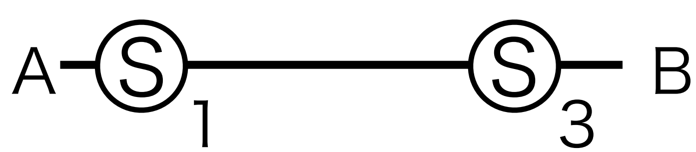
- 一直線上にあるとき



$$
\begin{aligned}
S_1 &= A_z X_1 Z_2 \\
S_2 &= Z_1 X_2 Z_3 \\
S_3 &= Z_2 X_3 B_Z
\end{aligned}
$$

$$S_1 S_2 = A_z Y_1 (Y_2) Z_3$$

$$S_2 S_3 = Z_1 (Y_2) Y_3 B_Z$$

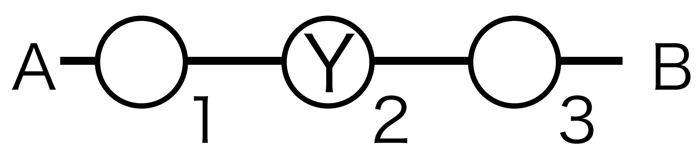


$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

グラフ状態とPauli測定 Y

◆ Y 基底の測定

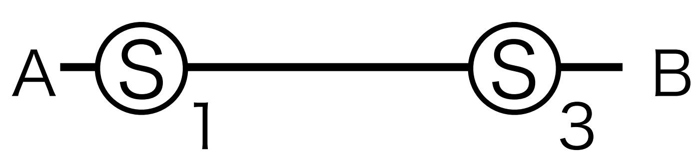
- 一直線上にあるとき



$$
\begin{aligned}
S_1 &= A_z X_1 Z_2 \\
S_2 &= Z_1 X_2 Z_3 \\
S_3 &= Z_2 X_3 B_Z
\end{aligned}
$$

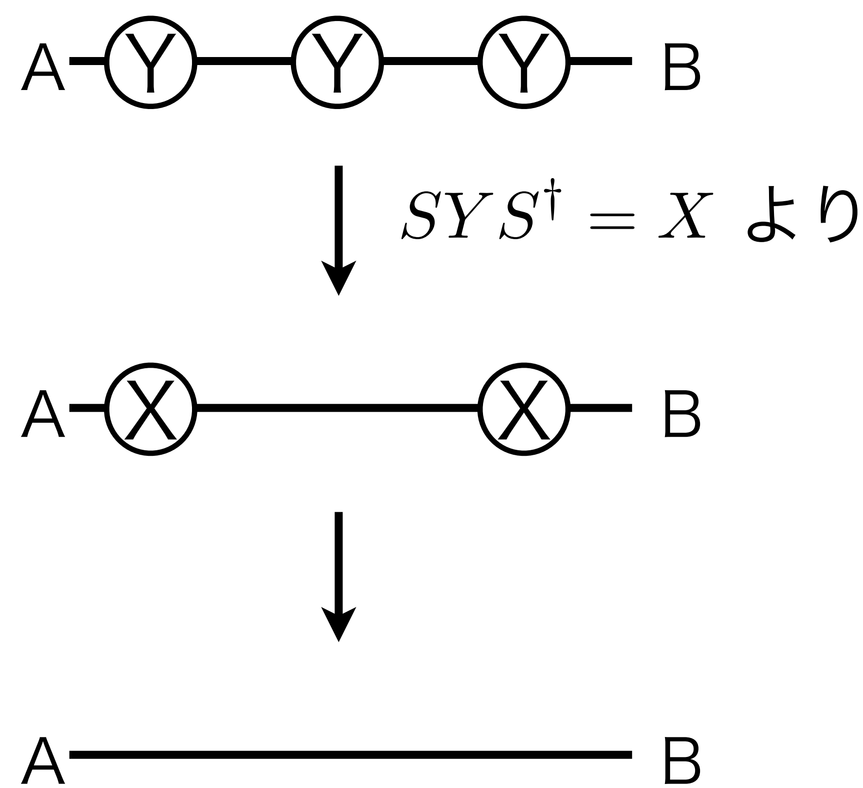
$$S_1 S_2 = A_z Y_1 (Y_2) Z_3$$

$$S_2 S_3 = Z_1 (Y_2) Y_3 B_Z$$



$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

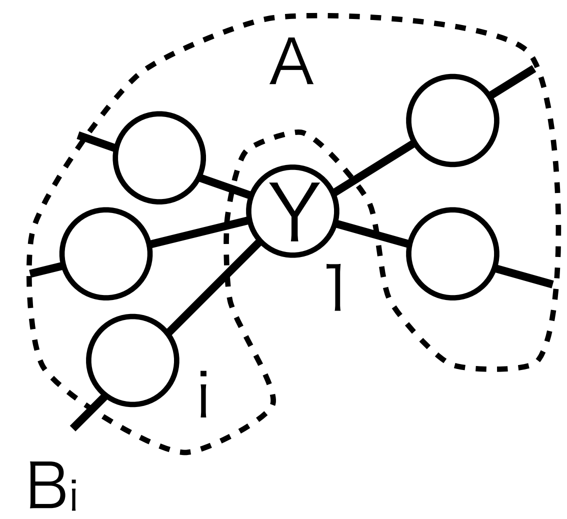
- 応用



グラフ状態とPauli測定 Y

◆ Y 基底の測定

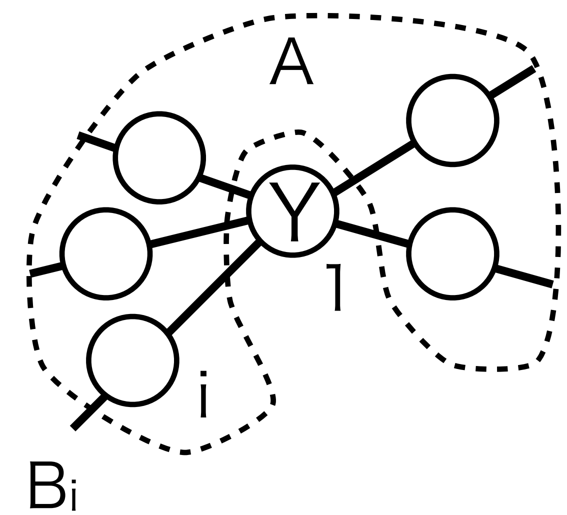
- 一般の場合



グラフ状態とPauli測定 Y

◆ Y基底の測定

- 一般の場合



$$S_1 = X_1 \prod_{i \in A} Z_i$$

$$S_i = X_i Z_1 \prod_{k \in B_i} Z_k$$

$$S_1 S_i = Y_i (Y_1) \prod_{j \in A/i} Z_j \prod_{k \in B_i} Z_k$$

グラフ状態とPauli測定 Y

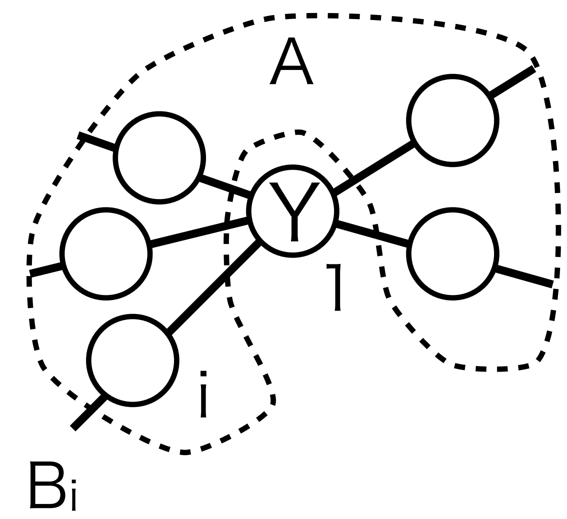
◆ Y基底の測定

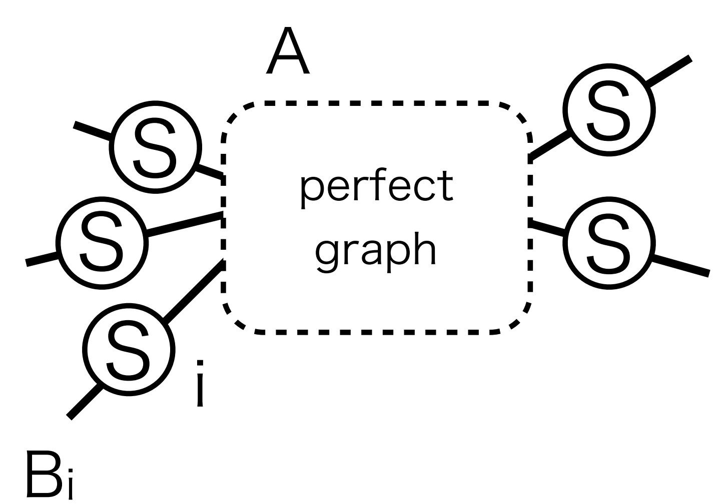
- 一般の場合

$$S_1 = X_1 \prod_{i \in A} Z_i$$

$$S_i = X_i Z_1 \prod_{k \in B_i} Z_k$$

$$S_1 S_i = Y_i (Y_1) \prod_{j \in A/i} Z_j \prod_{k \in B_i} Z_k$$





グラフ状態とstabilizer状態

- 任意のstabilizer状態はあるグラフ状態と local Clifford (LC) のもとで等価（LC-equivalence）

グラフ状態とstabilizer状態

◆ 任意のstabilizer状態はあるグラフ状態と local Clifford (LC) のもとで等価（LC-equivalence）

→２つのstabilizer状態のLC-等価性は効率よく調べれる.

→グラフ状態はstabilizer状態の標準形.

→証明は [Van den Nest et al., PRA **69**, 022316 (2004)]参照.

グラフ状態とstabilizer状態

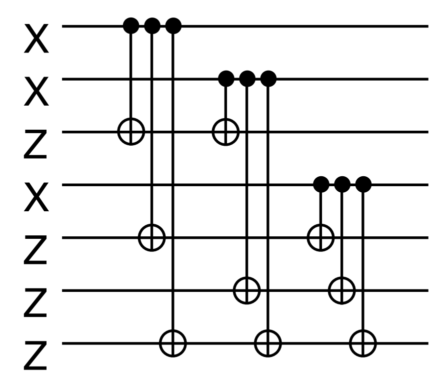
- 任意のstabilizer状態はあるグラフ状態と local Clifford (LC) のもとで等価（LC-equivalence）

→２つのstabilizer状態のLC-等価性は効率よく調べれる.

→グラフ状態はstabilizer状態の標準形.

→証明は [Van den Nest et al., PRA **69**, 022316 (2004)]参照.

$$(|00\rangle + |11\rangle)/\sqrt{2}$$



グラフ状態とstabilizer状態

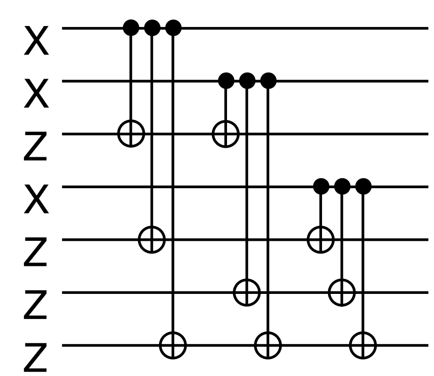
- 任意のstabilizer状態はあるグラフ状態と local Clifford (LC) のもとで等価（LC-equivalence）

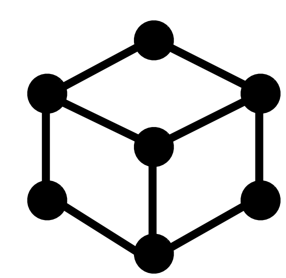
→２つのstabilizer状態のLC-等価性は効率よく調べれる.

→グラフ状態はstabilizer状態の標準形.

→証明は [Van den Nest et al., PRA **69**, 022316 (2004)]参照.

$$(|00\rangle + |11\rangle)/\sqrt{2}$$





グラフ状態とstabilizer状態

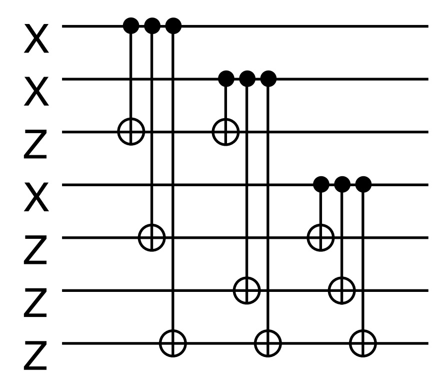
◆ 任意のstabilizer状態はあるグラフ状態と local Clifford (LC) のもとで等価（LC-equivalence）

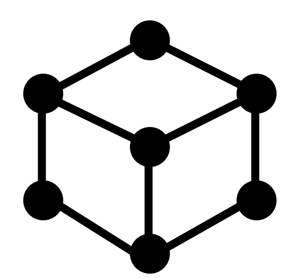
→２つのstabilizer状態のLC-等価性は効率よく調べれる.

→グラフ状態はstabilizer状態の標準形.

→証明は [Van den Nest et al., PRA **69**, 022316 (2004)]参照.

$$(|00\rangle + |11\rangle)/\sqrt{2}$$





◆ LU-LC conjecture: LU-eq. の必要十分条件は LC-eq.

効率よく検証できない.

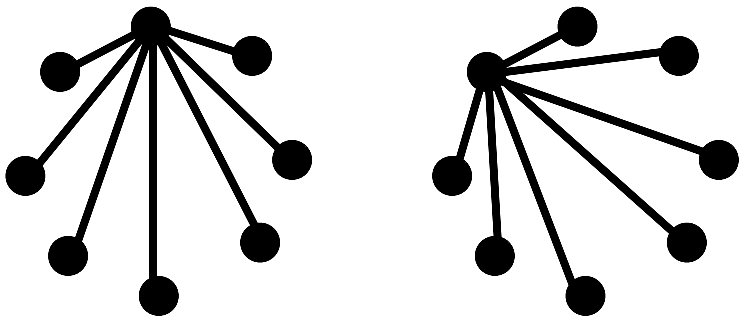
→反例が見つかる[Ji et al., Quant. Inf. Comput. **10** 97 (2010)].

→最近の動向[Sarvepalli-Raussendorf, PRA 82 022304 (2010)]とその参考文献.

グラフ状態とstabilizer状態

◆ しかし，あるstabilizer状態に対応するグラフが一意的に定義されるわけではない.

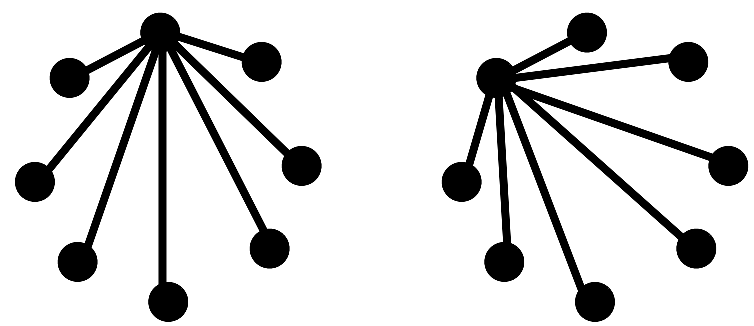
$$\frac{1}{\sqrt{N}}(|00\ldots0\rangle+|11\ldots1\rangle)$$



グラフ状態とstabilizer状態

◆ しかし，あるstabilizer状態に対応するグラフが一意的に定義されるわけではない.

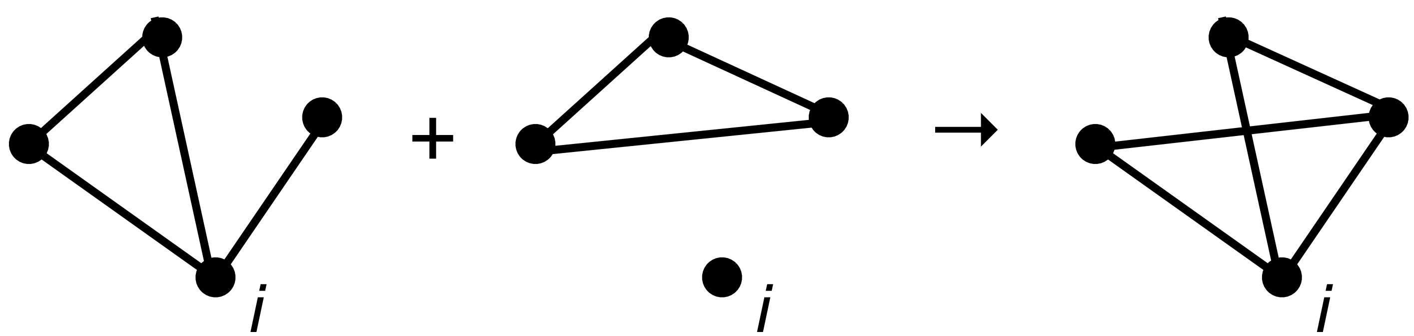
$$\frac{1}{\sqrt{N}}(|00\ldots0\rangle + |11\ldots1\rangle)$$



◆ Local complementarity: local Clifford $\sqrt{K_i}$

$$G \to G + \underline{G_i}$$

i に接続している点の完全グラフ.



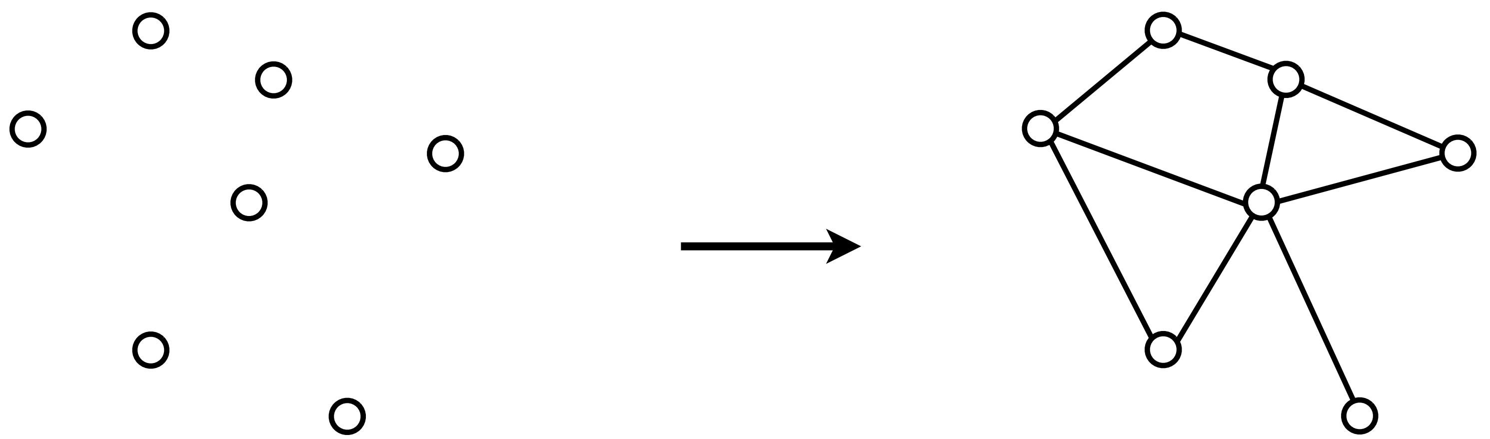
[Van den Nest et al., PRA **69**, 022316 (2004); Hein et al., quant-ph/0602096]

グラフ状態の生成

◆ グラフ状態の生成方法：

グラフ状態の生成

◆ グラフ状態の生成方法：

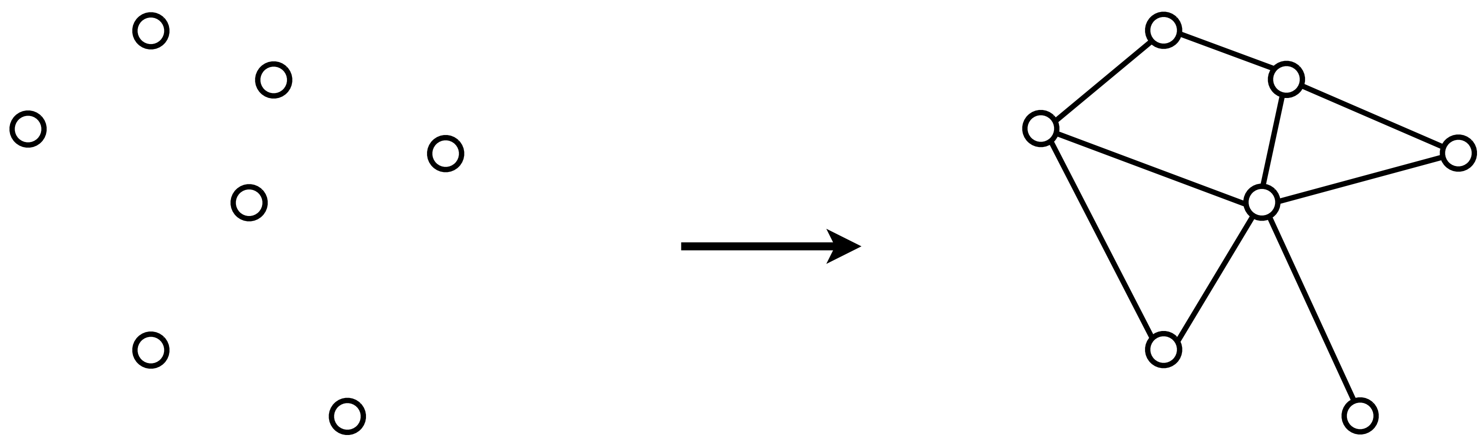


各頂点に対して $|+\rangle$ を準備.　　各辺の両端のqubitにCZを作用させる.

$$\prod_{e\in E} U_{\mathrm{CZ}}^{(e)} X_i U_{\mathrm{CZ}}^{(e)} = X_i \prod_{j\in V_i} Z_j$$

グラフ状態の生成

◆ グラフ状態の生成方法：



各頂点に対して $|+\rangle$ を準備.

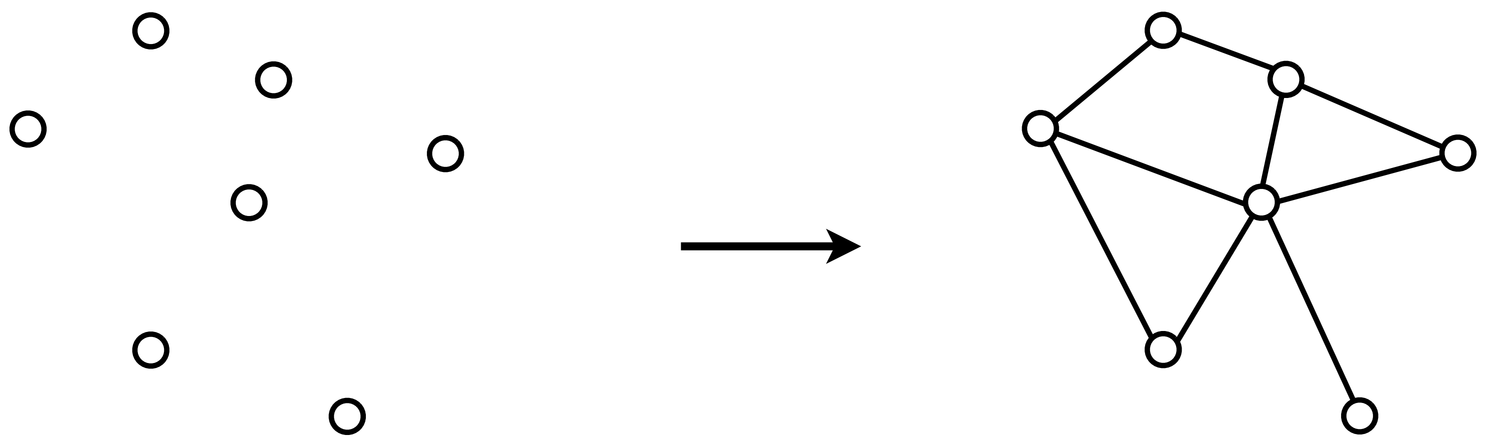
各辺の両端のqubitにCZを作用させる.

$$\prod_{e\in E} U_{\mathrm{CZ}}^{(e)} X_i U_{\mathrm{CZ}}^{(e)} = X_i \prod_{j\in V_i} Z_j$$

• CZは互いに可換なので，順序は重要でない.

グラフ状態の生成

◆ グラフ状態の生成方法：



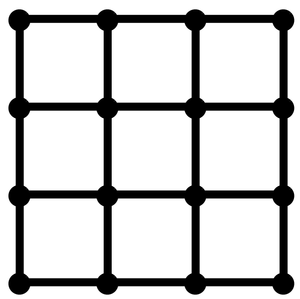
各頂点に対して $|+\rangle$ を準備．

各辺の両端のqubitにCZを作用させる．

$$\prod_{e \in E} U_{\mathrm{CZ}}^{(e)} X_i U_{\mathrm{CZ}}^{(e)} = X_i \prod_{j \in V_i} Z_j$$

• CZは互いに可換なので，順序は重要でない．

• degree（次数）が有限のグラフ（一直線，正方格子など）の場合は constant depth の回路で生成できる．

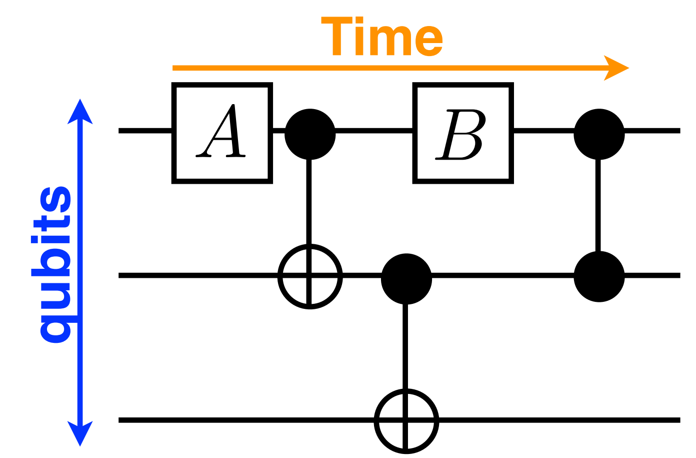


MBQC

Measurement-based quantum computation

（One-way quantum computation = 一方向量子計算, Cluster model）

Raussendorf-Briegel PRL **86** 910 (2001);
Raussendorf-Browne-Briegel PRA **68** 022312 (2003).



量子回路モデル

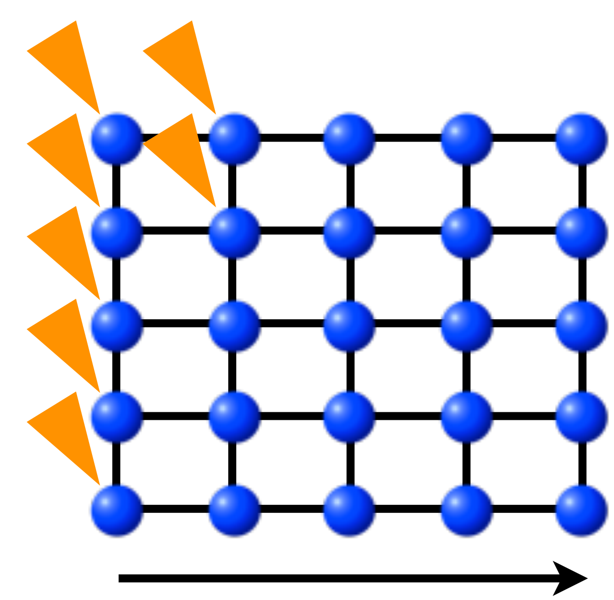
多体エンタングル状態

+

adoptive single-qubit measurement
(情報の読み出し)

=

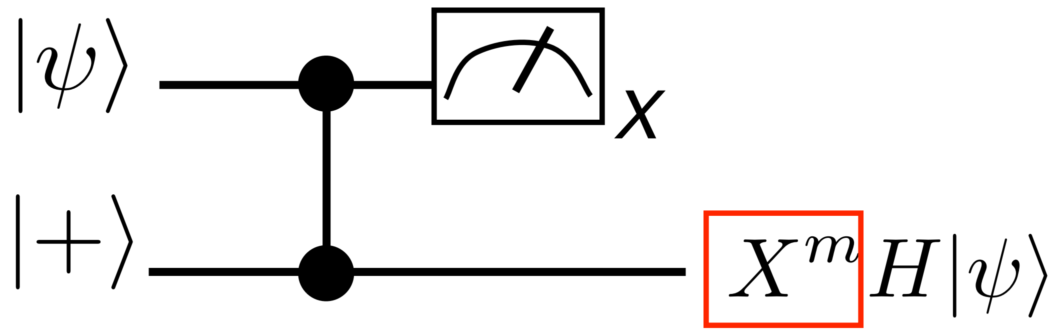
universal QC



2D cluster state

MBQC(Hadamard gate)

- 1-bit teleportation： Zhou-Leung-Chuang, Phys. Rev. A 62,052316 (2000).

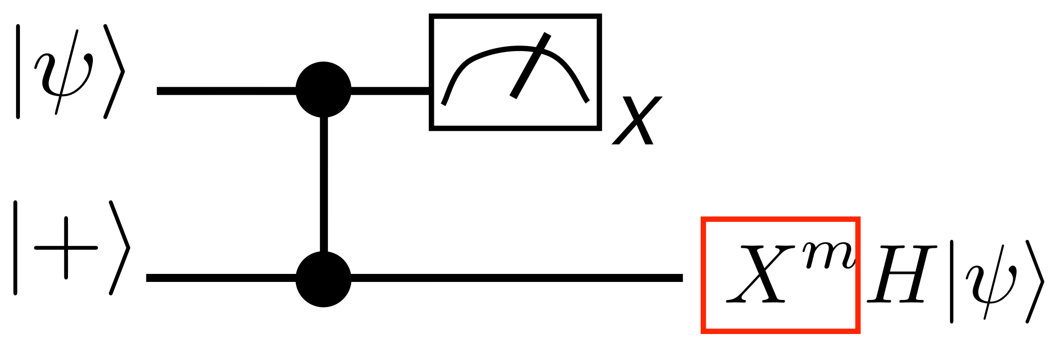


測定結果に依存してつく

“Pauli byproduct”

MBQC(Hadamard gate)

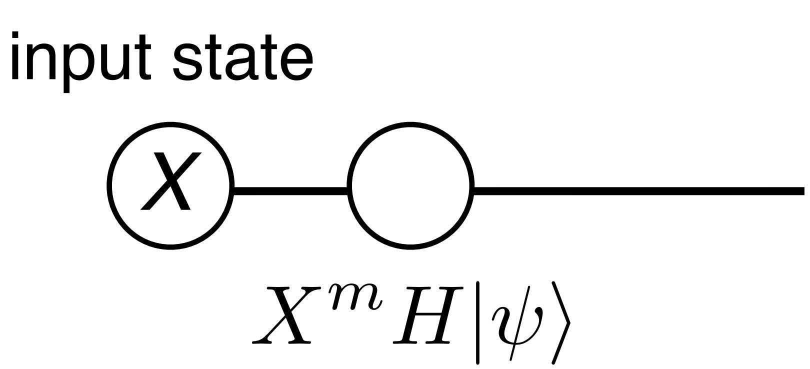
- 1-bit teleportation： Zhou-Leung-Chuang, Phys. Rev. A 62,052316 (2000).



測定結果に依存してつく

“Pauli byproduct”

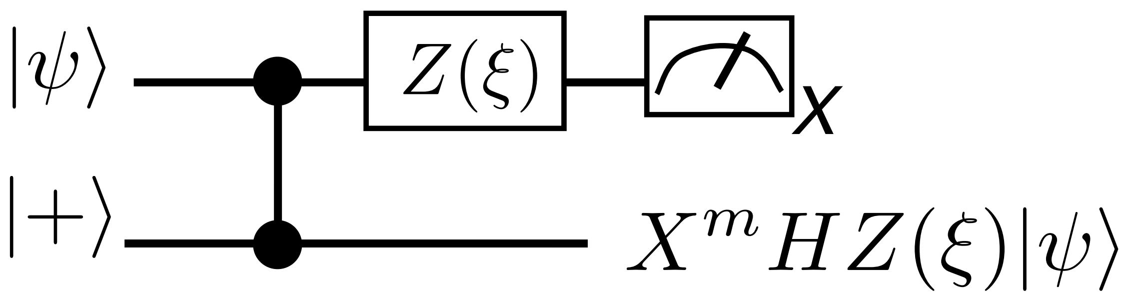
- 1D cluster上でのX基底の測定：



MBQC: single-qubit gate

一次元cluster状態に対して $Z(\xi) = e^{-i\xi Z/2}$ を作用させて

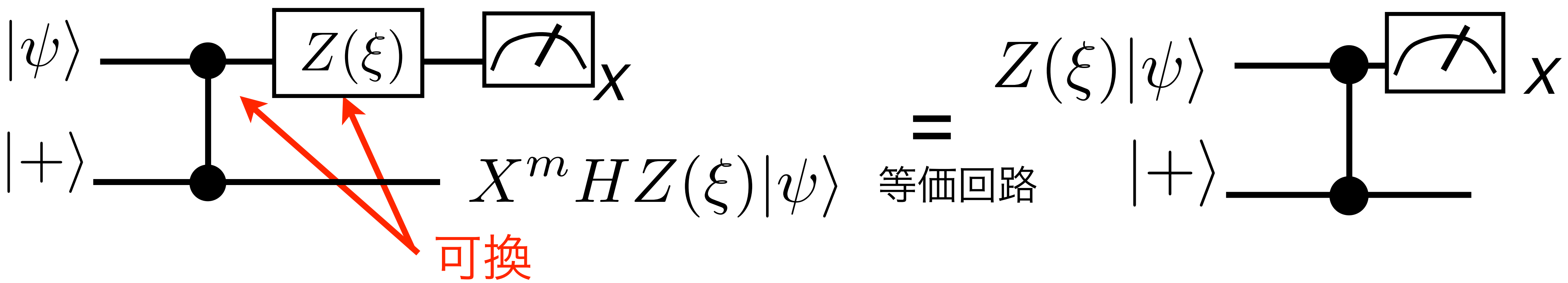
X 基底で測定→ $\{|0\rangle \pm e^{i\xi}|1\rangle\}$ 基底での測定 .



MBQC: single-qubit gate

一次元cluster状態に対して $Z(\xi) = e^{-i\xi Z/2}$ を作用させて

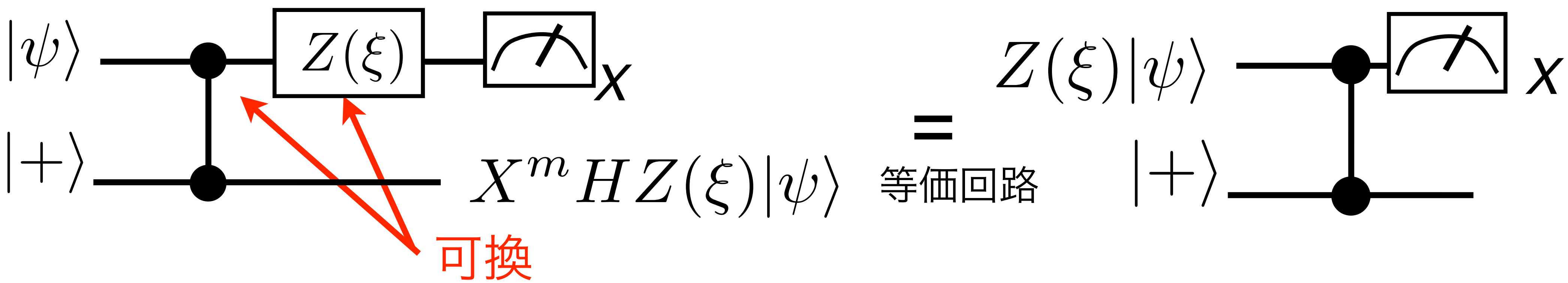
X 基底で測定→ $\{|0\rangle \pm e^{i\xi}|1\rangle\}$ 基底での測定 .

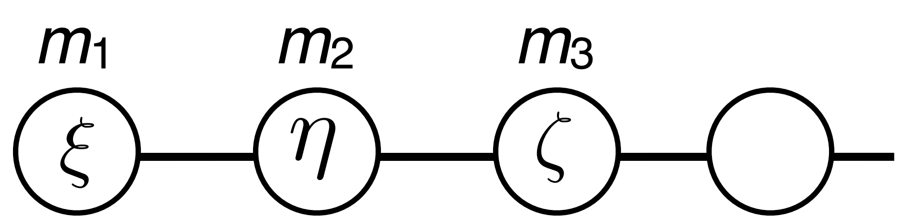


MBQC: single-qubit gate

一次元cluster状態に対して $Z(\xi)=e^{-i\xi Z/2}$ を作用させて

X 基底で測定→ $\{|0\rangle \pm e^{i\xi}|1\rangle\}$ 基底での測定 .



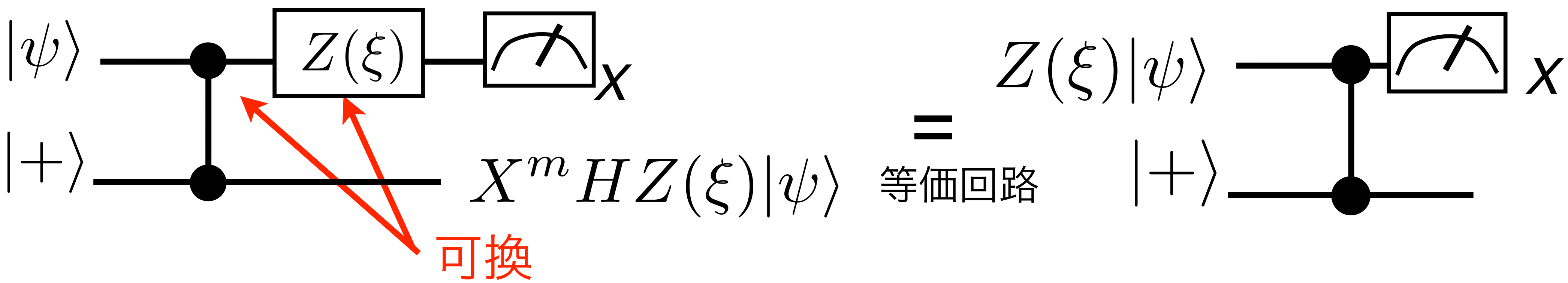


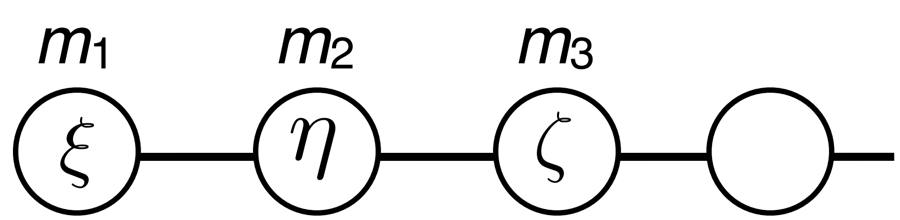
$$
\begin{aligned}
&X^{m_3}HZ(\zeta)X^{m_2}HZ(\eta)X^{m_1}HZ(\xi)|\psi\rangle \\
&= X^{m_3+m_1}Z^{m_2}HZ\big((-1)^{m_2}\zeta\big)HZ\big((-1)^{m_1}\eta\big)HZ(\xi)|\psi\rangle
\end{aligned}
$$

MBQC: single-qubit gate

一次元cluster状態に対して $Z(\xi)=e^{-i\xi Z/2}$ を作用させて

X 基底で測定→ $\{|0\rangle \pm e^{i\xi}|1\rangle\}$基底での測定 .





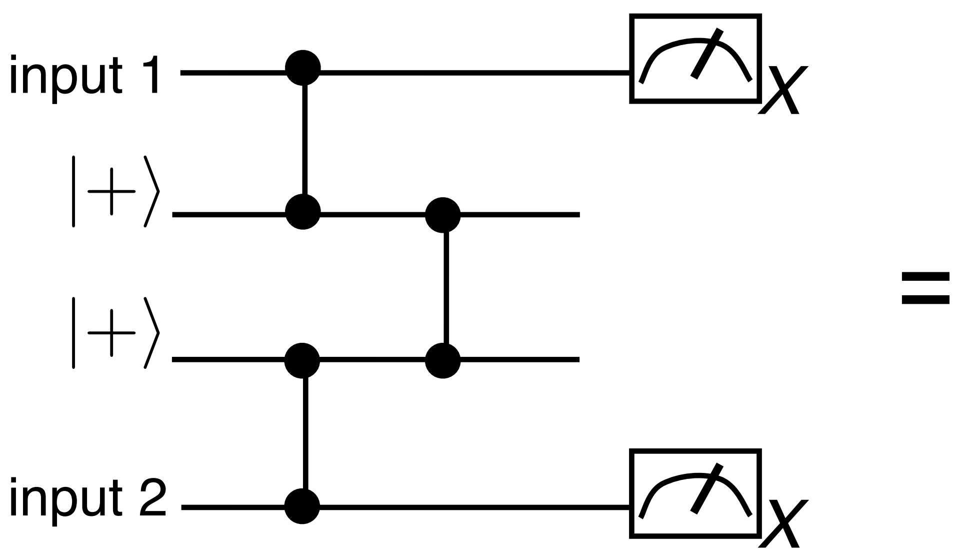
$$X^{m_3}HZ(\zeta)X^{m_2}HZ(\eta)X^{m_1}HZ(\xi)|\psi\rangle$$
$$=X^{m_3+m_1}Z^{m_2}HZ\big((-1)^{m_2}\zeta\big)HZ\big((-1)^{m_1}\eta\big)HZ(\xi)|\psi\rangle$$

測定結果に従って, $\eta=(-1)^{m_1}\tilde{\eta}, \zeta=(-1)^{m_2}\tilde{\zeta}$, とすれば,

測定結果によらず $HZ(\tilde{\zeta})\underbrace{HZ(\tilde{\eta})H}_{=X(\tilde{\eta})}Z(\xi)|\psi\rangle$ を作用させれる.

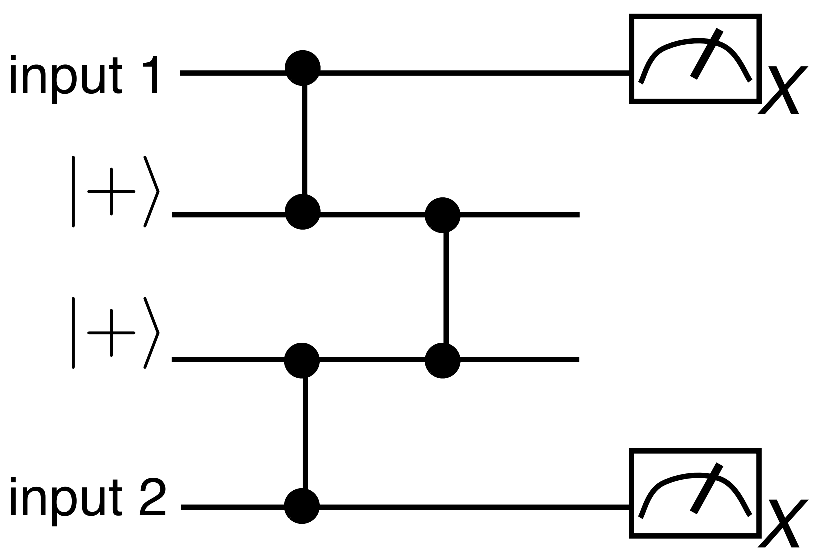
MBQC: two-qubit gate

- Gate teleportation : D. Gottesman and I. L. Chuang, Nature (London) **402**, 390 (1999).

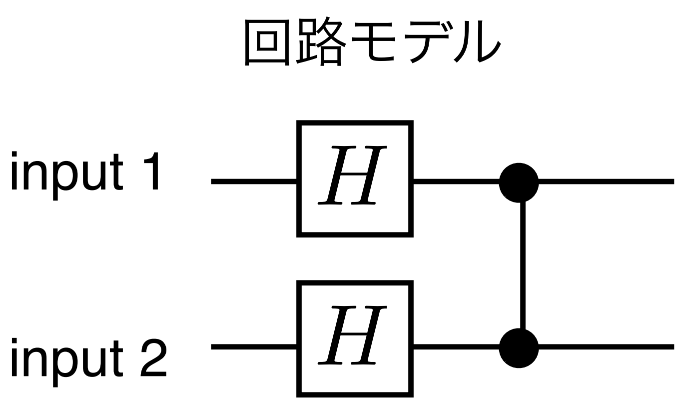


MBQC: two-qubit gate

- Gate teleportation：D. Gottesman and I. L. Chuang, Nature (London) **402**, 390 (1999).

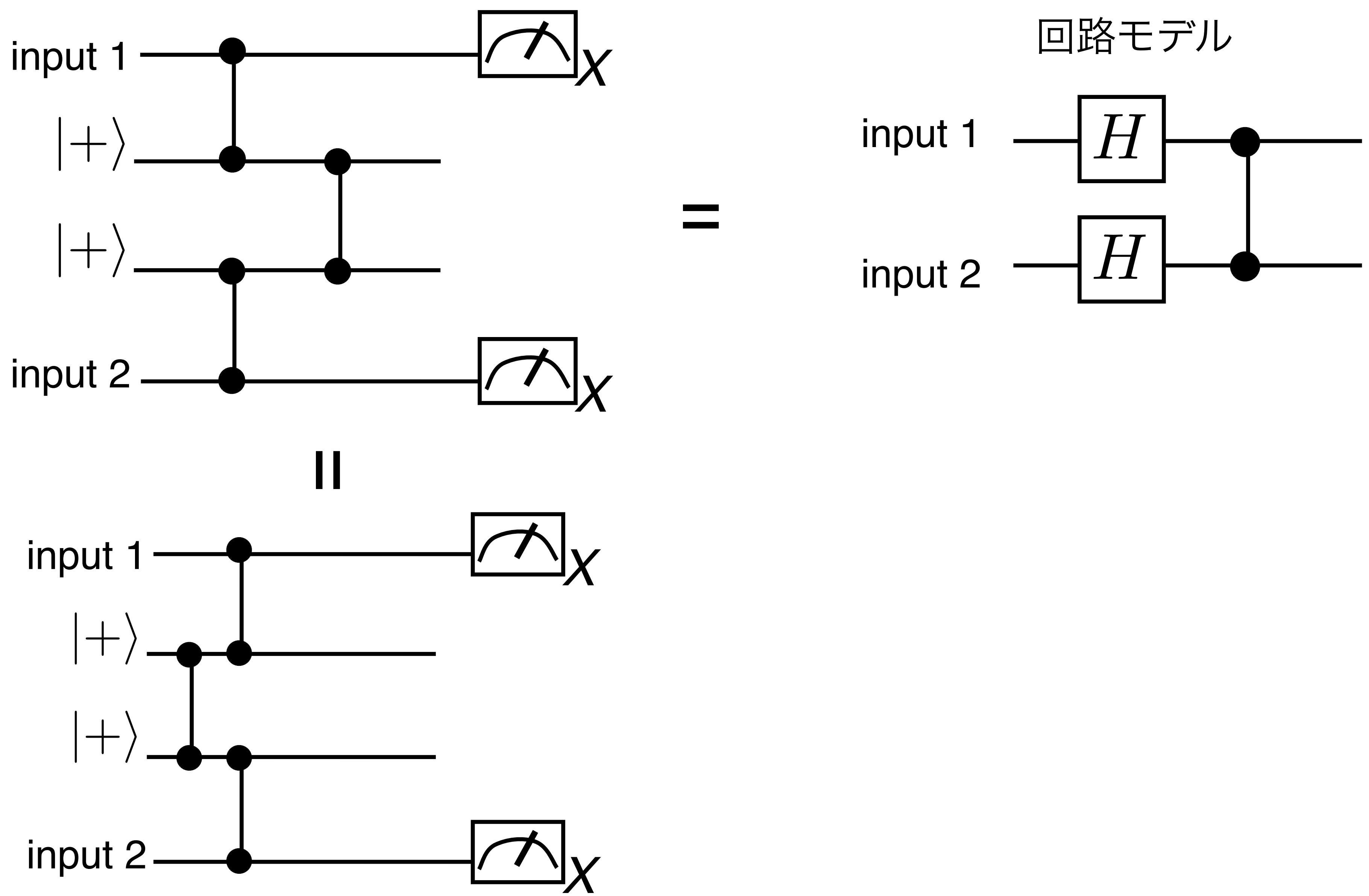


=



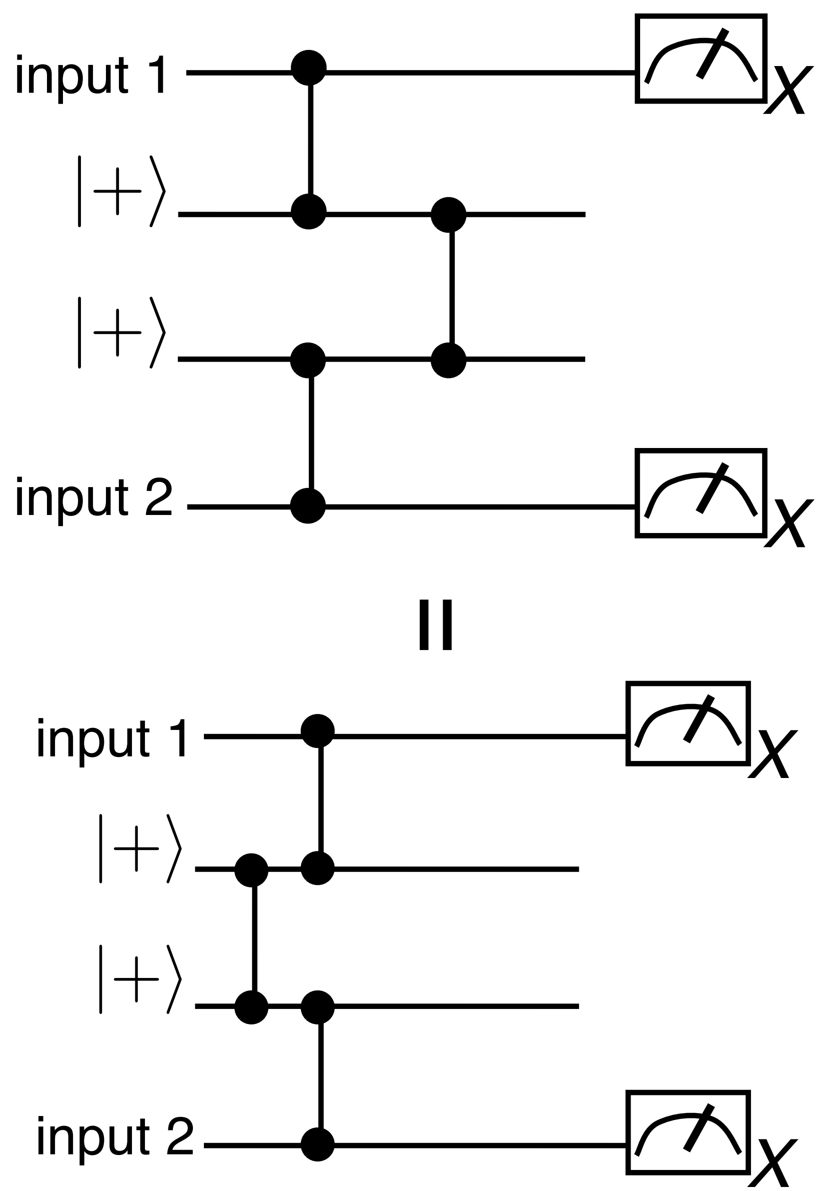
MBQC: two-qubit gate

- Gate teleportation：D. Gottesman and I. L. Chuang, Nature (London) **402**, 390 (1999).

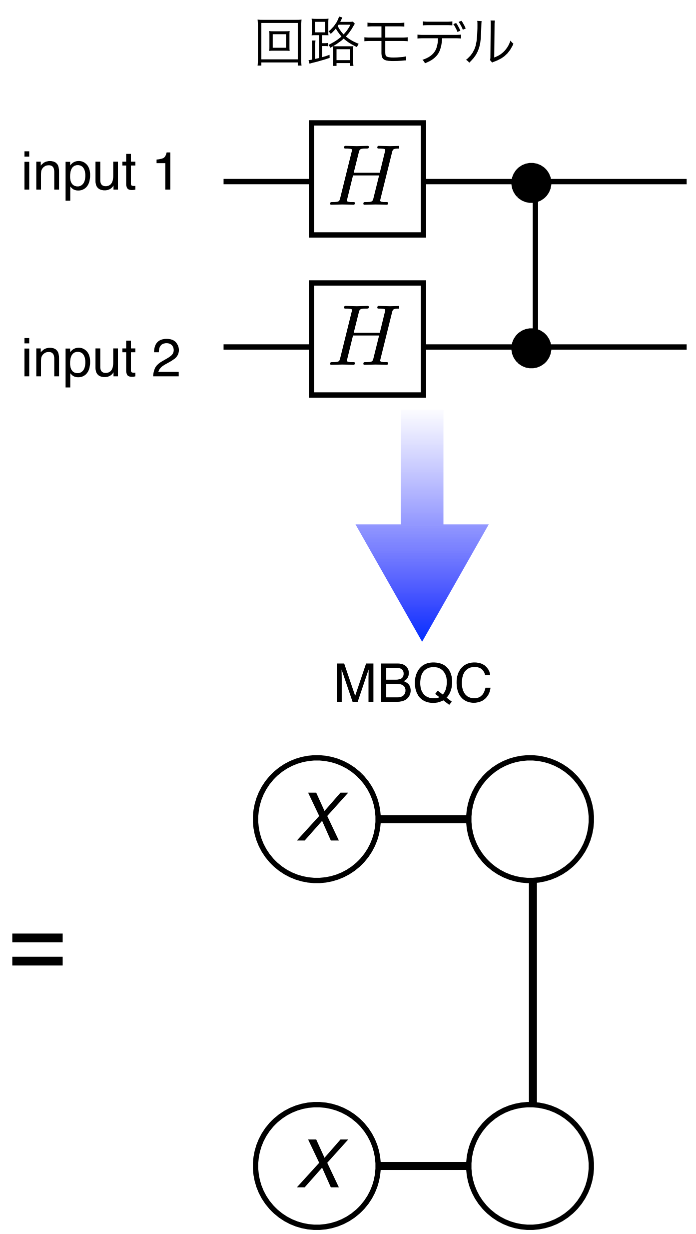


MBQC: two-qubit gate

- Gate teleportation：D. Gottesman and I. L. Chuang, Nature (London) **402**, 390 (1999).

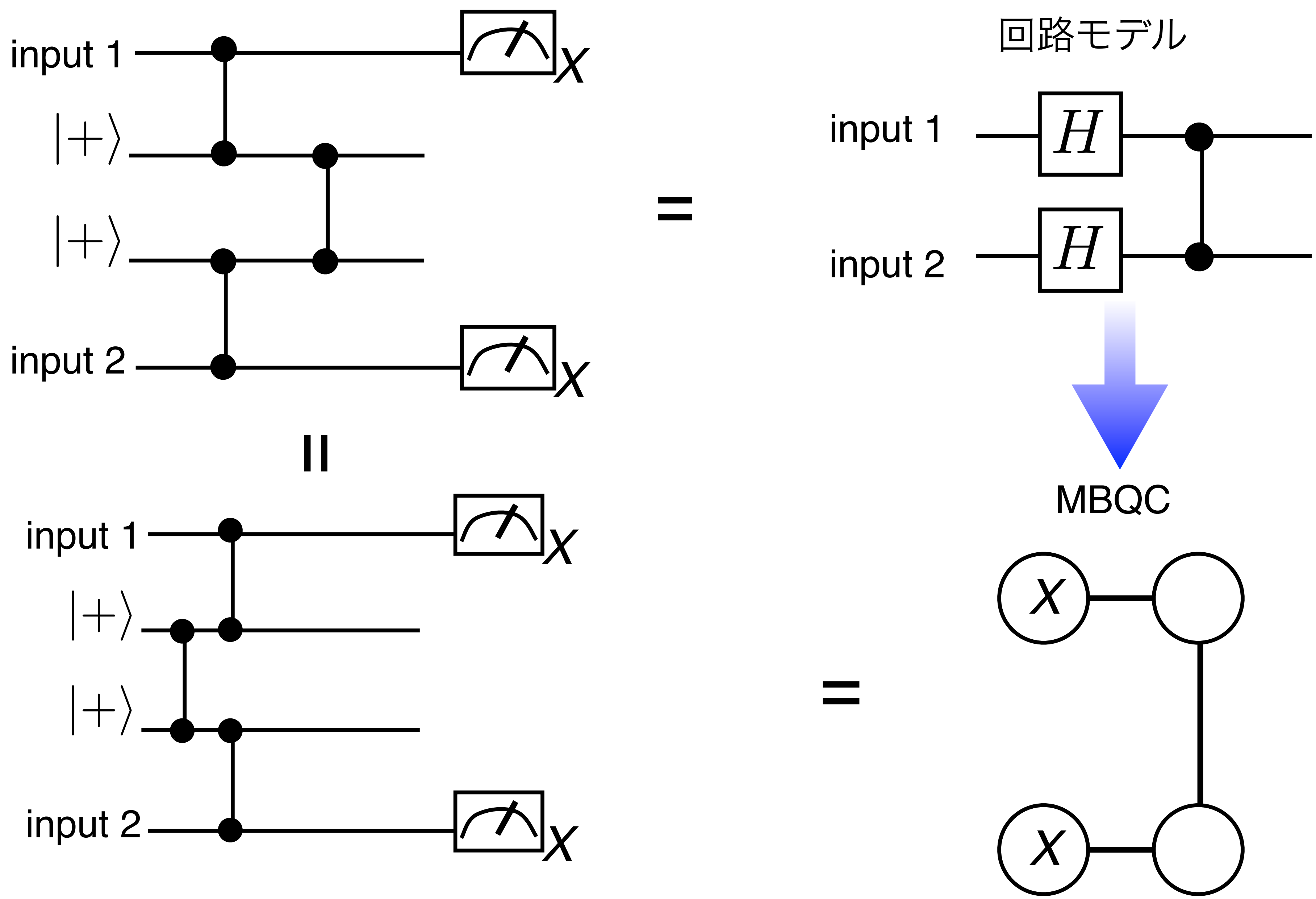


=



MBQC: two-qubit gate

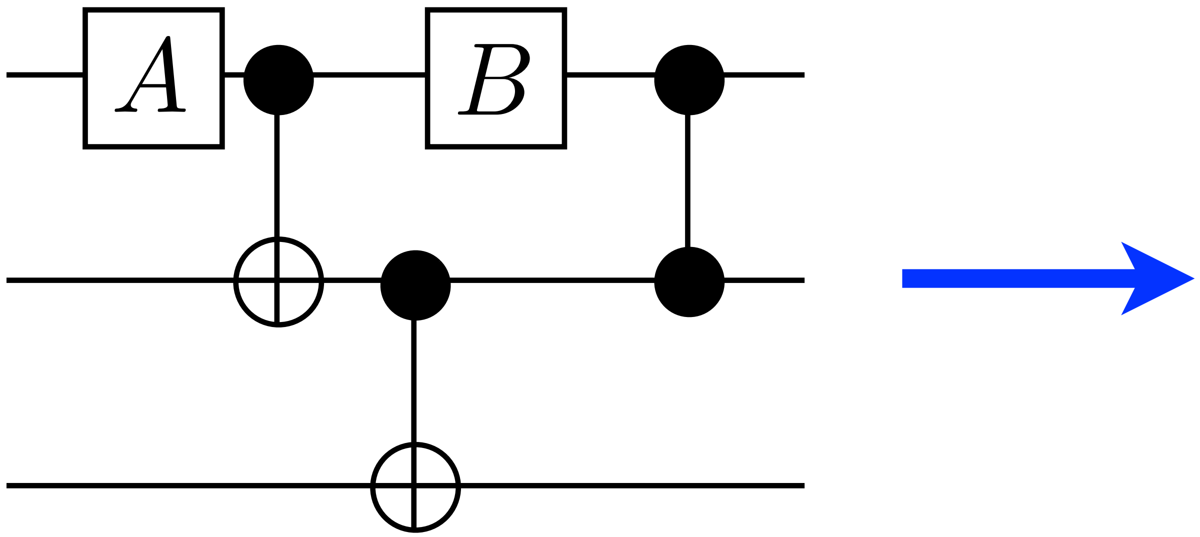
- Gate teleportation：D. Gottesman and I. L. Chuang, Nature (London) **402**, 390 (1999).



（byproductの計算）

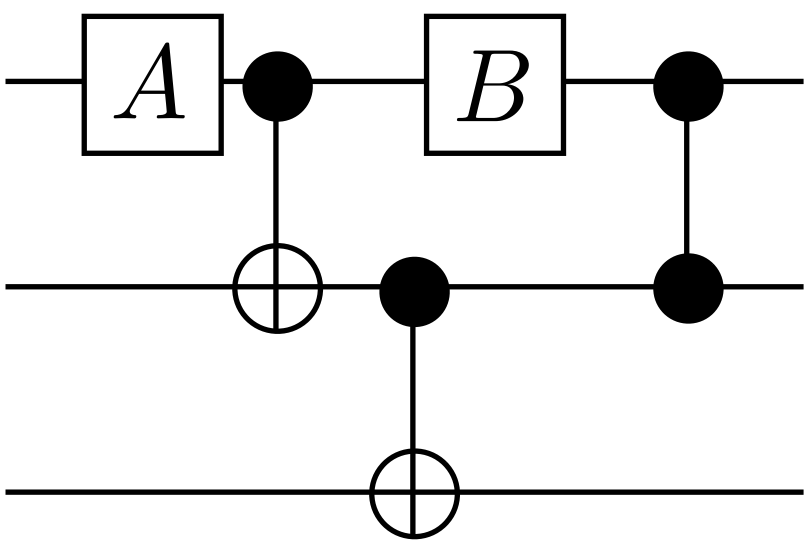
MBQC: universal QC

実行したい量子計算

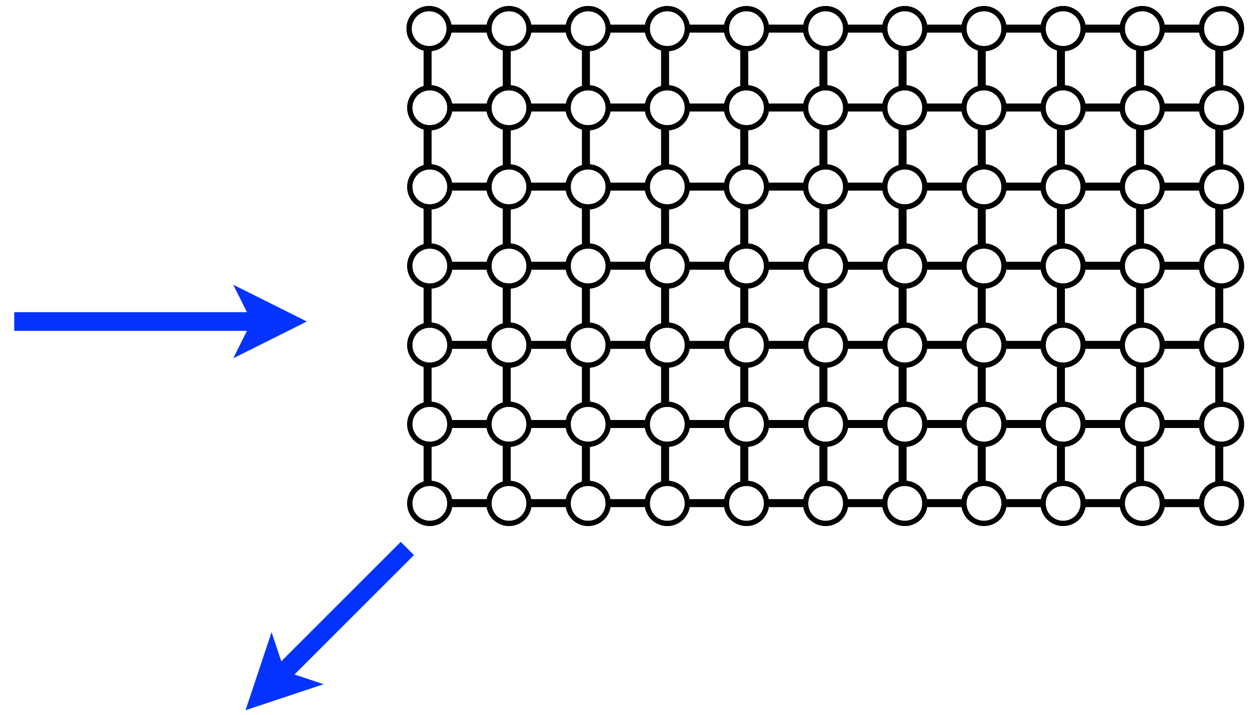


MBQC: universal QC

実行したい量子計算

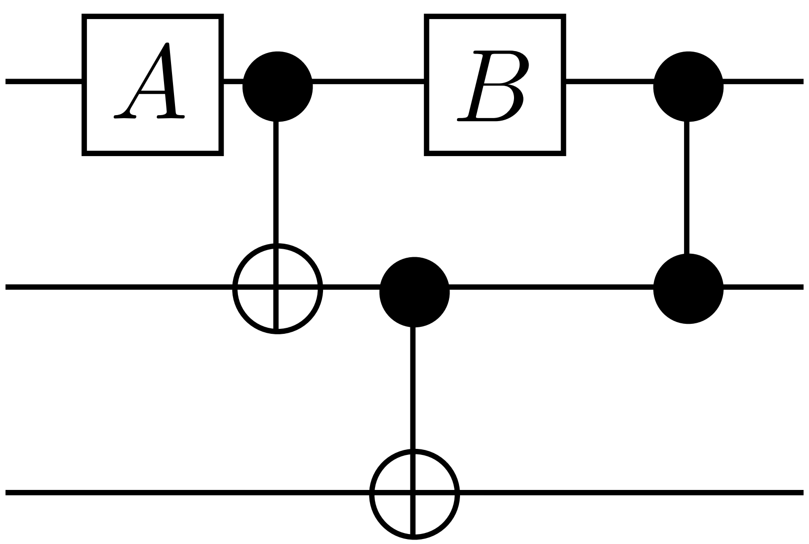


2Dクラスター状態

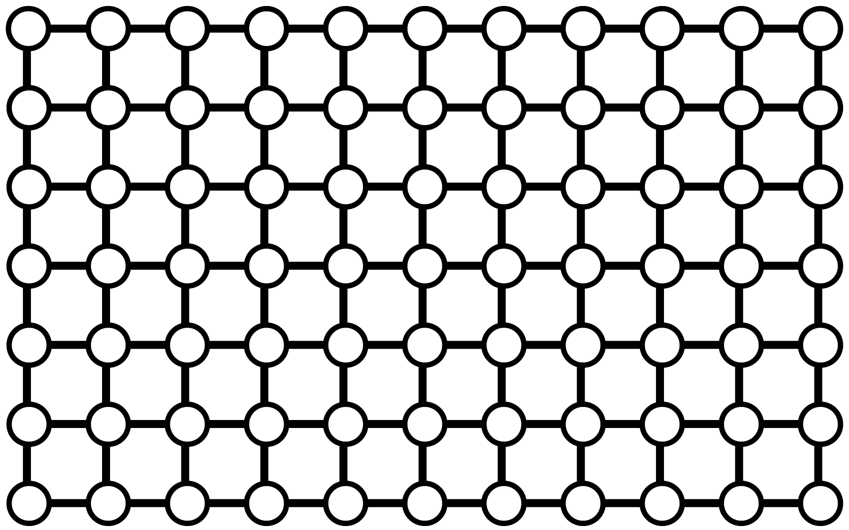


MBQC: universal QC

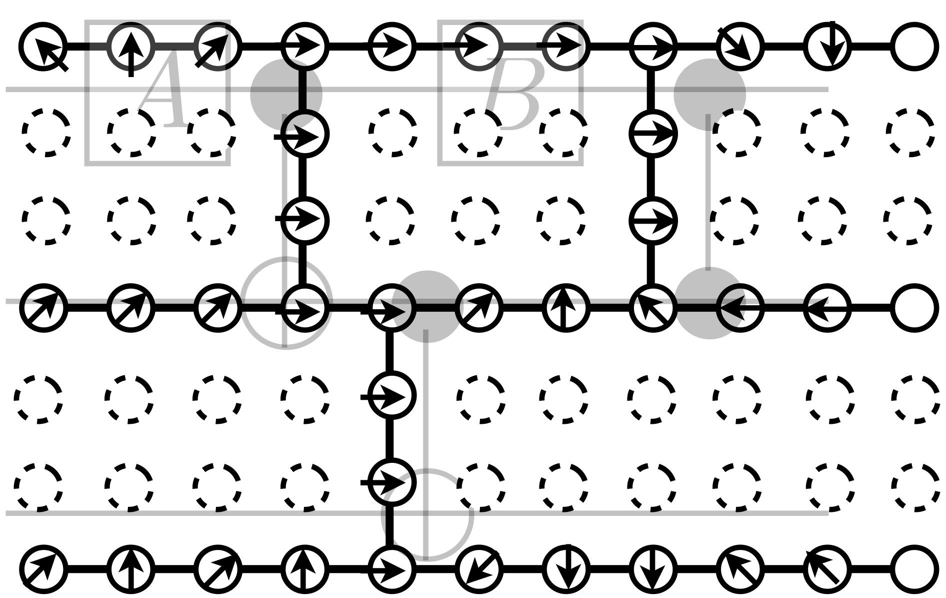
実行したい量子計算



2Dクラスター状態



MBQC！

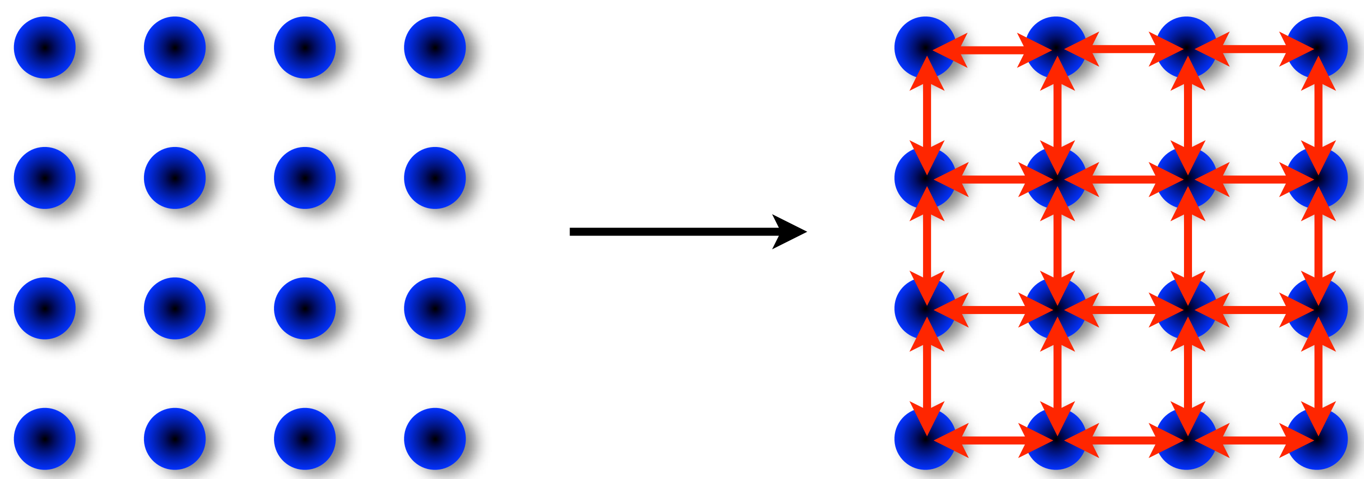


測定

グラフ状態の生成（Ising型相互作用）

$U_{\mathrm{CZ}}^{(1,2)} = e^{-i\pi/4(Z_1 Z_2 - Z_1 - Z_2 - I)}$であることを利用して，Ising型相互作用で状態生成 ← 光格子上の中性原子などに適している．

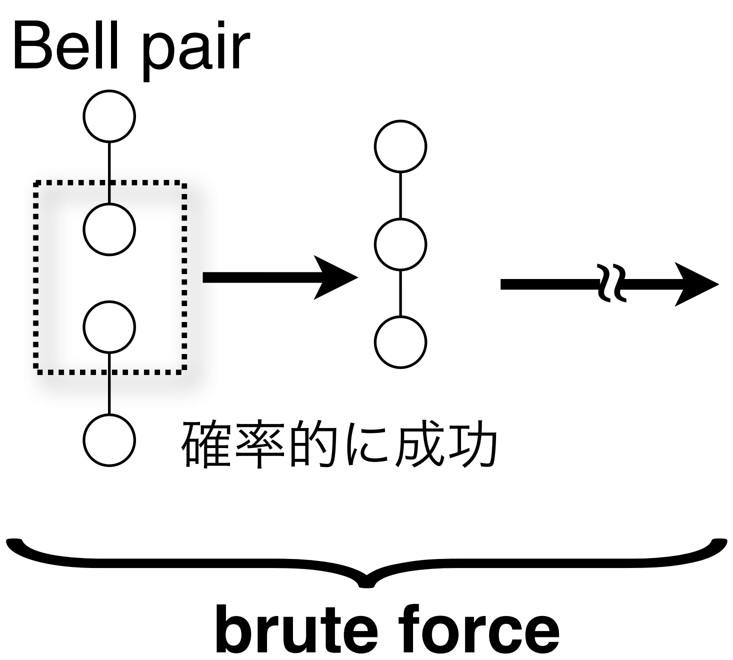
$|+\rangle^{\otimes n}$



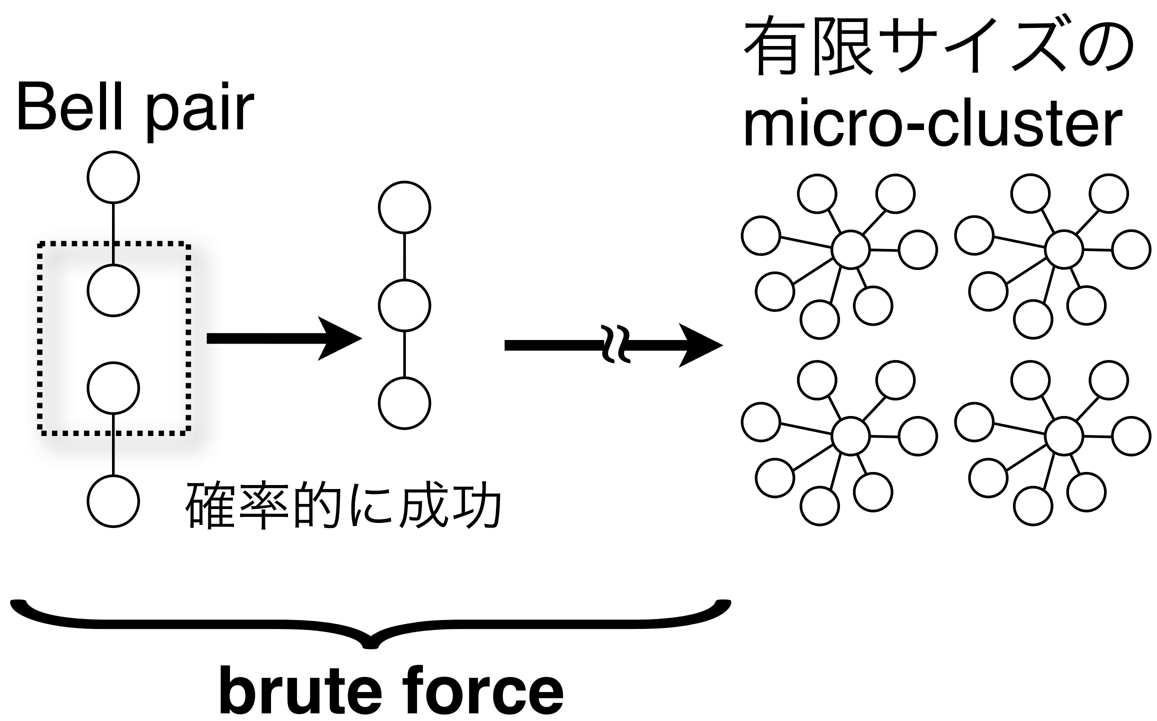
オリジナル論文では，光格子上の中性原子を意識している．

Briegel-Raussendorf, Phys. Rev. Lett. **86**, 910 (2001).
Raussendorf-Briegel PRL **86** 910 (2001);
Raussendorf-Browne-Briegel PRA **68** 022312 (2003).

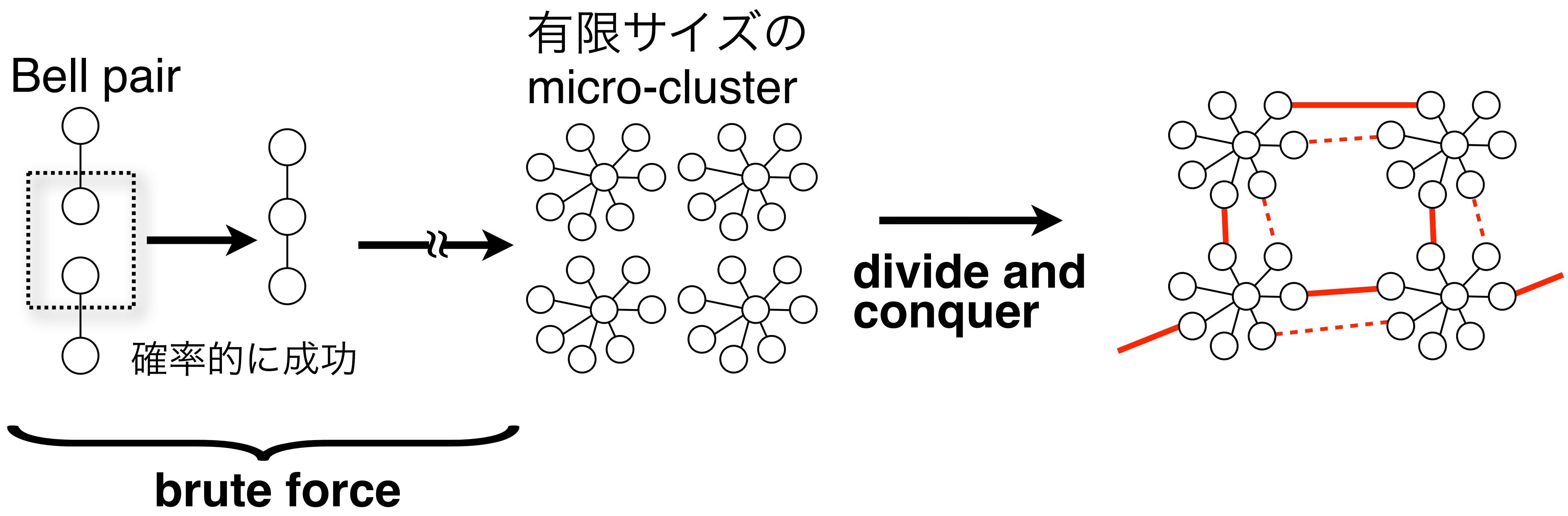
グラフ状態の生成（確率的操作）



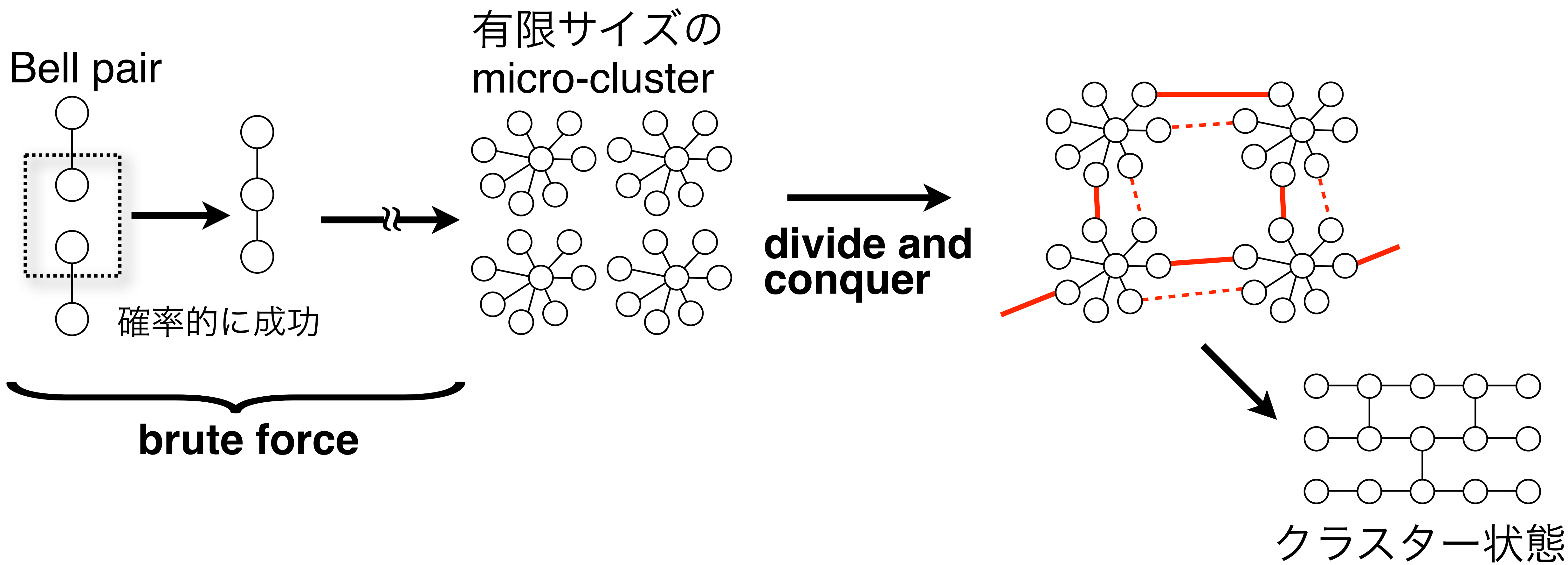
グラフ状態の生成（確率的操作）



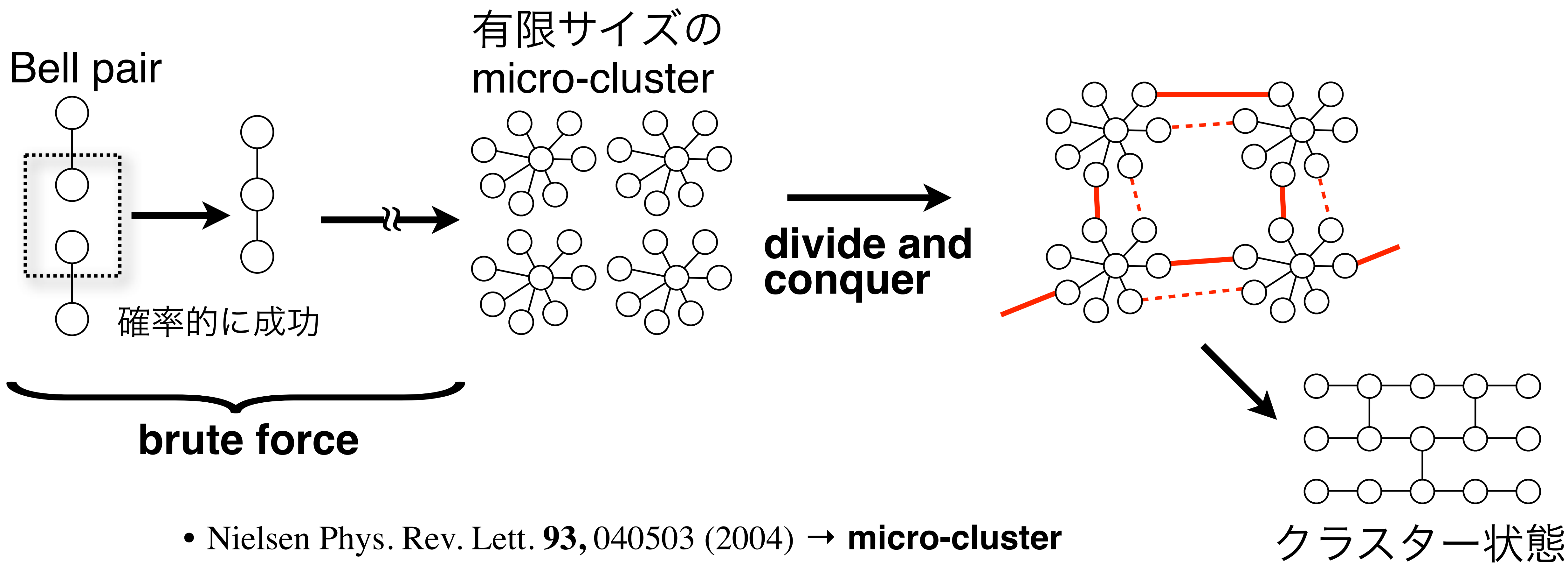
グラフ状態の生成（確率的操作）



グラフ状態の生成（確率的操作）



グラフ状態の生成（確率的操作）



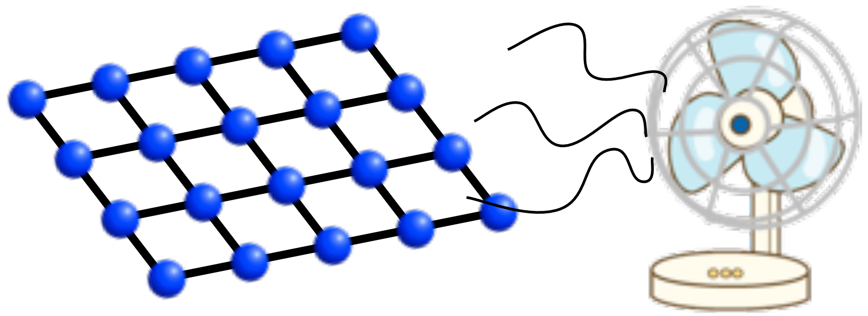
- Nielsen Phys. Rev. Lett. **93,** 040503 (2004) → **micro-cluster**
- Yoran-Reznik Phys. Rev. Lett. **91,** 037903 (2003)
- Browne-Rudolph Phys. Rev. Lett. **95**, 010501 (2005)→ **fusion gate**
- Duan-Raussendorf Phys. Rev. Lett. **95**, 080503 (2005) → **cross-strategy**
- K. Kieling, T. Rudolph, and J. Eisert Phys. Rev. Lett. **99**, 130501 (2007) → **percolation**
- Matsuzaki-Benjamin-Fitzsimons, Phys. Rev. Lett. **104**, 050501 (2010) → **tree graph**

→ 光子を用いた entangling operationに適している.

グラフ状態の生成（ハミルトニアン）

時間発展の定常状態として得られればロバスト

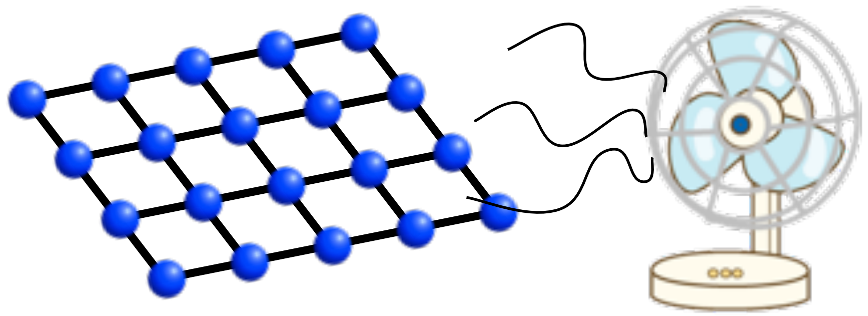
→ 基底状態や熱平行状態



グラフ状態の生成（ハミルトニアン）

時間発展の定常状態として得られればロバスト

→ 基底状態や熱平行状態



◆ stabilizer (cluster) Hamiltonian:

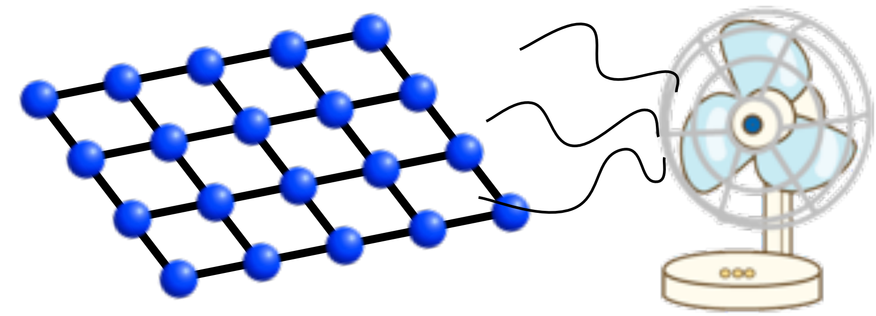
$$H_{\mathrm{fc}} = -J \sum_i K_i$$

基底状態が cluster state.

グラフ状態の生成（ハミルトニアン）

時間発展の定常状態として得られればロバスト

→ 基底状態や熱平行状態



◆ stabilizer (cluster) Hamiltonian:

$$H_{\rm fc} = -J \sum_i K_i$$

基底状態が cluster state.

$U_{\mathcal{L}} \equiv \prod_{\langle i,j \rangle \in E} U_{\rm CZ}^{(i,j)}$ を用いて変換すると，$U_{\mathcal{L}} H_{\rm fc} U_{\mathcal{L}} = -J \sum_i X_i$.

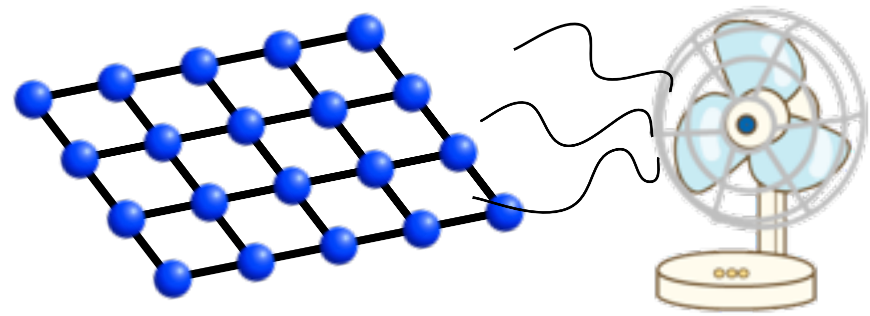
→自由模型なので熱平衡状態が計算できる.

Raussendorf-Bravyi-Harrington, PRA **71**, 062313 (2005);
Barrett *et al*., PRA **80**, 062328 (2009).

グラフ状態の生成（ハミルトニアン）

時間発展の定常状態として得られればロバスト

→ 基底状態や熱平行状態



◆ stabilizer (cluster) Hamiltonian:

$$H_{\mathrm{fc}} = -J \sum_i K_i$$

基底状態が cluster state.

$U_{\mathcal{L}} \equiv \prod_{\langle i,j \rangle \in E} U_{\mathrm{CZ}}^{(i,j)}$ を用いて変換すると，$U_{\mathcal{L}} H_{\mathrm{fc}} U_{\mathcal{L}} = -J \sum_i X_i$.

→自由模型なので熱平衡状態が計算できる.

Raussendorf-Bravyi-Harrington, PRA **71**, 062313 (2005);
Barrett *et al*., PRA **80**, 062328 (2009).

→多体効果を積極的に利用できないだろうか？

K. Fujii, Y. Nakata, M. Ohzeki, M. Murao, arXiv:1209.1265

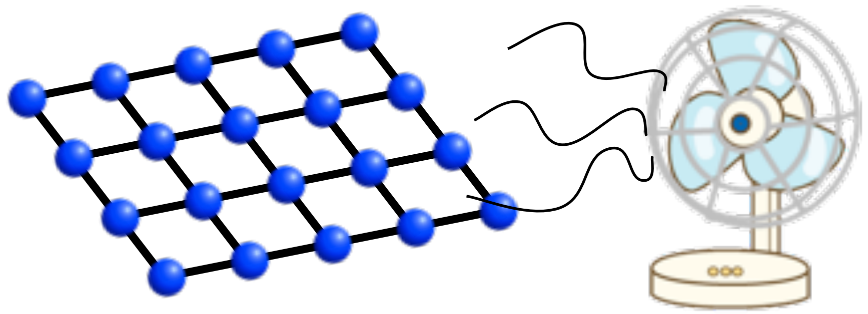
グラフ状態の生成（ハミルトニアン）

グラフ状態の生成（ハミルトニアン）

◆２体相互作用ハミルトニアン

クラスター状態は局所２体ハミルトニアンの基底状態になり得ない.

Nielsen, Rep. Math. Phys. **57,** 147 (2006);
Van den Nest *et al*., PRA **77,** 012301 (2008):
Chen *et al*., PRA **83,** 050301 (2011).

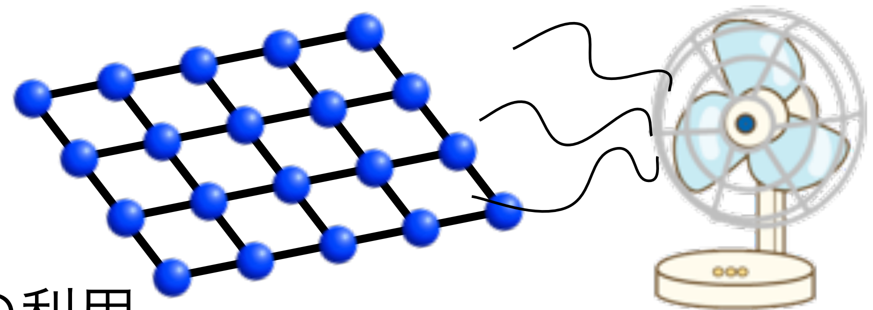


グラフ状態の生成（ハミルトニアン）

◆２体相互作用ハミルトニアン

クラスター状態は局所２体ハミルトニアンの基底状態になり得ない.

Nielsen, Rep. Math. Phys. **57,** 147 (2006);
Van den Nest *et al*., PRA **77,** 012301 (2008):
Chen *et al*., PRA **83,** 050301 (2011).



→**高次元粒子（qudit）**多体エンタングル状態の利用

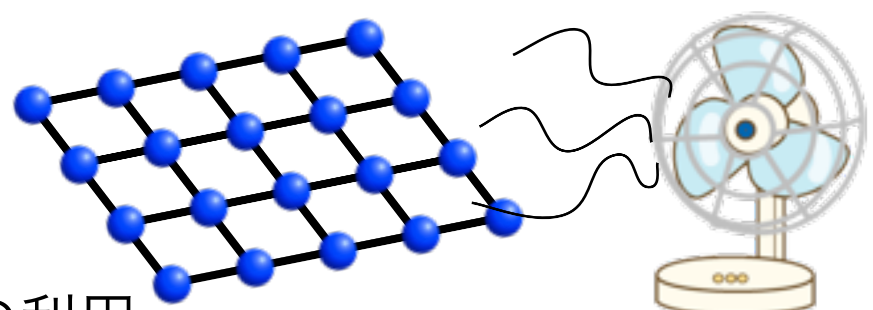
Gross-Eisert, PRL **98,** 220503 (2007); Gross *et al*. PRA **76,** 052315 (2007).

グラフ状態の生成（ハミルトニアン）

◆２体相互作用ハミルトニアン

クラスター状態は局所２体ハミルトニアンの基底状態になり得ない.

Nielsen, Rep. Math. Phys. **57,** 147 (2006);
Van den Nest *et al*., PRA **77,** 012301 (2008):
Chen *et al*., PRA **83,** 050301 (2011).



→**高次元粒子（qudit）**多体エンタングル状態の利用

Gross-Eisert, PRL **98,** 220503 (2007); Gross *et al*. PRA **76,** 052315 (2007).

dimension (spin)	model	resource	
d=6 (spin-5/2)	Tri-cluster by Chen *et al*., PRL '09	ground state	
d=4 (spin-3/2)	quasi 1D AKLT by Cai *et al*., PRA '10	ground state	
d=4 (spin-3/2)	2D AKLT by Miyake, Ann. Phys. '11	ground state	
d=4 (spin-3/2)	2D AKLT by Wei *et al*., PRL '11	ground state	
d=4 (spin-3/2)	2D honeycomb by Li *et al*., PRL '11	ground state	
d=5 (spin-2)	3D lattice by Li *et al*., PRL '11	thermal state	(T=0.21Δ)
d=4 (spin-3/2)	3D lattice by Fujii-Morimae, PRA '12	thermal state	(T=0.18Δ)

第三章: 量子誤り訂正とフォールトトレランス

ここで勉強する内容：

量子誤り訂正とは？

Stabilizer形式での量子誤り訂正符号の記述,

フォールトトレラント量子計算

誤り訂正の必要性

誤り訂正の必要性

Rolf Landauer @IBM

All papers on quantum computing should carry a footnote: *“This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form take into account all possible sources of noise, unreliability and manufacturing error, and probably will not work.”* (S. Lloyd Nature 400 720 (1999))

誤り訂正の必要性

Rolf Landauer @IBM

All papers on quantum computing should carry a footnote: *"This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form take into account all possible sources of noise, unreliability and manufacturing error, and probably will not work."* (S. Lloyd Nature 400 720 (1999))

誤り訂正の必要性

Rolf Landauer @IBM

All papers on quantum computing should carry a footnote: *"This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form take into account all possible sources of noise, unreliability and manufacturing error, and probably will not work."* (S. Lloyd Nature 400 720 (1999))

例えば，．．．

アナログ古典コンピューター=無限精度の実数の加減乗除

→NP完全問題が多項式時間で解ける

by A Schönhage *"On the power of random access machines."* ICALP, 520 (1979).

誤り訂正の必要性

Rolf Landauer @IBM

All papers on quantum computing should carry a footnote: *“This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form take into account all possible sources of noise, unreliability and manufacturing error, and probably will not work.”* (S. Lloyd Nature 400 720 (1999))

例えば, . . .

アナログ古典コンピュータ=無限精度の実数の加減乗除

→NP完全問題が多項式時間で解ける

by A Schönhage *“On the power of random access machines.”* ICALP, 520 (1979).

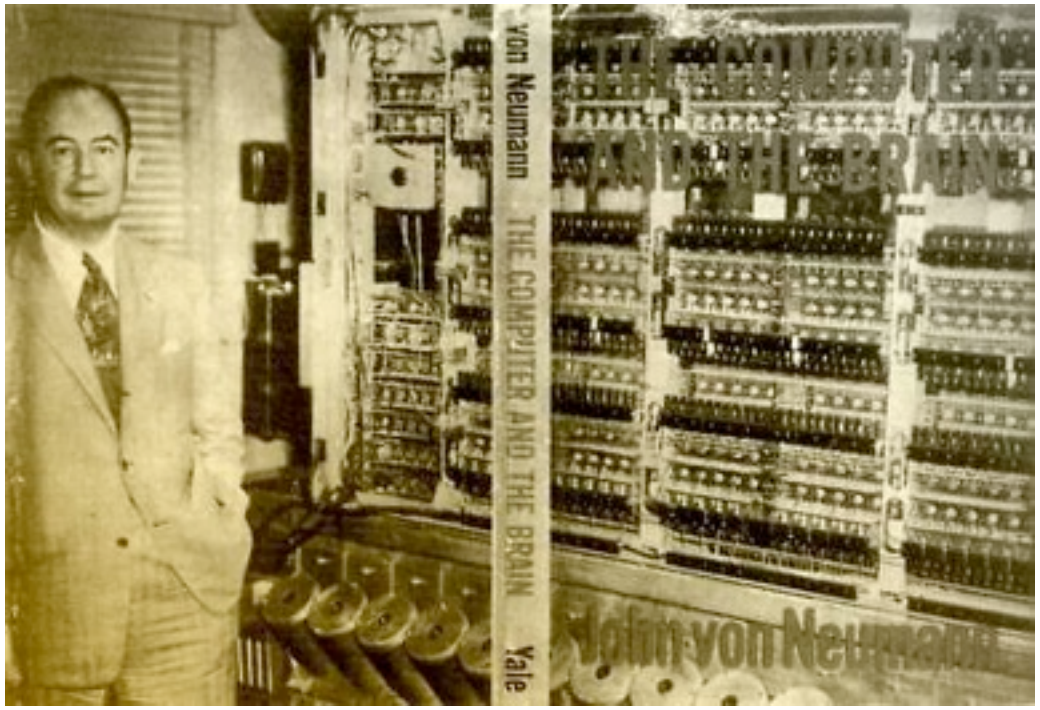
雑音の影響を考えると実現は不可能

→古典デジタルコンピュータの場合は？

古典計算機の黎明期

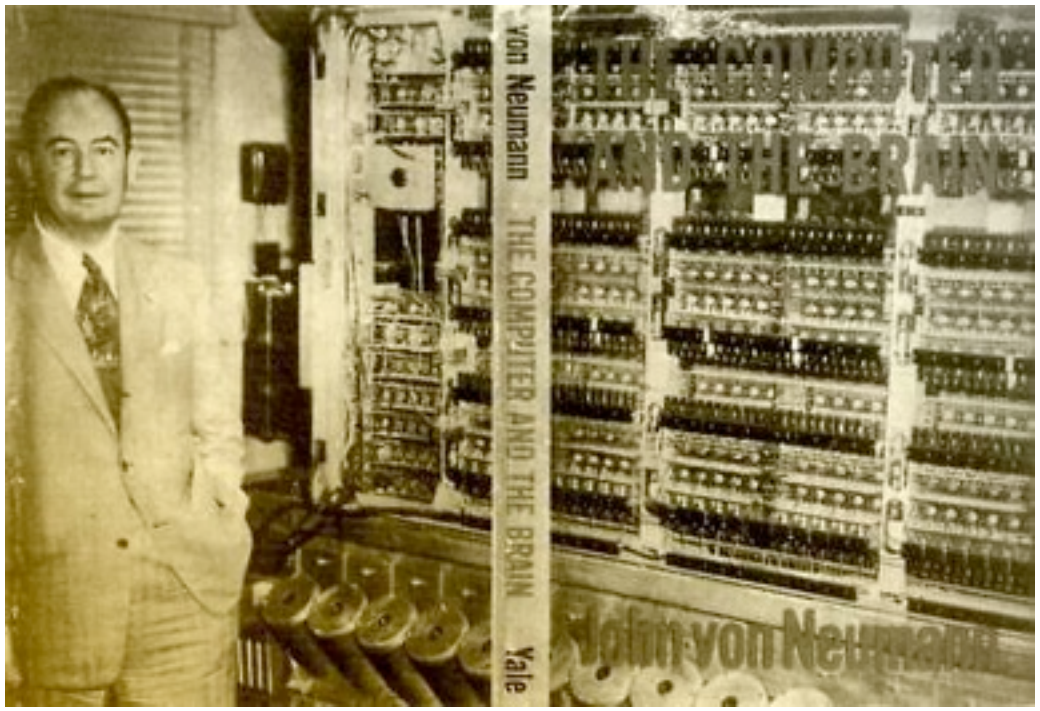
古典計算機の黎明期

EDVAC: Electric discrete variable automatic computer



古典計算機の黎明期

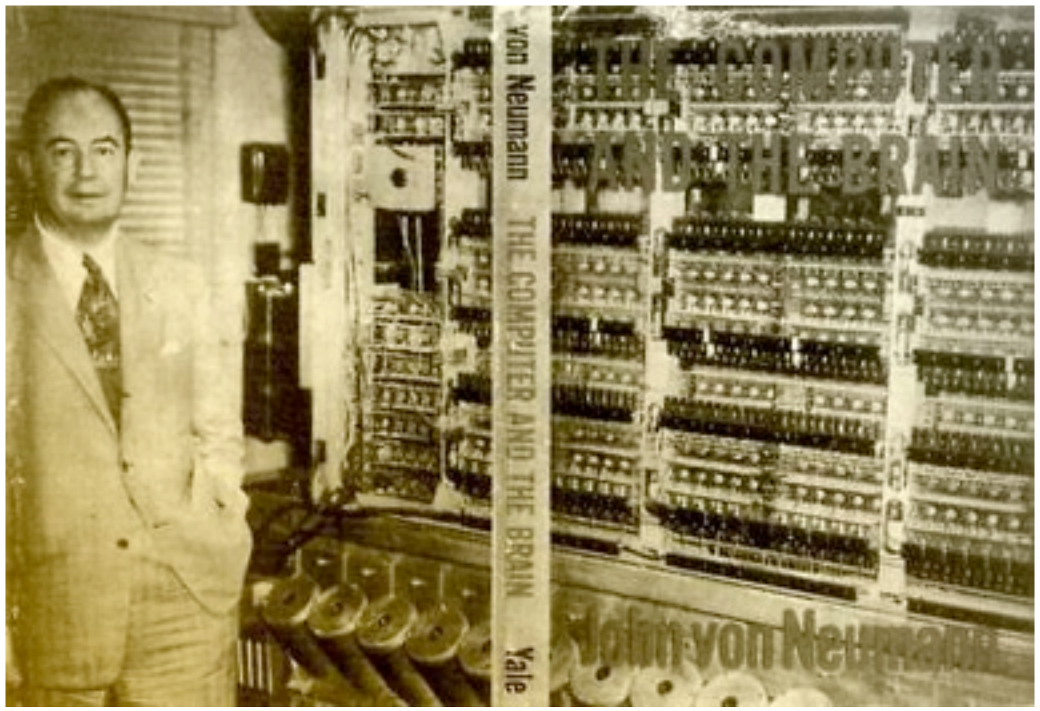
EDVAC: Electric discrete variable automatic computer



For the recognition and correction of such malfunctions **intelligent human intervention will in general be necessary**.
by J. von Neumann (1945)
"First draft of a report on the EDVAC"

古典計算機の黎明期

EDVAC: Electric discrete variable automatic computer



For the recognition and correction of such malfunctions **intelligent human intervention will in general be necessary**.

by J. von Neumann (1945)

"First draft of a report on the EDVAC"

約10年後

J. von Neumann (1956)

"Probabilistic logics and the synthesis of reliable organisms from unreliable components"

NAND-multiplexing

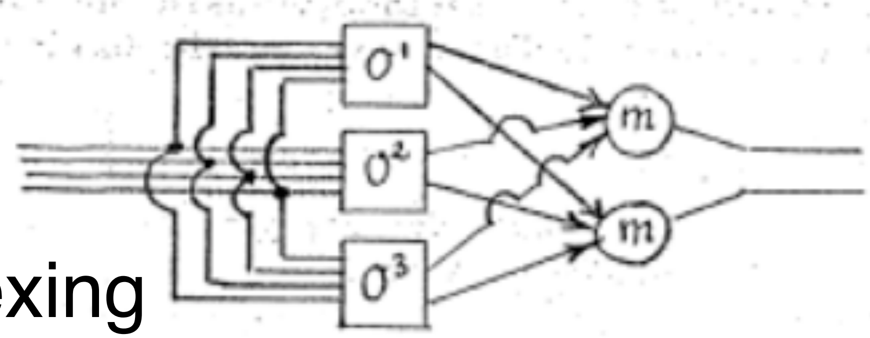
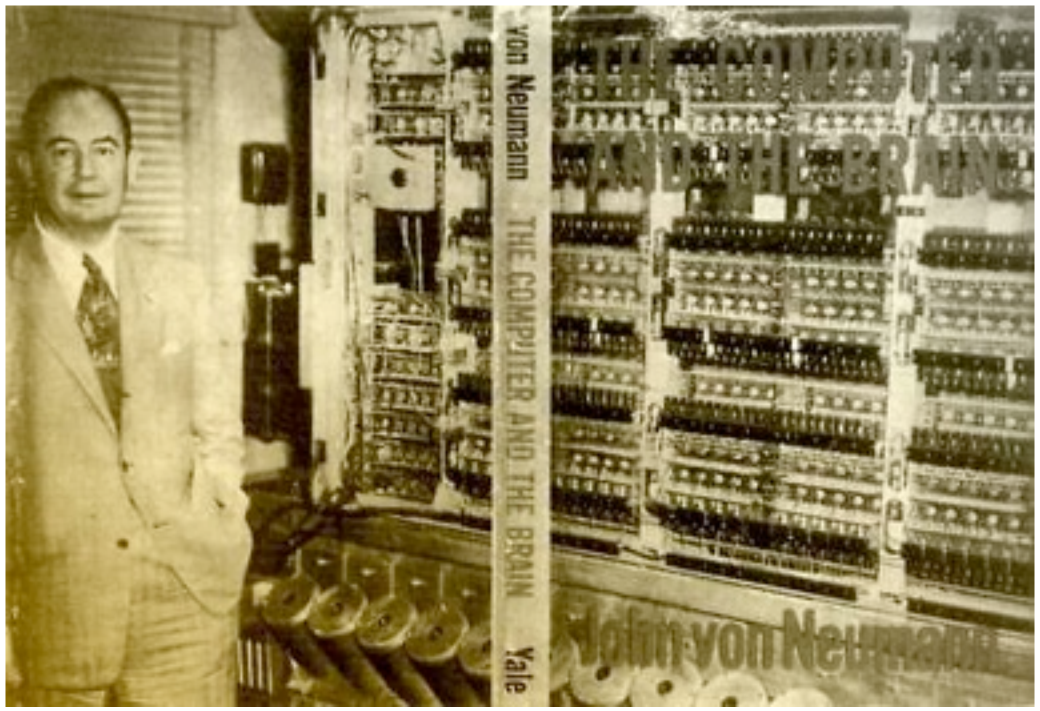


Fig. 26

古典計算機の黎明期

EDVAC: Electric discrete variable automatic computer



For the recognition and correction of such malfunctions **intelligent human intervention will in general be necessary**.
by J. von Neumann (1945)
"First draft of a report on the EDVAC"

約10年後

NAND-multiplexing

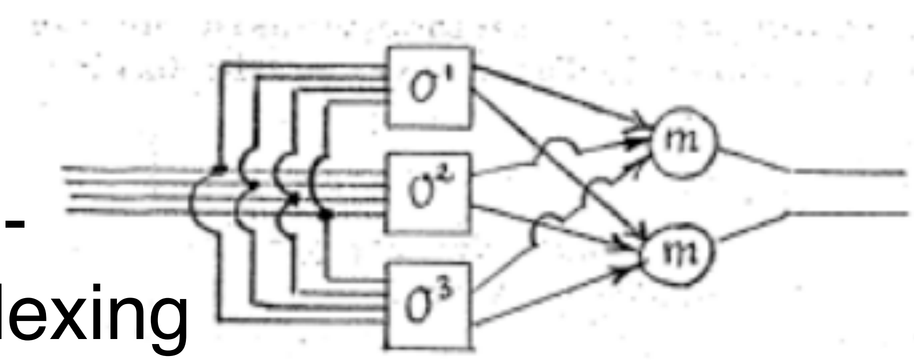


Fig. 26

J. von Neumann (1956)
"Probabilistic logics and the synthesis of reliable organisms from unreliable components"

約50年後

J. Han and P. Jonker (2003)
"A defect- and fault-tolerant architecture for nanocomputers"

古典計算機の黎明期

EDVA(

NAND-
multiple)

such
rvention

'e

量子計算における精度の保証

そもそも，計算するのが困難だから量子計算機を使う

量子計算における精度の保証

そもそも，計算するのが困難だから量子計算機を使う

→計算結果が正しいことをどう示すのか?

（human intervention?）

量子計算における精度の保証

そもそも，計算するのが困難だから量子計算機を使う

→計算結果が正しいことをどう示すのか?

（human intervention?）

→素因数分解は NP問題（多項式時間で解が正しいか検証できる）のでひとまずOK?

量子計算における精度の保証

そもそも，計算するのが困難だから量子計算機を使う

→計算結果が正しいことをどう示すのか?

（human intervention?）

→素因数分解は NP問題（多項式時間で解が正しいか検証できる）のでひとまずOK?

→例えば、NP問題ではない場合は？

Aspuru-Guzik *et al*. Science **309**, 1704 (2005)

D. Aharonov, et al. arXiv:quant-ph/0702008.

量子計算における精度の保証

そもそも，計算するのが困難だから量子計算機を使う

→計算結果が正しいことをどう示すのか?

（human intervention?）

→素因数分解は NP問題（多項式時間で解が正しいか検証できる）のでひとまずOK?

→例えば、NP問題ではない場合は？

Aspuru-Guzik *et al*. Science **309**, 1704 (2005)

D. Aharonov, et al. arXiv:quant-ph/0702008.

→汎用計算機としては，出力の精度を保証できないと．．．

量子計算における精度の保証

そもそも，計算するのが困難だから量子計算機を使う

→計算結果が正しいことをどう示すのか?
（human intervention?）

→素因数分解は NP問題（多項式時間で解が正しいか検証できる）のでひとまずOK?

→例えば、NP問題ではない場合は？

Aspuru-Guzik *et al*. Science **309**, 1704 (2005)

D. Aharonov, et al. arXiv:quant-ph/0702008.

→汎用計算機としては，出力の精度を保証できないと．．．

個々の要素に分解して全体の振る舞いに対する精度について言及できるべき．

量子誤り訂正

量子誤り訂正

・量子情報は no-cloning 定理があるため古典情報のように情報を複製することは出来ない.

$$|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$$

量子誤り訂正

・量子情報は no-cloning 定理があるため古典情報のように情報を複製することは出来ない.

$$\xcancel{|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle}$$

量子誤り訂正

・量子情報は no-cloning 定理があるため古典情報のように情報を複製することは出来ない.

$$\cancel{|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle}$$

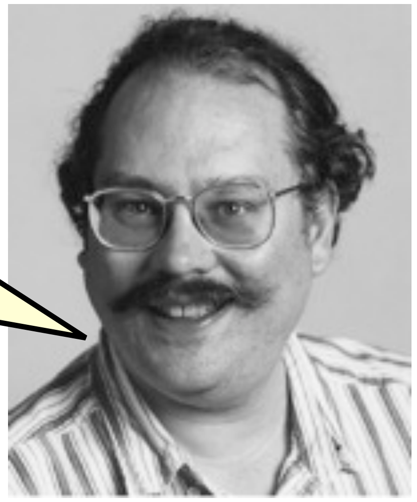
・量子ビットのエラーは連続的なもの→精確にもとの状態に戻せるのか？

$$\alpha|0\rangle + \beta|1\rangle \underset{?}{\longrightarrow} \alpha'|0\rangle + \beta'|1\rangle$$

量子誤り訂正

量子状態そのものをコピーするのではなく，基底を複数の物理量子ビットを用いて符号化すべし.

$$|0_L\rangle \rightarrow |0\rangle \otimes |0\rangle \otimes |0\rangle \qquad |1_L\rangle \rightarrow |1\rangle \otimes |1\rangle \otimes |1\rangle$$



$$\alpha|0\rangle + \beta|1\rangle \in \mathbb{C}^2 \xrightarrow{\text{encode}} \alpha|000\rangle + \beta|111\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$$

noise $(\sigma_x \otimes I \otimes I)$

$?(\alpha|100\rangle + \beta|011\rangle)$

quantum error correction

syndrome measurement

$\alpha|100\rangle + \beta|011\rangle$

recovery $\sigma_x \otimes I \otimes I$

部分空間内の重ね合わせ状態を破壊せずに，射影測定によってどの直交部分空間に飛んでしまったかを知れる.

Stabilizer basis

n qubit系において，stabilizer群の生成元が n 個与えられているとすると，stabilizer状態が一意的に定義される.

Stabilizer basis

n qubit系において，stabilizer群の生成元が *n* 個与えられているとすると，stabilizer状態が一意的に定義される.

n 個の生成元の固有値 ±1 でラベルされる *n* 個の量子状態は正規直交基底と成る→stabilizer basis

$$S_i|\Psi(s_1, s_2, \ldots, s_n)\rangle = (-1)^{s_i}|\Psi(s_1, s_2, \ldots, s_n)\rangle \qquad s_i \in \{0, 1\}$$

（stabilizer状態は $|\Psi(0, 0, \ldots, 0)\rangle$ ）

Stabilizer basis

n qubit系において，stabilizer群の生成元が *n* 個与えられているとすると，stabilizer状態が一意的に定義される.

n 個の生成元の固有値 ±1 でラベルされる *n* 個の量子状態は正規直交基底と成る→stabilizer basis

$$S_i|\Psi(s_1, s_2, \ldots, s_n)\rangle = (-1)^{s_i}|\Psi(s_1, s_2, \ldots, s_n)\rangle \qquad s_i \in \{0, 1\}$$

（stabilizer状態は $|\Psi(0, 0, \ldots, 0)\rangle$ ）

例）$S_1 = XX, S_2 = ZZ$

$$|\Psi(00)\rangle = (|00\rangle + |11\rangle)/\sqrt{2}, \quad |\Psi(01)\rangle = (|01\rangle + |10\rangle)/\sqrt{2},$$
$$|\Psi(10)\rangle = (|00\rangle - |11\rangle)/\sqrt{2}, \quad |\Psi(11)\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$$

→Bell基底

Pauli積Aのstabilizer basisへの作用

ある Pauli積 Aの stabilizer basis への作用を考える.

$$A|\Psi(s_1, s_2, \ldots, s_n)\rangle = ?$$

$AS_i = (-1)^{u_i} S_i A$ で, u_i を定義すると，つまり可換なら

$u_i = 0$ 非可換なら $u_i = 1$ とすると,

$$\begin{aligned} & A(-1)^{s_i} S_i |\Psi(s_1, s_2, ...s_n)\rangle \\ = \ & (-1)^{s_i \oplus u_i} S_i A |\Psi(s_1, s_2, ...s_n)\rangle \\ = \ & |\Psi(s_1 \oplus u_1, s_2 \oplus u_2, ...s_n \oplus u_n)\rangle \end{aligned}$$

Pauli積Aのstabilizer basisへの作用

ある Pauli積 Aの stabilizer basis への作用を考える.

$$A|\Psi(s_1, s_2, \ldots, s_n)\rangle = ?$$

$AS_i = (-1)^{u_i} S_i A$ で, u_iを定義すると，つまり可換なら
$u_i = 0$ 非可換なら $u_i = 1$ とすると,

$$\begin{aligned} & A(-1)^{s_i} S_i|\Psi(s_1, s_2, \ldots s_n)\rangle \\ = & (-1)^{s_i \oplus u_i} S_i A|\Psi(s_1, s_2, \ldots s_n)\rangle \\ = & |\Psi(s_1 \oplus u_1, s_2 \oplus u_2, \ldots s_n \oplus u_n)\rangle \end{aligned}$$

つまり，演算子S_iと非可換な演算子が，S_iに関する固有値をフリップする.

ZZ \ XX	Z: +1	Z: -1
X: +1	$(\|00\rangle + \|11\rangle)/\sqrt{2}$	$(\|00\rangle - \|11\rangle)/\sqrt{2}$
X: -1	$(\|01\rangle + \|10\rangle)/\sqrt{2}$	$(\|10\rangle - \|01\rangle)/\sqrt{2}$

Stabilizer subspace

stabilizer生成元の数 $|\mathcal{S}_G|$ が qubit 数 *n* よりも小さい場合

stabilizer状態は $2^{n-|\mathcal{S}_G|}$ 次元の部分空間となる.

例) $\langle XX, ZZ \rangle \rightarrow |\Psi(00)\rangle = (|00\rangle + |11\rangle)/\sqrt{2},$

ZZ \ XX	+1	-1
+1	$(\vert 00\rangle + \vert 11\rangle)/\sqrt{2}$	$(\vert 00\rangle - \vert 11\rangle)/\sqrt{2}$
-1	$(\vert 01\rangle + \vert 10\rangle)/\sqrt{2}$	$(\vert 10\rangle - \vert 01\rangle)/\sqrt{2}$

Stabilizer subspace

stabilizer生成元の数 $|\mathcal{S}_G|$ が qubit 数 *n* よりも小さい場合

stabilizer状態は $2^{n-|\mathcal{S}_G|}$ 次元の部分空間となる.

例) $\langle XX, ZZ\rangle \to |\Psi(00)\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$,

→*XX* を生成元から取り除くと，*XX* の固有値+1,-1の状態がともにstabilizer状態になる.

$\langle ZZ\rangle \to \{|\Psi(s_1, 0)\rangle\} = \{|00\rangle, |11\rangle\}$

ZZ \ XX	+1	-1
+1	$(\|00\rangle + \|11\rangle)/\sqrt{2}$	$(\|00\rangle - \|11\rangle)/\sqrt{2}$
-1	$(\|01\rangle + \|10\rangle)/\sqrt{2}$	$(\|10\rangle - \|01\rangle)/\sqrt{2}$

Stabilizer subspace

stabilizer生成元の数 $|\mathcal{S}_G|$ が qubit 数 *n* よりも小さい場合

stabilizer状態は $2^{n-|\mathcal{S}_G|}$ 次元の部分空間となる.

例) $\langle XX, ZZ\rangle \rightarrow |\Psi(00)\rangle = (|00\rangle + |11\rangle)/\sqrt{2},$

→*XX* を生成元から取り除くと，*XX* の固有値+1,-1の状態がともにstabilizer状態になる.

$\langle ZZ\rangle \rightarrow \{|\Psi(s_1, 0)\rangle\} = \{|00\rangle, |11\rangle\}$

ZZ \ XX	+1	-1
+1	$(\vert 00\rangle + \vert 11\rangle)/\sqrt{2}$	$(\vert 00\rangle - \vert 11\rangle)/\sqrt{2}$
-1	$(\vert 01\rangle + \vert 10\rangle)/\sqrt{2}$	$(\vert 10\rangle - \vert 01\rangle)/\sqrt{2}$

部分空間の指定は，*n* 個の生成元（従って状態を指定）から生成元を取り除く = 縮退を導入すると考えると分かりやすい.

$$|\Psi(\underbrace{s_1, s_2, \ldots, s_k}_{\text{取り除かれた生成元に対応する部分}}, 0, 0, \ldots, 0)\rangle$$

論理演算子

縮退した部分空間を記述する演算子.

→ stabilizer演算子と可換，かつstabilizer群に含まれない演算子

→ そしてそれら演算子からPauliの交換関係を満たす対を見つける

$$\Big(\,|\Psi(\underbrace{s_1, s_2, \ldots, s_k}_{\text{ここに作用}}, \underbrace{0, 0, \ldots, 0}_{\text{ここは変えない}})\rangle \longleftarrow$$ stabilizer状態から演算子を抜くと考えた場合 $\Big)$

$[L_i^A, S_j] = 0$ でかつ, $L_i^X L_i^Z = -L_i^Z L_i^X$ を満たす, L_i^X, L_i^Z を,

$n - |\mathcal{S}_G|$ 個見つけてくる.

eg) $\langle ZZ \rangle$

$$L_1^X = XX, \quad L_1^Z = IZ$$

L_1^X の固有状態 $|00\rangle \pm |11\rangle$ 　　L_1^Z の固有状態 $|00\rangle, |11\rangle$

eg) $\mathcal{S}_{\rm bit} = \langle ZZI, IZZ \rangle$

$$L_1^X = XXX, \; L_1^Z = ZII$$

Stabilizer符号

◆ 論理基底（logical basis）：

$\langle S_j, (-1)^{i_1} L_1^Z, \cdots, (-1)^{i_k} L_k^Z \rangle$ によって stabilize される状態を logical computational basis $|i_1, \cdots, i_k\rangle_L$ として定義する.

($k = n - |\mathcal{S}_G|$)

Stabilizer符号

◆ 論理基底（logical basis）：

$\langle S_j, (-1)^{i_1} L_1^Z, \cdots, (-1)^{i_k} L_k^Z \rangle$ によって stabilize される状態を logical computational basis $|i_1, \cdots, i_k\rangle_L$ として定義する.

($k = n - |\mathcal{S}_G|$)

注）L_i^Z, L_i^X は logical basis 上のlogical Pauli operatorになっている.

$$L_k^Z |i_1, \cdots, i_k\rangle_L = (-1)^{i_k} |i_1, \cdots, i_k\rangle_L$$

$$L_j^X |i_1, \cdots, i_k\rangle_L = |i_1, \cdots, i_j \oplus 1, \cdots, i_k\rangle_L$$

$$\left(L_j^Z (L_j^X |i_1, \cdots, i_k\rangle_L) = -L_j^X (-1)^{i_j} |i_1, \cdots, i_k\rangle_L = (-1)^{i_j \oplus 1} (L_j^X |i_1, \cdots, i_k\rangle_L) \right)$$

Stabilizer符号

◆ 論理基底（logical basis）：

$\langle S_j, (-1)^{i_1} L_1^Z, \cdots, (-1)^{i_k} L_k^Z \rangle$ によって stabilize される状態を logical computational basis $|i_1, \cdots, i_k\rangle_L$ として定義する.

($k = n - |\mathcal{S}_G|$)

注）L_i^Z, L_i^X は logical basis 上のlogical Pauli operatorになっている.

$$L_k^Z |i_1, \cdots, i_k\rangle_L = (-1)^{i_k} |i_1, \cdots, i_k\rangle_L$$

$$L_j^X |i_1, \cdots, i_k\rangle_L = |i_1, \cdots, i_j \oplus 1, \cdots, i_k\rangle_L$$

$$\left(L_j^Z (L_j^X |i_1, \cdots, i_k\rangle_L) = -L_j^X (-1)^{i_j} |i_1, \cdots, i_k\rangle_L = (-1)^{i_j \oplus 1} (L_j^X |i_1, \cdots, i_k\rangle_L) \right)$$

eg) $\langle ZZI, IZZ \rangle,\ L_1^Z = ZII \rightarrow |0\rangle_L = |000\rangle,\ |1\rangle_L = |111\rangle$

$$L_1^Z |1\rangle_L = -|1\rangle_L$$
$$L_1^X |0\rangle_L = |1\rangle_L$$

Stabilizer符号

◆ 論理基底（logical basis）：

$\langle S_j, (-1)^{i_1} L_1^Z, \cdots, (-1)^{i_k} L_k^Z \rangle$ によって stabilize される状態を logical computational basis $|i_1, \cdots, i_k\rangle_L$ として定義する.

($k = n - |\mathcal{S}_G|$)

注）L_i^Z, L_i^X は logical basis 上のlogical Pauli operatorになっている.

$$L_k^Z |i_1, \cdots, i_k\rangle_L = (-1)^{i_k} |i_1, \cdots, i_k\rangle_L$$

$$L_j^X |i_1, \cdots, i_k\rangle_L = |i_1, \cdots, i_j \oplus 1, \cdots, i_k\rangle_L$$

$$\left(L_j^Z (L_j^X |i_1, \cdots, i_k\rangle_L) = -L_j^X (-1)^{i_j} |i_1, \cdots, i_k\rangle_L = (-1)^{i_j \oplus 1} (L_j^X |i_1, \cdots, i_k\rangle_L) \right)$$

eg) $\langle ZZI, IZZ \rangle,\ L_1^Z = ZII \to |0\rangle_L = |000\rangle,\ |1\rangle_L = |111\rangle$

$$L_1^Z |1\rangle_L = -|1\rangle_L$$

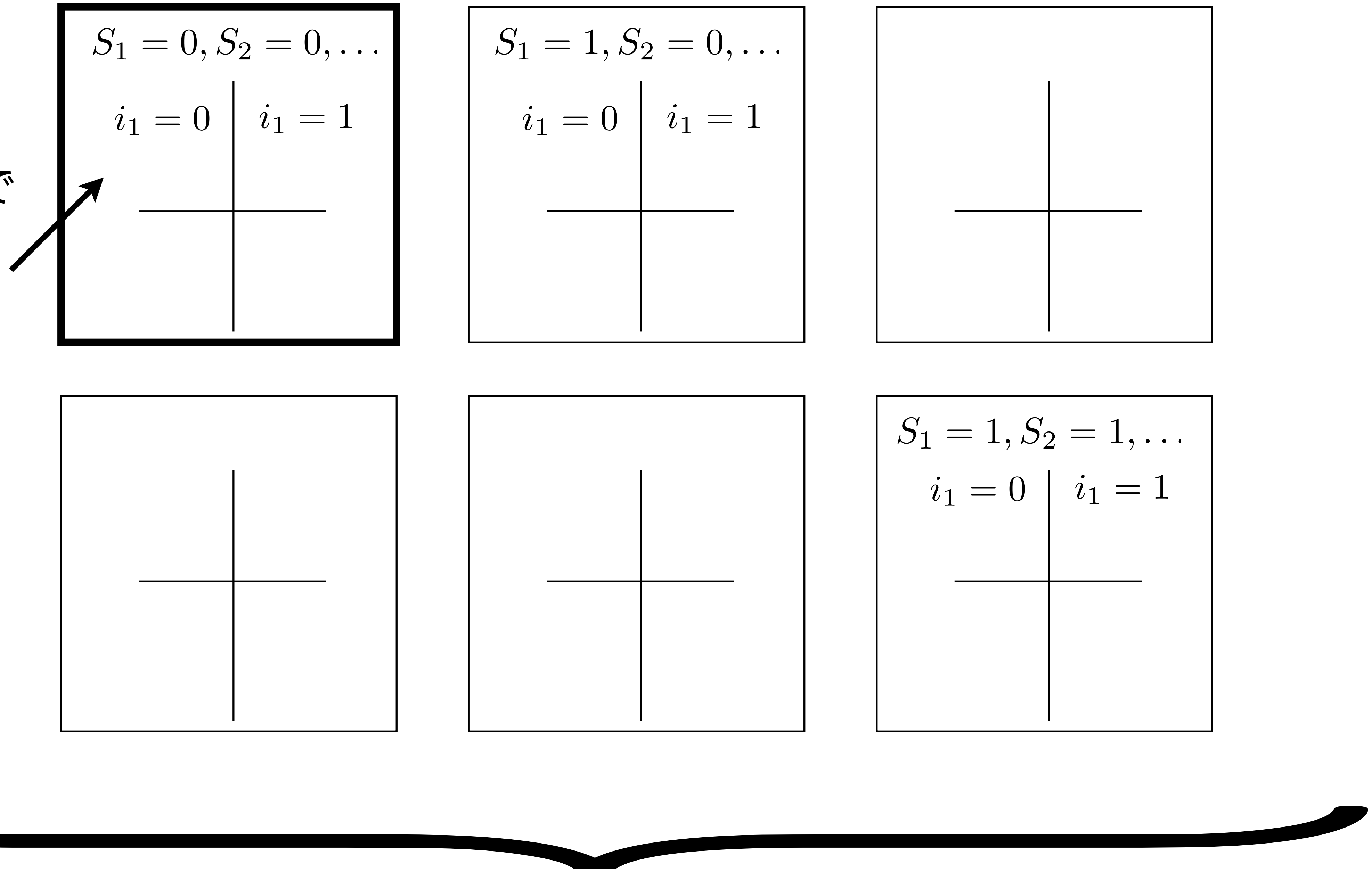
$$L_1^X |0\rangle_L = |1\rangle_L$$

これが stabilizer 符号の定義！

Stabilizer符号の構造

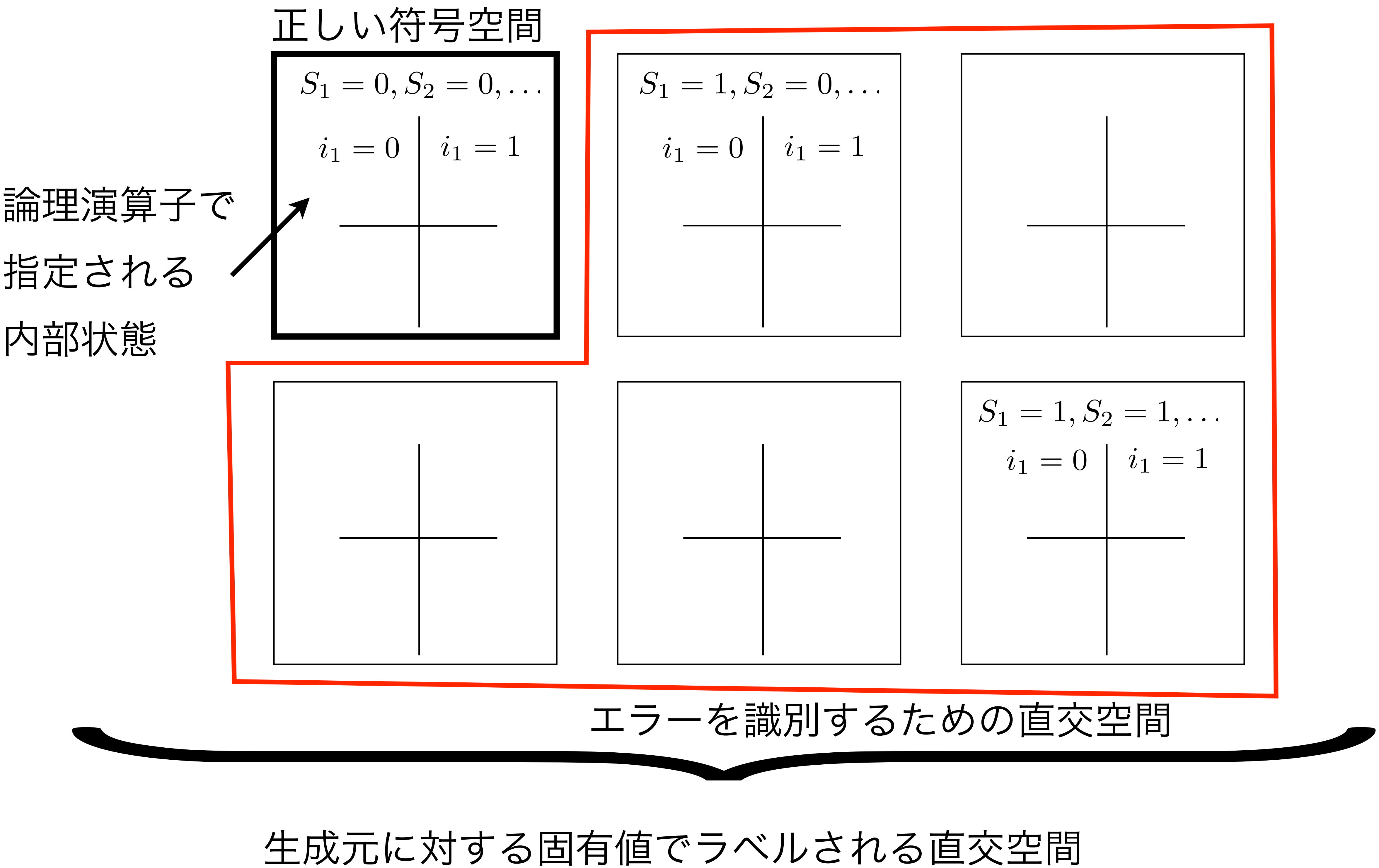
正しい符号空間

論理演算子で
指定される
内部状態



生成元に対する固有値でラベルされる直交空間

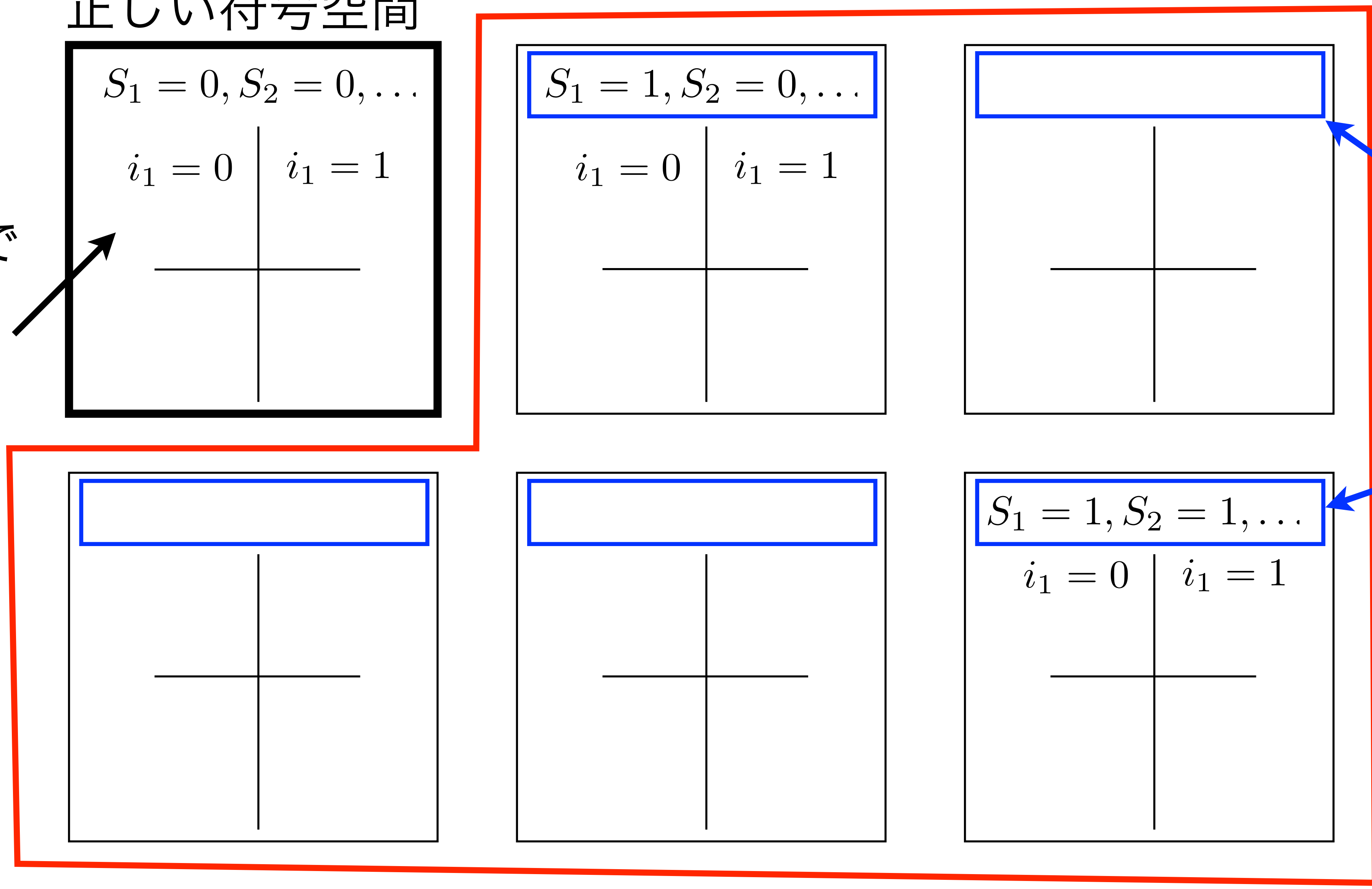
Stabilizer符号の構造



Stabilizer符号の構造

正しい符号空間

論理演算子で
指定される
内部状態



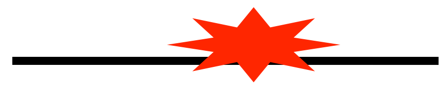
直交部分空間
のラベル＝
error syndrome

エラーを識別するための直交空間

生成元に対する固有値でラベルされる直交空間

エラーの作用

- Pauli エラー:

input  output

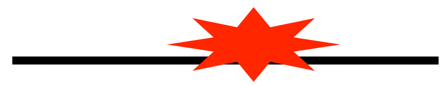
bit-flip error: $(1-p)\rho + pX\rho X$

phase-flip error: $(1-p)\rho + pZ\rho Z$

depolarizing error: $(1-p)\rho + p/3(X\rho X + Y\rho Y + Z\rho Z)$

エラーの作用

◆ Pauli エラー:



bit-flip error: $(1-p)\rho + pX\rho X$

phase-flip error: $(1-p)\rho + pZ\rho Z$

depolarizing error: $(1-p)\rho + p/3(X\rho X + Y\rho Y + Z\rho Z)$

◆ Pauli エラーの符号空間への作用:

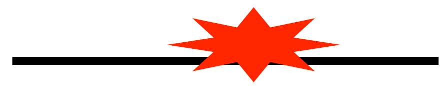
A がstabilizer である．logical stateを変化させない．

$[A, S_i] = 0$ 全てのstabilizerと可換．

A がlogical operator である．logical stateを変化させる．

エラーの作用

◆ Pauli エラー:

input  output

bit-flip error: $(1-p)\rho + pX\rho X$

phase-flip error: $(1-p)\rho + pZ\rho Z$

depolarizing error: $(1-p)\rho + p/3(X\rho X + Y\rho Y + Z\rho Z)$

◆ Pauli エラーの符号空間への作用:

$[A, S_i] = 0$ 全てのstabilizerと可換.

→ *A* がstabilizer である．logical stateを変化させない.

→ *A* がlogical operator である．logical stateを変化させる.

$[A, S_j] \neq 0$ stabilizerのどれかと反可換 → 状態がstabilizer subspaceの外（直交補空間）に出る.

エラーの作用

◆ Pauli エラー:

input ——— output

bit-flip error: $(1-p)\rho + pX\rho X$

phase-flip error: $(1-p)\rho + pZ\rho Z$

depolarizing error: $(1-p)\rho + p/3(X\rho X + Y\rho Y + Z\rho Z)$

◆ Pauli エラーの符号空間への作用:

$[A, S_i] = 0$ 全てのstabilizerと可換.
- ↗ *A* がstabilizer である．logical stateを変化させない.
- ↘ *A* がlogical operator である．logical stateを変化させる.

$[A, S_j] \neq 0$ stabilizerのどれかと反可換 → 状態がstabilizer subspaceの外（直交補空間）に出る.

例）

$\alpha|000\rangle + \beta|111\rangle \xrightarrow{IZZ} \alpha|000\rangle + \beta|111\rangle$ 何もなかった。

$\alpha|000\rangle + \beta|111\rangle \xrightarrow{IIZ} \alpha|000\rangle - \beta|111\rangle$ 状態が壊れた。

$\alpha|000\rangle + \beta|111\rangle \xrightarrow{IIX} \alpha|001\rangle + \beta|110\rangle$ 状態が直交補空間へ飛んだ。

量子誤り訂正

(1) どのsyndrome subspaceに状態がいるかを知る(= error syndrome を知る)

(2) error syndrome からどのようなエラーが起きているか推定し，訂正する.

量子誤り訂正

(1) どのsyndrome subspaceに状態がいるかを知る(= error syndrome を知る)

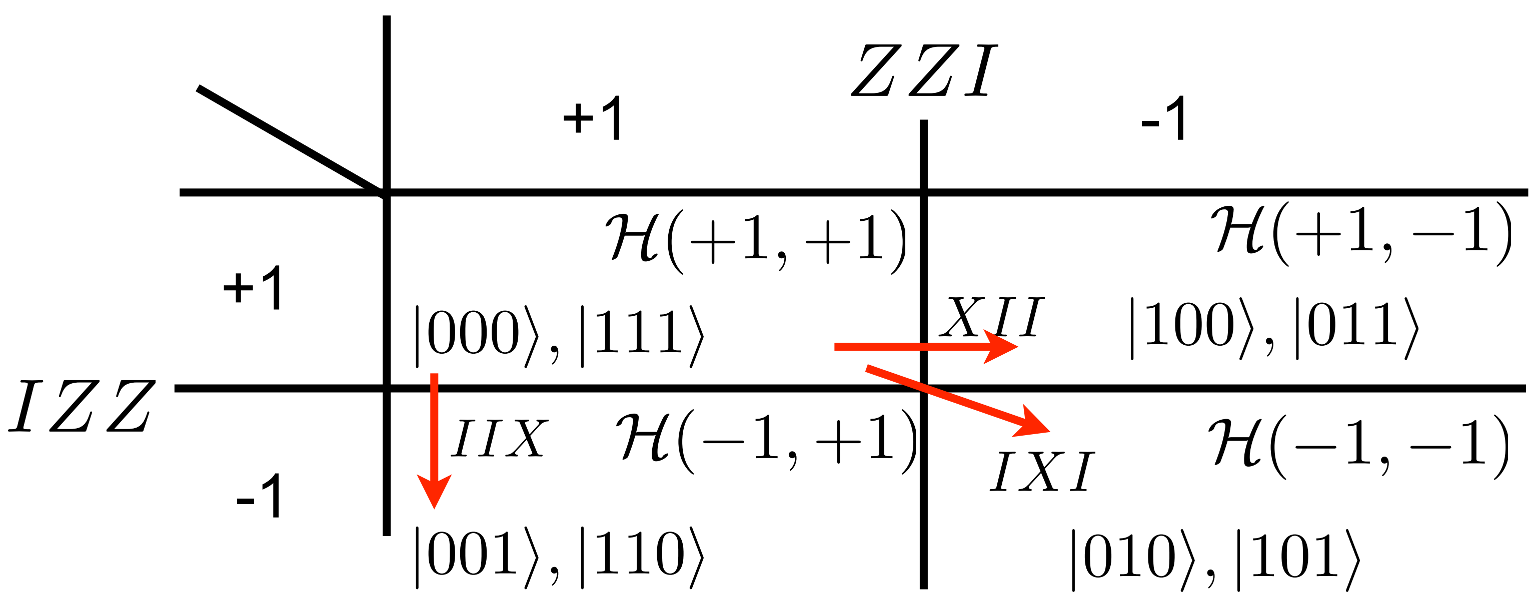
(2) error syndrome からどのようなエラーが起きているか推定し，訂正する.

eg) bit-flip error が3つのqubitに独立に作用するような場合を考える.

もし，error syndrome (+1,-1)が得られたとき,

XII with probability ~*p* or IXX with probability ~*p*²

most likely



エラーの発生確率 *p* が十分小さければこの操作によってeffectiveなエラー確率が減少する.

Syndromeの測定

▶ Indirect projective measurement:

*A*を固有値 ±1 のエルミート演算子とする．

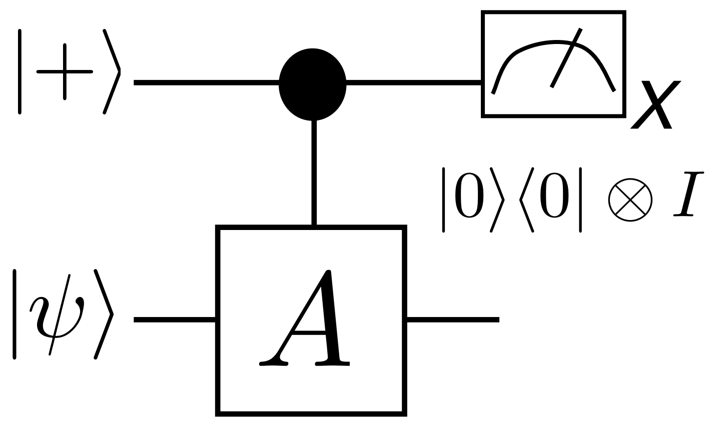
$$(|0\rangle_m|\psi\rangle + |1\rangle_m A|\psi\rangle)/\sqrt{2}$$

$$\Big\downarrow |+\rangle\langle+|$$

$$|+\rangle \underline{\left(\frac{I+A}{2}\right)} |\psi\rangle$$

*A*の+1の固有状態への

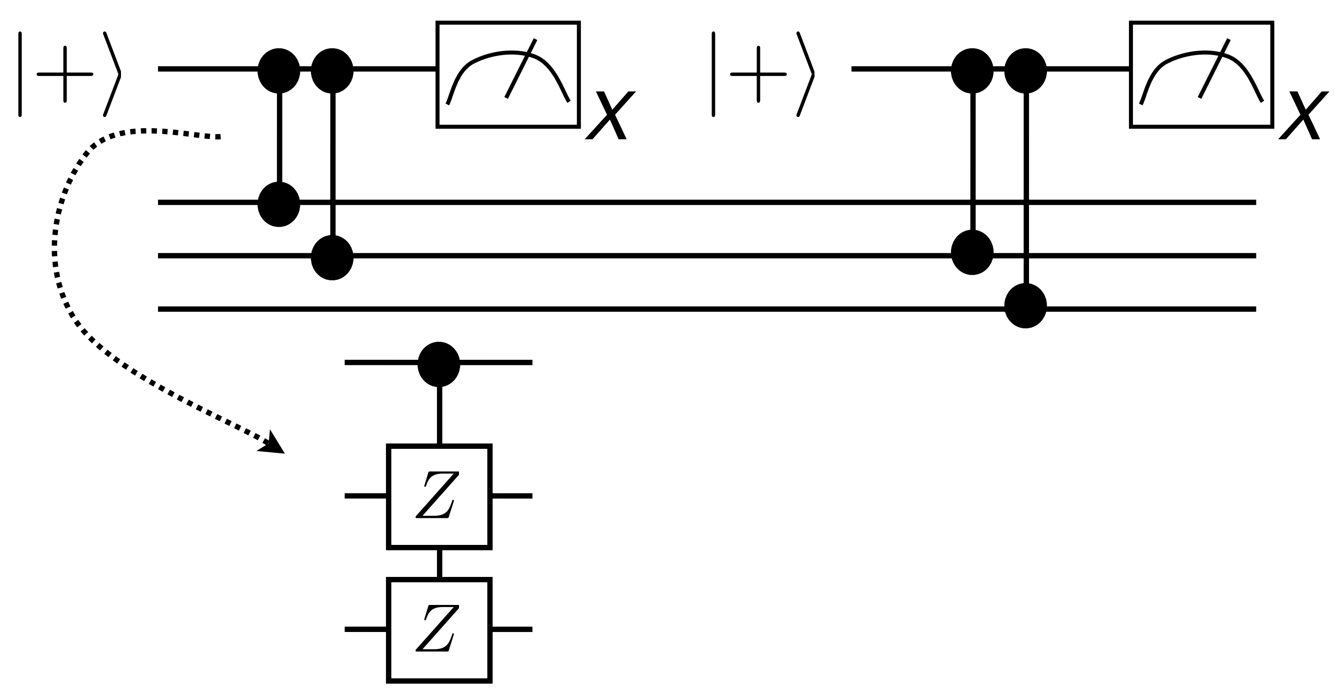
射影演算子



$$|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes A$$

projective measurement of *A*

例) $\mathcal{S}_{\mathrm{bit}} = \langle ZZI, IZZ \rangle$



Stean’s 7-qubit code

1つのqubitに作用する*X* error もしくは *Z* errorを訂正できる.

Stabilizer generators: $S_1 = ZIZIZIZ$ $S_4 = XIXIXIX$

$S_2 = IZZIIZZ$ $S_5 = IXXIIXX$

$S_3 = IIIZZZZ$ $S_6 = IIIXXXX$

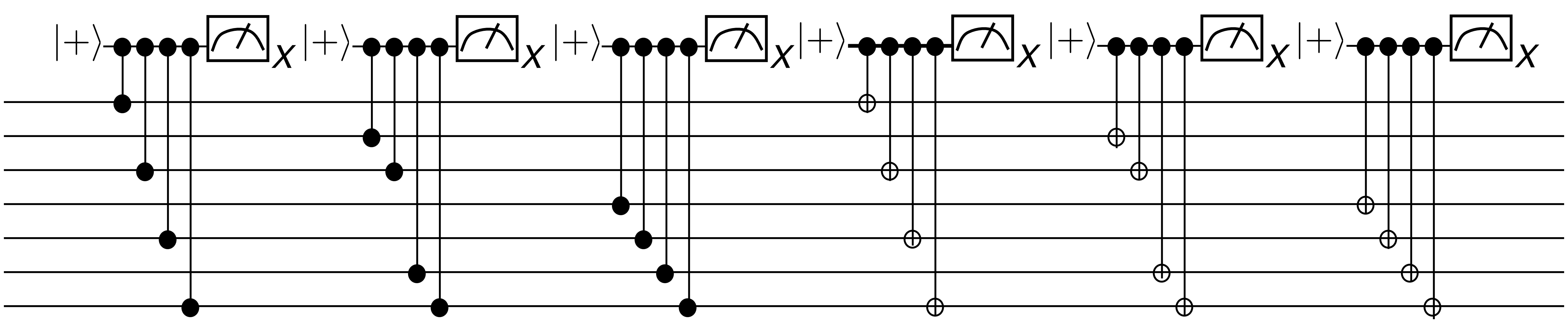
Syndrome subspaceの数は 2^6=64

errorの種類は (1+7)(1+7)=64 →異なるエラーが異なる直交補空間に対応している.

一般に *t* 個のエラーを訂正するためには

$$\left[\sum_{i=0}^{i=k} \binom{n}{t}\right]^2 \leq 2^{n-1}$$

等号成立は (n,t)=(7,1),(23,3)



(時間があれば、5-qubit)

フォールトトレランス

Rolf Landauer @IBM

All papers on quantum computing should carry a footnote: *"This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form* ***take into account all possible sources of noise, unreliability and manufacturing error****, and probably will not work."* (S. Lloyd Nature 400 720 (1999))

フォールトトレランス

Rolf Landauer @IBM

All papers on quantum computing should carry a footnote: *"This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form* ***take into account all possible sources of noise, unreliability and manufacturing error****, and probably will not work."* (S. Lloyd Nature 400 720 (1999))

・誤り訂正を行う**操作の間にもエラー**はもちろん発生する.

フォールトトレランス

Rolf Landauer @IBM

All papers on quantum computing should carry a footnote: *"This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form* ***take into account all possible sources of noise, unreliability and manufacturing error****, and probably will not work."* (S. Lloyd Nature 400 720 (1999))

- 誤り訂正を行う**操作の間にもエラー**はもちろん発生する.
- 局所的な２量子ビット演算.

フォールトトレランス

Rolf Landauer @IBM

All papers on quantum computing should carry a footnote: *"This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form* ***take into account all possible sources of noise, unreliability and manufacturing error****, and probably will not work."* (S. Lloyd Nature 400 720 (1999))

- 誤り訂正を行う**操作の間にもエラー**はもちろん発生する.
- 局所的な２量子ビット演算.
- 特定の物理系では**演算操作が確率的**になることもある.

フォールトトレランス

Rolf Landauer @IBM

All papers on quantum computing should carry a footnote: *"This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form* ***take into account all possible sources of noise, unreliability and manufacturing error****, and probably will not work."* (S. Lloyd Nature 400 720 (1999))

- 誤り訂正を行う**操作の間にもエラー**はもちろん発生する.
- 局所的な２量子ビット演算.
- 特定の物理系では**演算操作が確率的**になることもある.
- 単一の物理系だけでは量子ビット数を増やせない→**スケーラビリティーの問題**

（ふやすと制御が指数関数的に難しくなる）

フォールトトレランス

Rolf Landauer @IBM

All papers on quantum computing should carry a footnote: *"This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form* ***take into account all possible sources of noise, unreliability and manufacturing error****, and probably will not work."* (S. Lloyd Nature 400 720 (1999))

・誤り訂正を行う**操作の間にもエラー**はもちろん発生する.

・局所的な２量子ビット演算

・特定の物理系では演算操作が確率的になることもある

・単一の物理系だけでは量子ビット数を増やせない→**スケーラビリティーの問題**

（ふやすと制御が指数関数的に難しくなる）

これら全てを取り込んでも機能するものでないと意味がない.

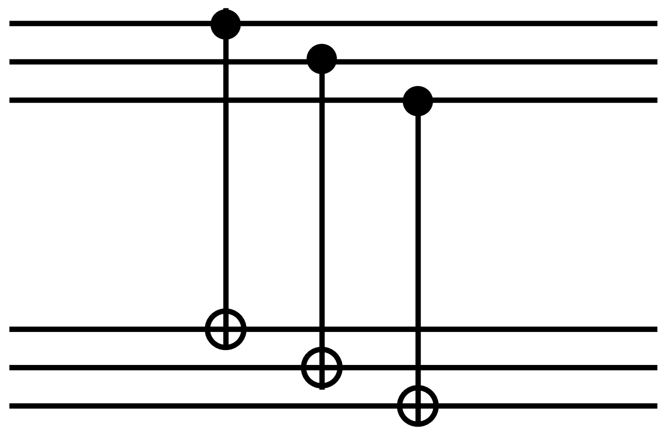
→フォールトトレラント量子計算

FT Clifford gate

◆ CSS符号の場合：

= Stabilizerが X だけの積と Z だけの積に分離されている場合

logical CNOT gate =

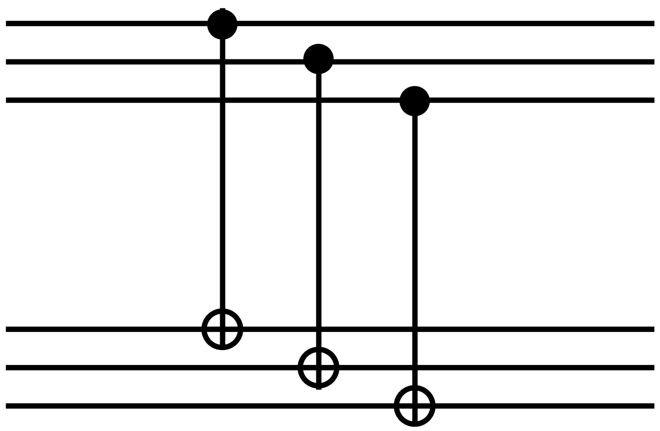


（確認）

FT Clifford gate

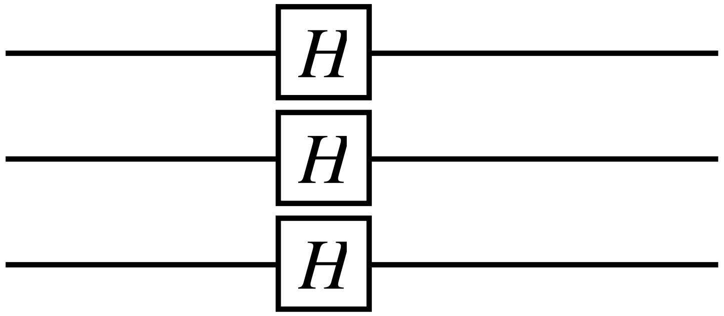
◆ CSS符号の場合：

= Stabilizerが X だけの積と Z だけの積に分離されている場合

logical CNOT gate = 

（確認）

◆ さらに X と Z の場所が同じ場合（i.e., 7-qubit code）：

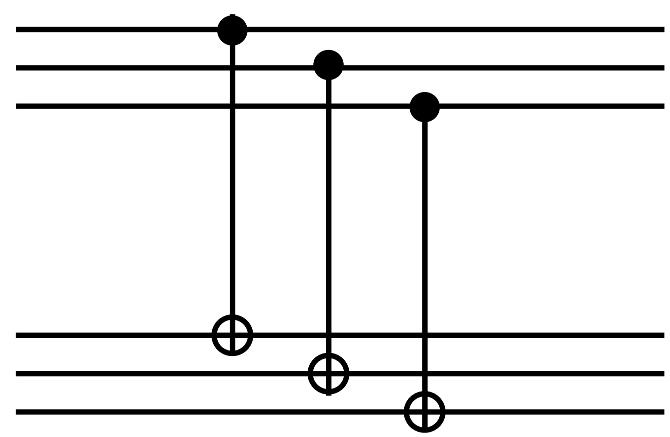
logical H gate = 

（確認）

FT Clifford gate

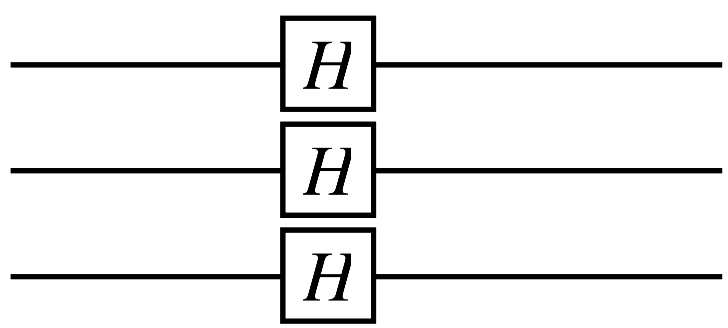
◆ CSS符号の場合：

= Stabilizerが X だけの積と Z だけの積に分離されている場合

logical CNOT gate = 

（確認）

◆ さらに X と Z の場所が同じ場合（i.e., 7-qubit code）：

logical H gate = 

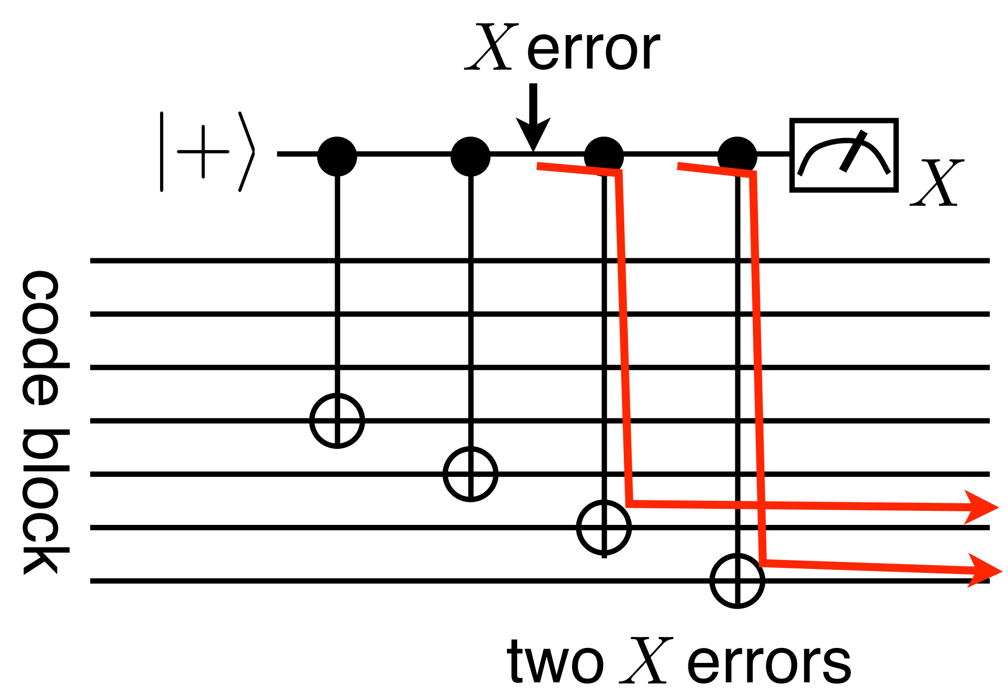
（確認）

◆ 一般のstabilizer符号の場合：

D. Gottesman, Phys. Rev. A **57**, 127 (1998).

FT syndrome measurement

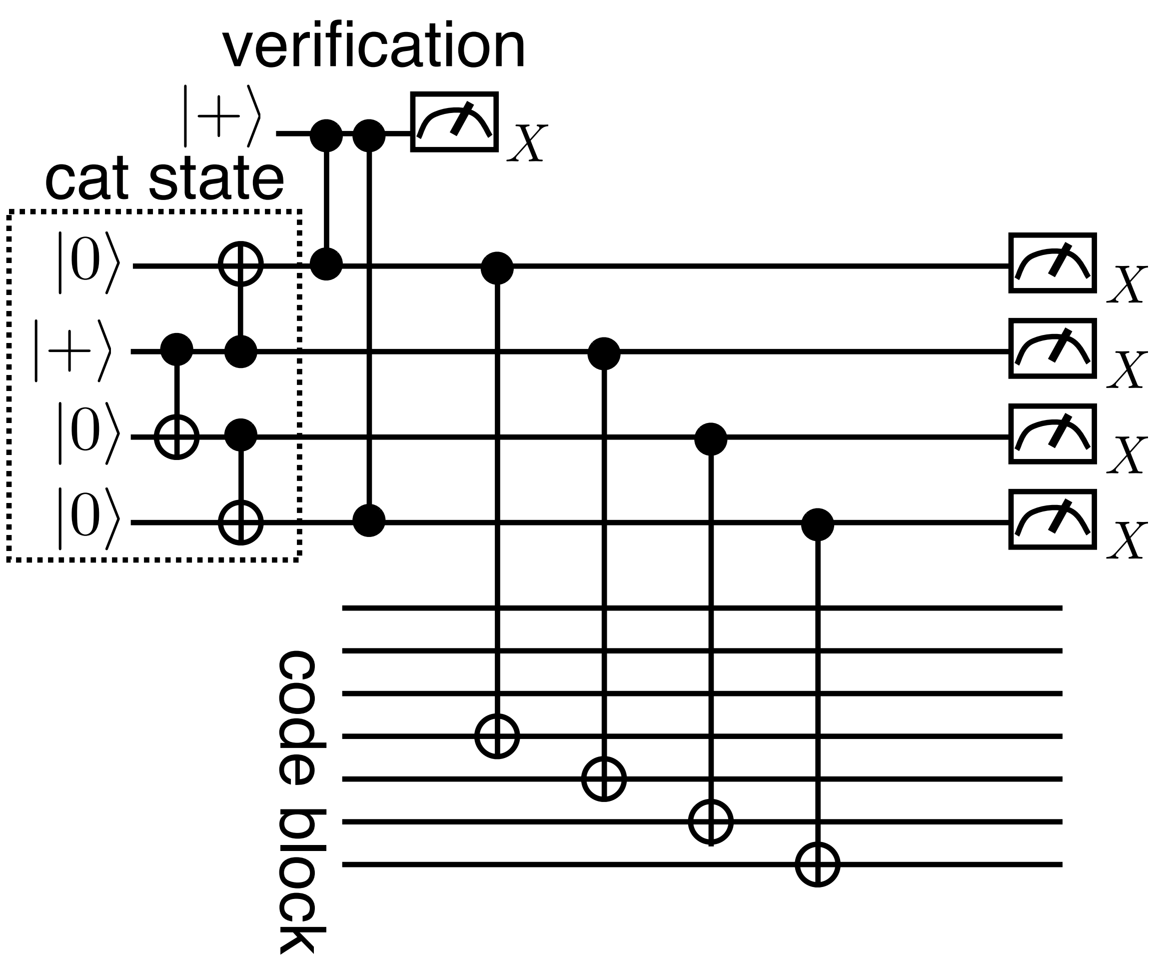
◆ Syndrome measurementにおけるエラーの伝搬:



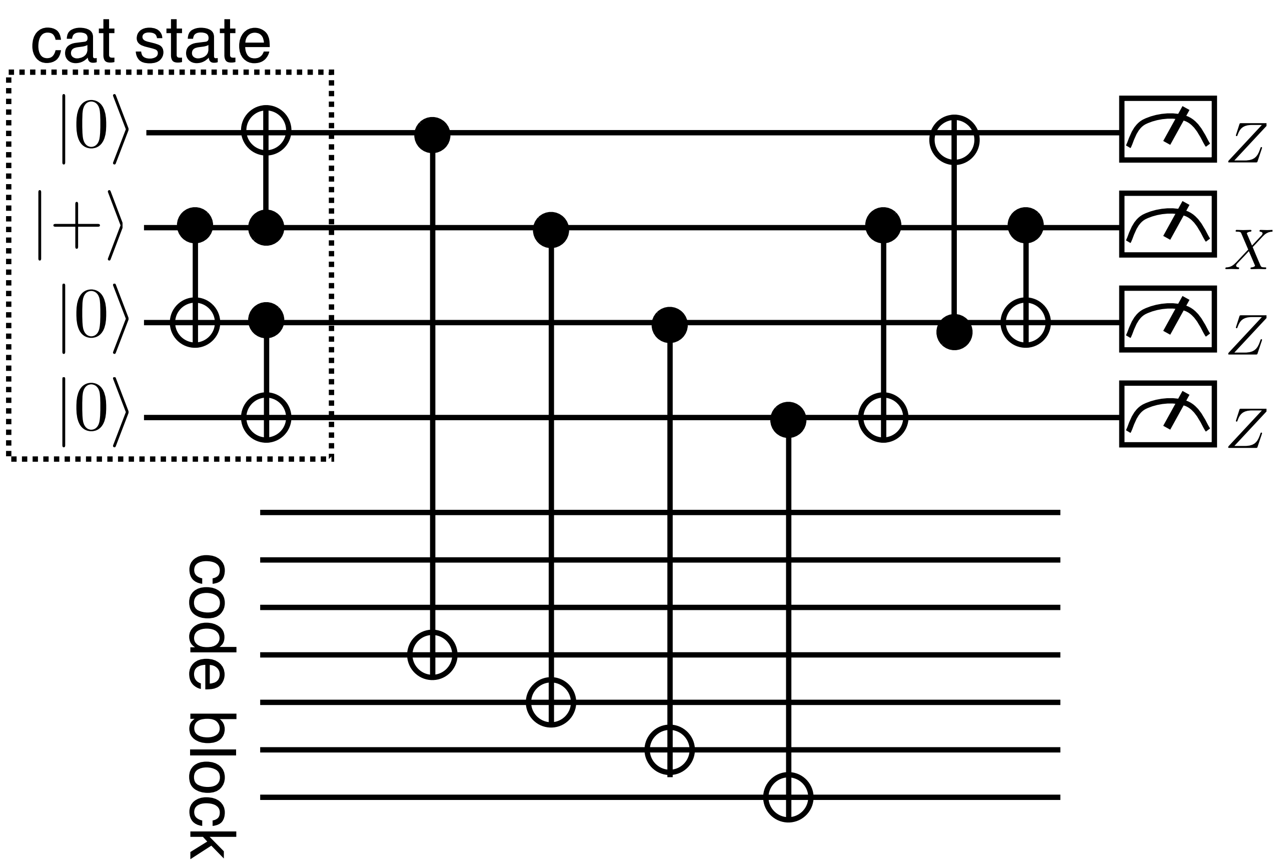
フォールトトレラントではない！

FT syndrome measurement

- Shor's gadget (Cat state):



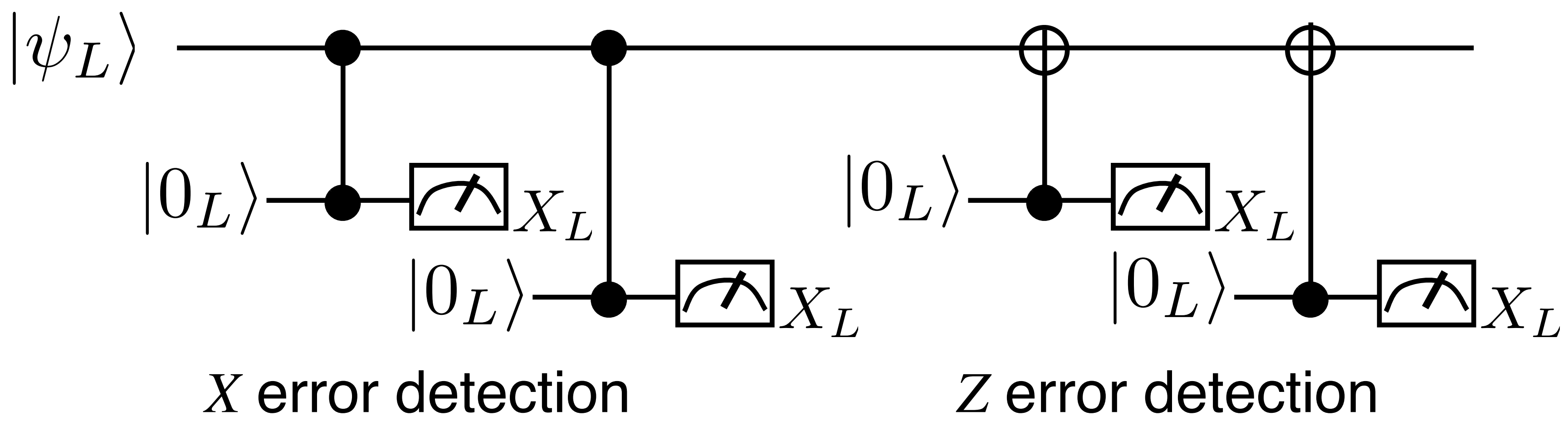
D. P. DiVincenzo and P. W. Shor,
Phys. Rev. Lett. **77**, 3260 (1996).



D. P. DiVincenzo and P. Aliferis,
Phys. Rev. Lett. **98**, 020501 (2007).

FT syndrome measurement

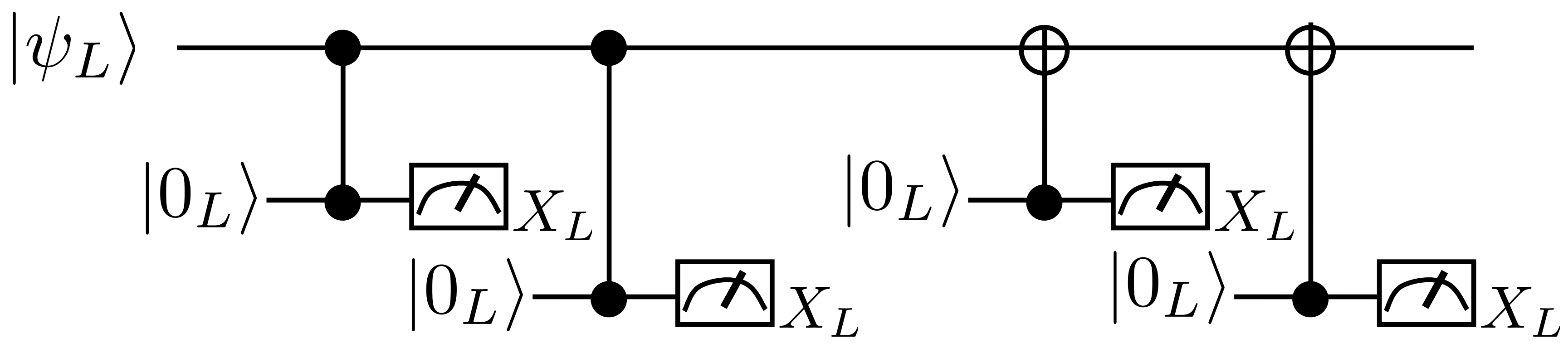
- Steane's gadget (logical ancilla):



A. M. Steane, Phys. Rev. Lett. **78**, 2252 (1997).

FT syndrome measurement

◆ Steane's gadget (logical ancilla):



X error detection　　　　Z error detection

A. M. Steane, Phys. Rev. Lett. **78**, 2252 (1997).

アンシラの論理状態は，エンタングルメント純粋化等を用いて
フォールトトレラントに準備（失敗しても捨てれる）

W. Dür, H. Aschauer, and H. J. Briegel, Phys. Rev. Lett. **91**, 107903 (2003).

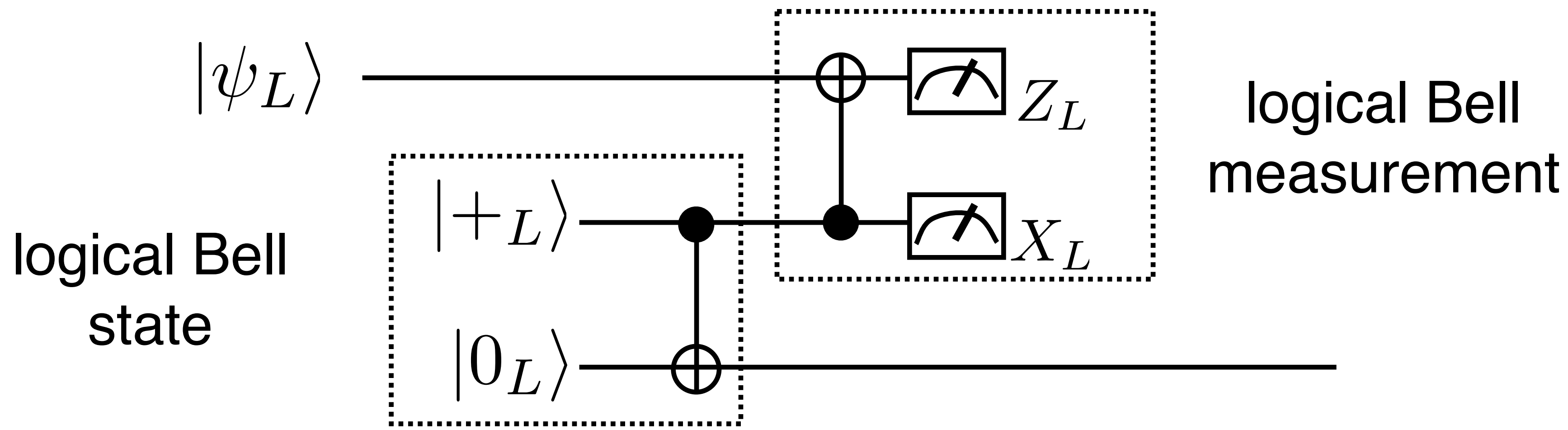
H. Aschauer, W. Dür, and H. J. Briegel, Phys. Rev. A **71**, 012319 (2005).

K. Fujii and K. Yamamoto, Phys. Rev. A **80**, 042308 (2009).

FT syndrome measurement

FT syndrome measurement

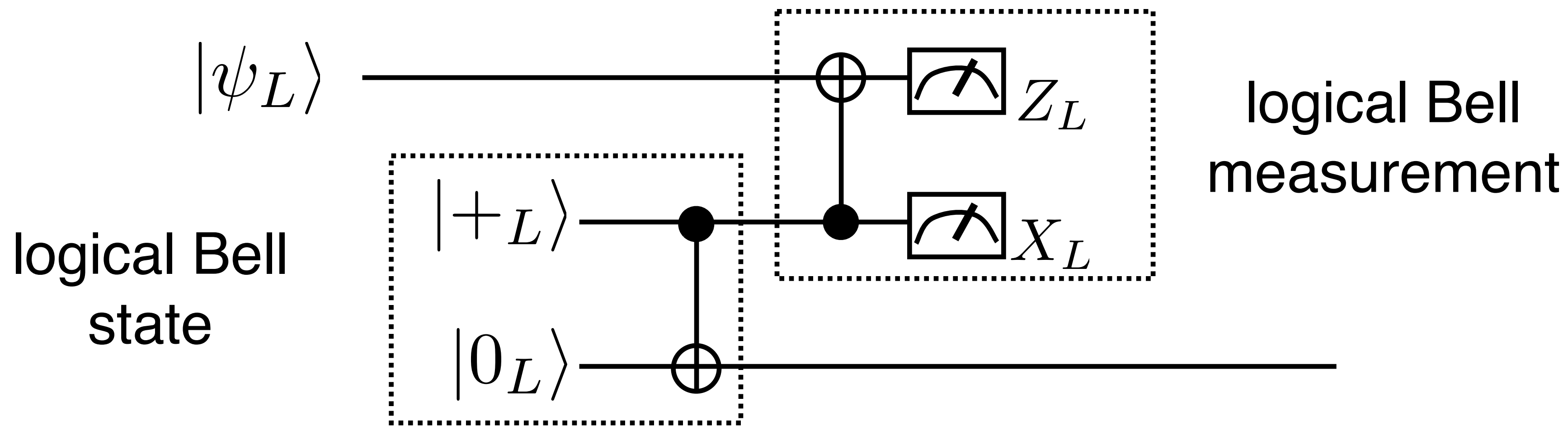
- Knill's gadget (tele-correction):



E. Knill, Nature (London) **434**, 39 (2005).

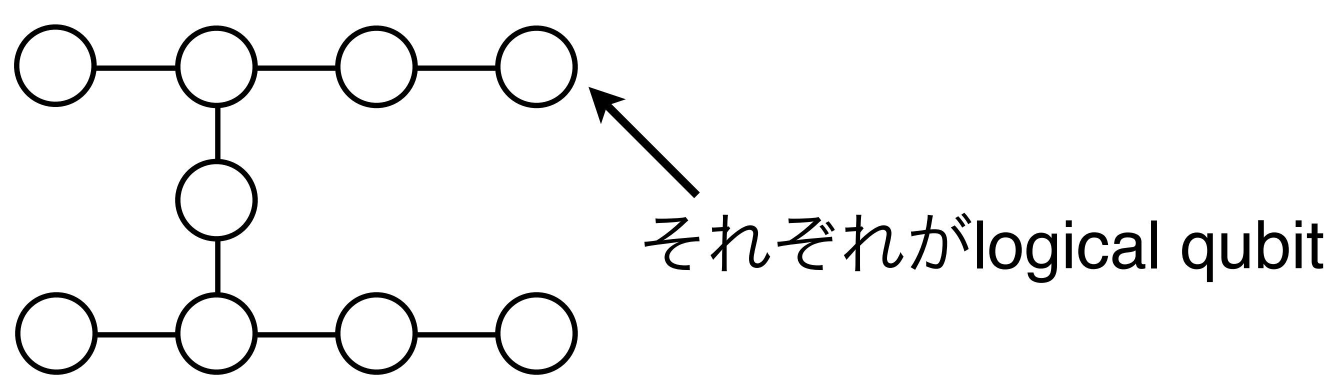
FT syndrome measurement

- Knill's gadget (tele-correction):



E. Knill, Nature (London) **434**, 39 (2005).

- Verified logical cluster state (pre-correction):



K. Fujii and K. Yamamoto, Phys. Rev. A **81**, 042324 (2010).
K. Fujii and K. Yamamoto, Phys. Rev. A **82**, 060301 (2010).

FT non-Clifford gate

◆ 一種類のnon-Clifford gateがあればOK

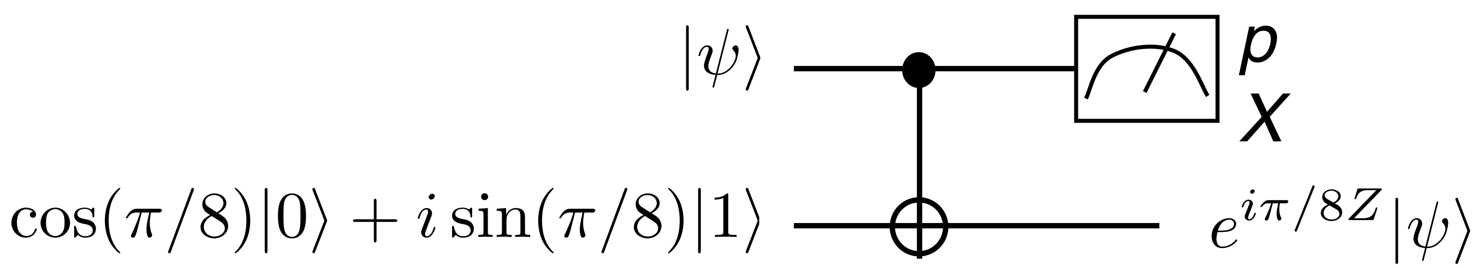
by Solovay-Kitaev theorem

FT non-Clifford gate

◆ 一種類のnon-Clifford gateがあればOK

by Solovay-Kitaev theorem

◆ 特殊なアンシラ状態（magic state）さえあれば，non-Clifford gateが 1-bit tel. （Clifford演算）を用いて実行できる.

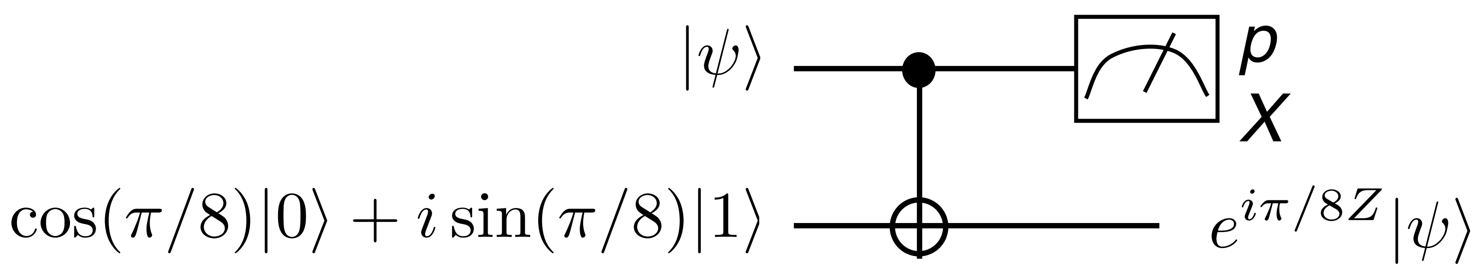


FT non-Clifford gate

◆ 一種類のnon-Clifford gateがあればOK

by Solovay-Kitaev theorem

◆ 特殊なアンシラ状態（magic state）さえあれば，non-Clifford gateが 1-bit tel. （Clifford演算）を用いて実行できる.



◆ ある程度きれいな magic state があれば，**Clifford演算のみ**を使って，理想的な magic stateをdistillationできる

by Bravyi-Kitaev PRA **71**, 022316 (2005)

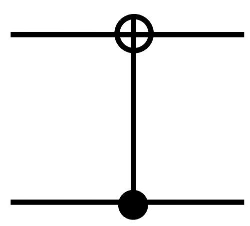
noisy ancilla + Clifford gate = universal

Threshold theorem

Concatenated quantum computation:

Threshold theorem

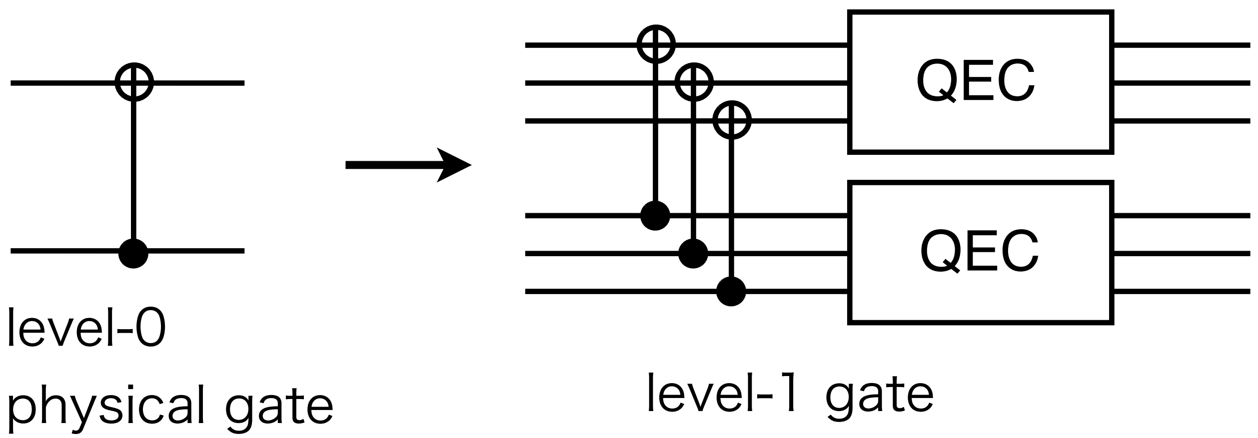
Concatenated quantum computation:



level-0
physical gate

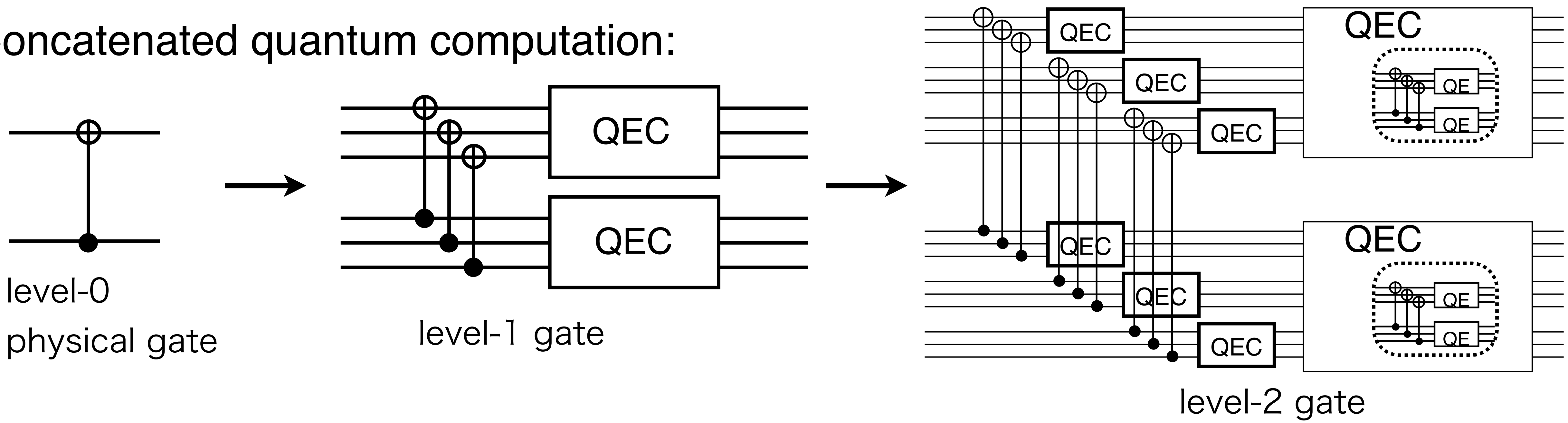
Threshold theorem

Concatenated quantum computation:



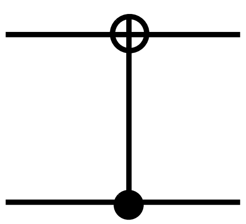
Threshold theorem

Concatenated quantum computation:

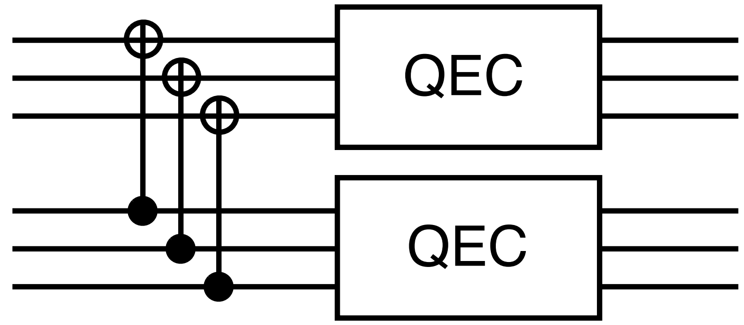


Threshold theorem

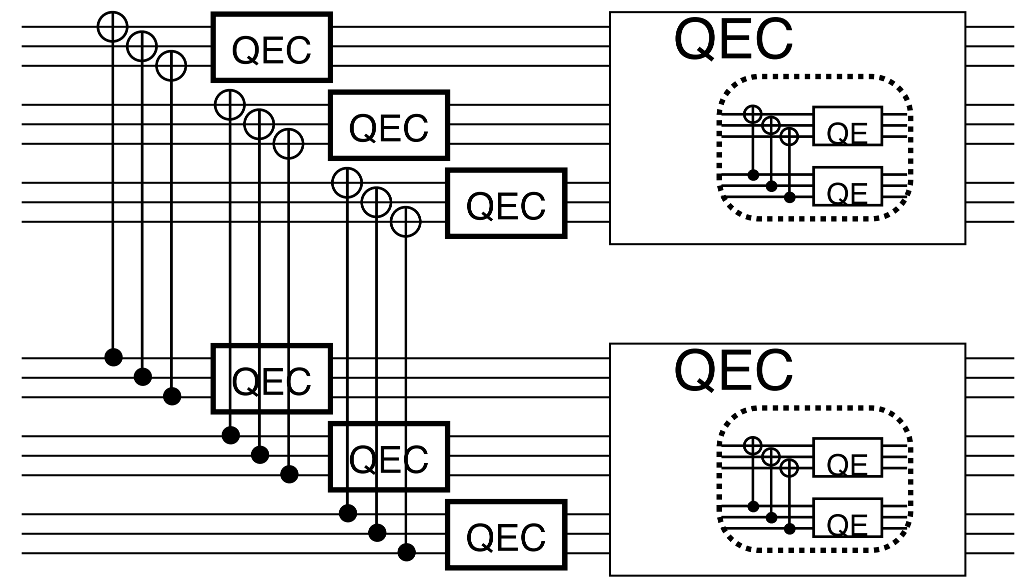
Concatenated quantum computation:



level-0
physical gate



level-1 gate



level-2 gate

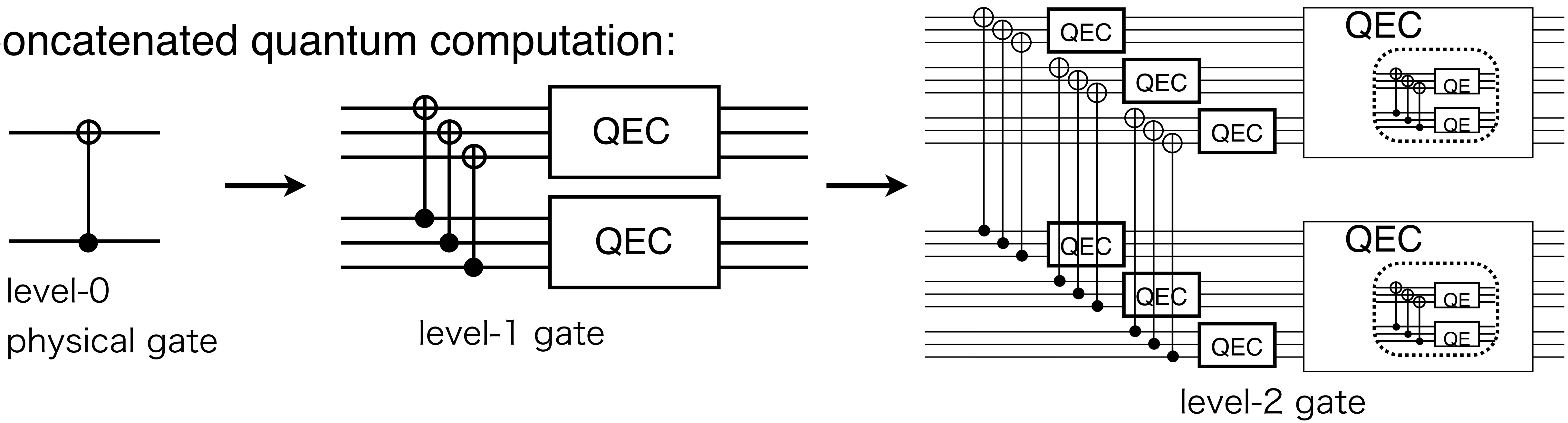
エラー確率：

$$p \rightarrow Cp^2 \rightarrow C(Cp^2)^2 \rightarrow \cdots \rightarrow (Cp)^{2^l}/C$$

Level-1 Level-2 Level- l

Threshold theorem

Concatenated quantum computation:



エラー確率：

$$p \to Cp^2 \to C(Cp^2)^2 \to \cdots \to (Cp)^{2^l}/C$$

Level-1 Level-2 Level- l

Threshold Theorem

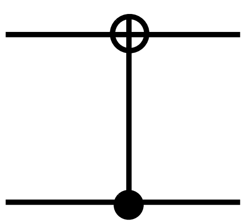
誤り確率が**しきい値**より低ければ論理誤り確率を**任意に小さく**できる.

$$p < \underline{1/C(\equiv p_{\mathrm{th}})} \Rightarrow (Cp)^{2^l}/C \to 0$$

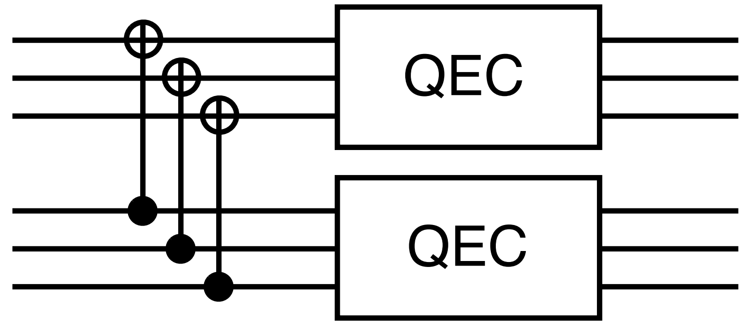
しきい値（Noise threshold）

Threshold theorem

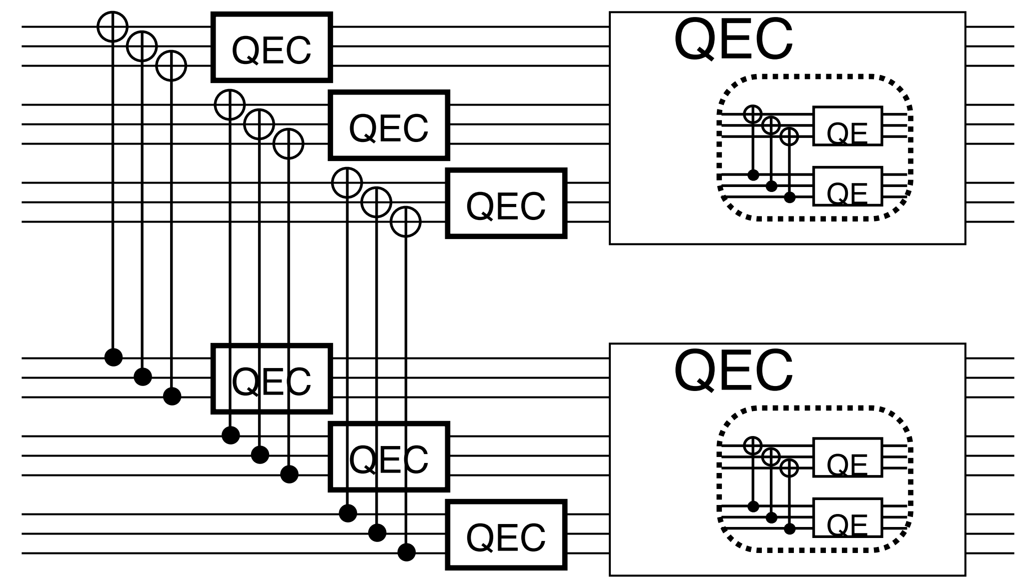
Concatenated quantum computation:



level-0
physical gate



level-1 gate



level-2 gate

エラー確率：

$$p \rightarrow Cp^2 \rightarrow C(Cp^2)^2 \rightarrow \cdots \rightarrow (Cp)^{2^l}/C$$

Level-1 Level-2 Level- l

Threshold Theorem

誤り確率が**しきい値**より低ければ論理誤り確率を**任意に小さく**できる.

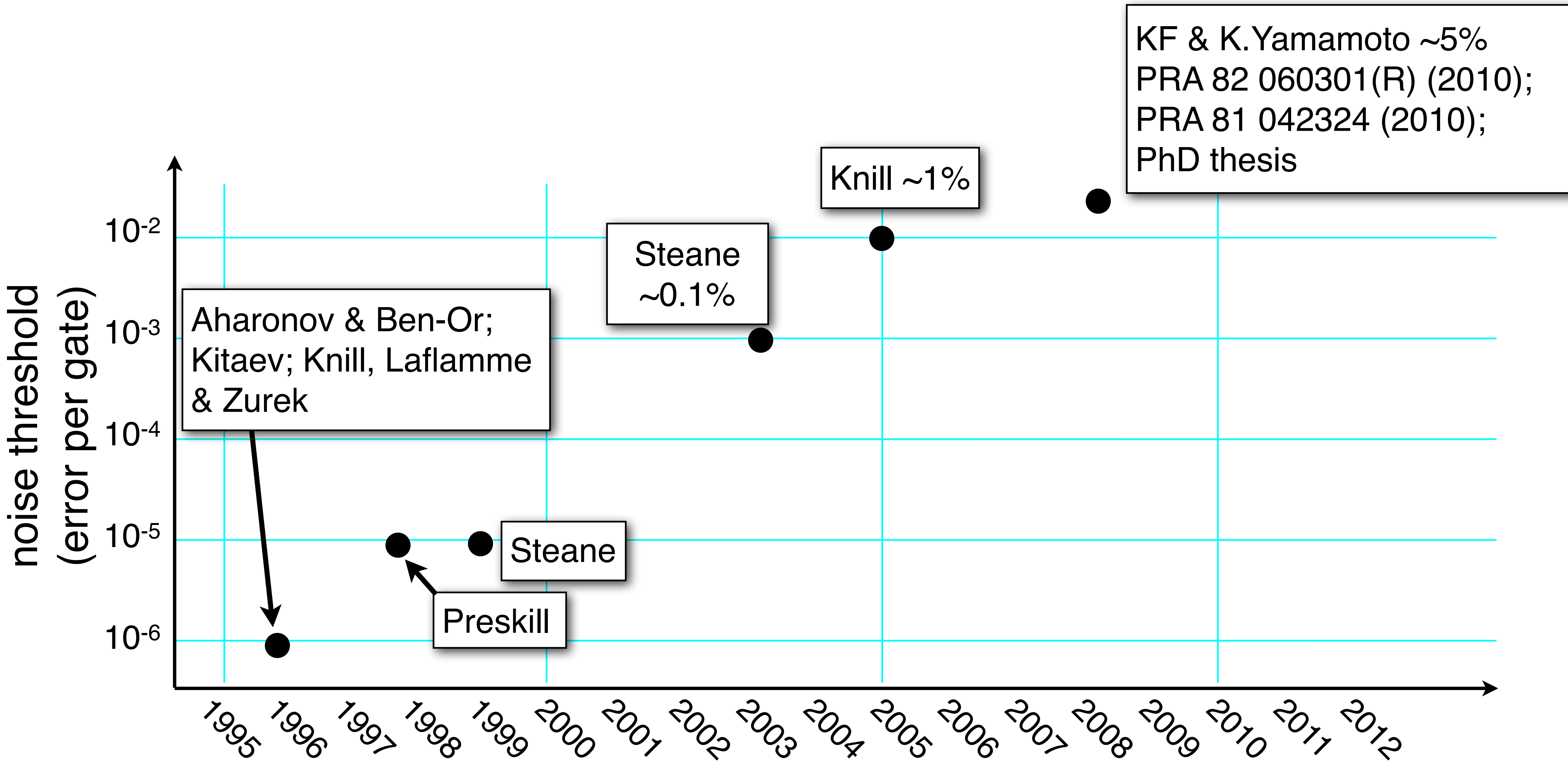
$$p < \underline{1/C(\equiv p_{\mathrm{th}})} \quad \Rightarrow \quad (Cp)^{2^l}/C \rightarrow 0$$

しきい値（Noise threshold）

最近ではトポロジカルフォールトトレラント量子計算の方が多い

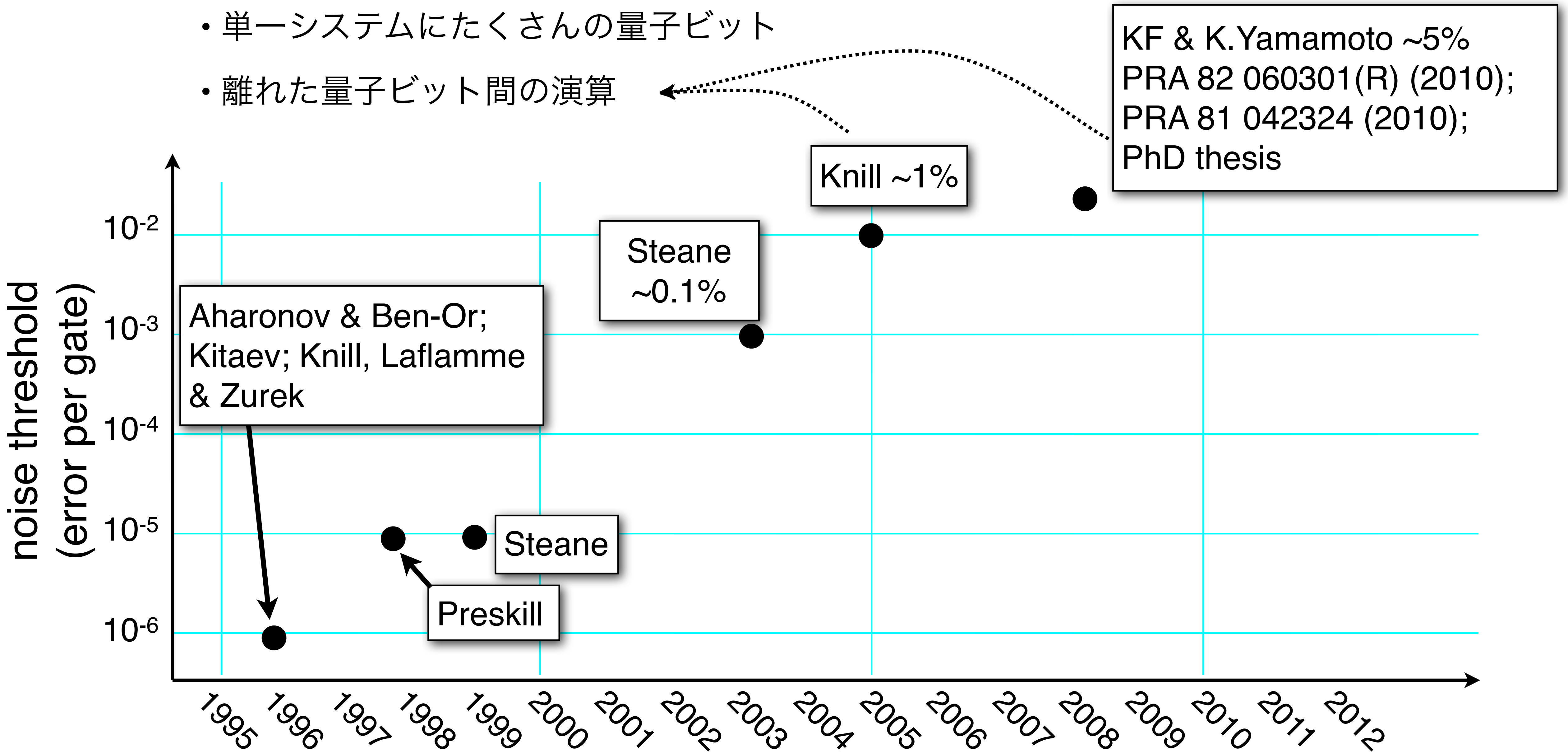
リソースの話

Noise thresholdの歴史



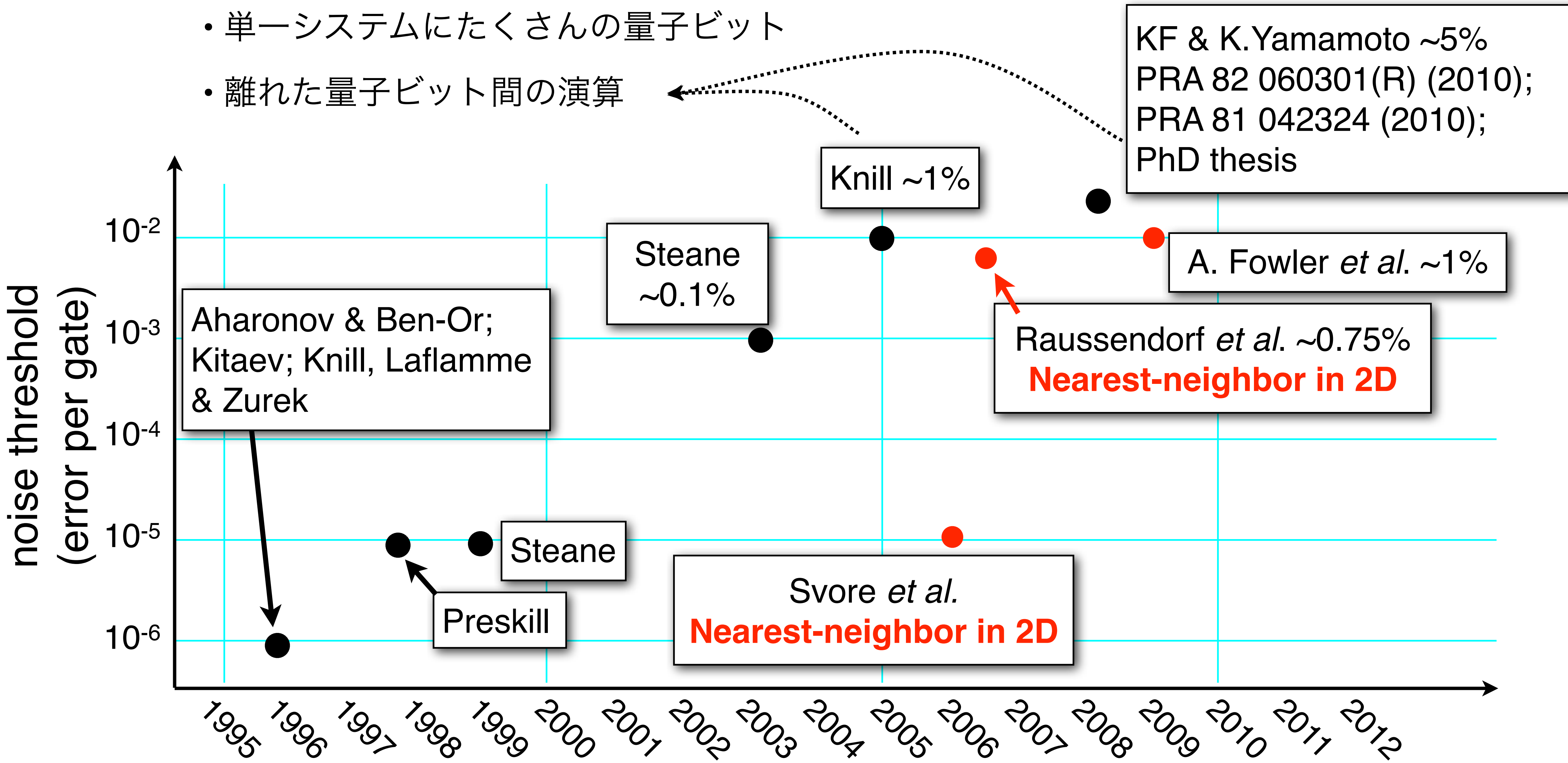
Noise thresholdの歴史

- 単一システムにたくさんの量子ビット
- 離れた量子ビット間の演算



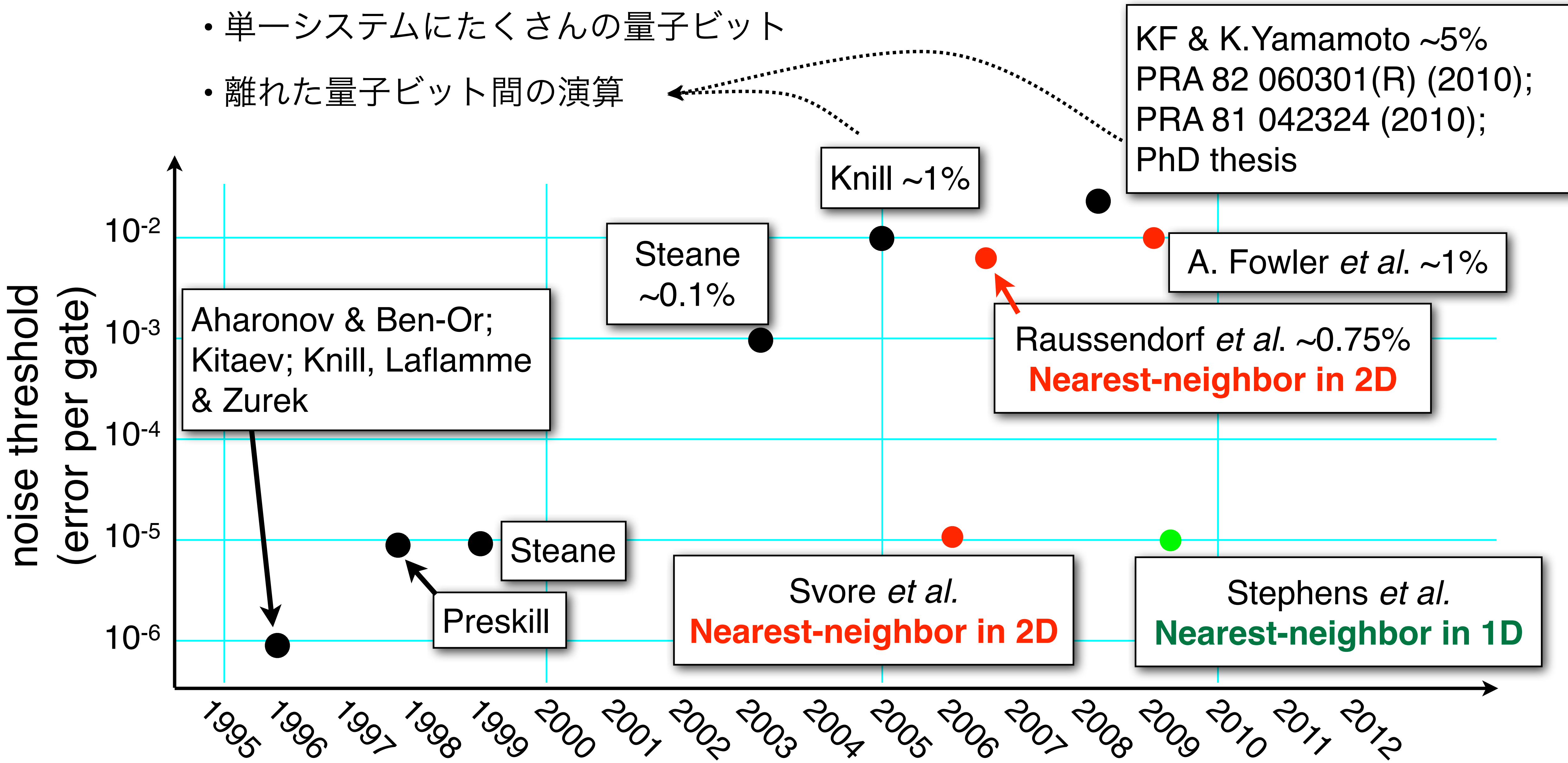
Noise thresholdの歴史

- 単一システムにたくさんの量子ビット
- 離れた量子ビット間の演算



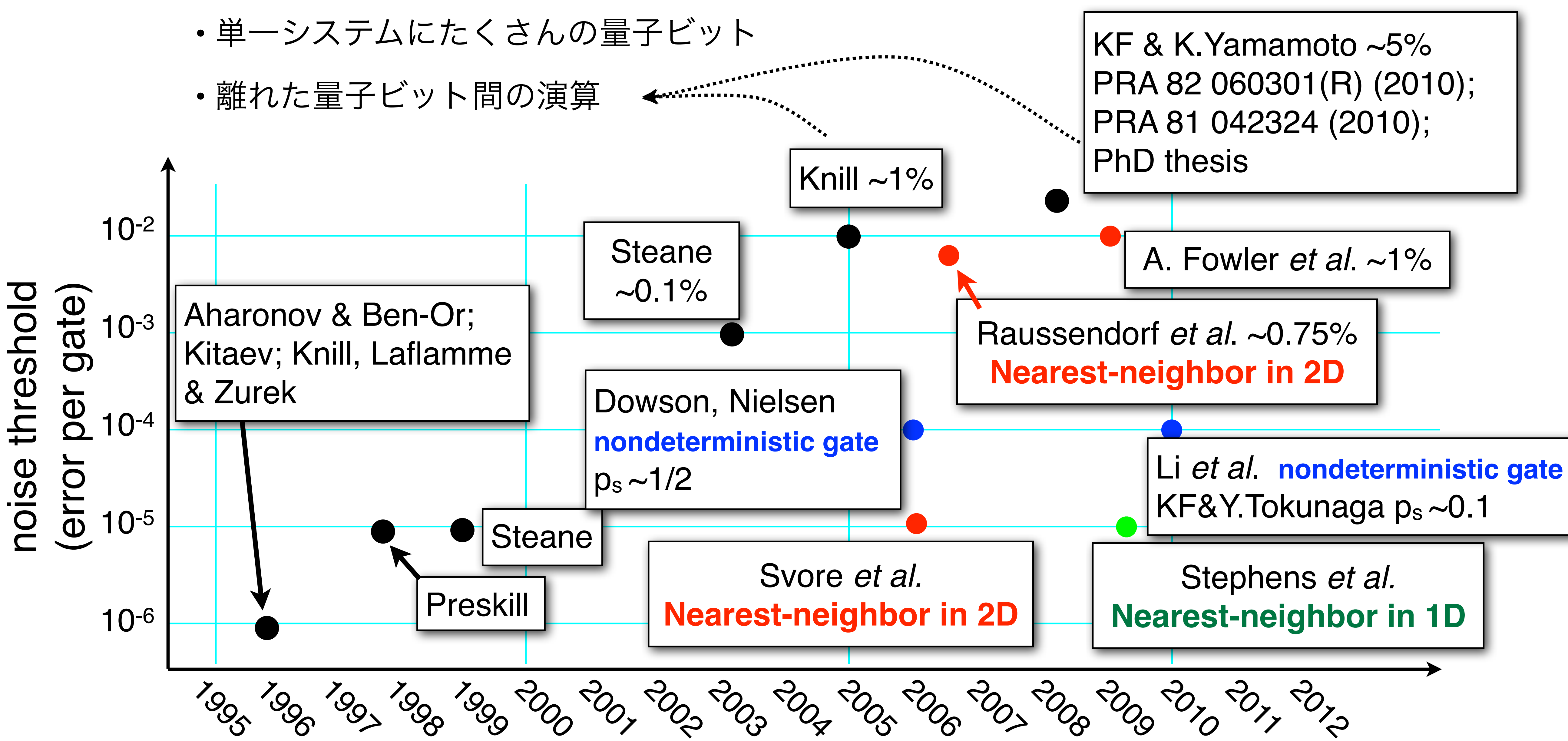
Noise thresholdの歴史

- 単一システムにたくさんの量子ビット
- 離れた量子ビット間の演算



Noise thresholdの歴史

- 単一システムにたくさんの量子ビット
- 離れた量子ビット間の演算



第四章: Surface符号上での量子誤り訂正

勉強する内容：

Surface codeの導入.

誤り訂正法，性能評価の方法,

スピングラス模型との対応について

準備：chain complex

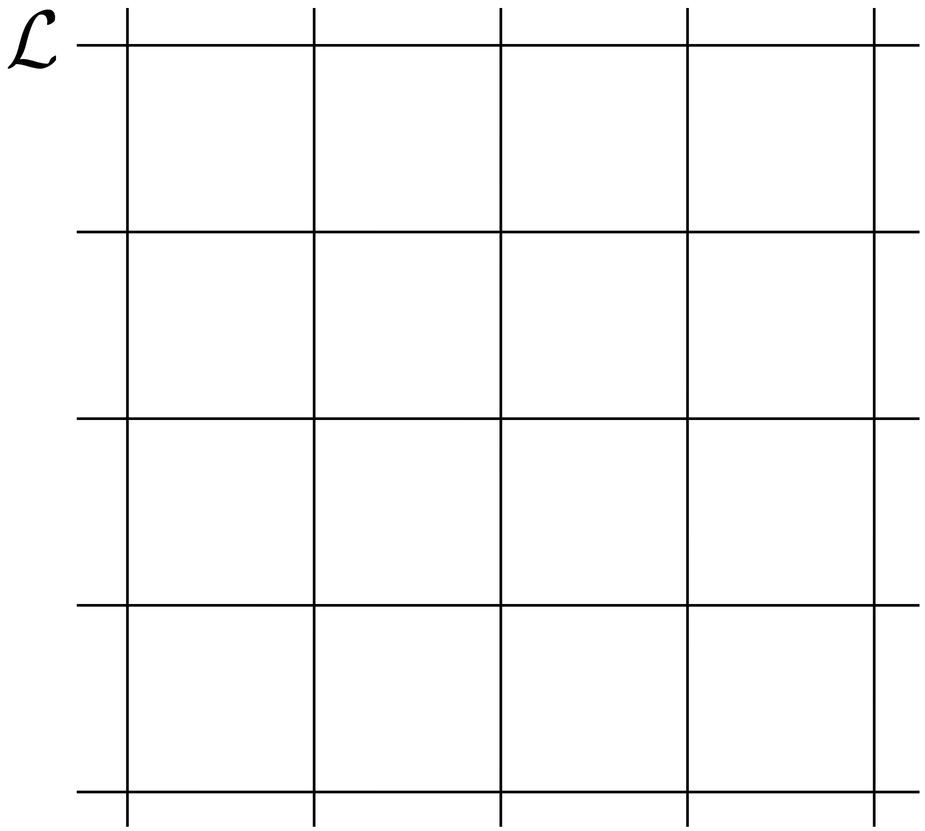
◆格子上の鎖複体（chain complex）を定義しておく：

格子: $\mathcal{L}$

頂点の集合： $V = \{v_k\}$

辺の集合: $E = \{e_l\}$

面の集合: $F = \{f_i\}$



準備：chain complex

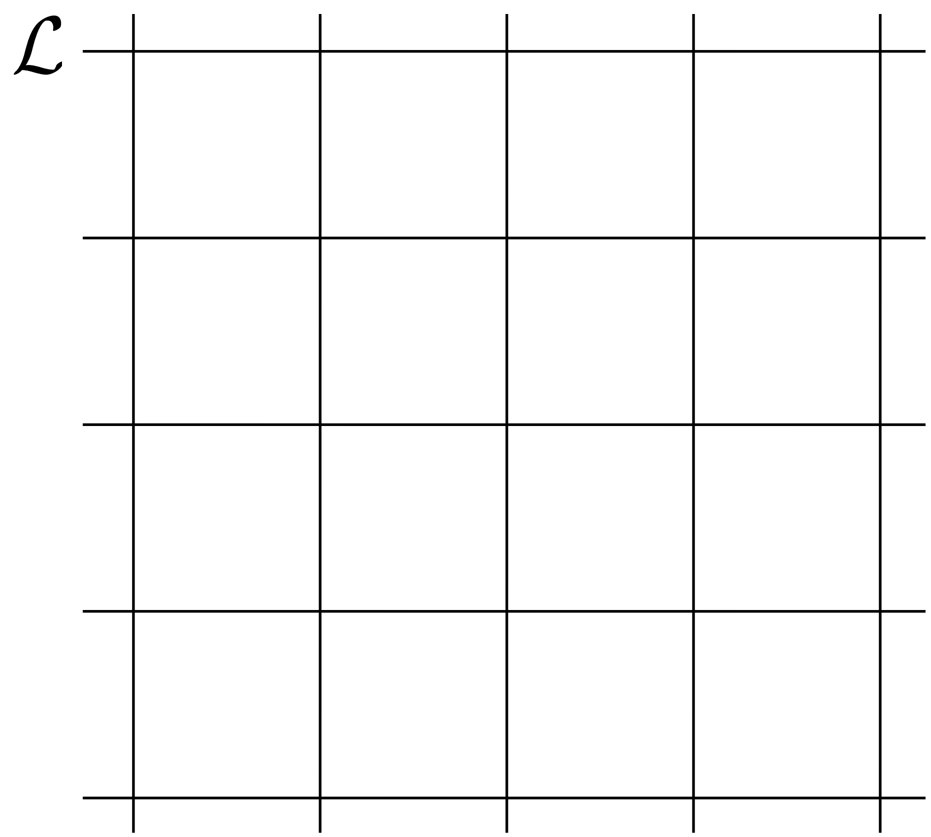
◆格子上の鎖複体（chain complex）を定義しておく：

格子: $\mathcal{L}$

頂点の集合： $V = \{v_k\}$

辺の集合: $E = \{e_l\}$

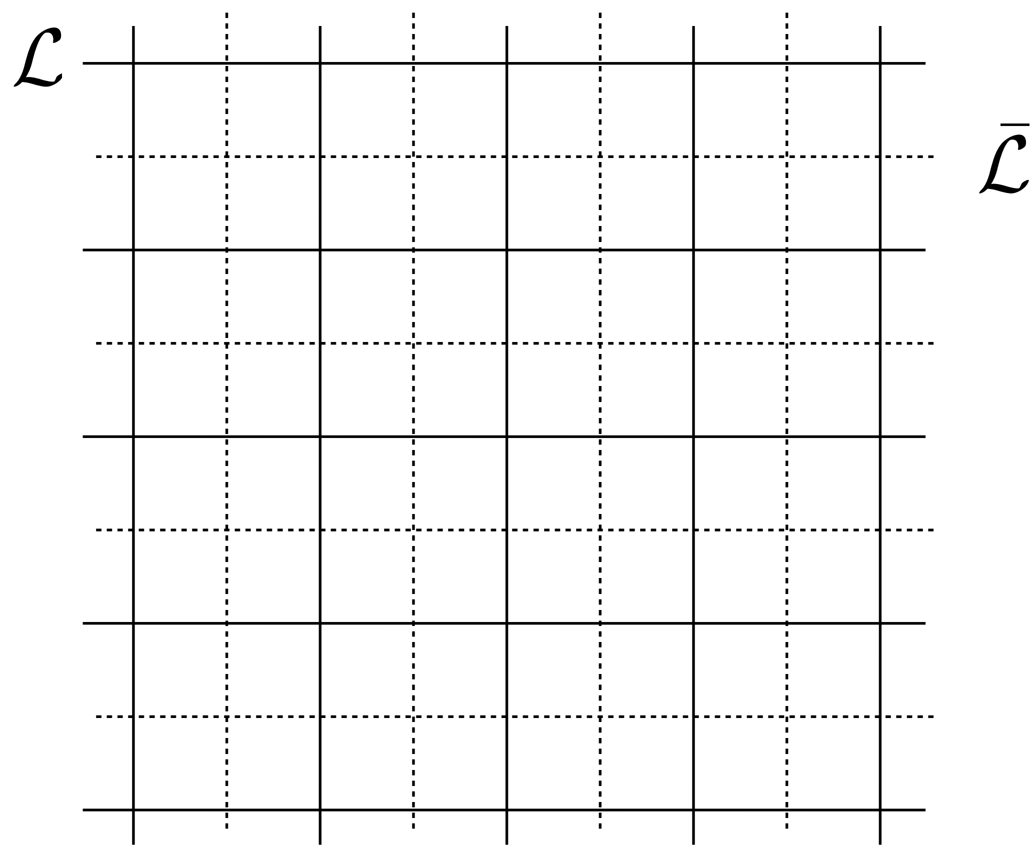
面の集合: $F = \{f_i\}$



準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

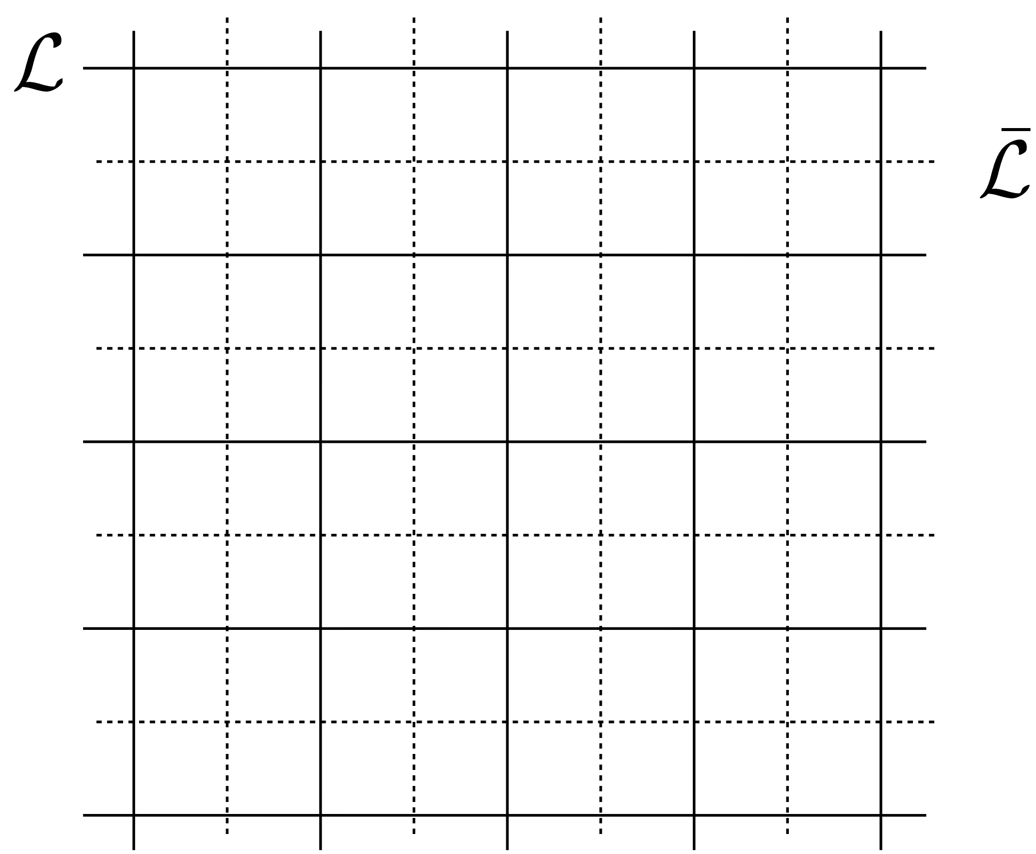
格子: $\mathcal{L}$	双対格子: $\bar{\mathcal{L}}$
頂点の集合： $V = \{v_k\}$	頂点の集合： $\bar{V} = \{\bar{v}_k\}$
辺の集合: $E = \{e_l\}$	辺の集合: $\bar{E} = \{\bar{e}_l\}$
面の集合: $F = \{f_i\}$	面の集合: $\bar{F} = \{\bar{f}_i\}$



準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

係数の和で、chainの足し算が定義される→ chainは加群

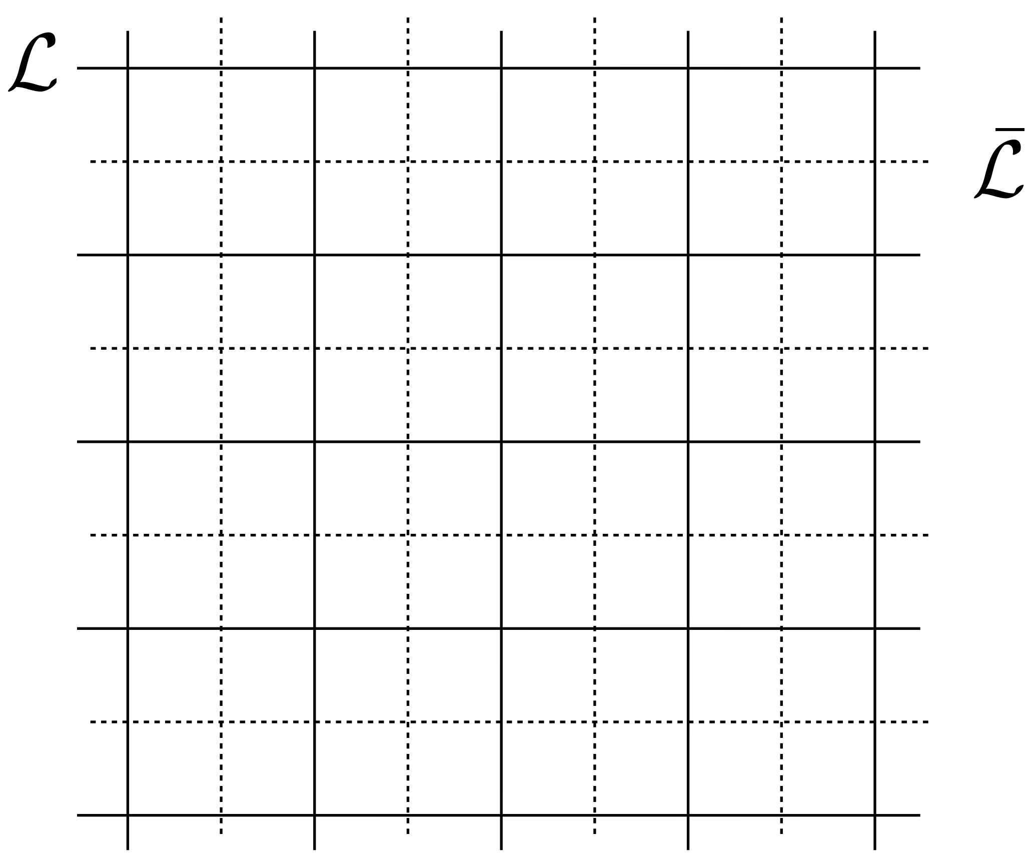


準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

- 0-chain: $c_0 = \sum_k z_k v_k \quad \in C_0$ 　　($z_l \in \{0,1\}$)

係数の和で、chainの足し算が定義される→ chainは加群

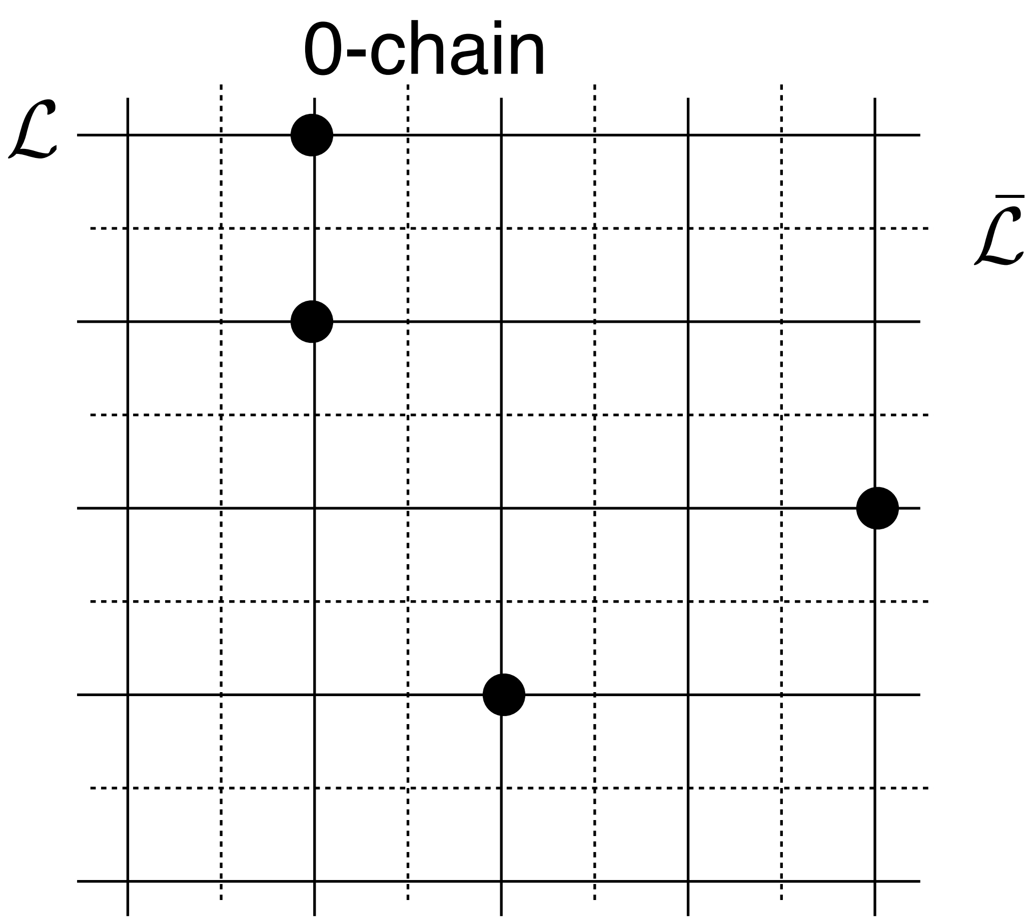


準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

- 0-chain: $c_0 = \sum_k z_k v_k \quad \in C_0$ ($z_l \in \{0,1\}$)

係数の和で、chainの足し算が定義される→ chainは加群

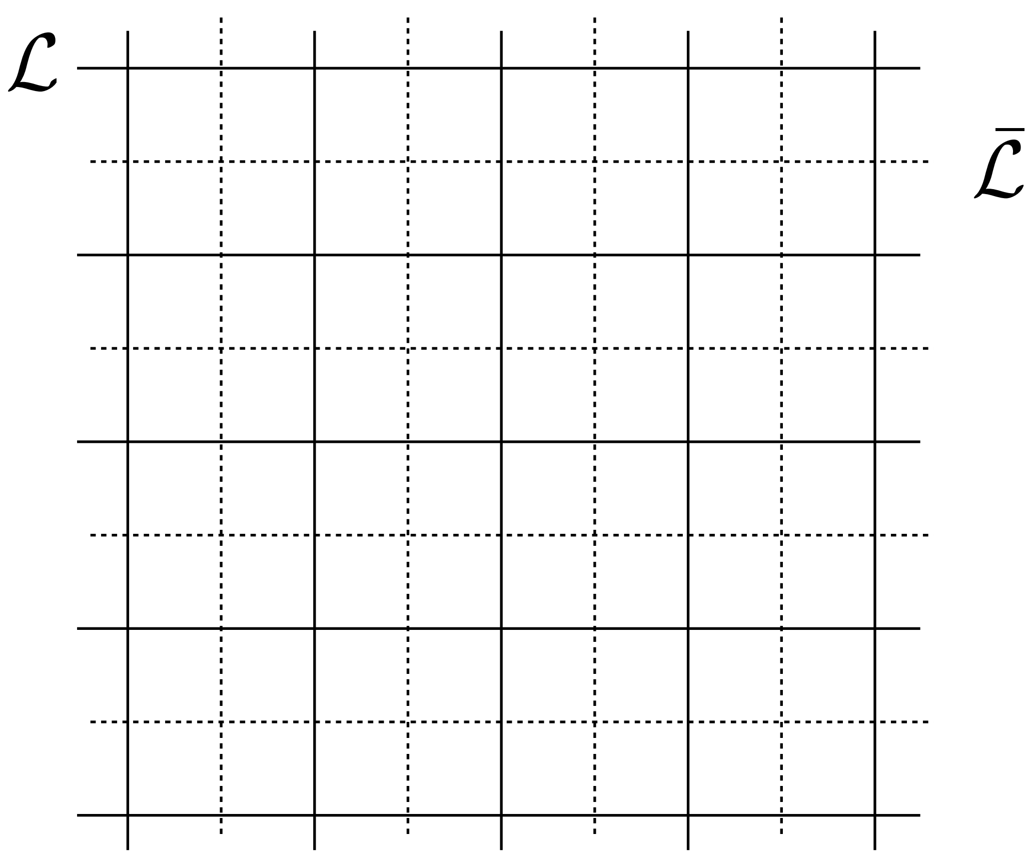


準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

- 0-chain: $c_0 = \sum_k z_k v_k \in C_0$ 　　　($z_l \in \{0, 1\}$)

係数の和で、chainの足し算が定義される→ chainは加群

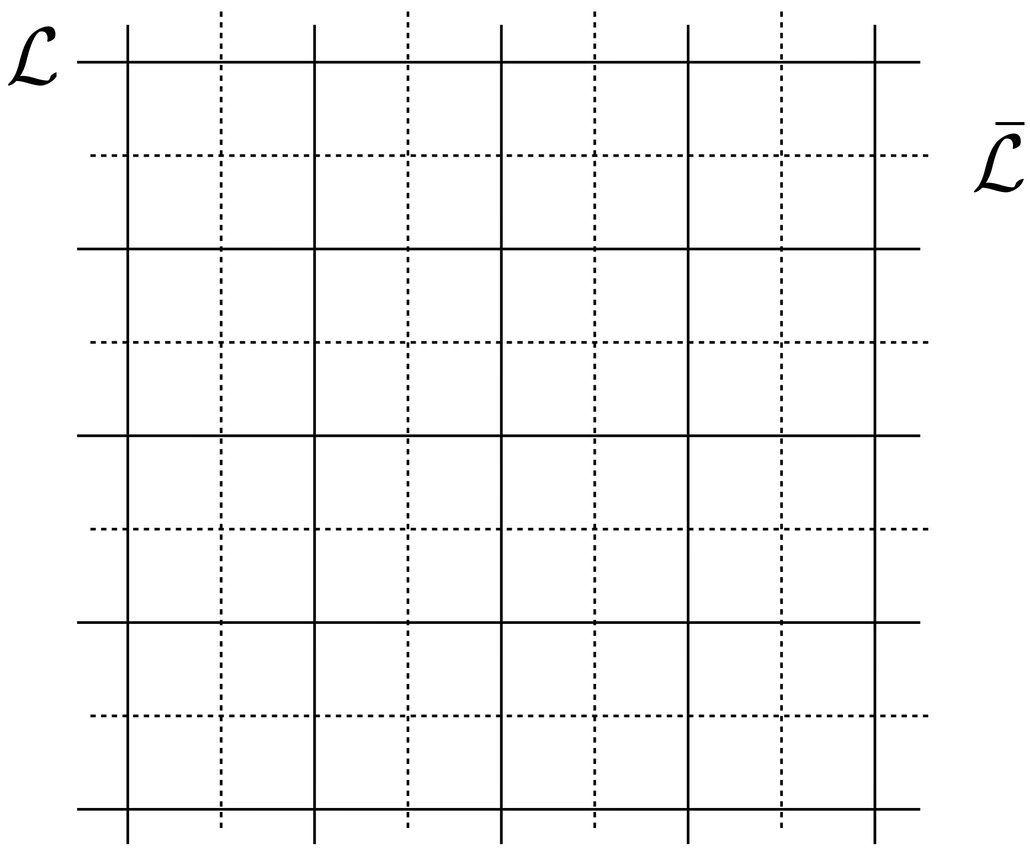


準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

- 0-chain: $c_0 = \sum_k z_k v_k \quad \in C_0$ ($z_l \in \{0, 1\}$)
- 1-chain: $c_1 = \sum_l z_l e_l \quad \in C_1$

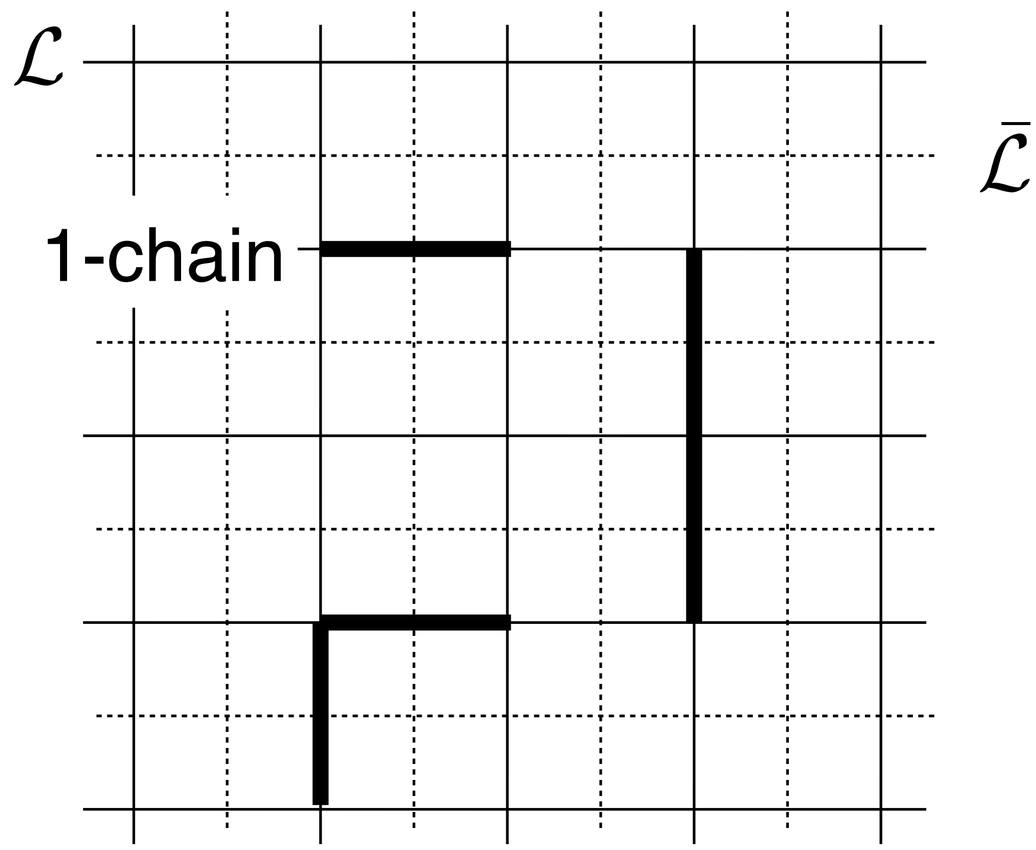
係数の和で、chainの足し算が定義される→ chainは加群



準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

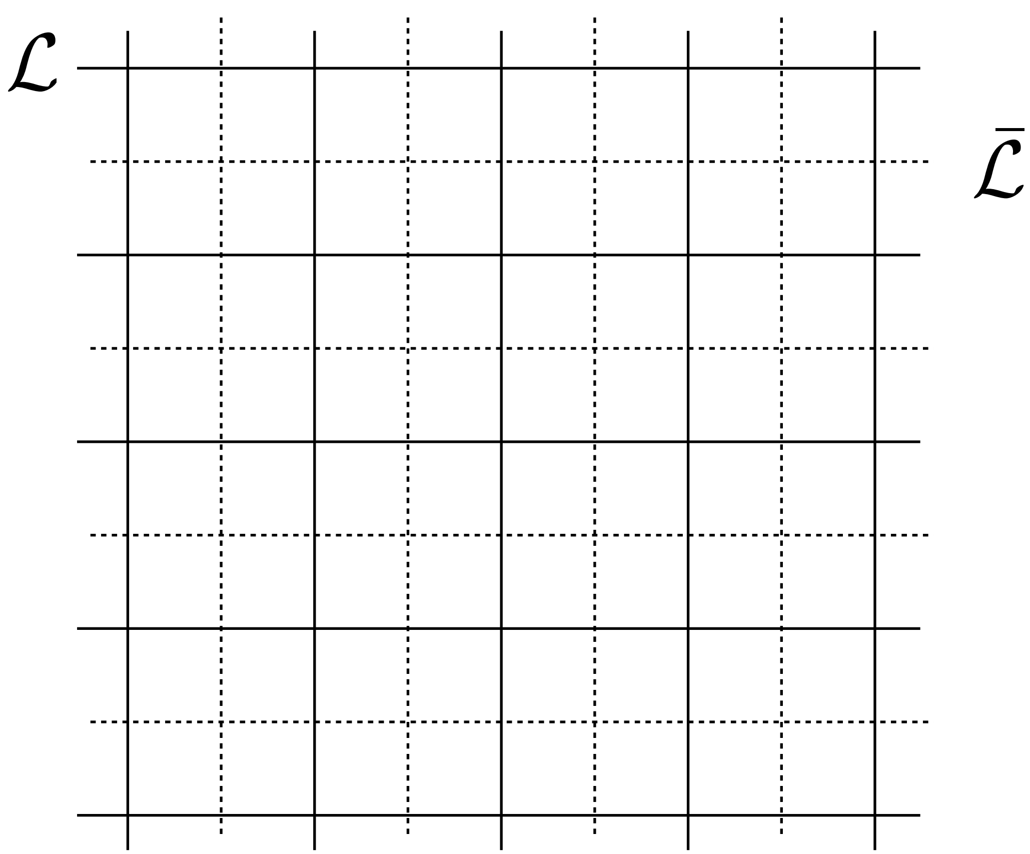
- 0-chain: $c_0 = \sum_k z_k v_k \quad \in C_0$ 　　($z_l \in \{0,1\}$)
- 1-chain: $c_1 = \sum_l z_l e_l \quad \in C_1$ 　　係数の和で、chainの足し算が定義される→ chainは加群



準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

- 0-chain: $c_0 = \sum_k z_k v_k \quad \in C_0$ 　　($z_l \in \{0,1\}$)
- 1-chain: $c_1 = \sum_l z_l e_l \quad \in C_1$ 　　係数の和で、chainの足し算が定義される→ chainは加群



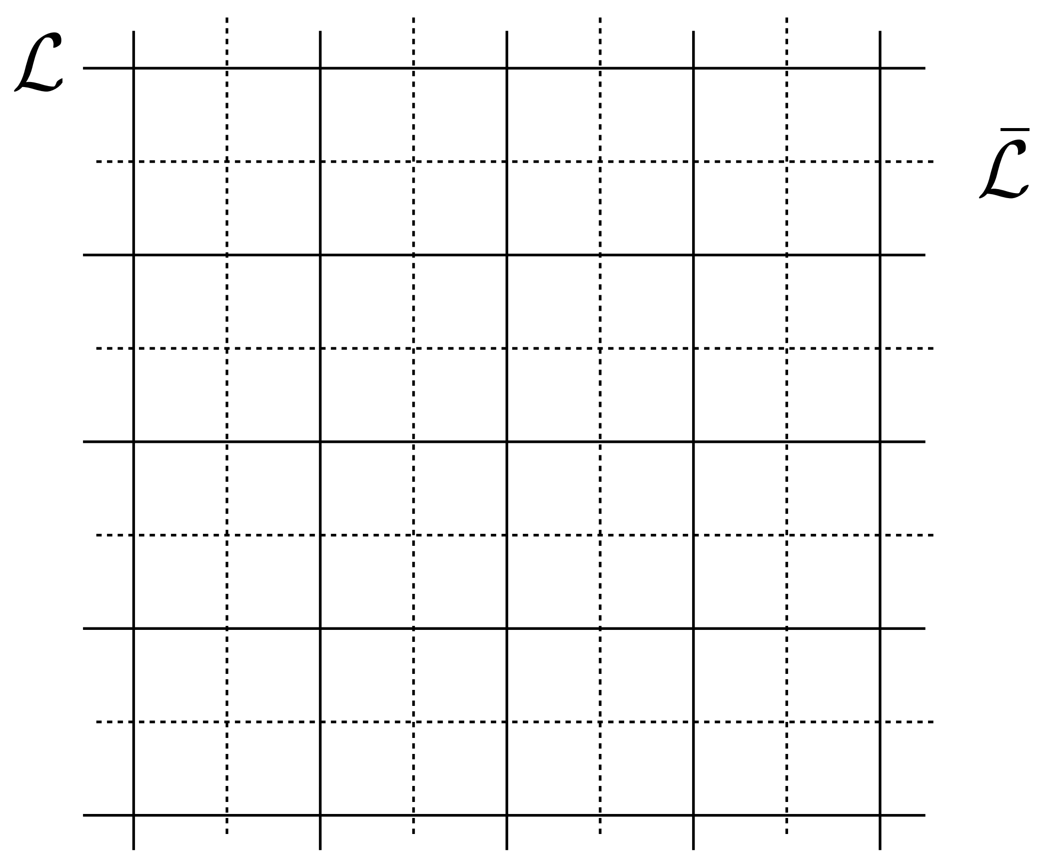
準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

- 0-chain: $c_0 = \sum_k z_k v_k \quad \in C_0$
- 1-chain: $c_1 = \sum_l z_l e_l \quad \in C_1$
- 2-chain: $c_2 = \sum_i z_i f_i \quad \in C_2$

($z_l \in \{0, 1\}$)

係数の和で、chainの足し算が定義される→ chainは加群

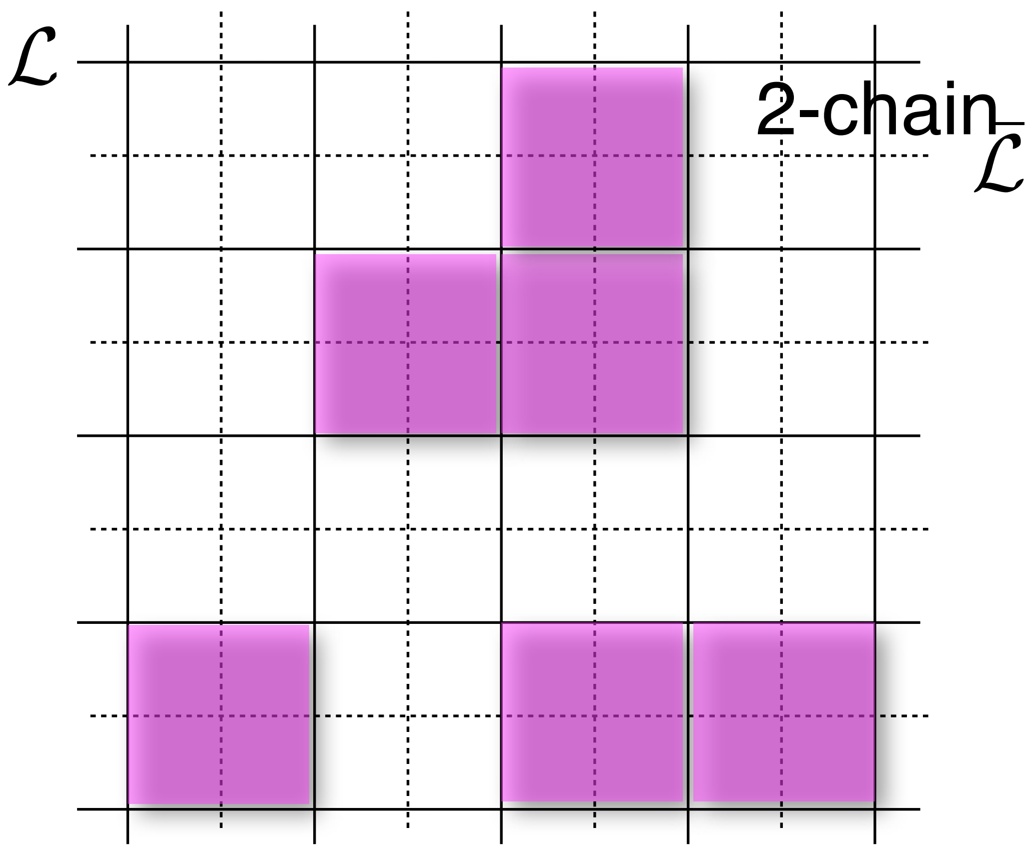


準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

- 0-chain: $c_0 = \sum_k z_k v_k \quad \in C_0$ 　　($z_l \in \{0, 1\}$)
- 1-chain: $c_1 = \sum_l z_l e_l \quad \in C_1$
- 2-chain: $c_2 = \sum_i z_i f_i \quad \in C_2$

係数の和で、chainの足し算が定義される→ chainは加群

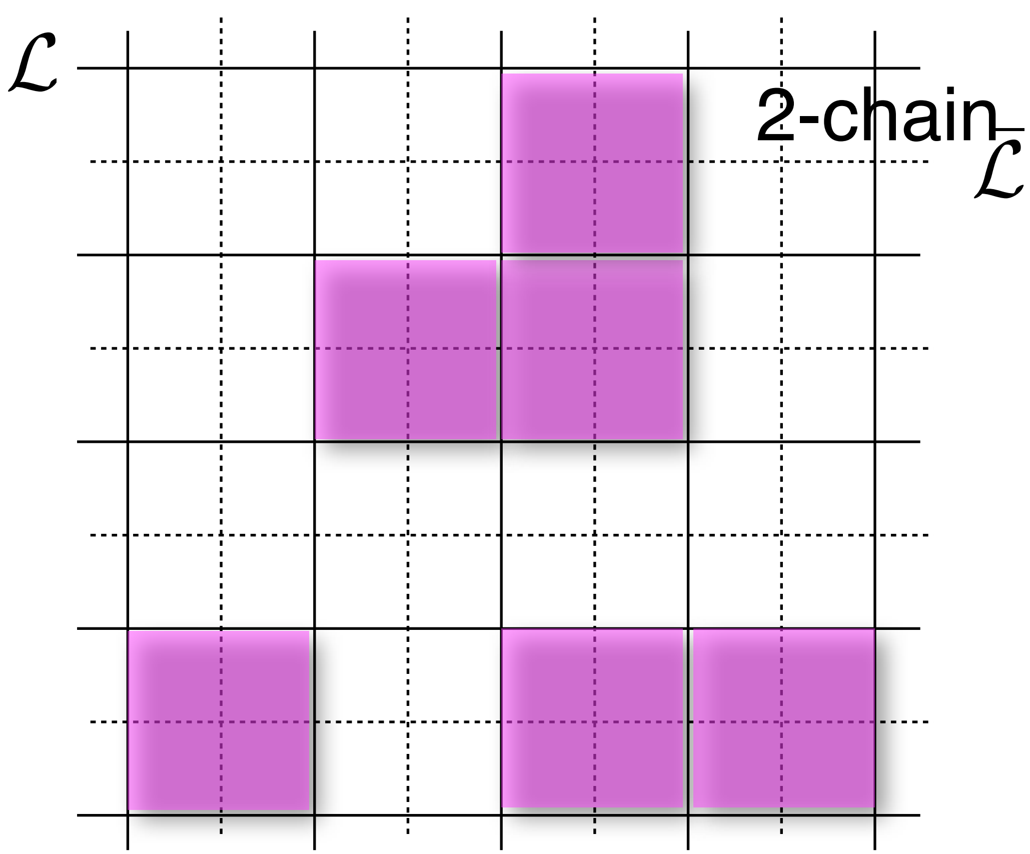


準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

- 0-chain: $c_0 = \sum_k z_k v_k \quad \in C_0$ $\qquad$ ($z_l \in \{0,1\}$)
- 1-chain: $c_1 = \sum_l z_l e_l \quad \in C_1$
- 2-chain: $c_2 = \sum_i z_i f_i \quad \in C_2$

係数の和で、chainの足し算が定義される→ chainは加群



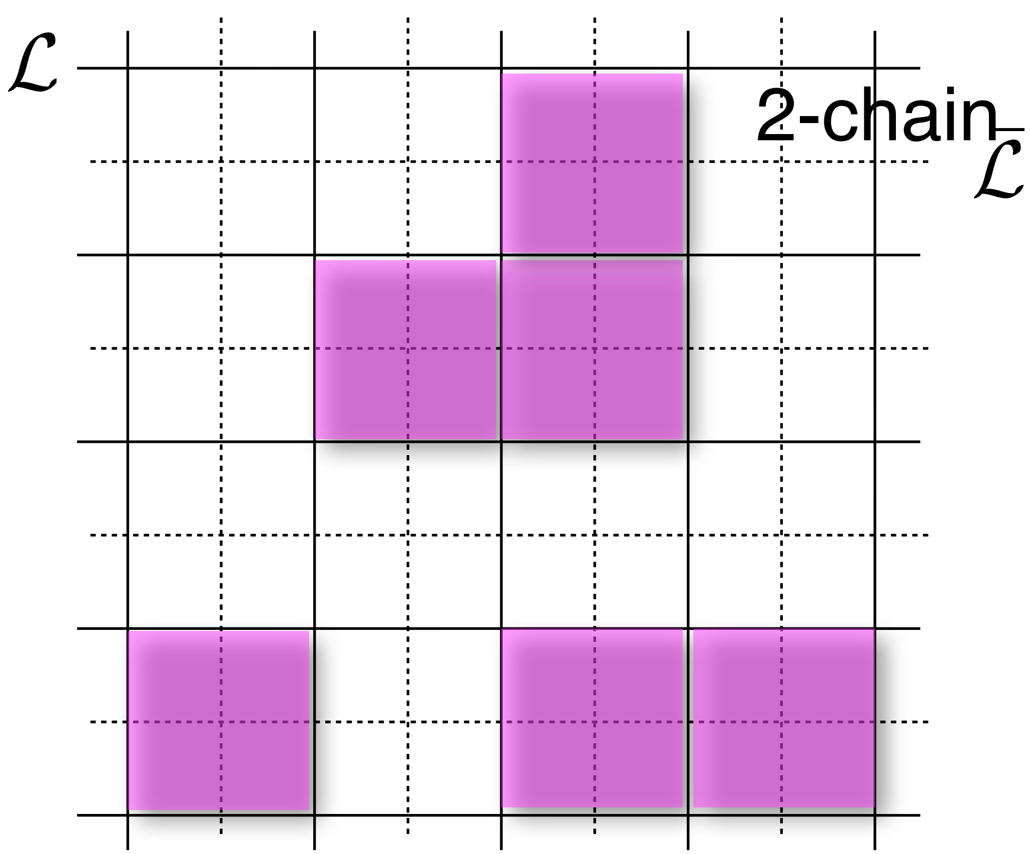
- 辺上にqubitが配置されたときの演算子: $A(c_1) = \prod_l A_l^{z_l}$

準備：chain complex

◆格子上の鎖複体（chain complex）を定義しておく：

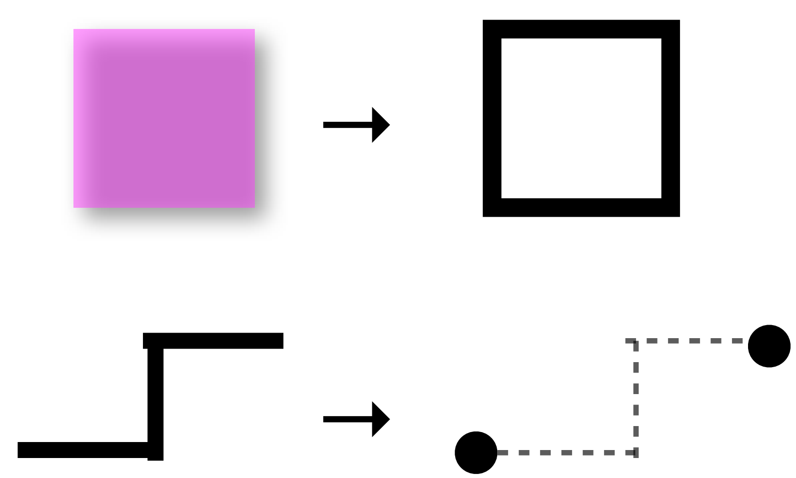
- 0-chain: $c_0 = \sum_k z_k v_k \in C_0$ 　　$(\ z_l \in \{0,1\}\ )$
- 1-chain: $c_1 = \sum_l z_l e_l \in C_1$
- 2-chain: $c_2 = \sum_i z_i f_i \in C_2$

係数の和で、chainの足し算が定義される→ chainは加群



- 辺上にqubitが配置されたときの演算子: $A(c_1) = \prod_l A_l^{z_l}$
- boundary map:

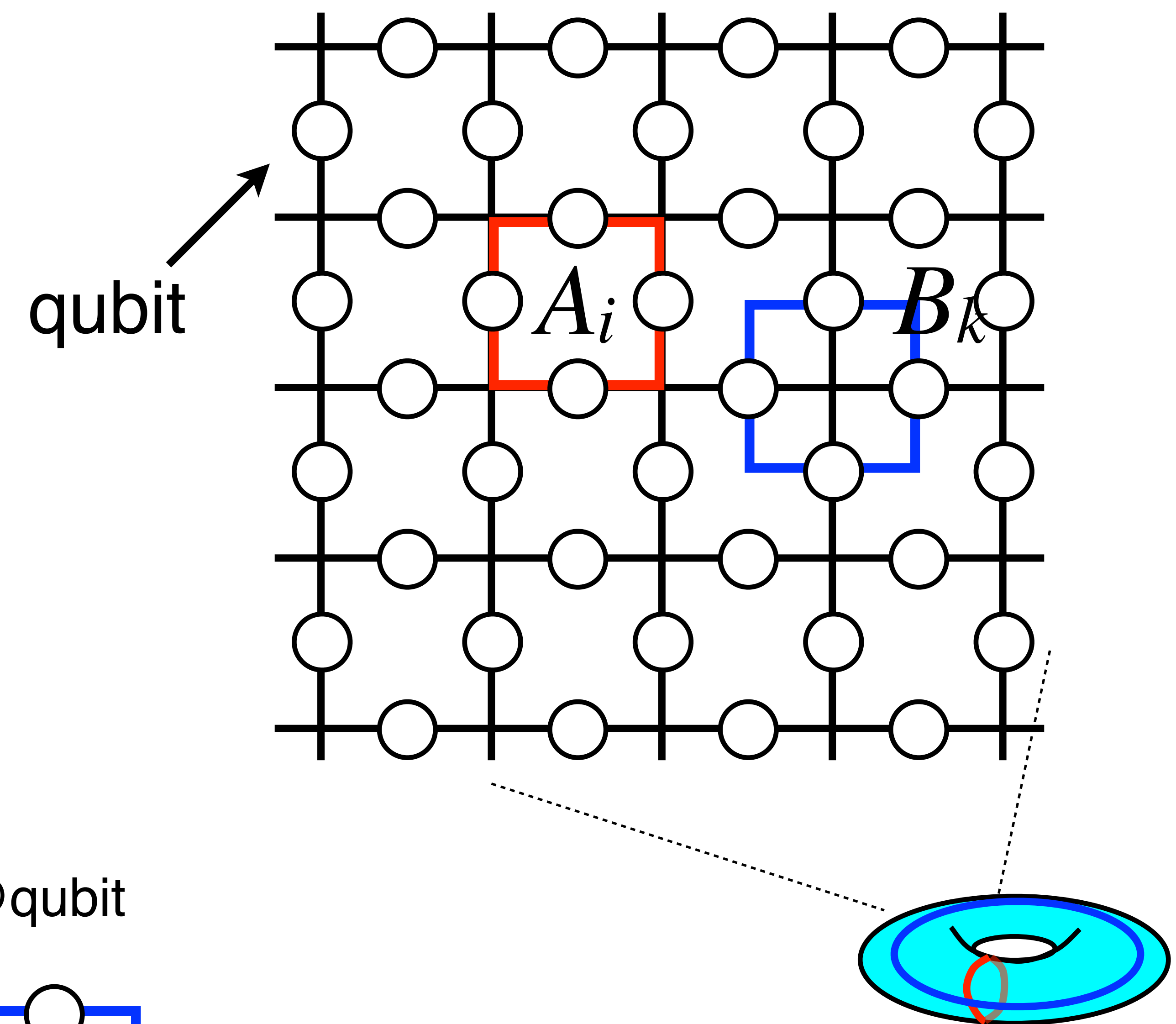
$\partial : C_{i+1} \to C_i$ 　　$\partial \circ \partial = 0$



Surface code状態

Surface code (Kitaev's toric code)

Kitaev, Annals Phys. **303**, 2 (2003)
arXiv:1997



スタビライザー演算子

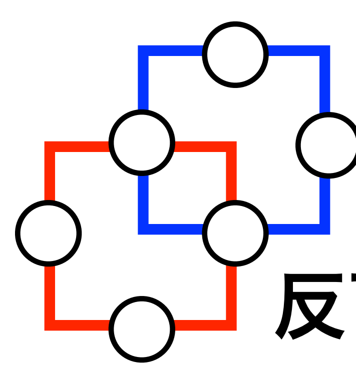
$$A_i = \prod_{l \in \partial f_i} Z_l = Z(\partial f_i),$$

面上の４つのqubit

$$B_k = \prod_{\bar{l} \in \partial \bar{f}_k} X_{\bar{l}} = X(\partial \bar{f}_k).$$

頂点上に隣接する４つのqubit

これらの演算子はすべて可換！

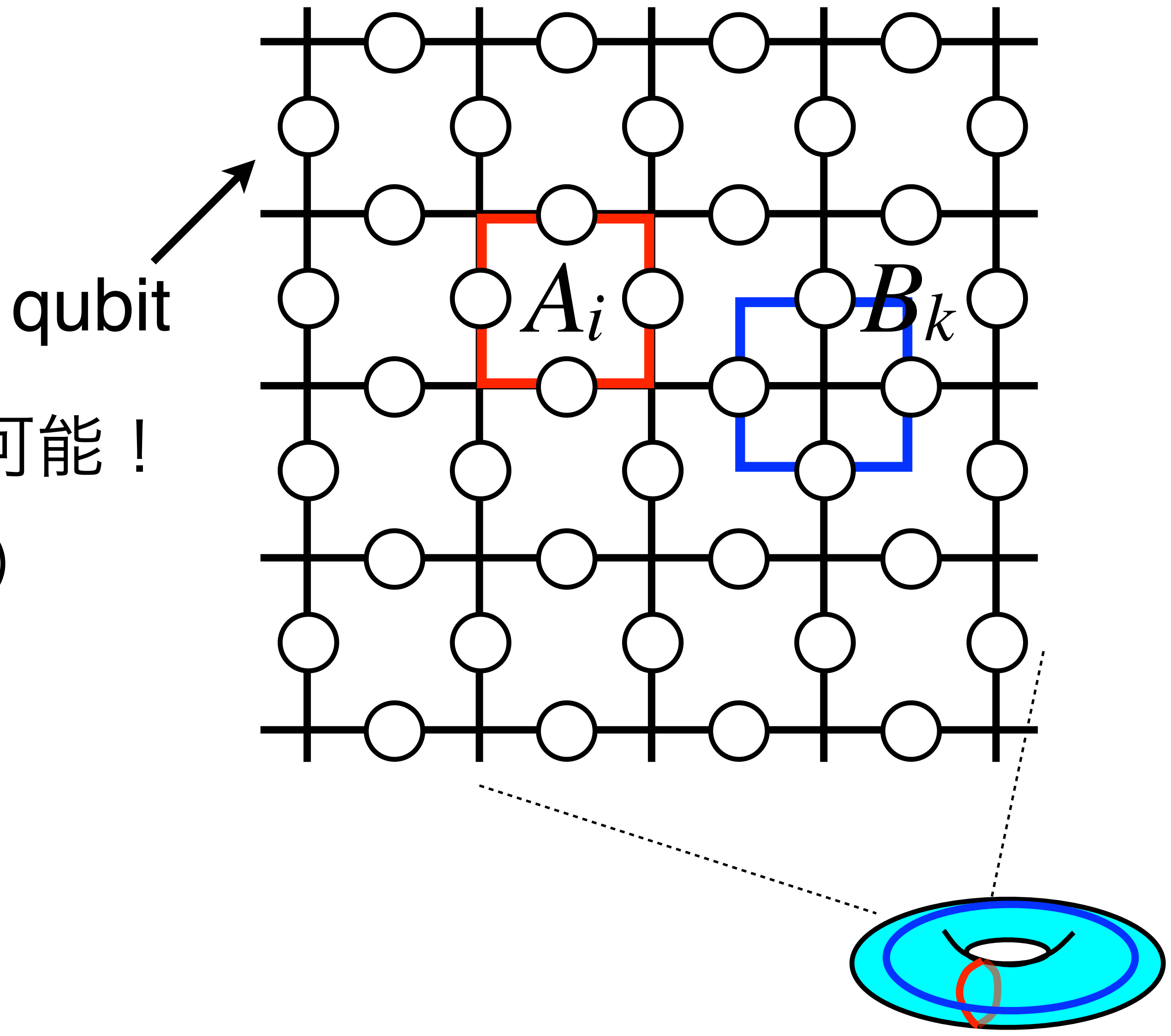


反可換×反可換＝可換

Surface code状態

Surface code (Kitaev's toric code)

Kitaev, Annals Phys. **303**, 2 (2003)



可換なエルミート演算子→同時対角化可能！

（パウリ演算子の固有値は+1 or -1）

すべての面，頂点の演算子に対して

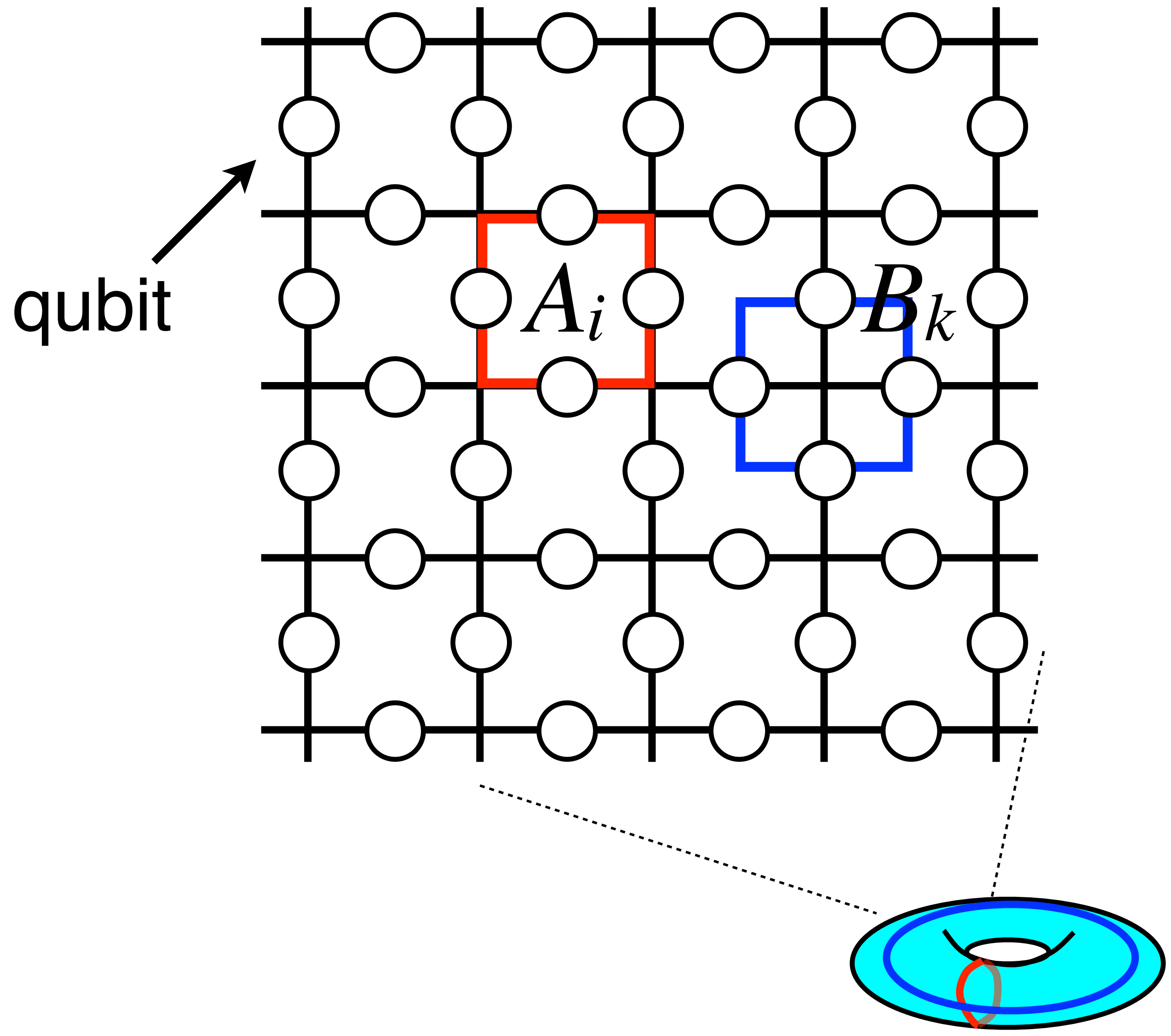
$$A_i|\Psi\rangle = |\Psi\rangle, \quad B_k|\Psi\rangle = |\Psi\rangle$$

を満たす状態→**surface code 状態**

Surface code状態

Surface code (Kitaev's toric code)

Kitaev, Annals Phys. **303**, 2 (2003)



状態が一意的に定まるのだろうか？

自由度の勘定：

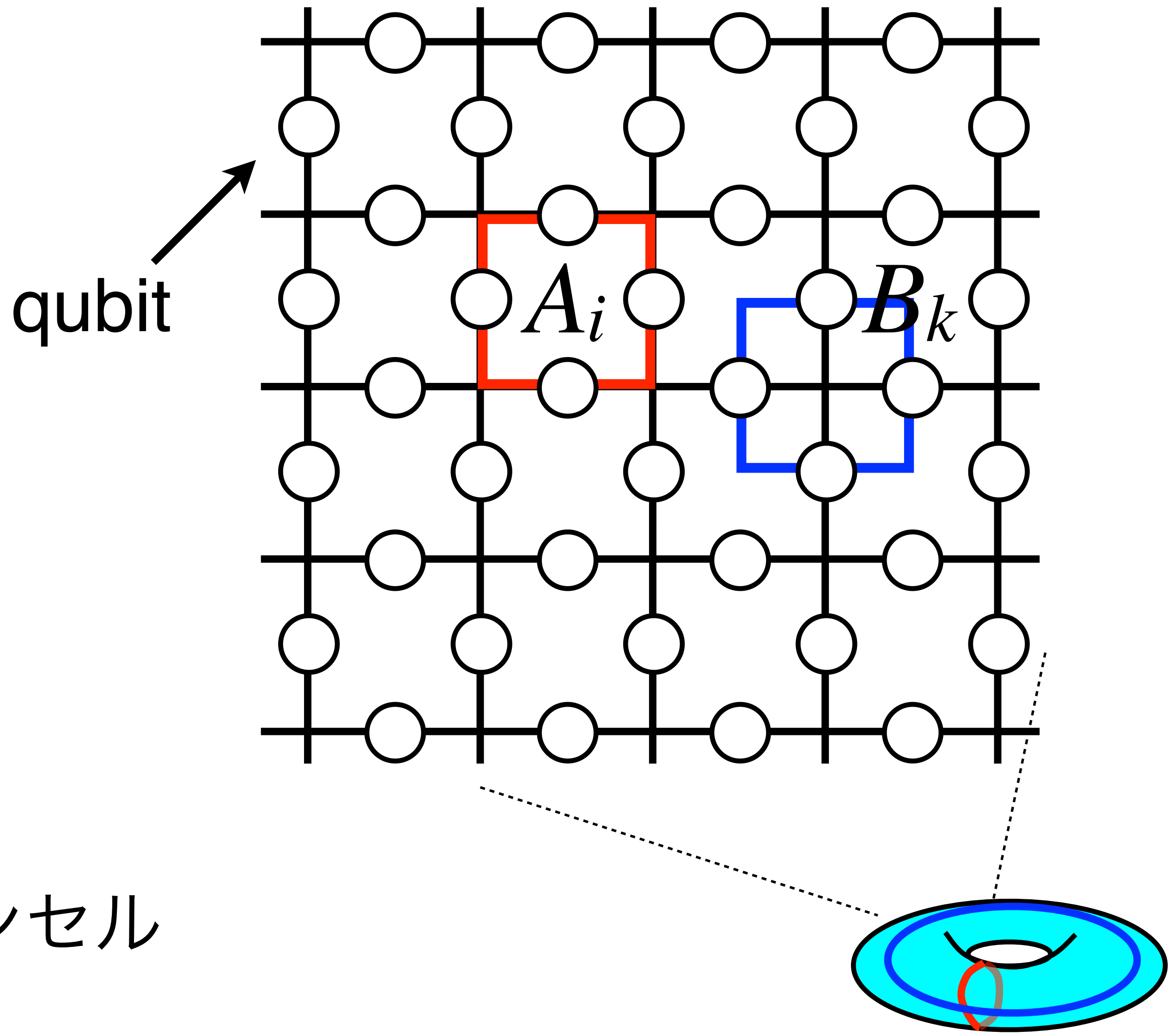
#qubit = |E| on N×N torus → $2N^2$

#generator = (|F| + |V| - 2) → $2N^2$ - 2

Surface code状態

Surface code (Kitaev's toric code)

Kitaev, Annals Phys. **303**, 2 (2003)



状態が一意的に定まるのだろうか？

自由度の勘定：

#qubit = $|E|$ on $N\times N$ torus → $2N^2$

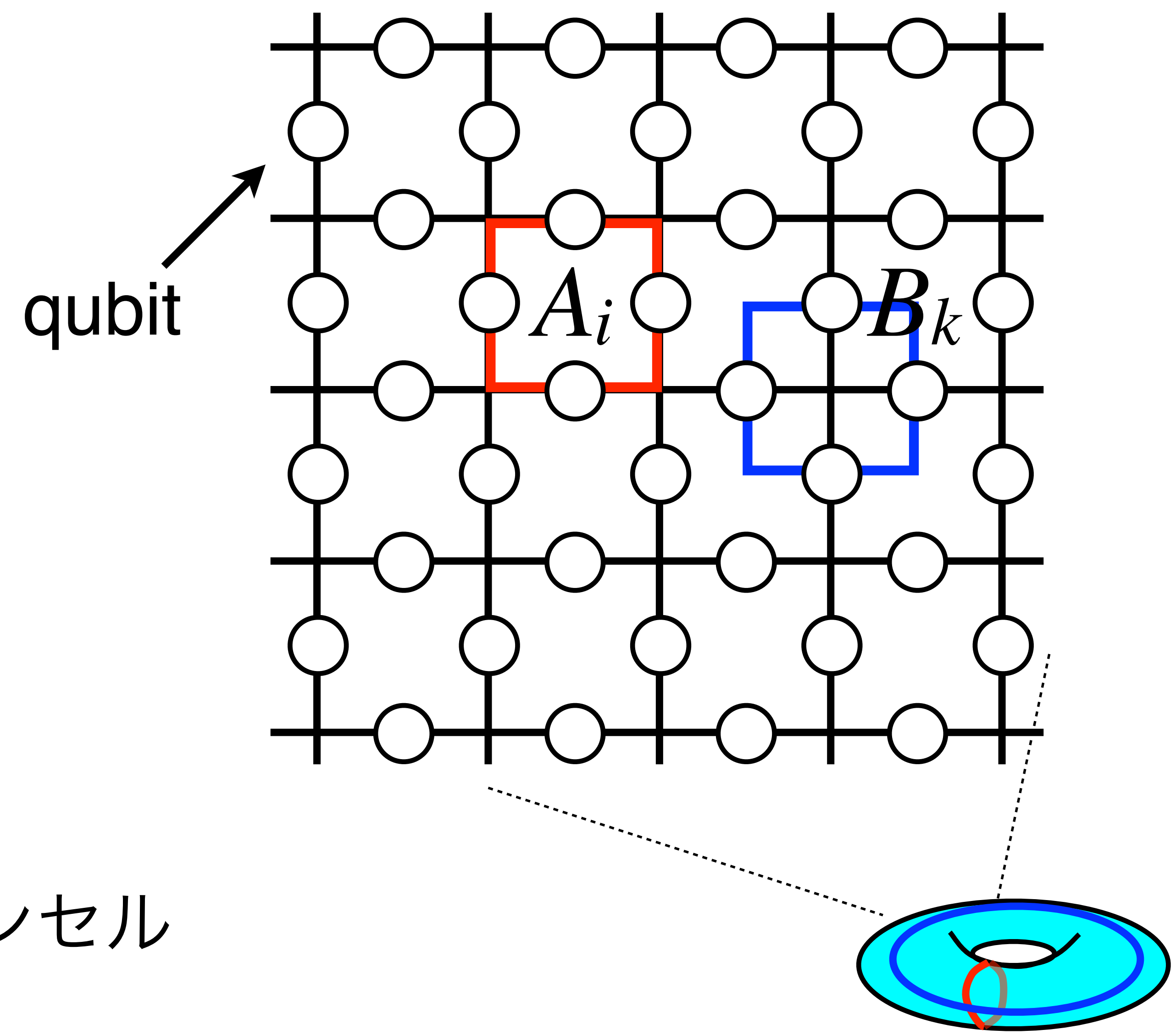
#generator = $(|F| + |V| - \boxed{2})$ → $2N^2 - 2$

全部掛け合わせるとキャンセル

Surface code状態

Surface code (Kitaev's toric code)

Kitaev, Annals Phys. **303**, 2 (2003)



状態が一意的に定まるのだろうか？

自由度の勘定：

#qubit = $|E|$ on $N\times N$ torus → $2N^2$

#generator = $(|F| + |V| - \boxed{2})$ → $2N^2 - 2$

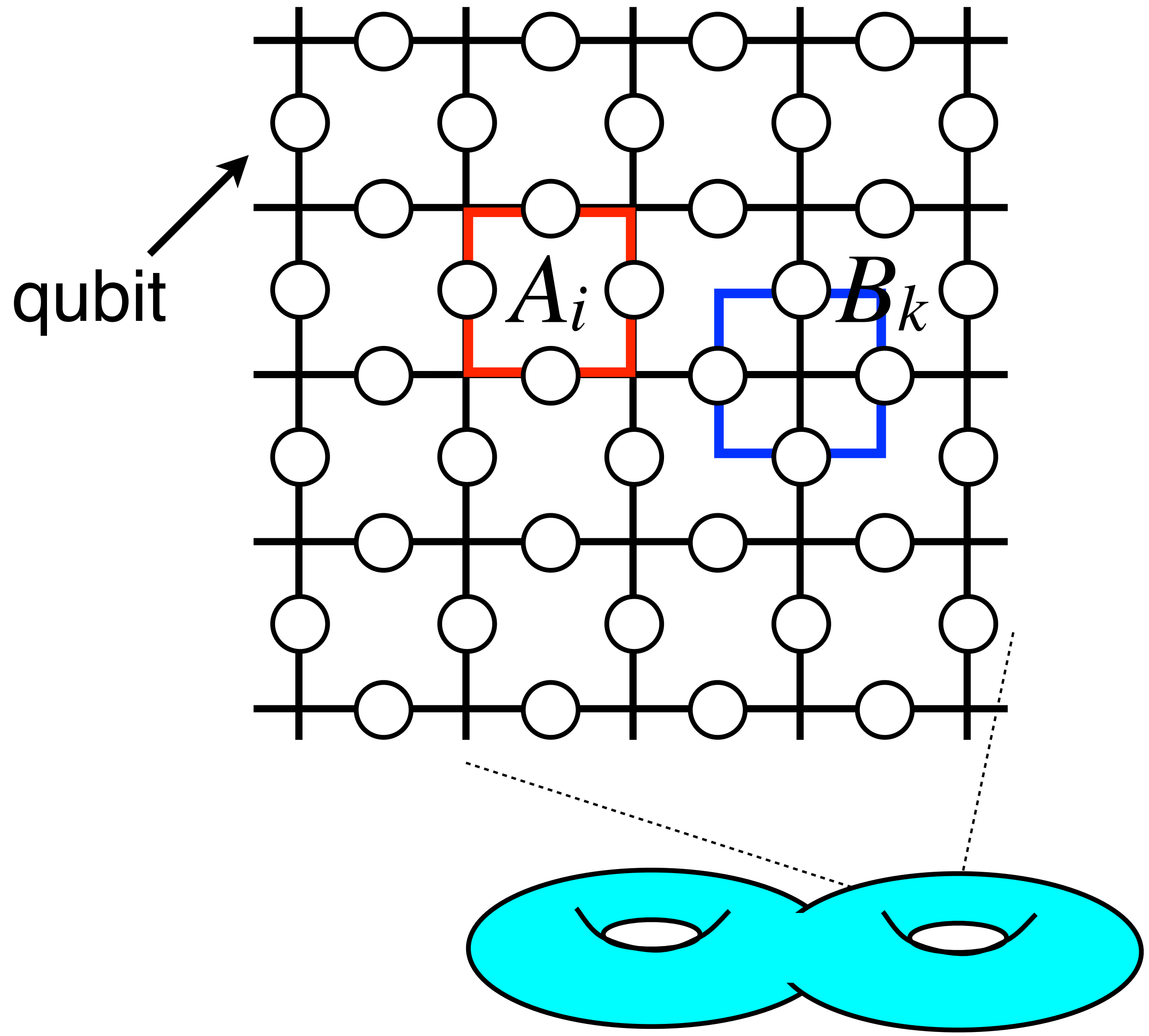
全部掛け合わせるとキャンセル

→4次元の縮退がある（qubit ２個分）

Surface code状態

Surface code (Kitaev's toric code)

Kitaev, Annals Phys. **303**, 2 (2003)



状態が一意的に定まるのだろうか？

自由度の勘定：

$$\underbrace{|F| + |V| - |E|}_{\text{Euler標数}} = 2 - 2g \qquad (g = \text{種数})$$

#logical qubits= $|E| - (|F| + |V| - 2) = 2g$

→2g個の qubit の縮退がある.

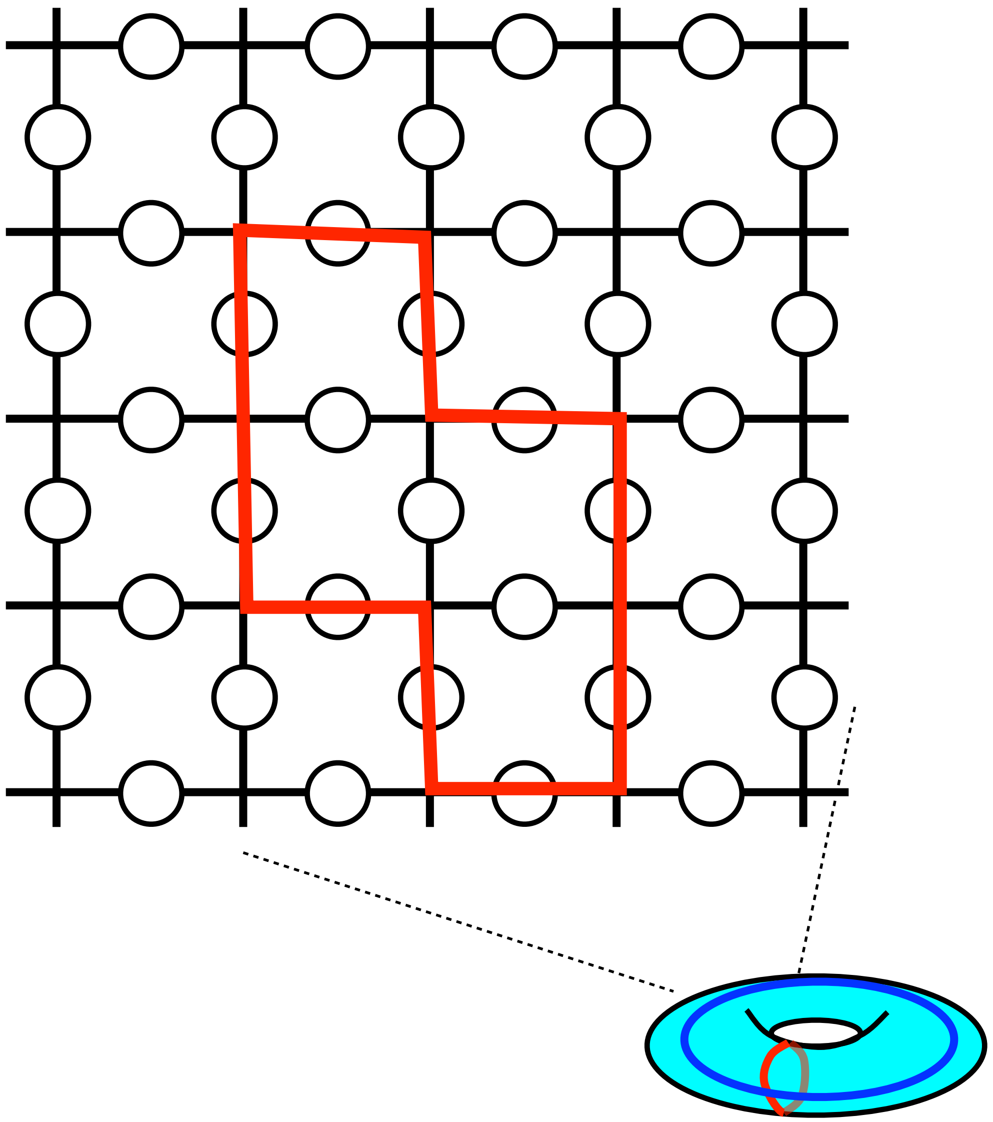
Surface code状態

Surface code (Kitaev's toric code)

Kitaev, Annals Phys. **303**, 2 (2003)

縮退をとく演算子=論理演算子は？

→スタビライザー演算子と可換な演算子



Surface code状態

Surface code (Kitaev's toric code)

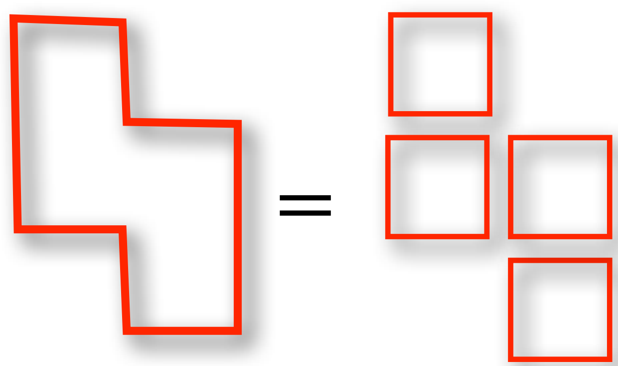
Kitaev, Annals Phys. **303**, 2 (2003)

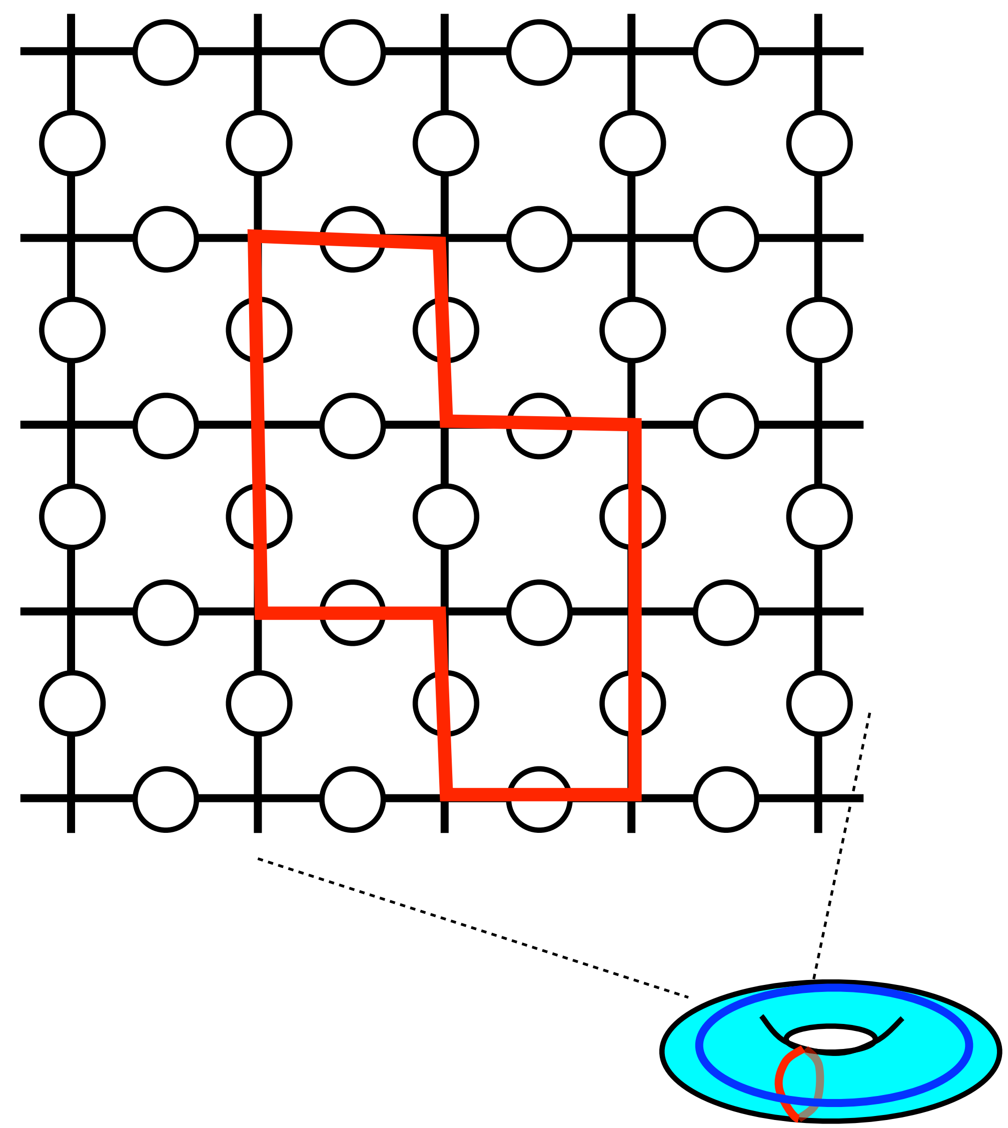
縮退をとく演算子=論理演算子は？

→スタビライザー演算子と可換な演算子

注）スタビライザーは群なので，スタビライザーの

積, $Z(\partial c_2), X(\partial \bar{c}_2)$ はスタビライザー

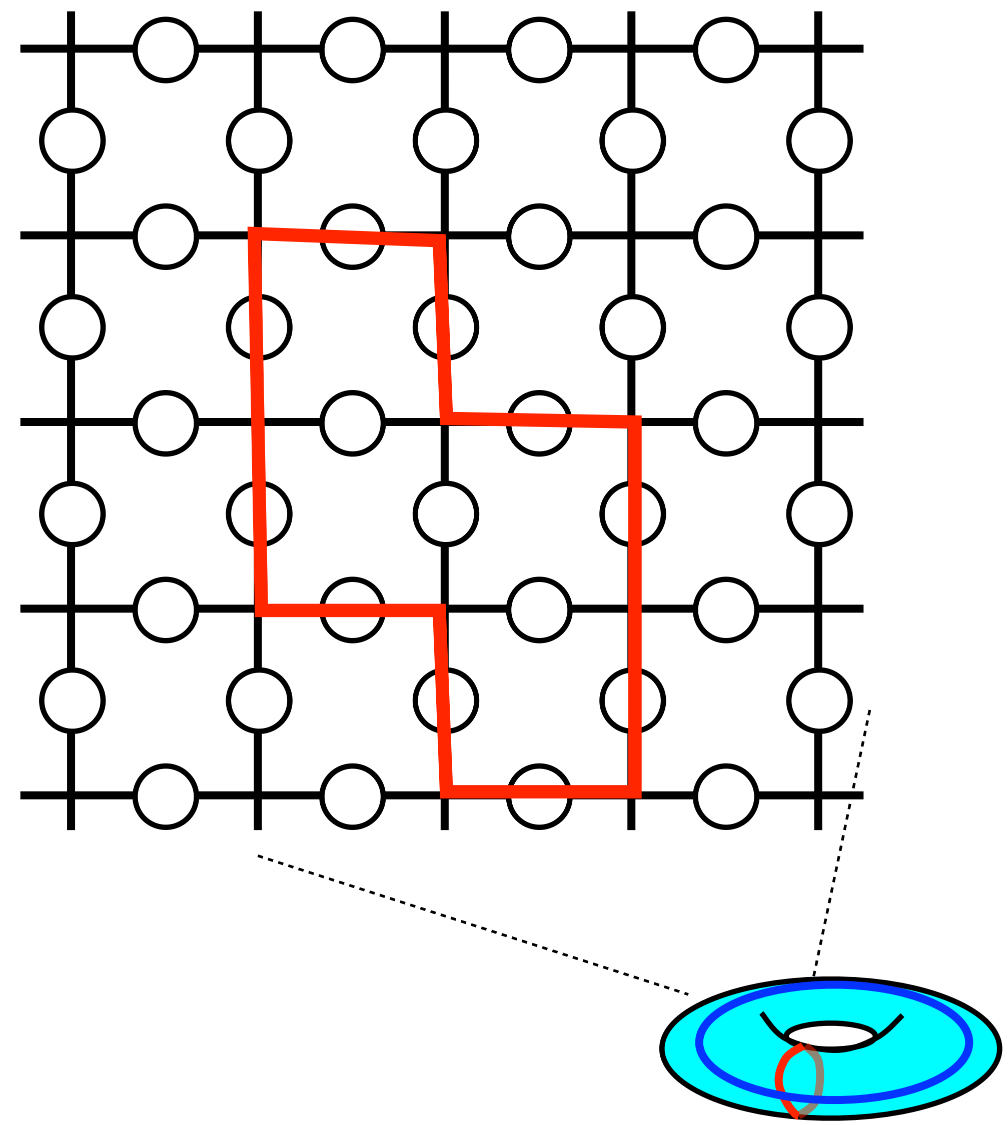




Surface code状態

Surface code (Kitaev's toric code)

Kitaev, Annals Phys. **303**, 2 (2003)

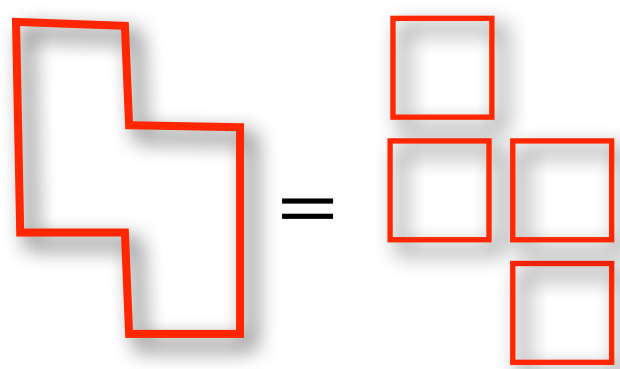


縮退をとく演算子=論理演算子は？

→スタビライザー演算子と可換な演算子

注）スタビライザーは群なので，スタビライザーの

積, $Z(\partial c_2), X(\partial \bar{c}_2)$ はスタビライザー



自明な（穴のない）ループ演算子はすべてスタビライザー

= $Z(c_1), X(\bar{c}_1)$ としたとき $c_1, \bar{c}_1$ が trivial cycle

(演算子の作用は chain の連続的変形に対して不変)

Surface code状態

Surface code (Kitaev's toric code)

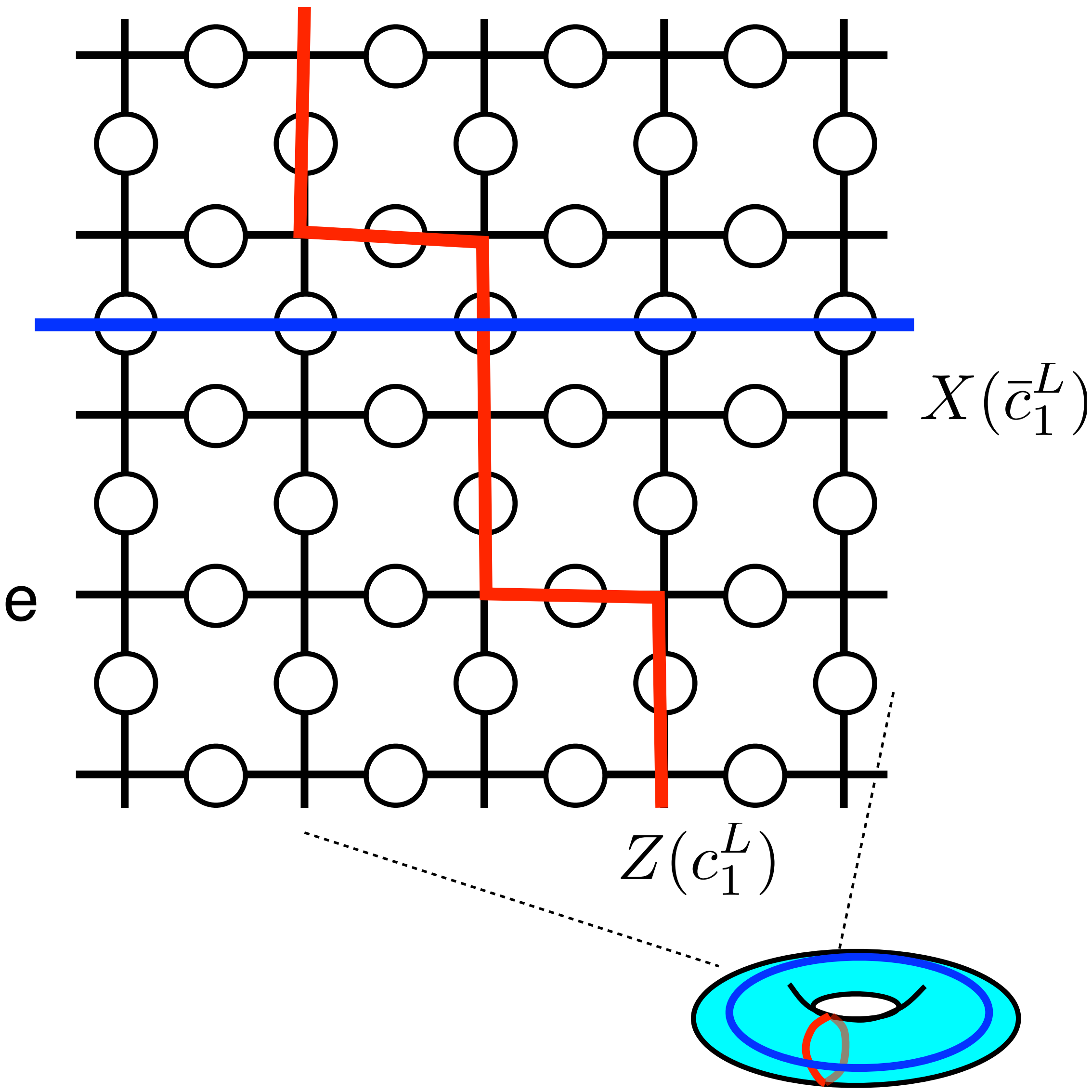
Kitaev, Annals Phys. **303**, 2 (2003)

縮退をとく演算子は？

非自明な（穴のある）ループ演算子

= $Z(c_1), X(\bar{c}_1)$としたとき $c_1, \bar{c}_1$ が non-trivial cycle

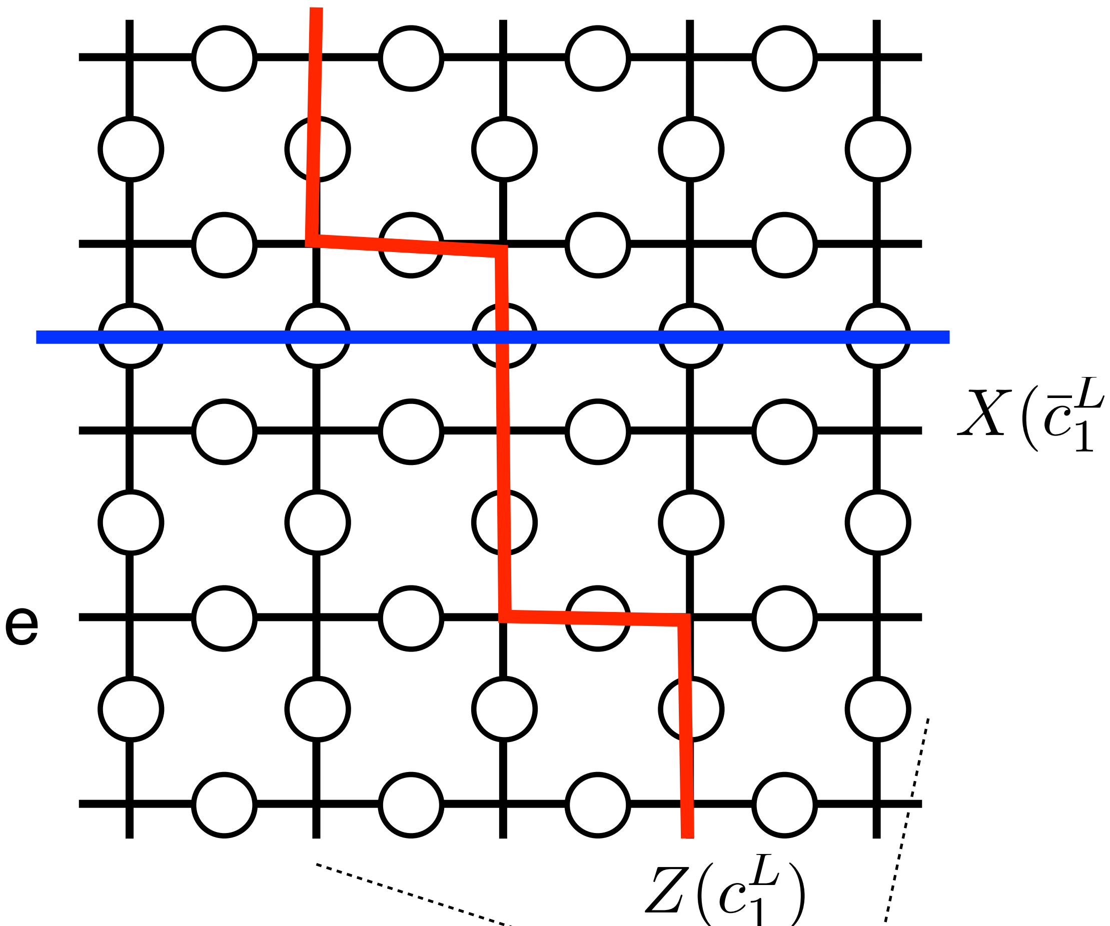
(つまり，ループ演算子の作用は，cycleの homology class にのみ依存する．）



Surface code状態

Surface code (Kitaev’s toric code)

Kitaev, Annals Phys. **303**, 2 (2003)

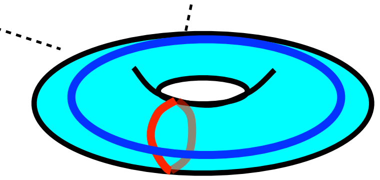


縮退をとく演算子は？

非自明な（穴のある）ループ演算子

= $Z(c_1), X(\bar{c}_1)$としたとき $c_1, \bar{c}_1$ が non-trivial cycle

(つまり，ループ演算子の作用は，cycleの homology class にのみ依存する．）



$Z(c_1^L), X(\bar{c}_1^L)$ はPauli 演算子と同じ交換関係$\{Z(c_1^L), X(\bar{c}_1^L)\} = 0$ (反可換)

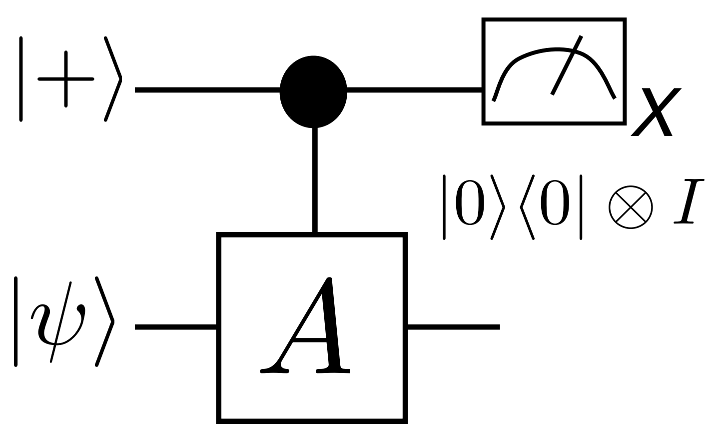
→符号空間上の論理 Pauli 演算子

Syndromeの測定

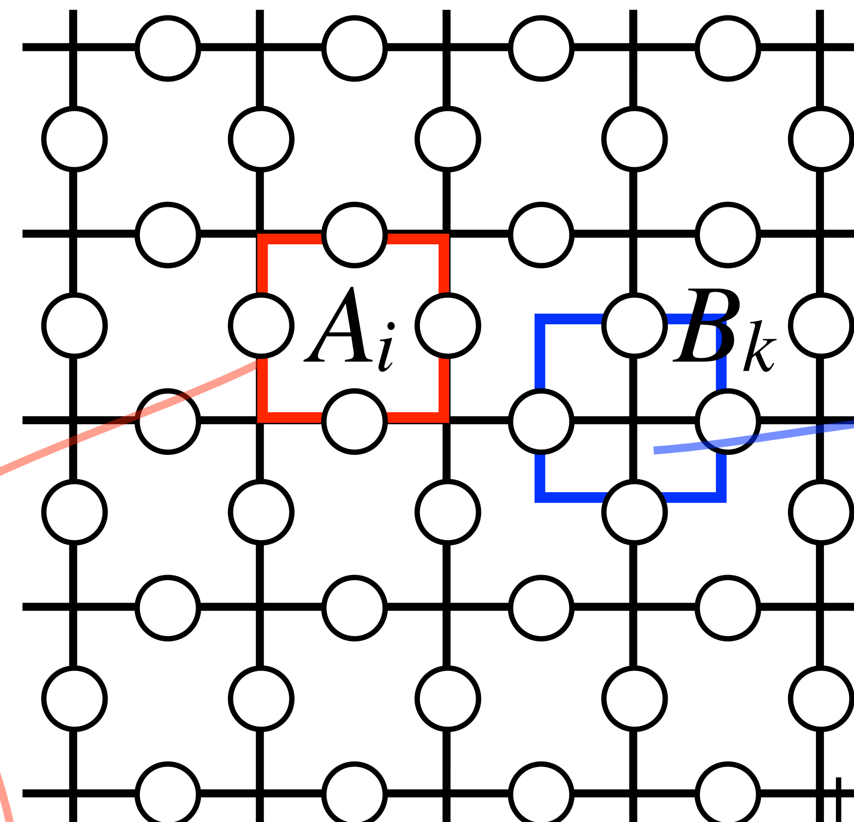
エラーが発生するとスタビライザー状態ではなくなってしまう

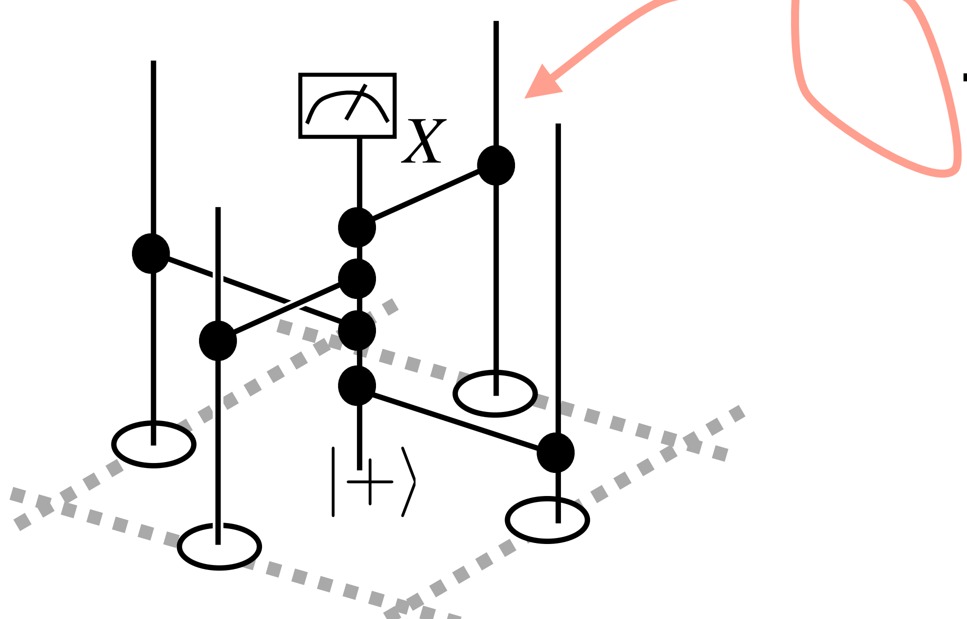
→スタビライザー演算子を測定する

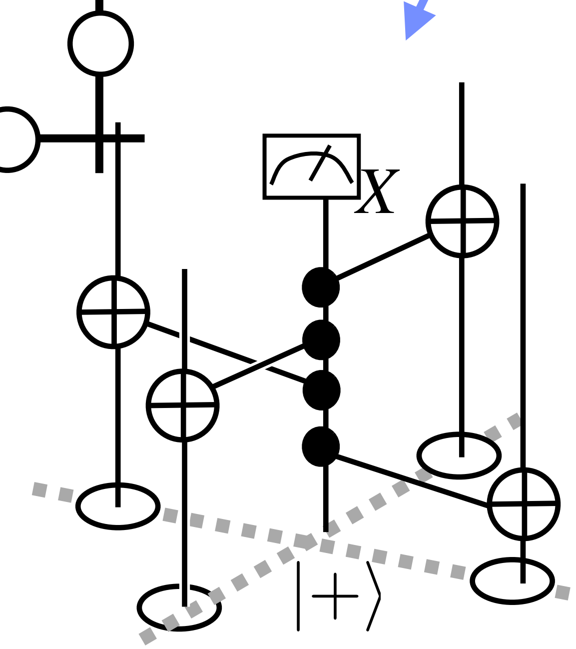
演算子*A*（エルミート&固有値±1）の測定



$$|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes A$$







エラーの訂正

エラーと反可換なスタビライザー演算子の

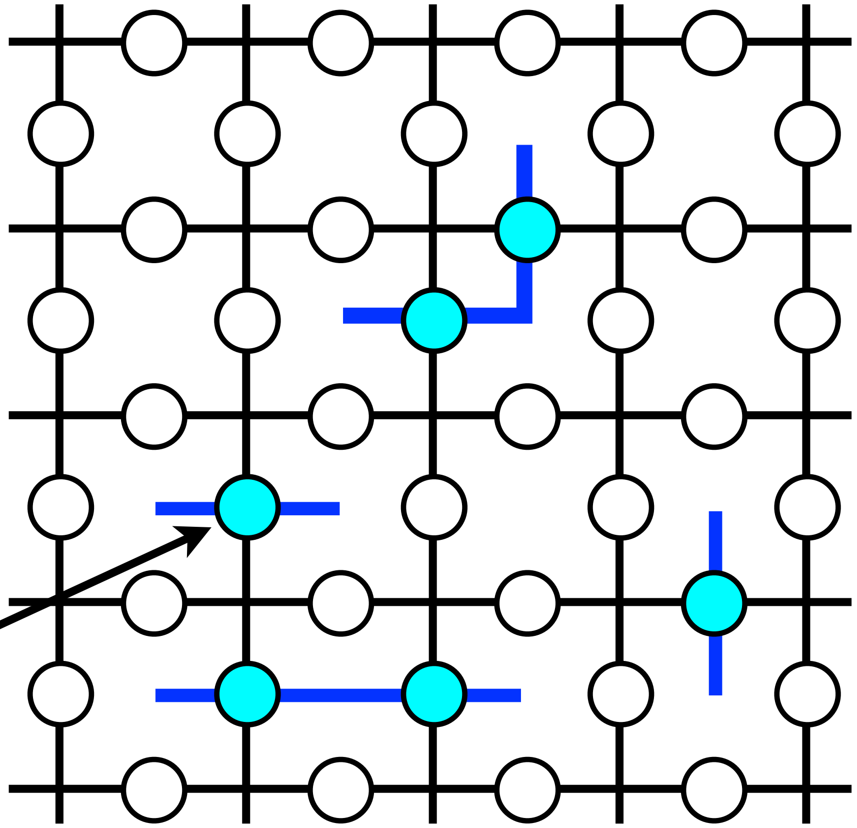
固有値が-1になる.

dual 1-chain $\bar{c}_1^e$ 上に

X エラーが発生した

とする.

X error: $X(\bar{c}_1^e)$



エラーの訂正

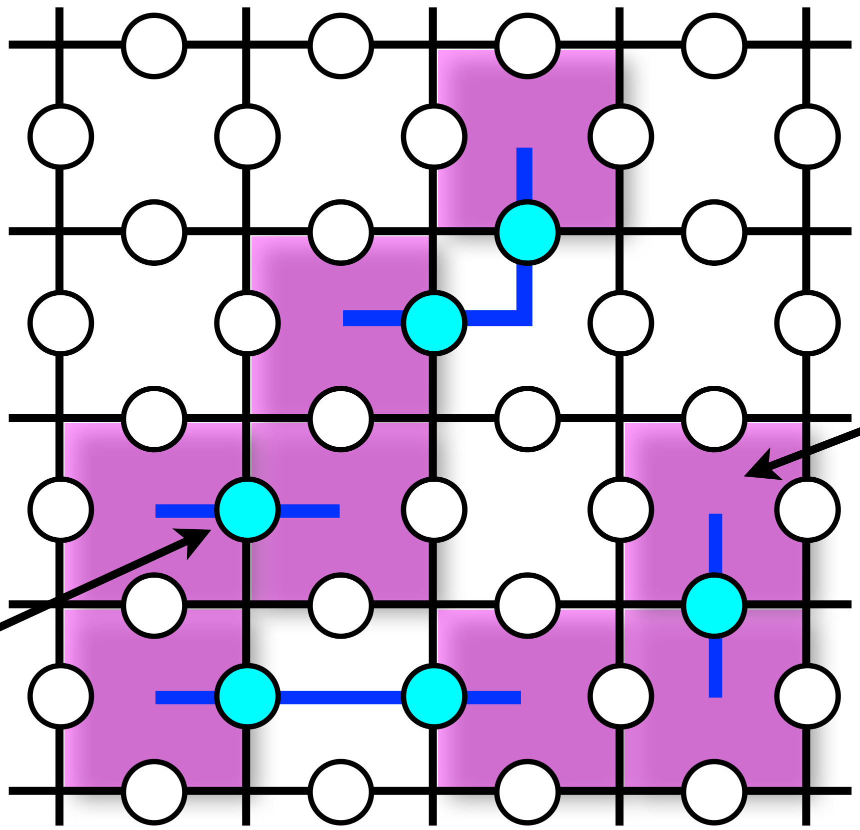
エラーと反可換なスタビライザー演算子の

固有値が-1になる.

dual 1-chain $\bar{c}_1^e$ 上に

X エラーが発生した

とする.



X error: $X(\bar{c}_1^e)$

incorrect
error syndrome

$c_2^e = \bar{c}_0^e = \partial \bar{c}_1^e$

for all $f_i \in c_2^e$,
$[Z(\partial f_i), X(\bar{c}_1^e)] \neq 0$

エラーの訂正

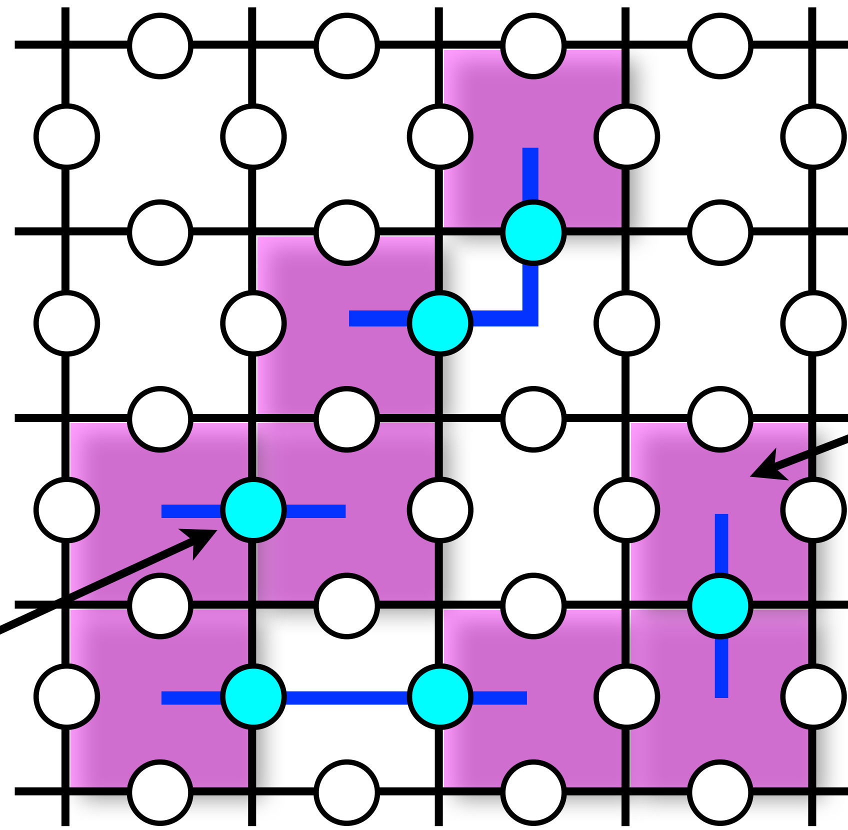
エラーと反可換なスタビライザー演算子の

固有値が-1になる.

dual 1-chain $\bar{c}_1^e$ 上に

X エラーが発生した

とする.



X error: $X(\bar{c}_1^e)$

incorrect
error syndrome

$c_2^e = \bar{c}_0^e = \partial \bar{c}_1^e$

for all $f_i \in c_2^e$,

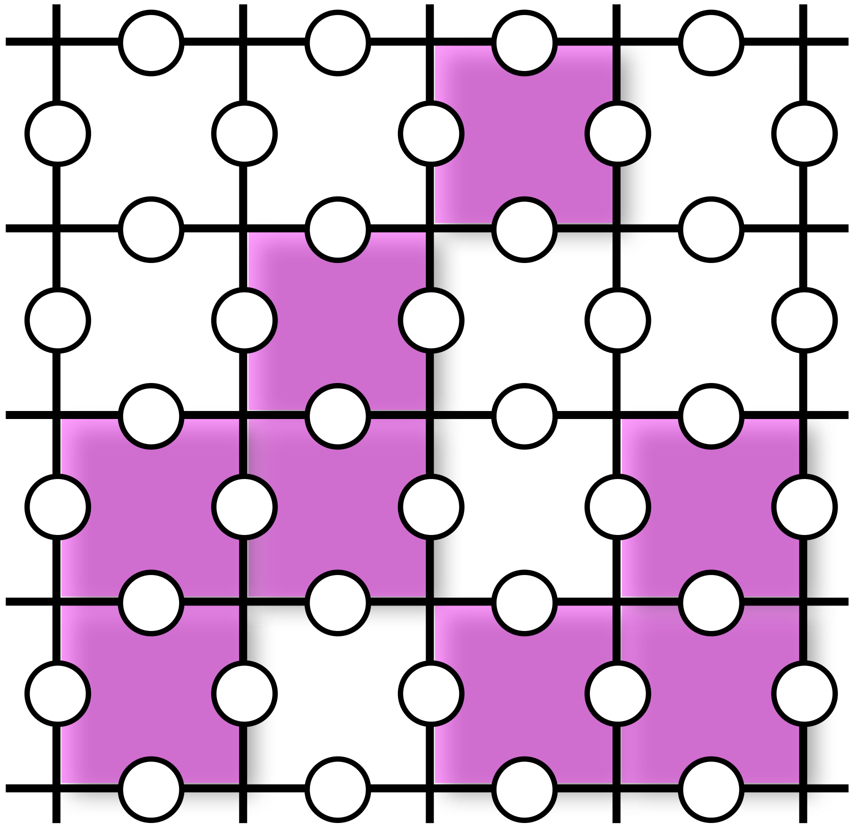
$[Z(\partial f_i), X(\bar{c}_1^e)] \neq 0$

エラー鎖 $\bar{c}_1^e$ の端点 $\partial \bar{c}_1^e$のスタビライザー

に異常が見つかる.

エラーの訂正

人間にはエラーの場所は見えない

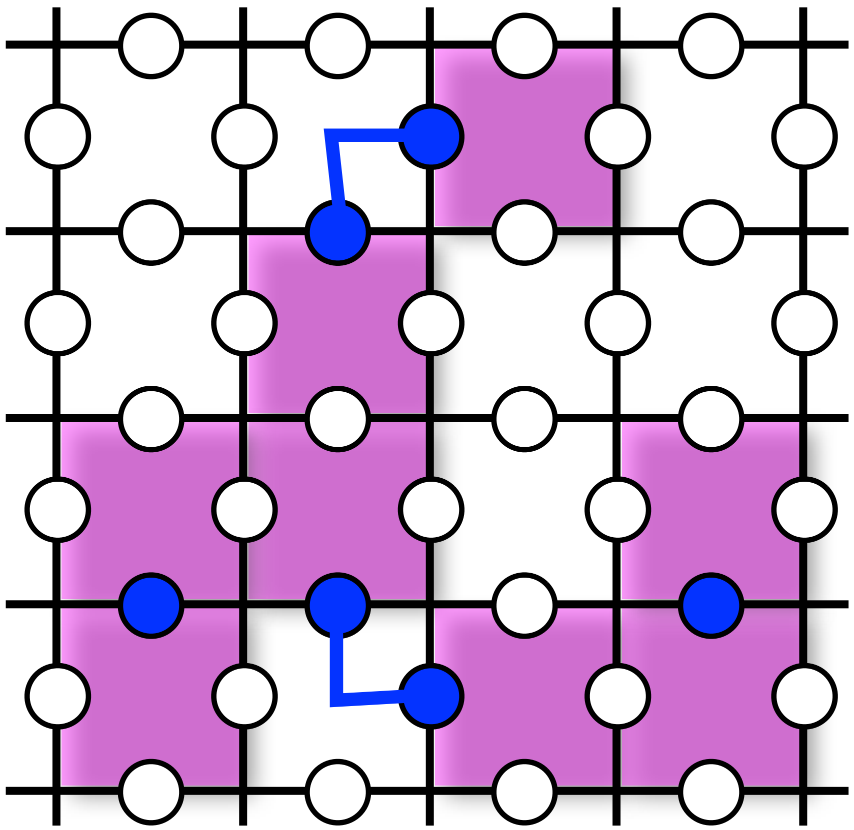


シンドローム $\partial \vec{c}_1^e$ からエラーの位置を推定する.

つまり, $\partial \vec{c}_1^e = \partial \vec{c}_1^r$ (境界条件を満たすような) $\vec{c}_1^r$ を探す.

エラーの訂正

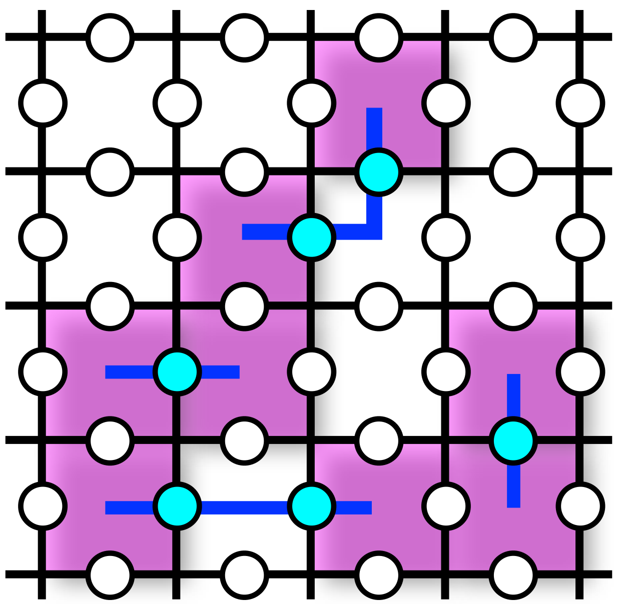
推定されたエラー鎖 $\vec{c}_1^r$



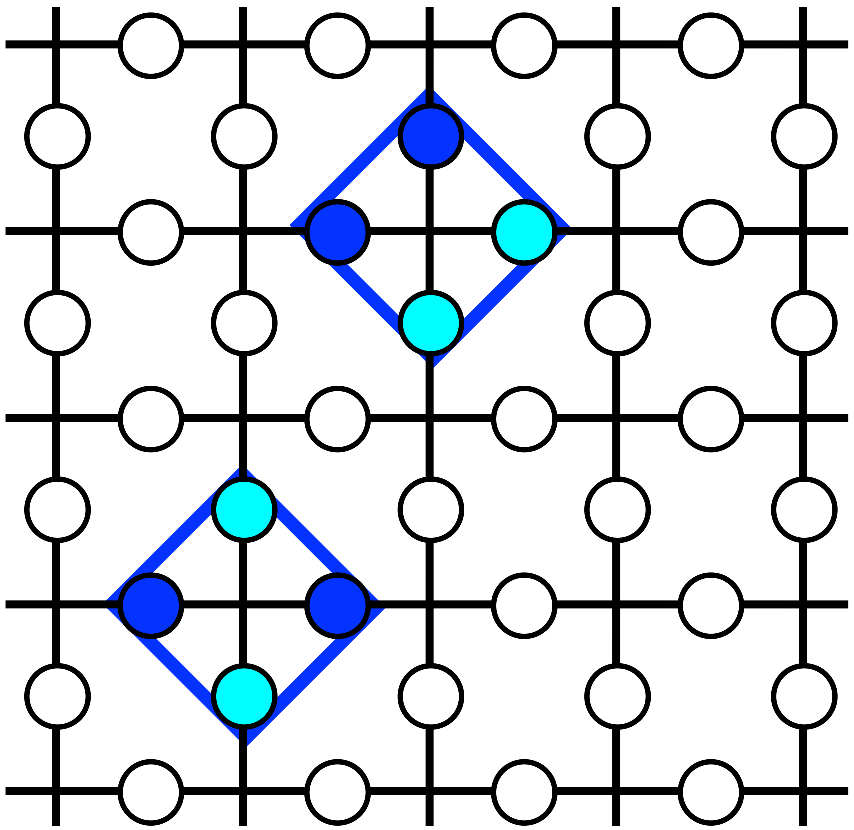
これを作用させると結果的に、、、

エラーの訂正

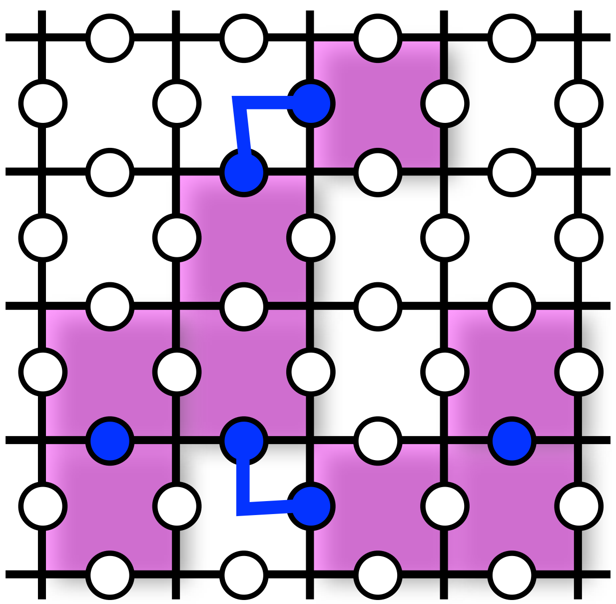
エラー＋回復操作＝自明ループ演算子になる→スタビライザー
(元の状態は固有状態＝エラー訂正が成功)



真のエラー　$\bar{c}_1^e$



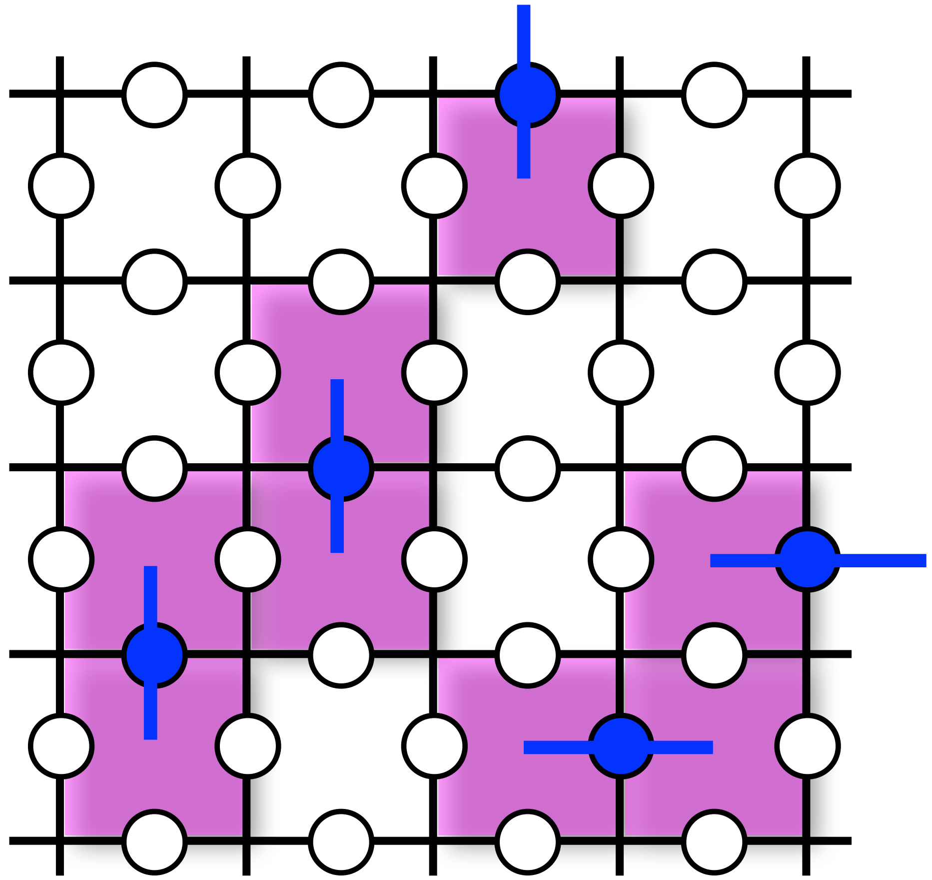
$\bar{c}_1^e + \bar{c}_1^r$ が trivial cycleに.
→ $X(\bar{c}_1^e + \bar{c}_1^r)$ は stabilizer



推定したもの　$\bar{c}_1^r$

エラーの訂正

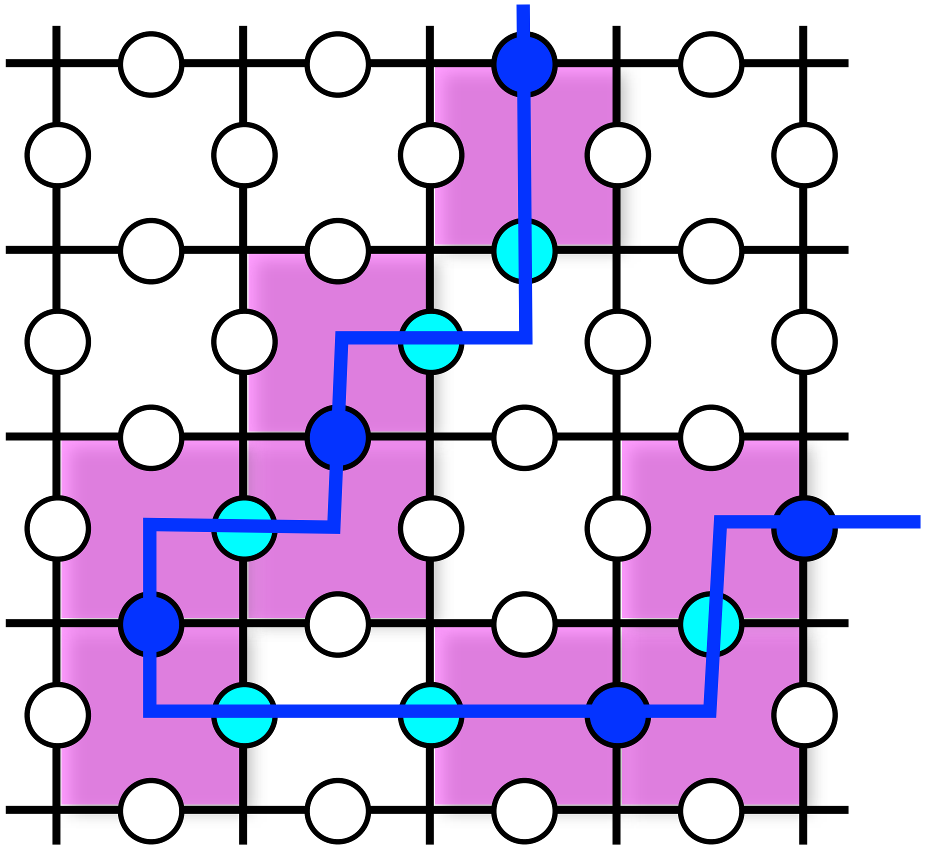
例えば、このようなエラー鎖 $\bar{c}_1^{r'}$ で回復させると、



エラーの訂正

非自明なループ演算子→符号状態を変える

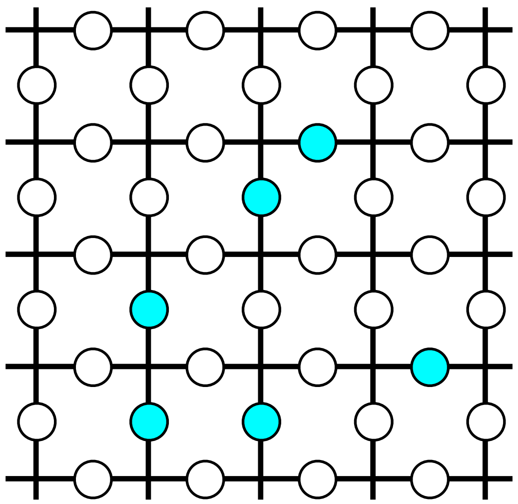
誤り訂正の失敗



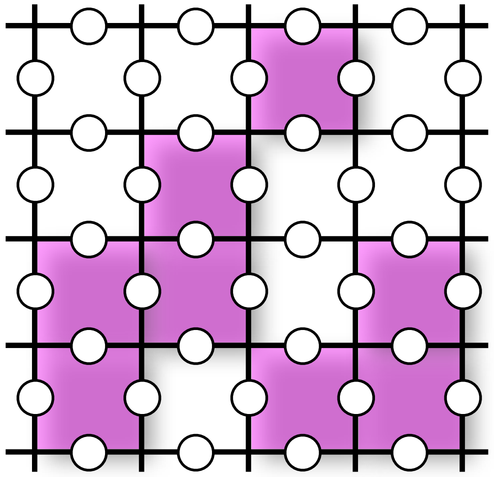
$\bar{c}_1^e + \bar{c}_1^{r'}$ は non-trivial cycle

→ $X(\bar{c}_1^e + \bar{c}_1^{r'})$ は論理演算子

訂正アルゴリズム

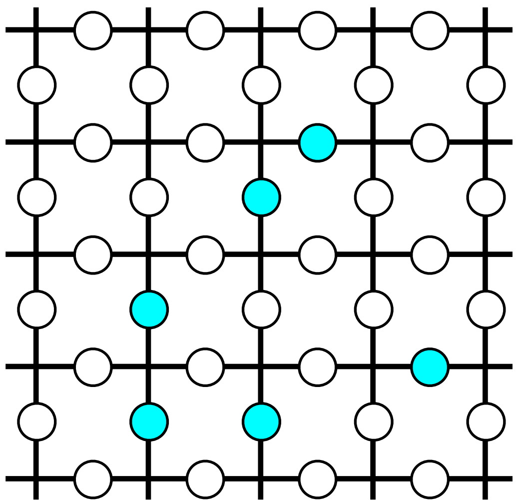


エラー $\bar{c}_1^e$

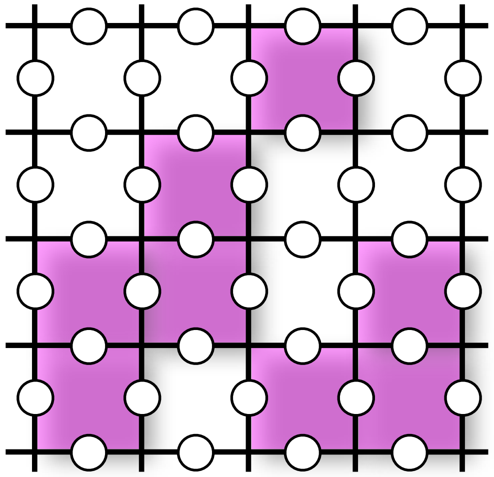


シンドローム $\partial\bar{c}_1^e$

訂正アルゴリズム



エラー $\bar{c}_1^e$

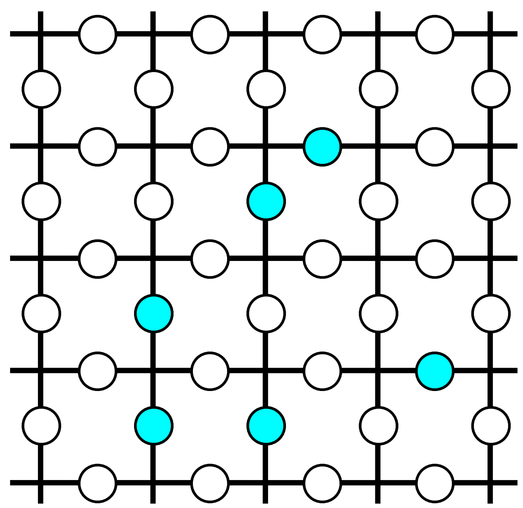


シンドローム $\partial\bar{c}_1^e$

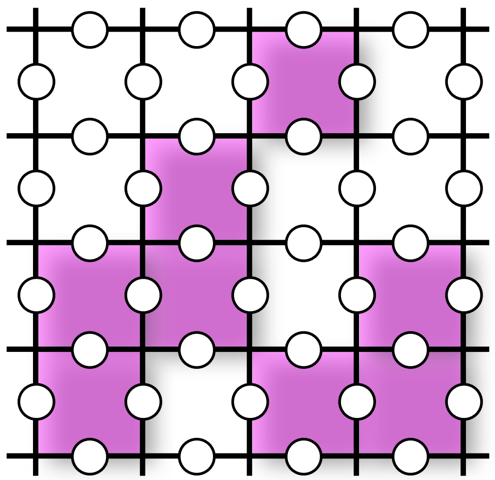
→ もっとも発生確率が高いエラーを推定して訂正しよう！

$\partial\bar{c}_1^e = \partial\bar{c}_1^r$ の条件の下で, $p[X(\bar{c}_1^r)]$ が最大になるような $\bar{c}_1^r$.

訂正アルゴリズム



エラー $\bar{c}_1^e$



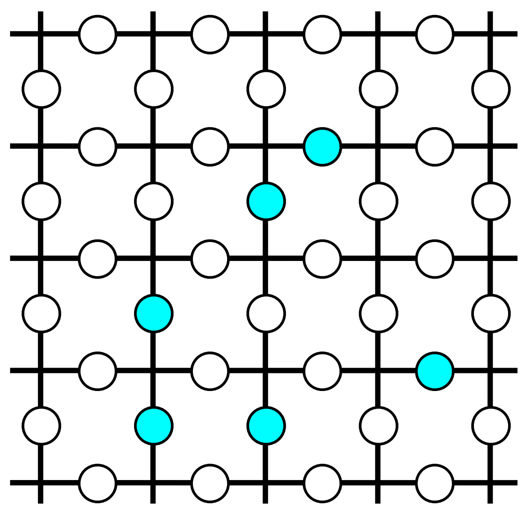
シンドローム $\partial\bar{c}_1^e$

→ もっとも発生確率が高いエラーを推定して訂正しよう！

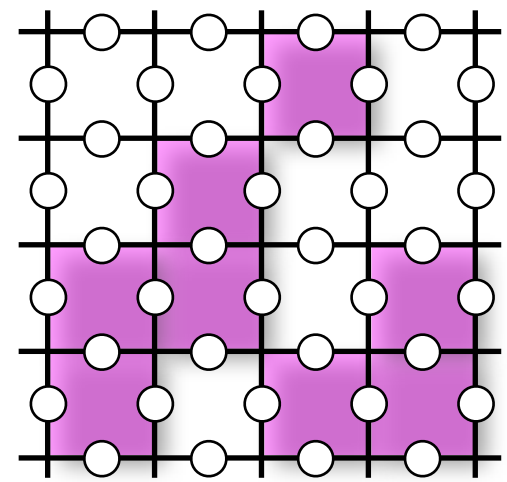
$\partial\bar{c}_1^e = \partial\bar{c}_1^r$ の条件の下で, $p[X(\bar{c}_1^r)]$ が最大になるような $\bar{c}_1^r$.

$$p[X(\bar{c}_1^r)] = (1-p)^{|E|}\prod_{\bar{l}}\left(\frac{p}{1-p}\right)^{z_{\bar{l}}^r}$$ のもとで $\arg\max_{\bar{c}_1^r}\ p[X(\bar{c}_1^r)]|_{\partial\bar{c}_1^e=\partial\bar{c}_1^r}$

訂正アルゴリズム



エラー $\bar{c}_1^e$



シンドローム $\partial\bar{c}_1^e$

→ もっとも発生確率が高いエラーを推定して訂正しよう！

$\partial\bar{c}_1^e = \partial\bar{c}_1^r$ の条件の下で, $p[X(\bar{c}_1^r)]$ が最大になるような $\bar{c}_1^r$.

$$p[X(\bar{c}_1^r)] = (1-p)^{|E|}\prod_{\bar{l}}\left(\frac{p}{1-p}\right)^{z_l^r}$$ のもとで $$\arg\max_{\bar{c}_1^r}\ p[X(\bar{c}_1^r)]|_{\partial\bar{c}_1^e=\partial\bar{c}_1^r}$$

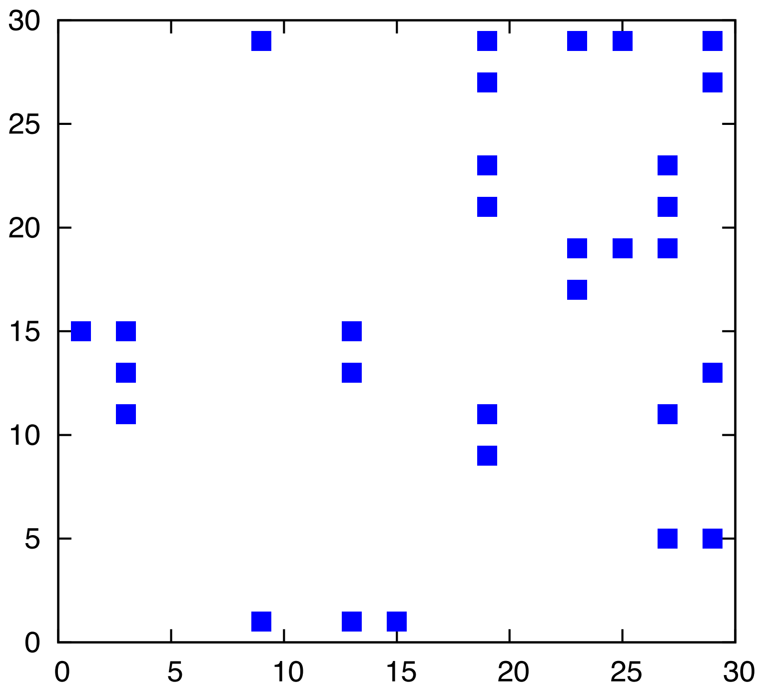
→つまりエラーの個数 $\sum_{\bar{l}} z_l^r$ が最も少なくなるようにすれば良い.

→ minimum-weight perfect match algorithm (**MWPMA**)

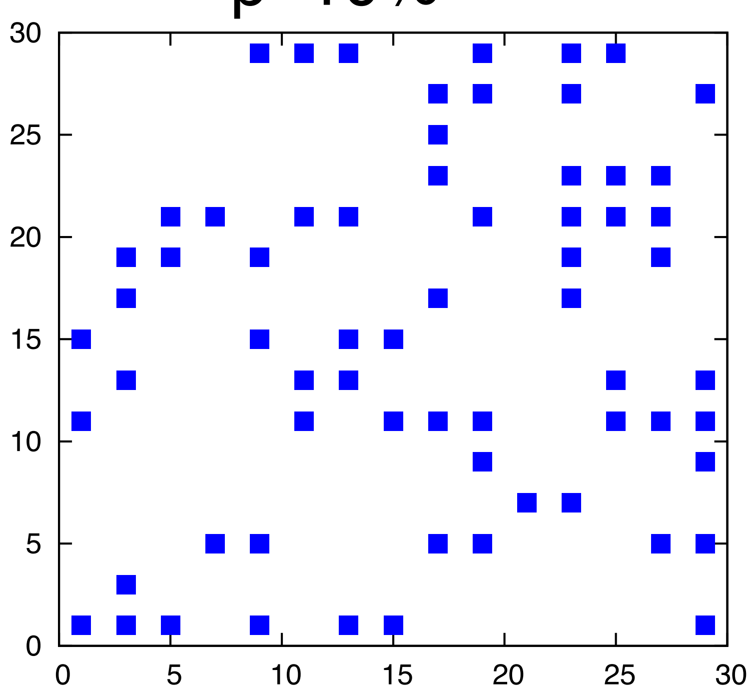
E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J.Math. Phys. **43**, 4452 (2002).

MWPMAを用いた場合の誤り訂正限界

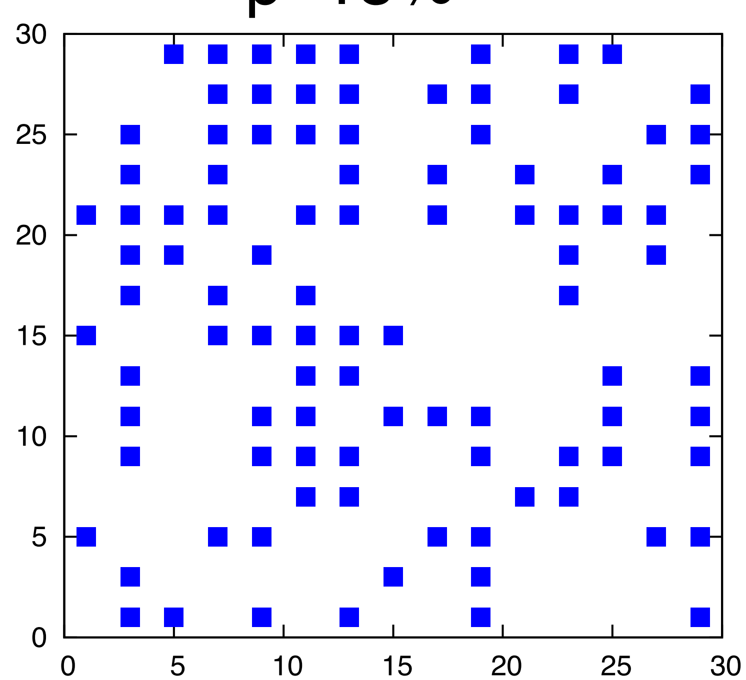
p=3%

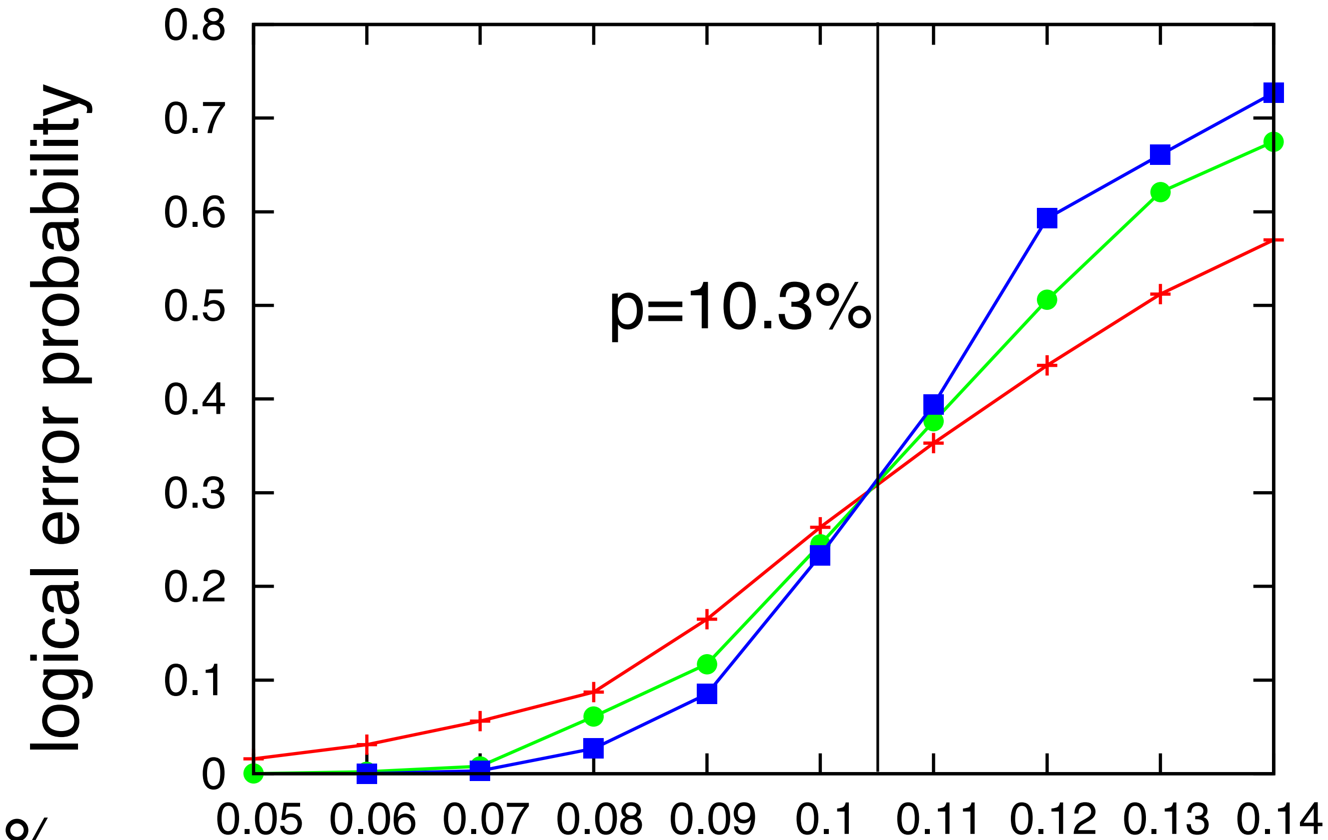


p=10%



p=15%



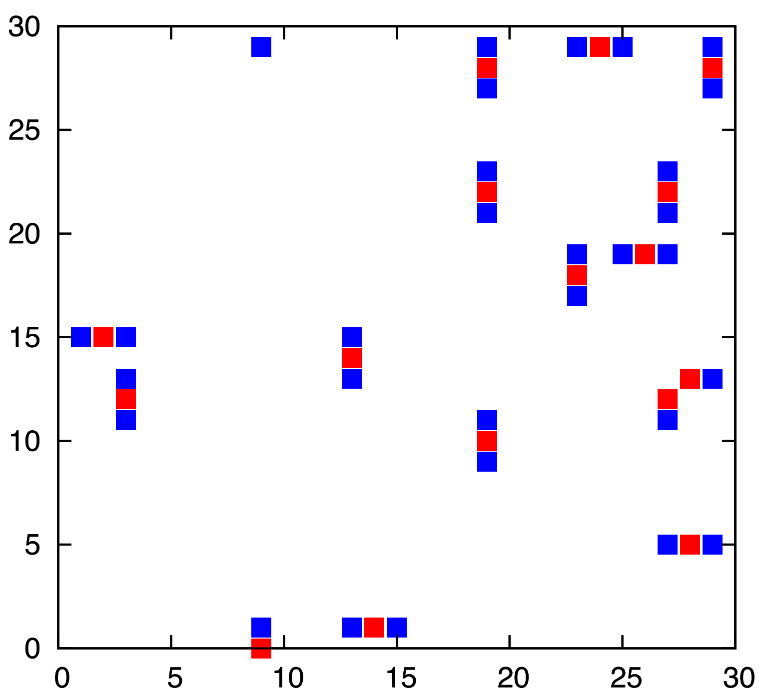


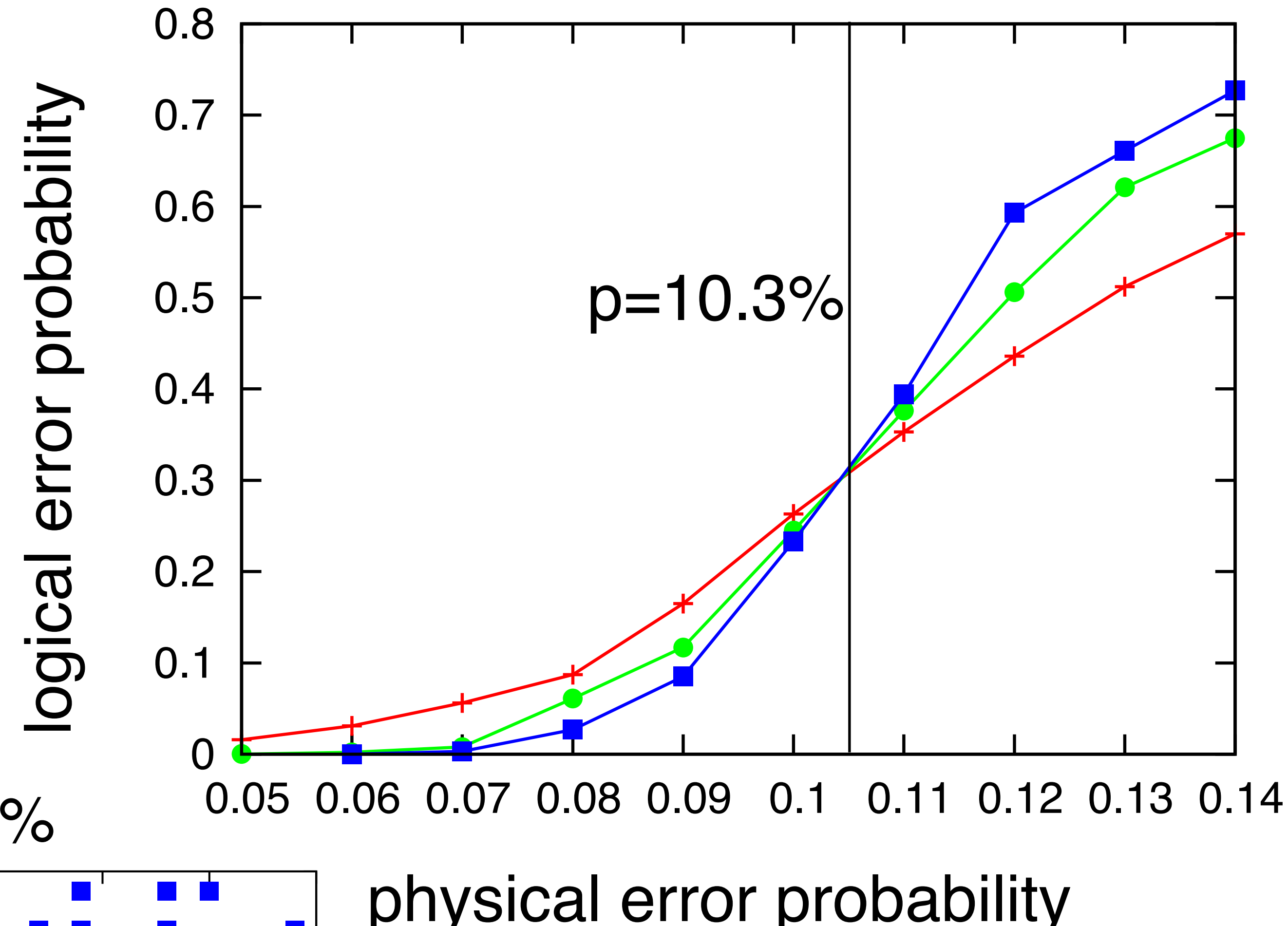
physical error probability

ある種のパーコレーション問題
→何らかの統計力学模型に対応？

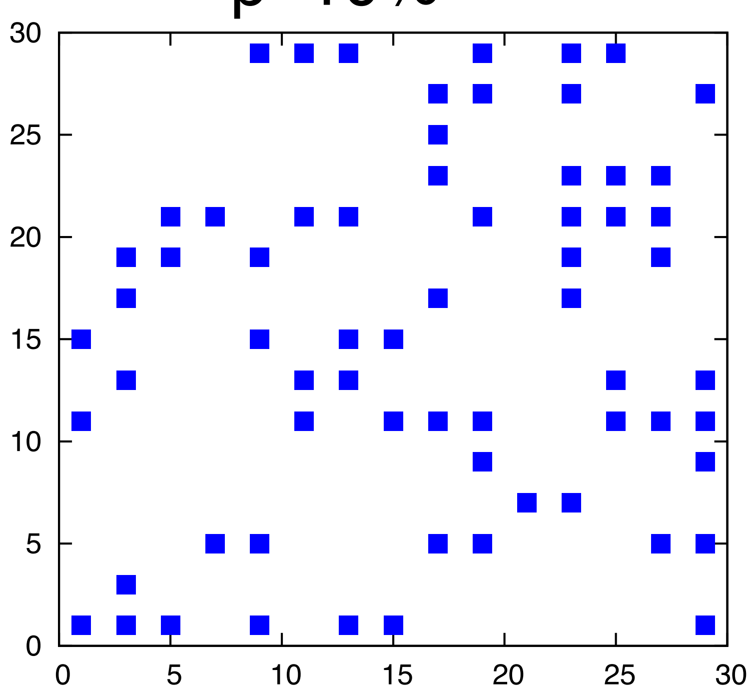
MWPMAを用いた場合の誤り訂正限界

p=3%

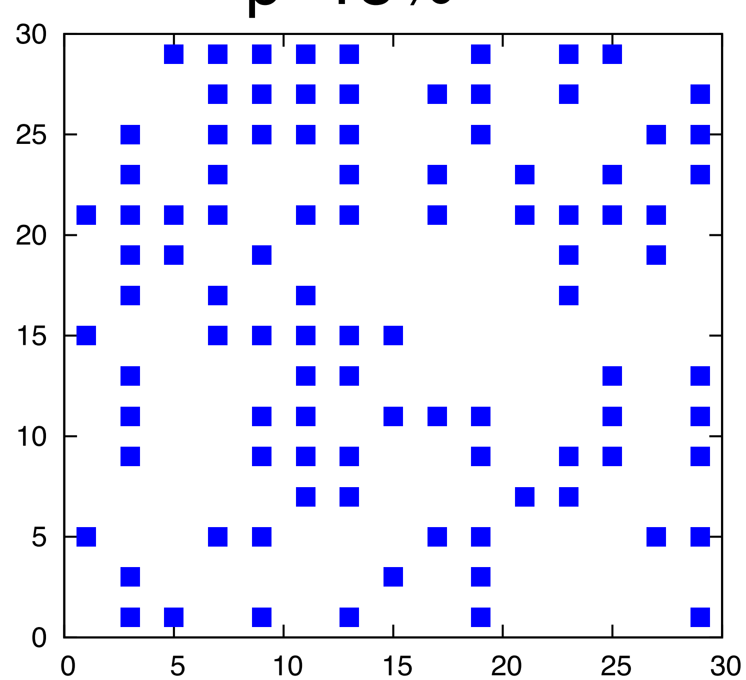




p=10%



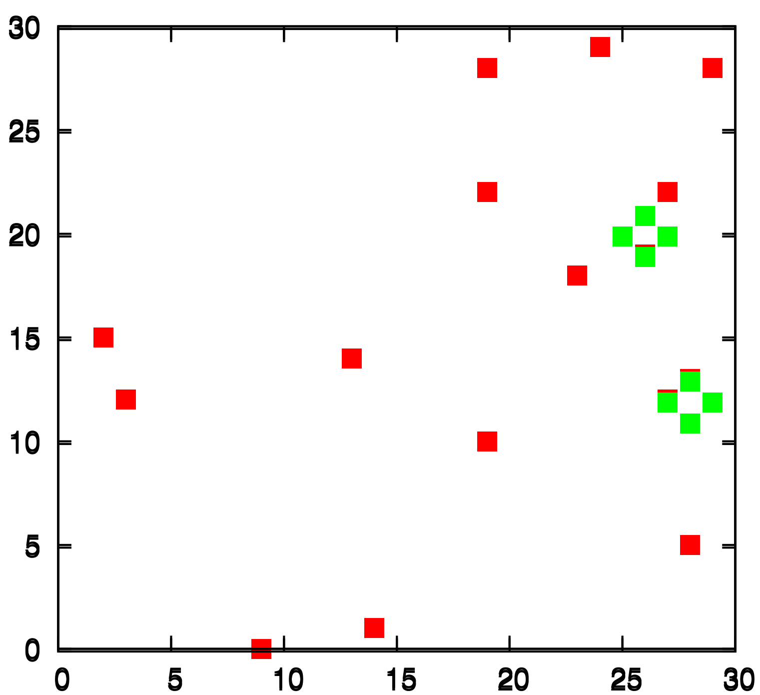
p=15%

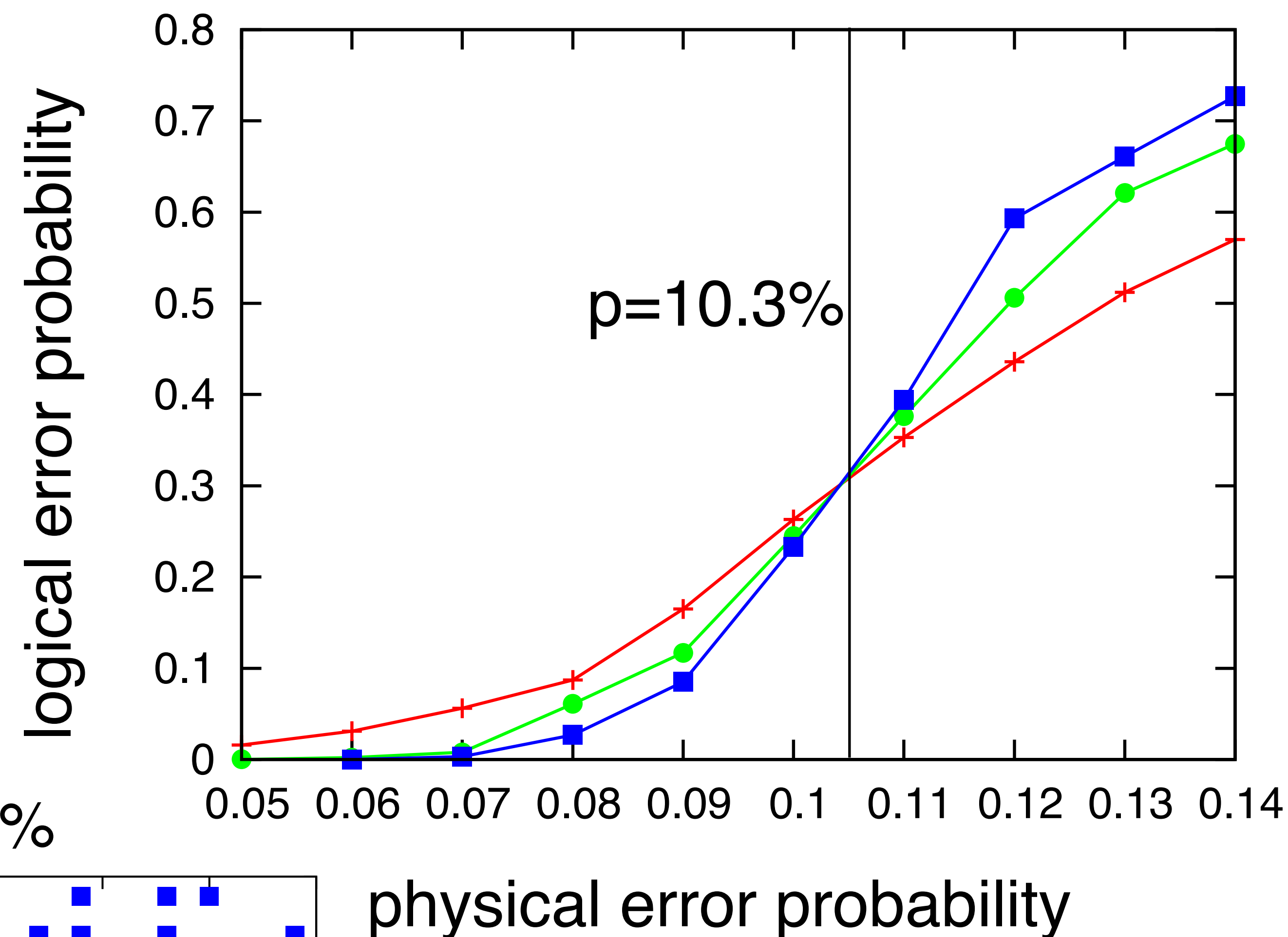


ある種のパーコレーション問題
→何らかの統計力学模型に対応？

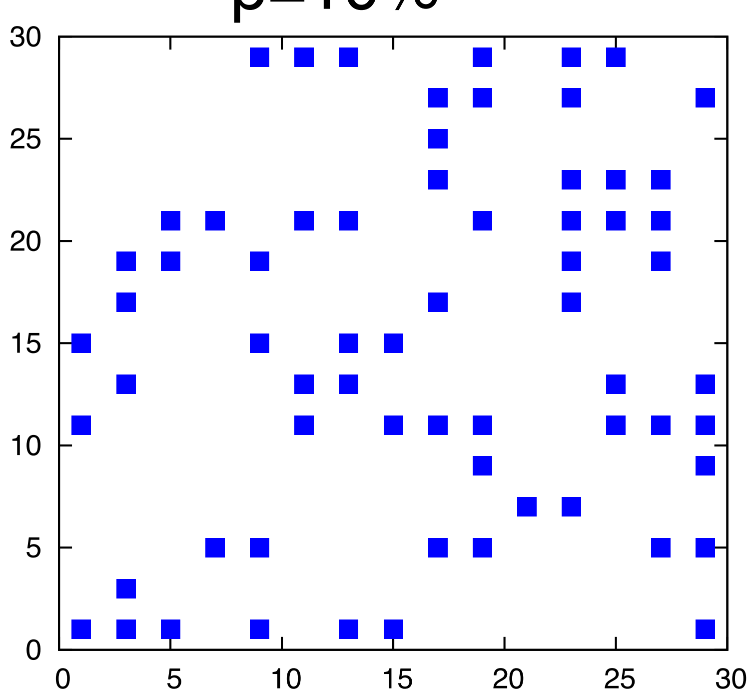
MWPMAを用いた場合の誤り訂正限界

p=3%

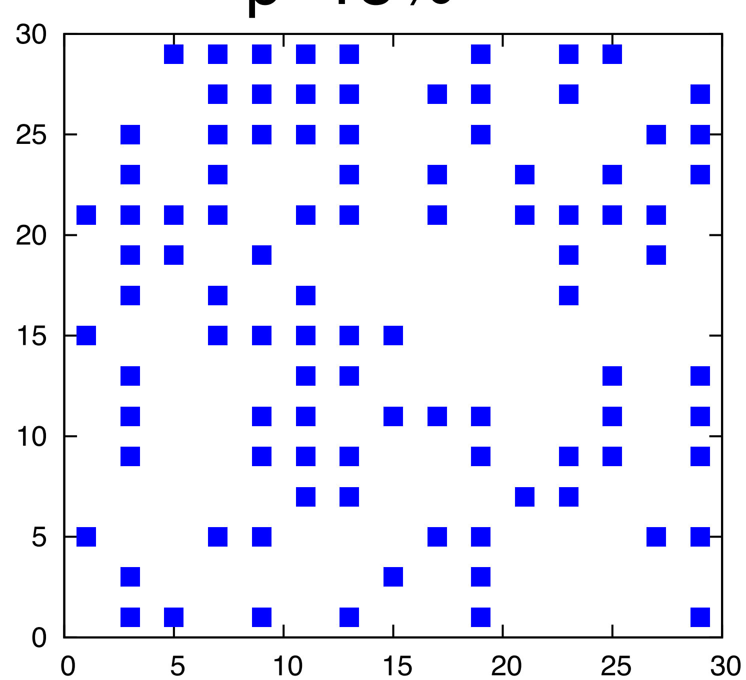




p=10%



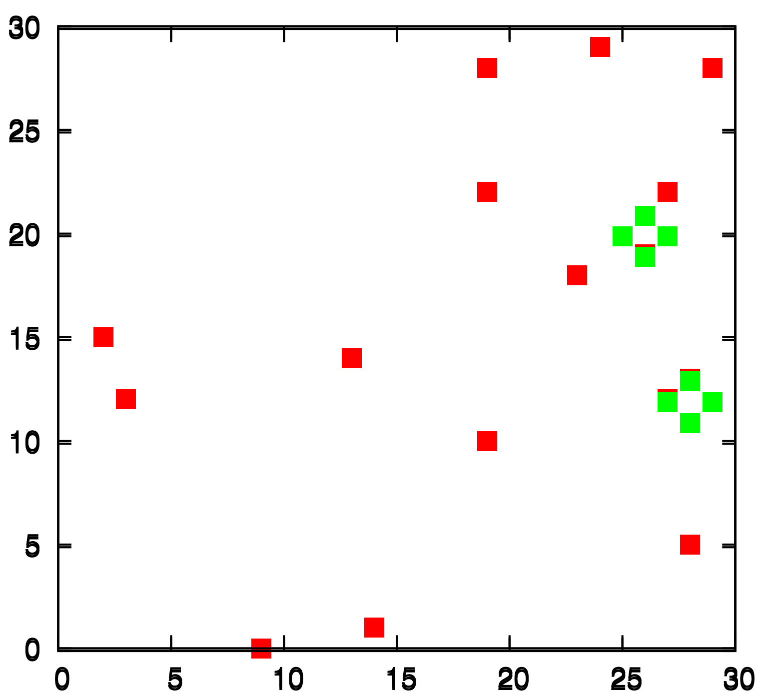
p=15%

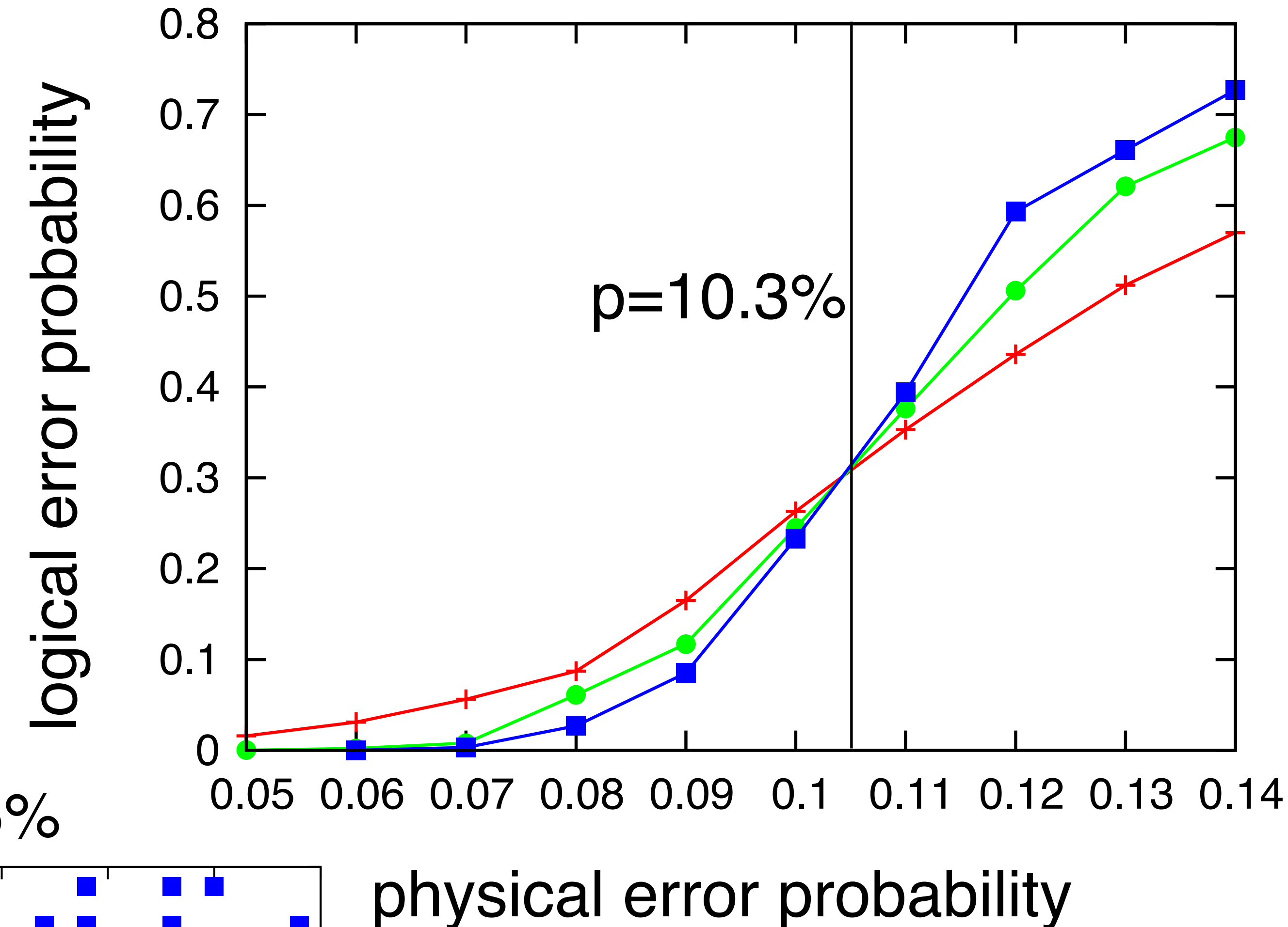


ある種のパーコレーション問題
→何らかの統計力学模型に対応？

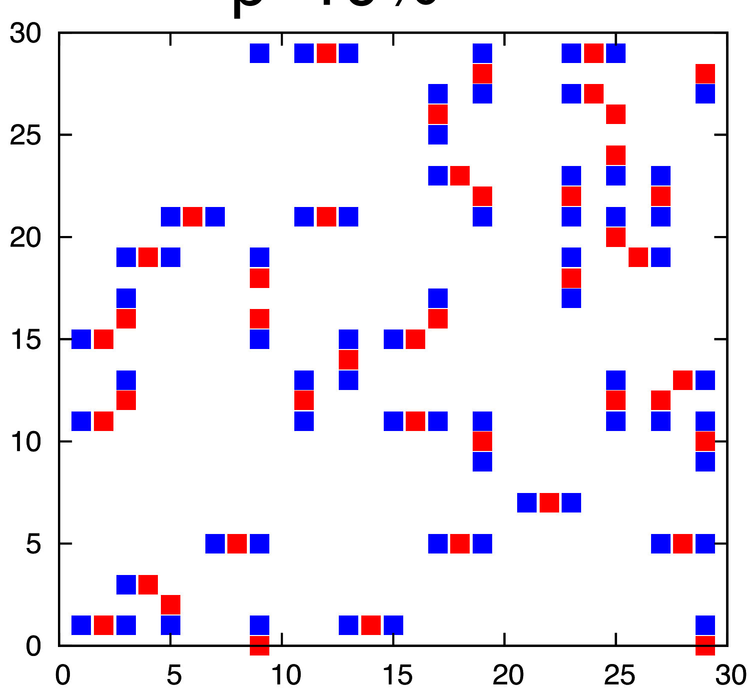
MWPMAを用いた場合の誤り訂正限界

p=3%

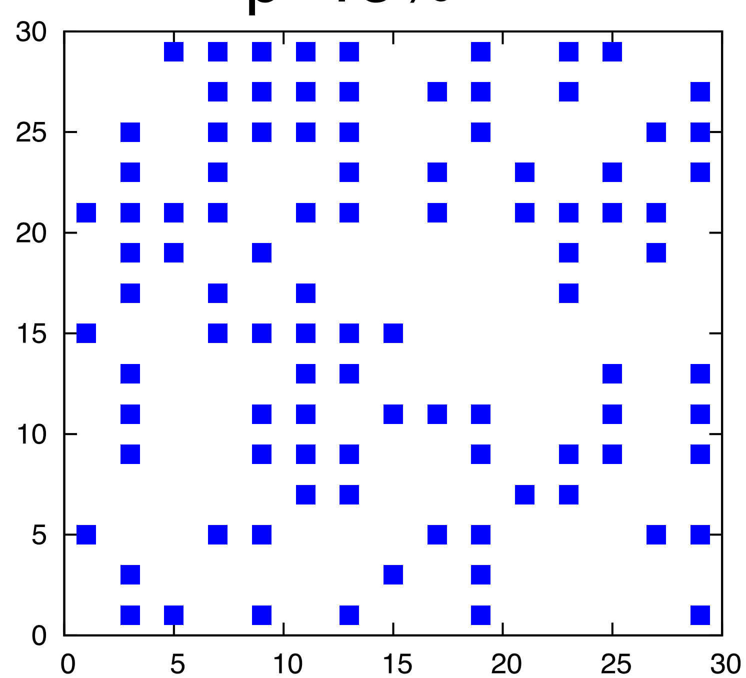




p=10%



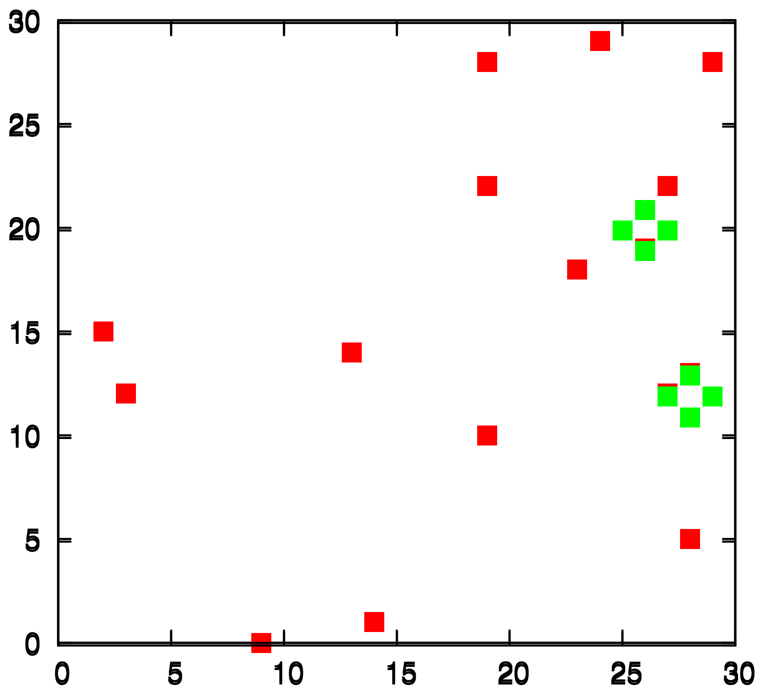
p=15%

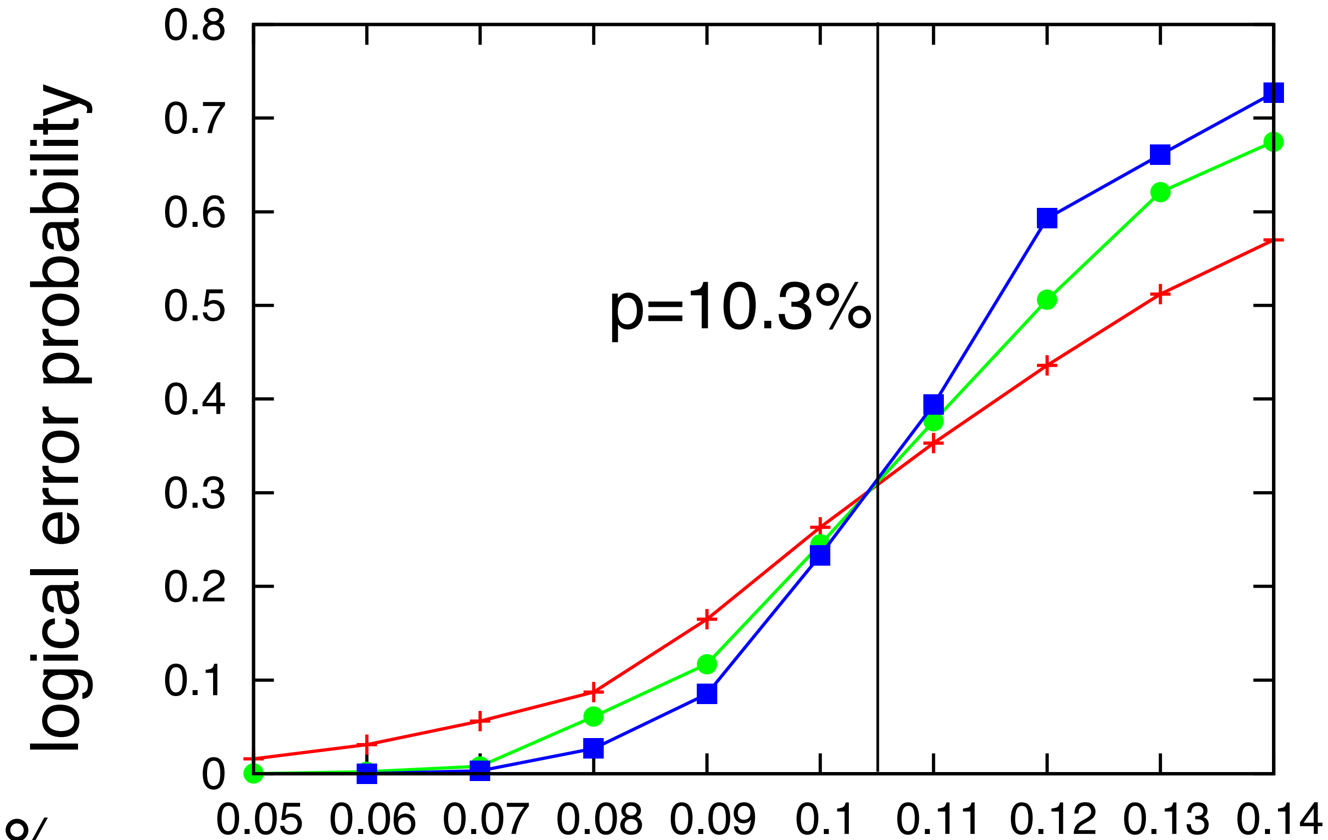


ある種のパーコレーション問題
→何らかの統計力学模型に対応？

MWPMAを用いた場合の誤り訂正限界

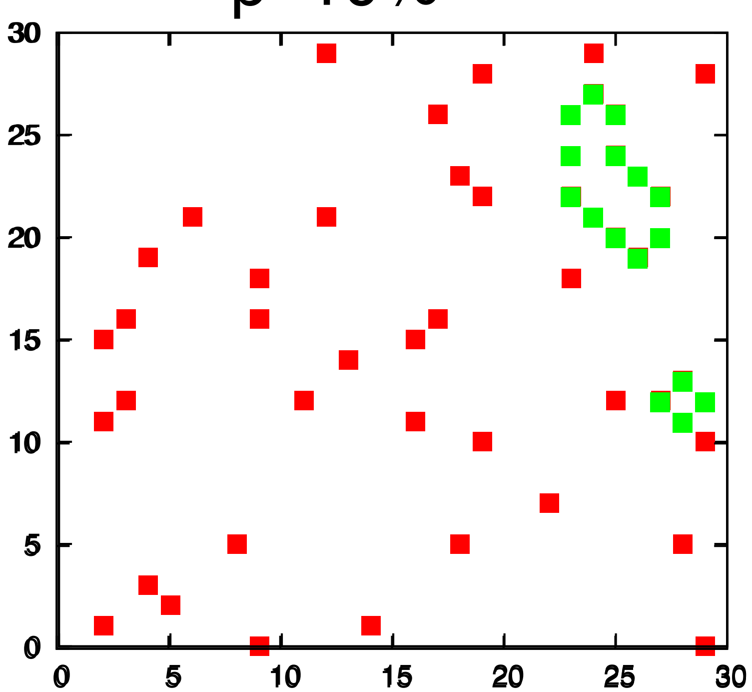
p=3%



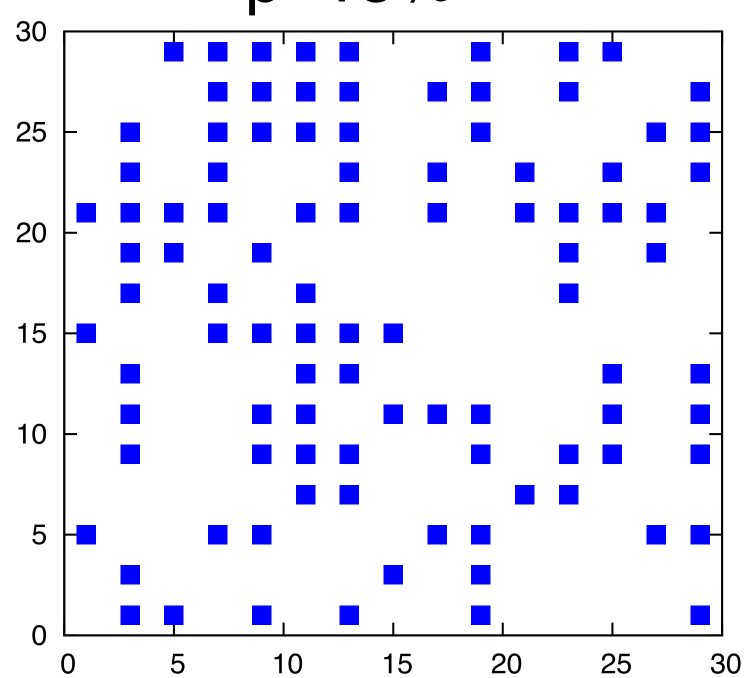


physical error probability

p=10%



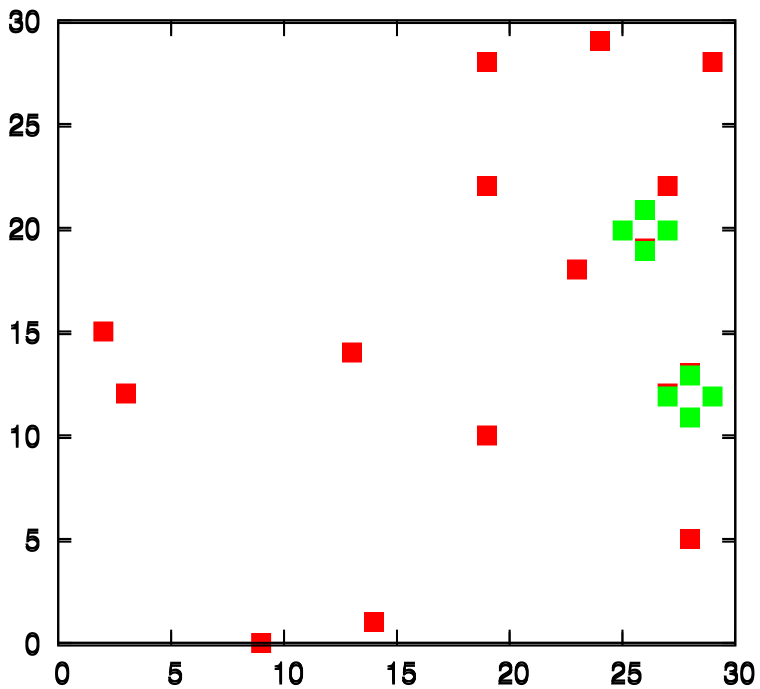
p=15%

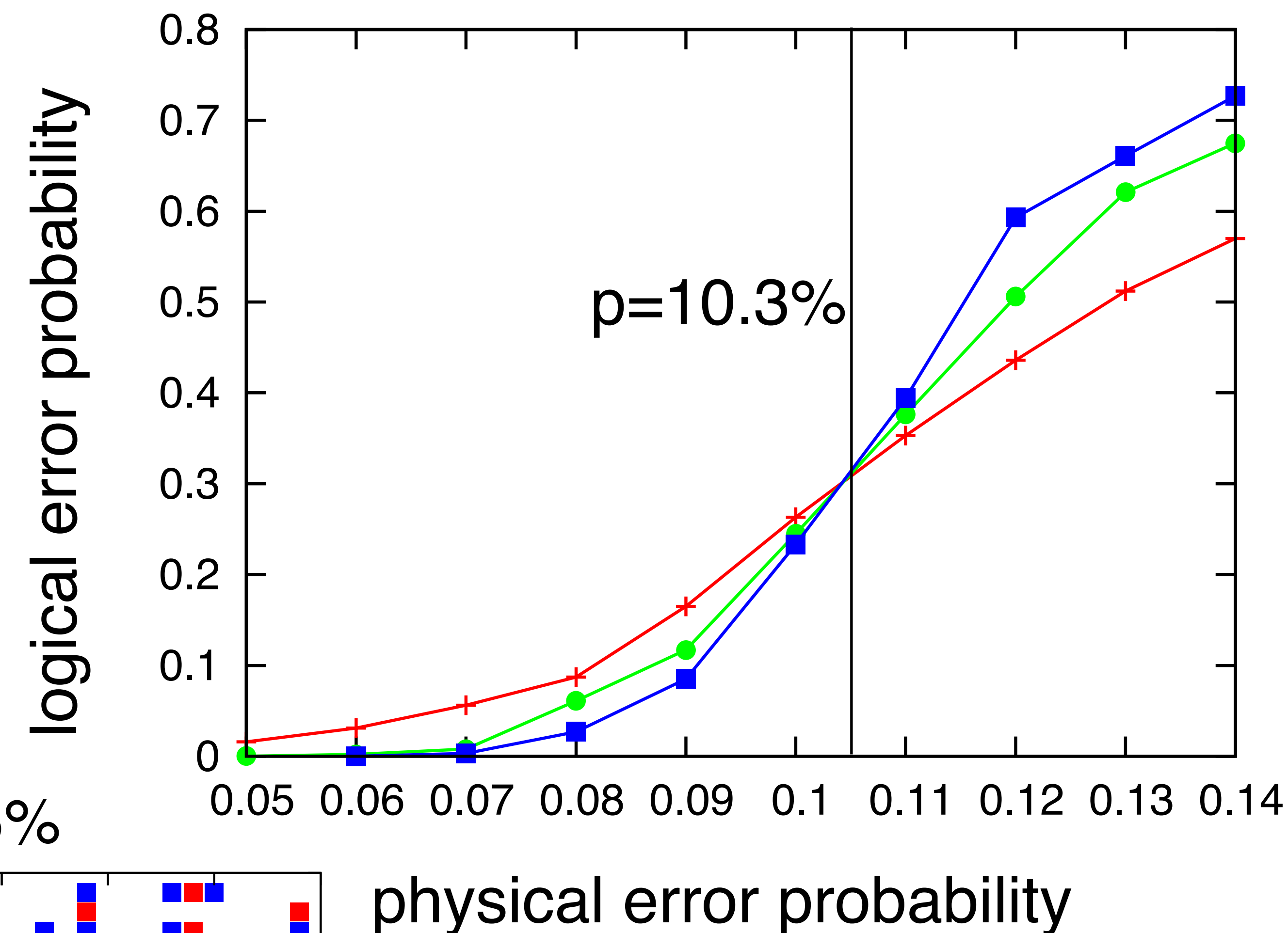


ある種のパーコレーション問題
→何らかの統計力学模型に対応？

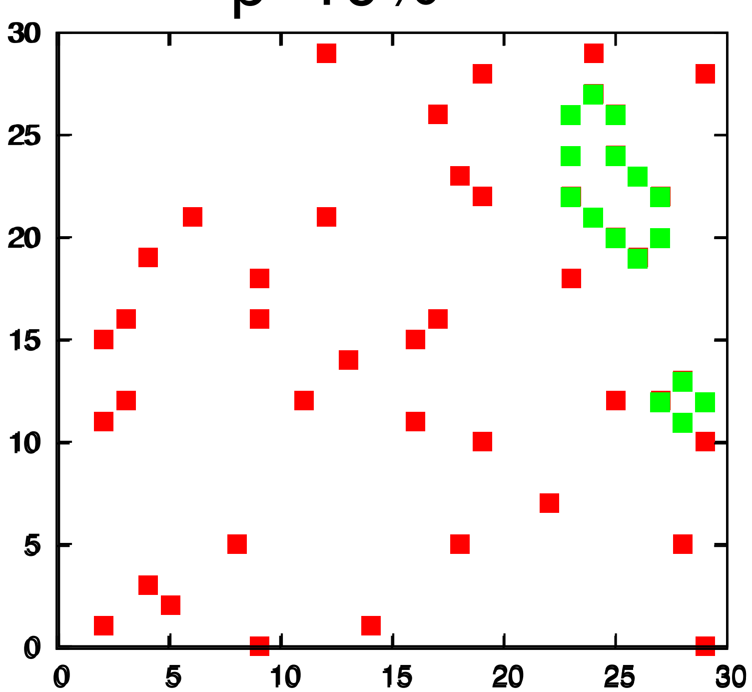
MWPMAを用いた場合の誤り訂正限界

p=3%

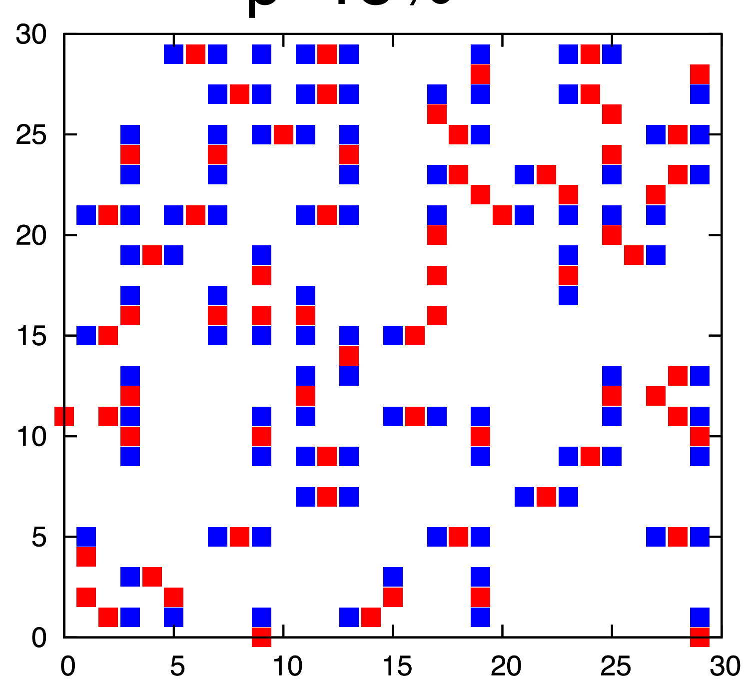




p=10%



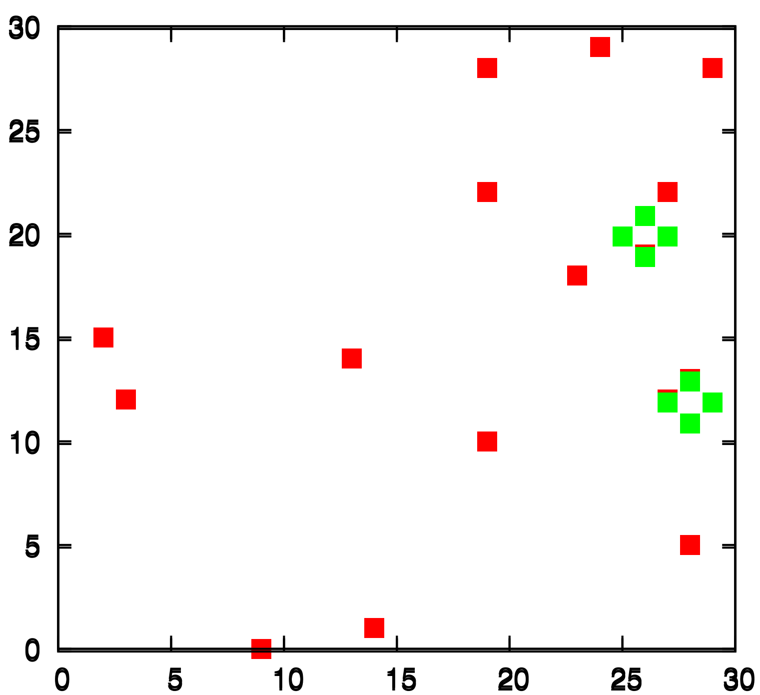
p=15%

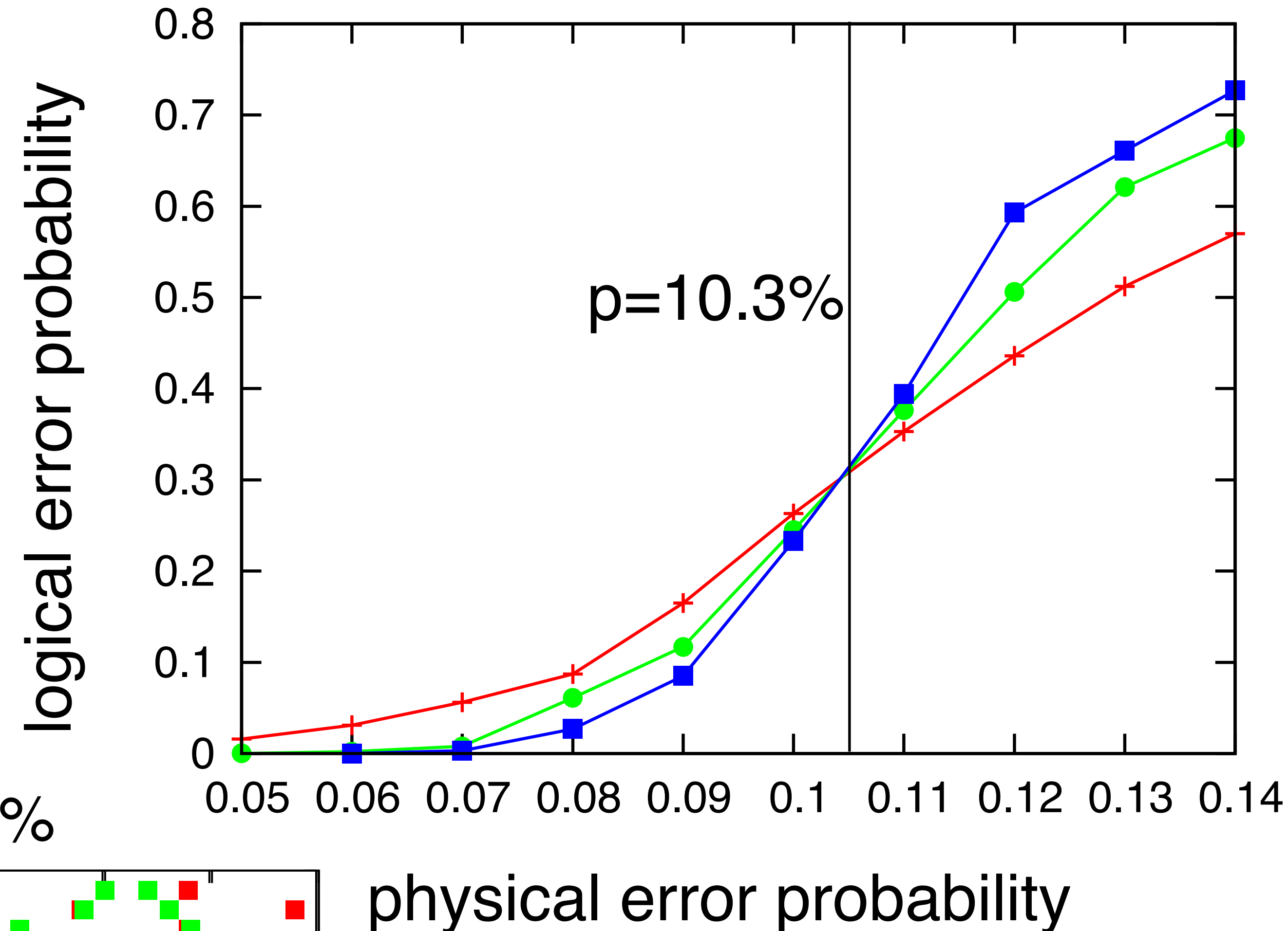


ある種のパーコレーション問題
→何らかの統計力学模型に対応？

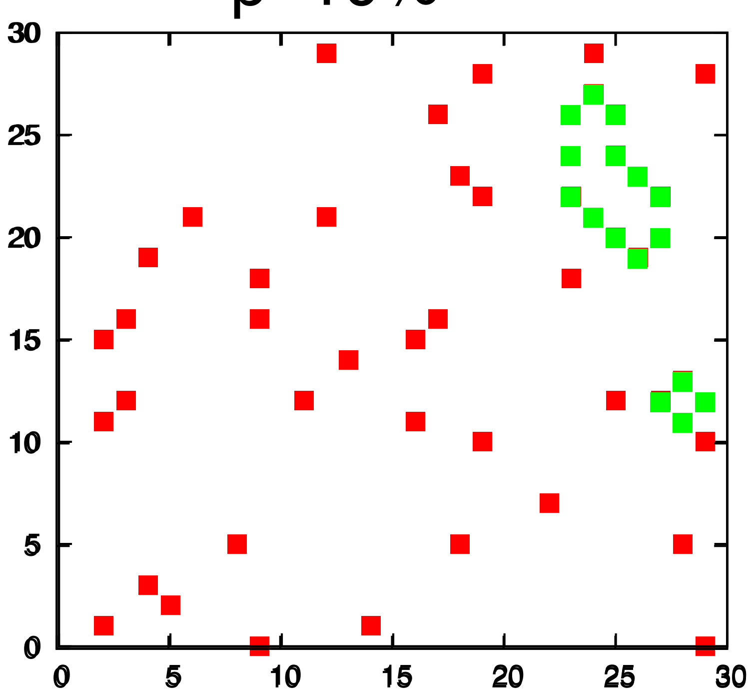
MWPMAを用いた場合の誤り訂正限界

p=3%

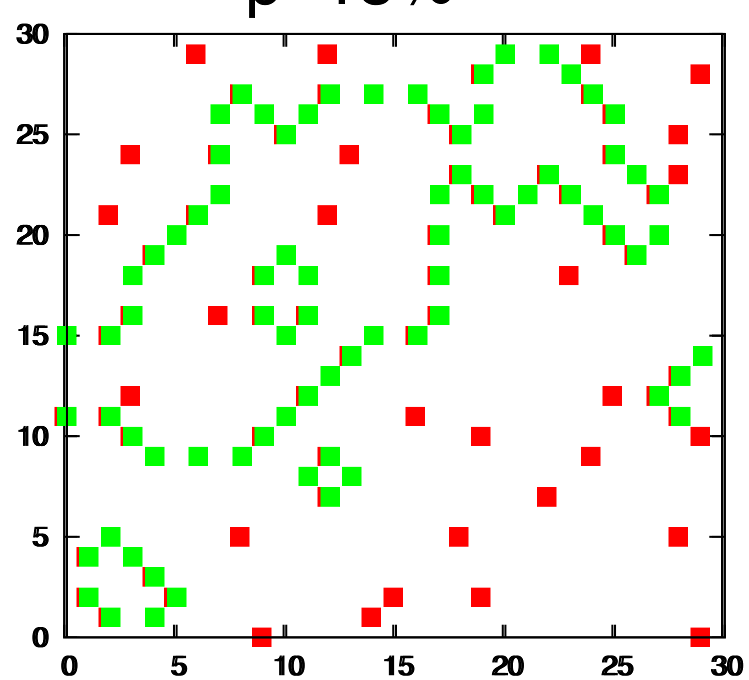




p=10%

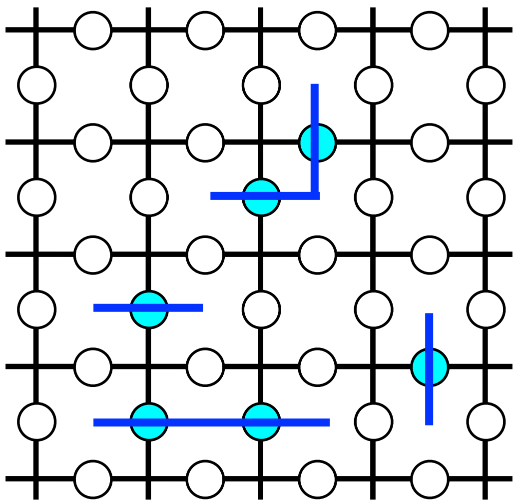


p=15%



ある種のパーコレーション問題
→何らかの統計力学模型に対応？

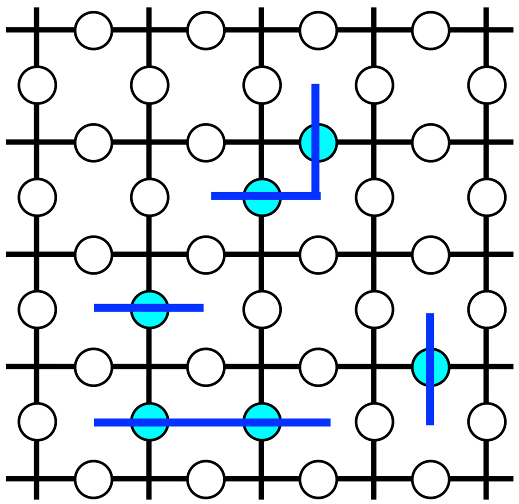
Surface codeとスピングラス模型



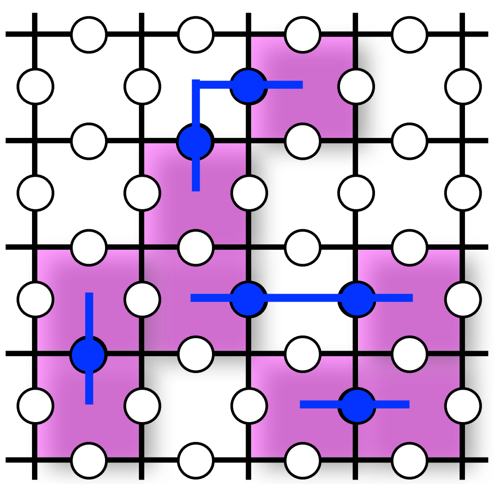
エラー: $\bar{c}_1^e$

エラー $\bar{c}_1^e$（dual 1-chain）が発生したとする（発生確率は各qubitに独立に確率 p とする）.

Surface codeとスピングラス模型



エラー: $\bar{c}_1^e$



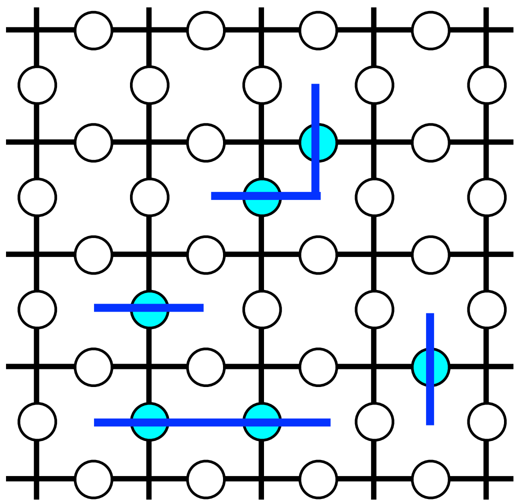
syndrome: $\partial \bar{c}_1^e$

推定したエラー: $\bar{c}_1^r$

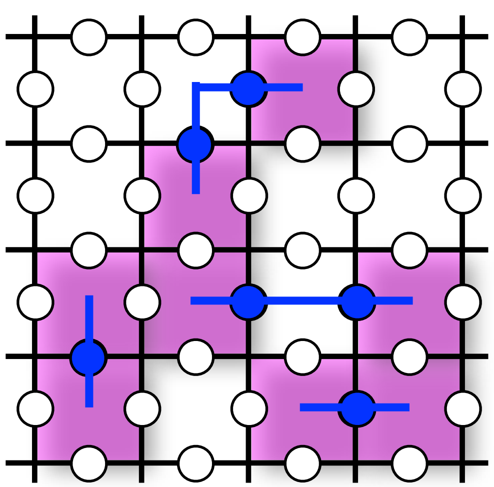
エラー $\bar{c}_1^e$（dual 1-chain）が発生したとする（発生確率は各qubitに独立に確率 p とする）.

エラー $\bar{c}_1^e$と同じ syndrome をもつ $\bar{c}_1^r$（$\partial \bar{c}_1^e = \partial \bar{c}_1^r$）を推定する問題（$\bar{c}_1^e + \bar{c}_1^r$が trivial cycleになってほしい）.

Surface codeとスピングラス模型



エラー: $\bar{c}_1^e$



syndrome: $\partial \bar{c}_1^e$

推定したエラー: $\bar{c}_1^r$

エラー $\bar{c}_1^e$（dual 1-chain）が発生したとする（発生確率は各qubitに独立に確率 p とする）.

エラー $\bar{c}_1^e$ と同じ syndrome をもつ $\bar{c}_1^r$（$\partial \bar{c}_1^e = \partial \bar{c}_1^r$）を推定する問題（$\bar{c}_1^e + \bar{c}_1^r$ が trivial cycleになってほしい）.

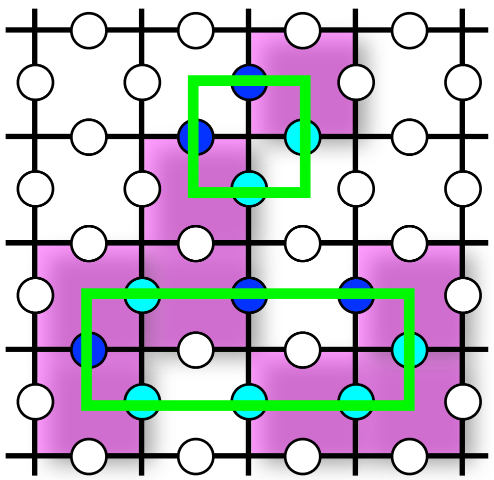
その出現確率を,

$$
\begin{aligned}
p(\bar{c}_1^r | \partial \bar{c}_1^e) &= (1-p')^{|\bar{E}|} \prod_l \left(\frac{p'}{1-p'} \right)^{z_l^r} \Big|_{\partial \bar{c}_1^e = \partial \bar{c}_1^r} \\
&= \mathcal{N} \exp \left[\beta' \sum_l u_l^r \right] \Big|_{\partial \bar{c}_1^e = \partial \bar{c}_1^r}
\end{aligned}
$$

ここで $e^{-\beta'} = \sqrt{\dfrac{p'}{1-p'}}$ ←推定のためのパラメータ

$u_l^r = (-1)^{z_l^r}$ ←±1 変数にした

Surface codeとスピングラス模型

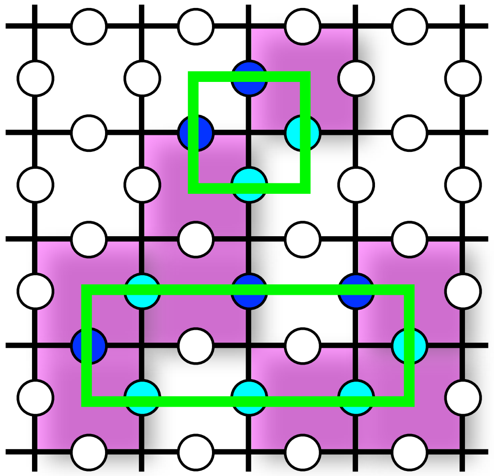


syndrome: $\partial \bar{c}_1^e$

$$p(\bar{c}_1^r | \partial \bar{c}_1^e) = \mathcal{N} \exp\left[\beta' \sum_l u_l^r\right]\Bigg|_{\boxed{\partial \bar{c}_1^e = \partial \bar{c}_1^r}}$$

解きたい.

Surface codeとスピングラス模型



syndrome: $\partial \bar{c}_1^e$

$$p(\bar{c}_1^r | \partial \bar{c}_1^e) = \mathcal{N} \exp \left[\beta' \sum_l u_l^r \right] \Bigg|_{\boxed{\partial \bar{c}_1^e = \partial \bar{c}_1^r}}$$

解きたい.

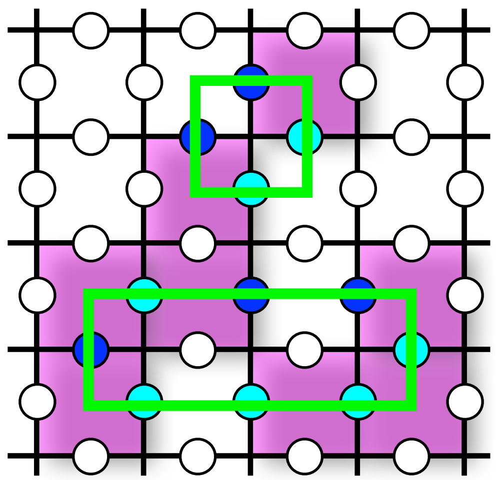
$$\bar{c}_1^e + \bar{c}_1^r = L$$

(trivial cycle)

$$u_l^e u_l^r = u_l^L$$

$$p(\bar{c}_1^r | \partial \bar{c}_1^e) = \mathcal{N} \exp \left[\beta' \sum_l u_l^e u_l^L \right]$$

Surface codeとスピングラス模型



syndrome: $\partial \bar{c}_1^e$

$$p(\bar{c}_1^r|\partial \bar{c}_1^e) = \mathcal{N} \exp\left[\beta' \sum_l u_l^r\right]\Bigg|_{\boxed{\partial \bar{c}_1^e = \partial \bar{c}_1^r}}$$

解きたい.

$$\bar{c}_1^e + \bar{c}_1^r = L$$

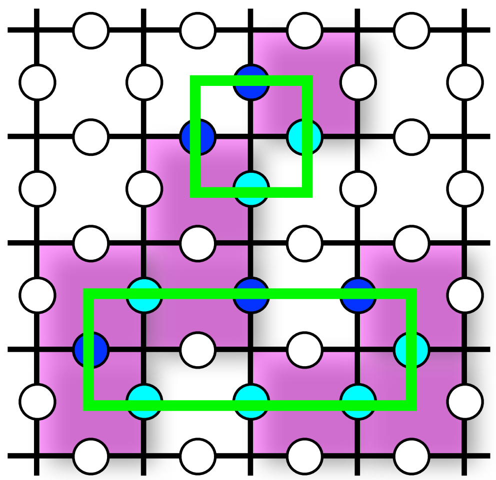
(trivial cycle)

$$u_l^e u_l^r = u_l^L$$

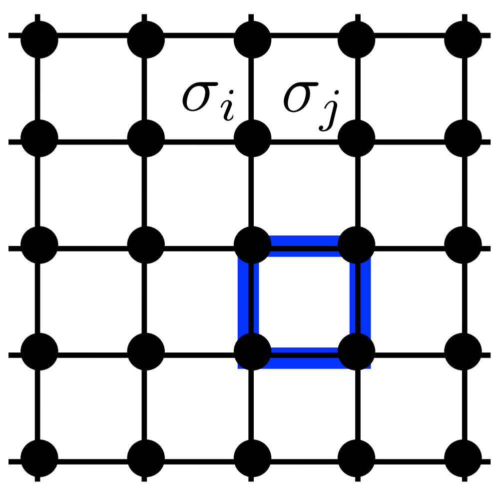
$$p(\bar{c}_1^r|\partial \bar{c}_1^e) = \mathcal{N} \exp\left[\beta' \sum_l u_l^e \boxed{u_l^L}\right]$$

trivial cycleの条件を
自動的に課したい.

Surface codeとスピングラス模型



syndrome: $\partial \bar{c}_1^e$



$$p(\bar{c}_1^r|\partial\bar{c}_1^e) = \mathcal{N}\exp\left[\beta'\sum_l u_l^r\right]\Bigg|_{\boxed{\partial\bar{c}_1^e=\partial\bar{c}_1^r}}$$

解きたい.

$$\bar{c}_1^e + \bar{c}_1^r = L$$

(trivial cycle)

$$u_l^e u_l^r = u_l^L$$

$$p(\bar{c}_1^r|\partial\bar{c}_1^e) = \mathcal{N}\exp\left[\beta'\sum_l u_l^e \boxed{u_l^L}\right]$$

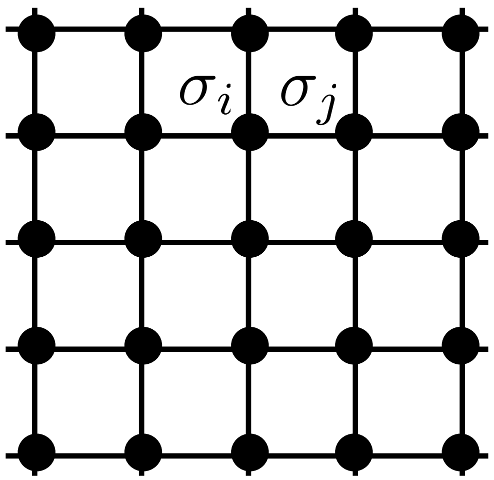
trivial cycleの条件を
自動的に課したい.

頂点上にゲージスピン σ_i を導入

$$u_l^L = \sigma_i\sigma_j \qquad\qquad l = \langle ij\rangle$$

とすれば, $\prod_{l\in\partial f_i} u_l^L = 0$ となり，trivial cycle条件が自動的に満たせる.

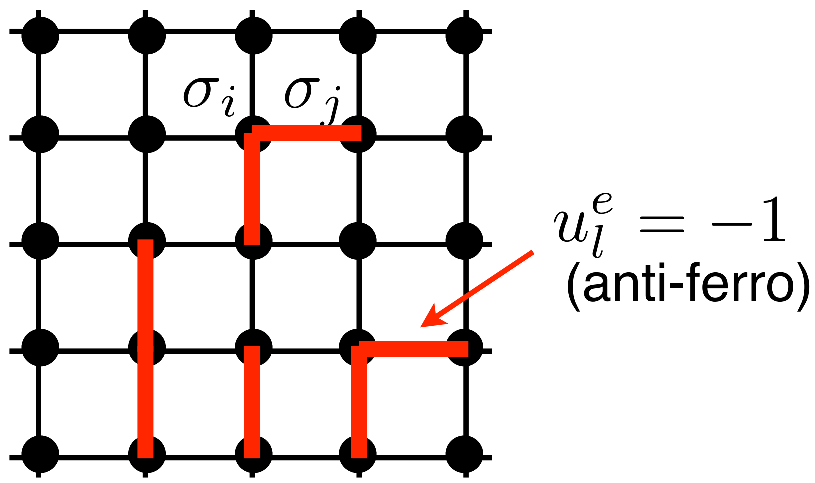
Surface codeとスピングラス模型



Random-bond Ising model (RBIM) のボルツマン因子：

$$p(\bar{c}_1^r|\partial\bar{c}_1^e) = \mathcal{N}\exp\left[\beta' \sum_{\langle ij \rangle} u_l^e \sigma_i \sigma_j\right] \equiv p(\{\sigma\})$$

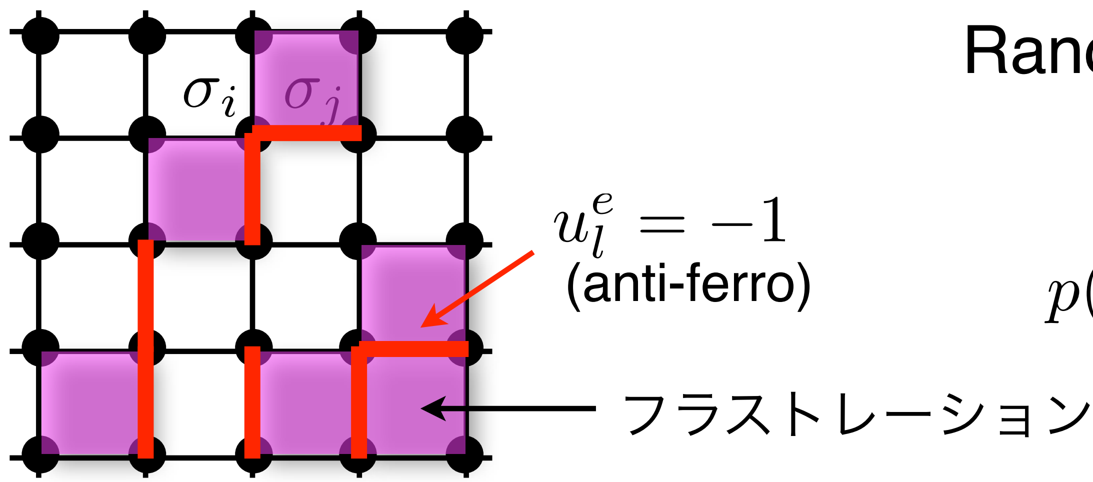
Surface codeとスピングラス模型



Random-bond Ising model (RBIM) のボルツマン因子：

$$p(\bar{c}_1^r|\partial\bar{c}_1^e) = \mathcal{N}\exp\left[\beta' \sum_{\langle ij\rangle} u_l^e \sigma_i \sigma_j\right] \equiv p(\{\sigma\})$$

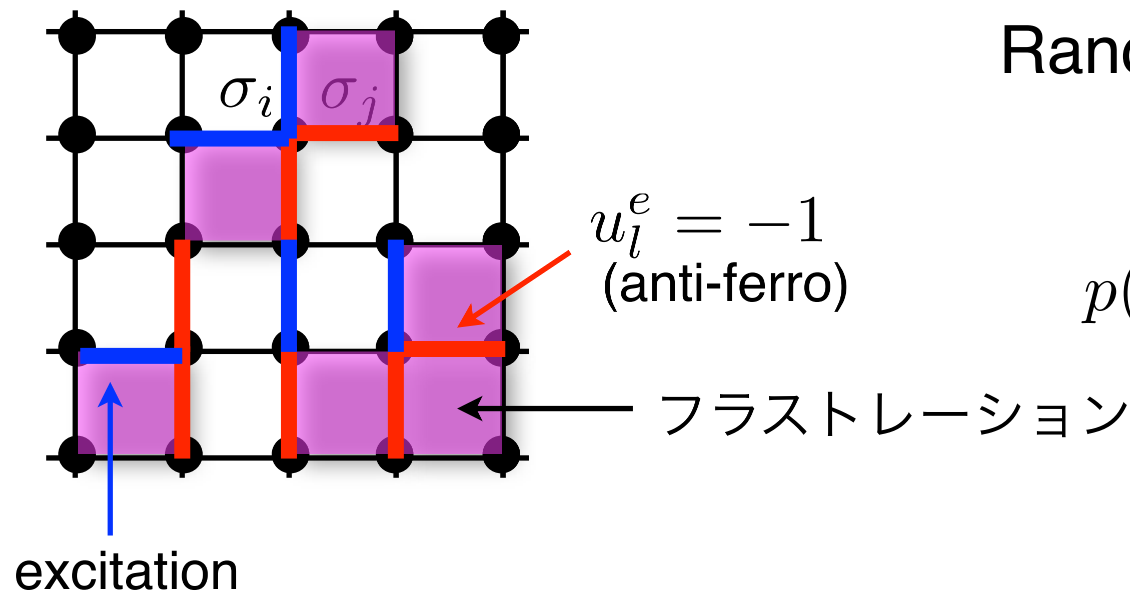
Surface codeとスピングラス模型



Random-bond Ising model (RBIM) のボルツマン因子：

$$p(\bar{c}_1^r|\partial\bar{c}_1^e) = \mathcal{N}\exp\left[\beta'\sum_{\langle ij\rangle}u_l^e\sigma_i\sigma_j\right] \equiv p(\{\sigma\})$$

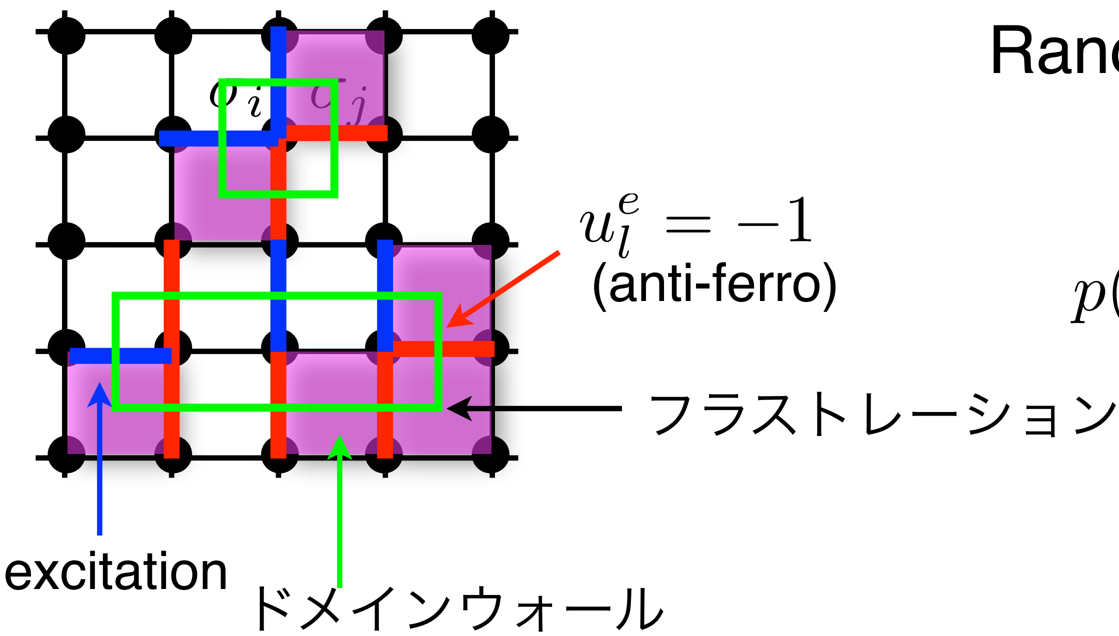
Surface codeとスピングラス模型



Random-bond Ising model (RBIM) のボルツマン因子：

$$p(\bar{c}_1^r | \partial \bar{c}_1^e) = \mathcal{N} \exp\left[\beta' \sum_{\langle ij \rangle} u_l^e \sigma_i \sigma_j\right] \equiv p(\{\sigma\})$$

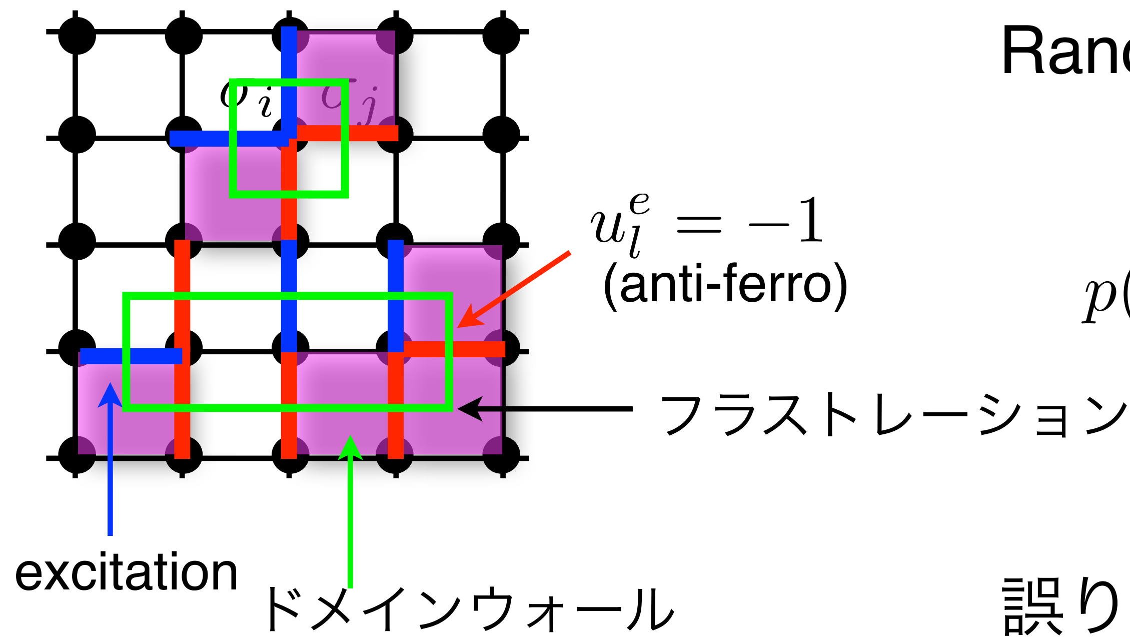
Surface codeとスピングラス模型



Random-bond Ising model (RBIM) のボルツマン因子：

$$p(\bar{c}_1^r|\partial\bar{c}_1^e) = \mathcal{N}\exp\left[\beta' \sum_{\langle ij\rangle} u_l^e \sigma_i \sigma_j\right] \equiv p(\{\sigma\})$$

Surface codeとスピングラス模型



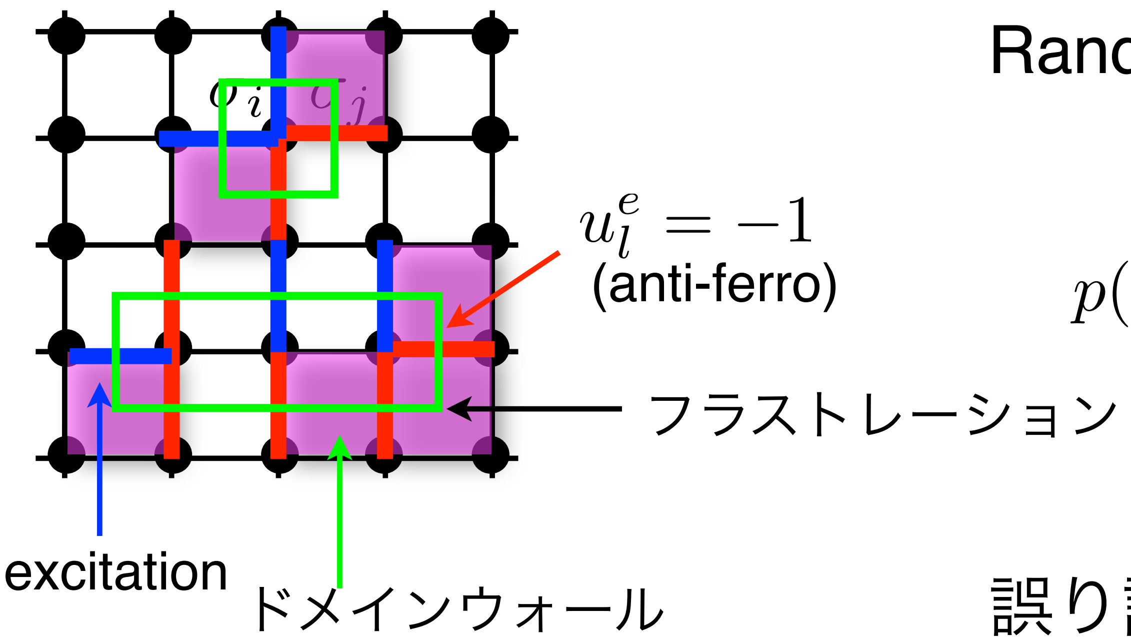
Random-bond Ising model (RBIM) のボルツマン因子：

$$p(\bar{c}_1^r | \partial \bar{c}_1^e) = \mathcal{N} \exp\left[\beta' \sum_{\langle ij \rangle} u_l^e \sigma_i \sigma_j\right] \equiv p(\{\sigma\})$$

誤り訂正が成功する確率は，全ての trivial cycleでの確率の和なので：

$$p_{\mathrm{suc}} \propto \sum_{\{\sigma\}} \mathcal{N} \exp\left[\beta' \sum_{\langle ij \rangle} u_l^e \sigma_i \sigma_j\right]$$

Surface codeとスピングラス模型



Random-bond Ising model (RBIM) のボルツマン因子：

$$p(\bar{c}_1^r|\partial\bar{c}_1^e) = \mathcal{N}\exp\left[\beta'\sum_{\langle ij\rangle}u_l^e\sigma_i\sigma_j\right] \equiv p(\{\sigma\})$$

誤り訂正が成功する確率は，全ての trivial cycleでの確率の和なので：

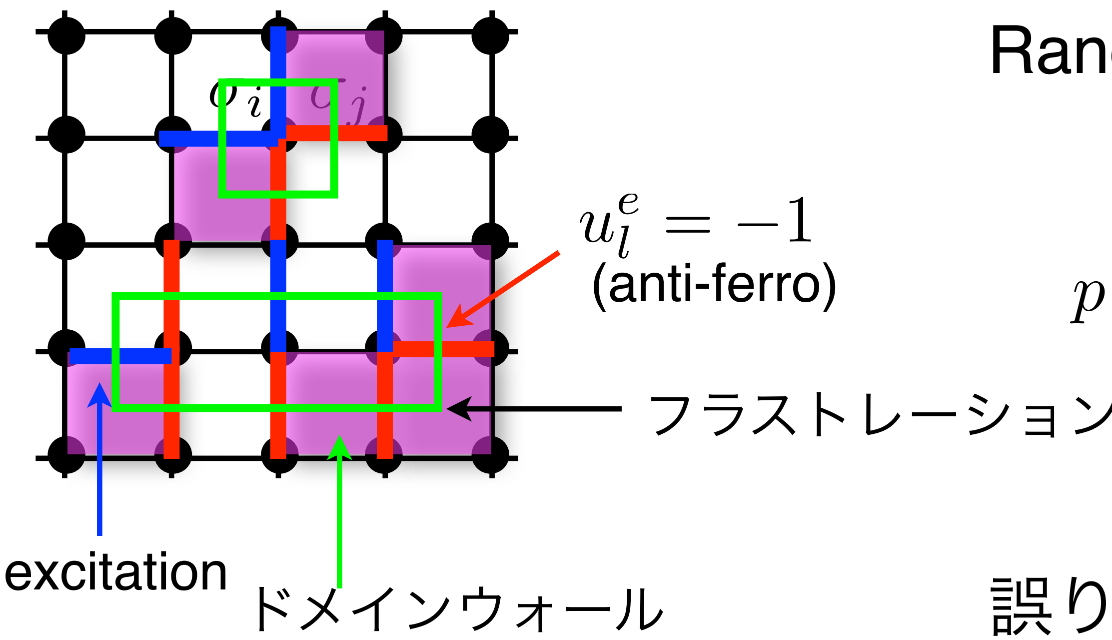
$$p_{\rm suc} \propto \sum_{\{\sigma\}}\mathcal{N}\exp\left[\beta'\sum_{\langle ij\rangle}u_l^e\sigma_i\sigma_j\right]$$

→RBIMの分配関数.

誤り訂正可能→ $p_{\rm suc} \to 1$

誤り訂正不可能→ $p_{\rm suc} \to {\rm const.} < 1$

Surface codeとスピングラス模型



Random-bond Ising model (RBIM) のボルツマン因子：

$$p(\bar{c}_1^r | \partial \bar{c}_1^e) = \mathcal{N} \exp \left[\beta' \sum_{\langle ij \rangle} u_l^e \sigma_i \sigma_j \right] \equiv p(\{\sigma\})$$

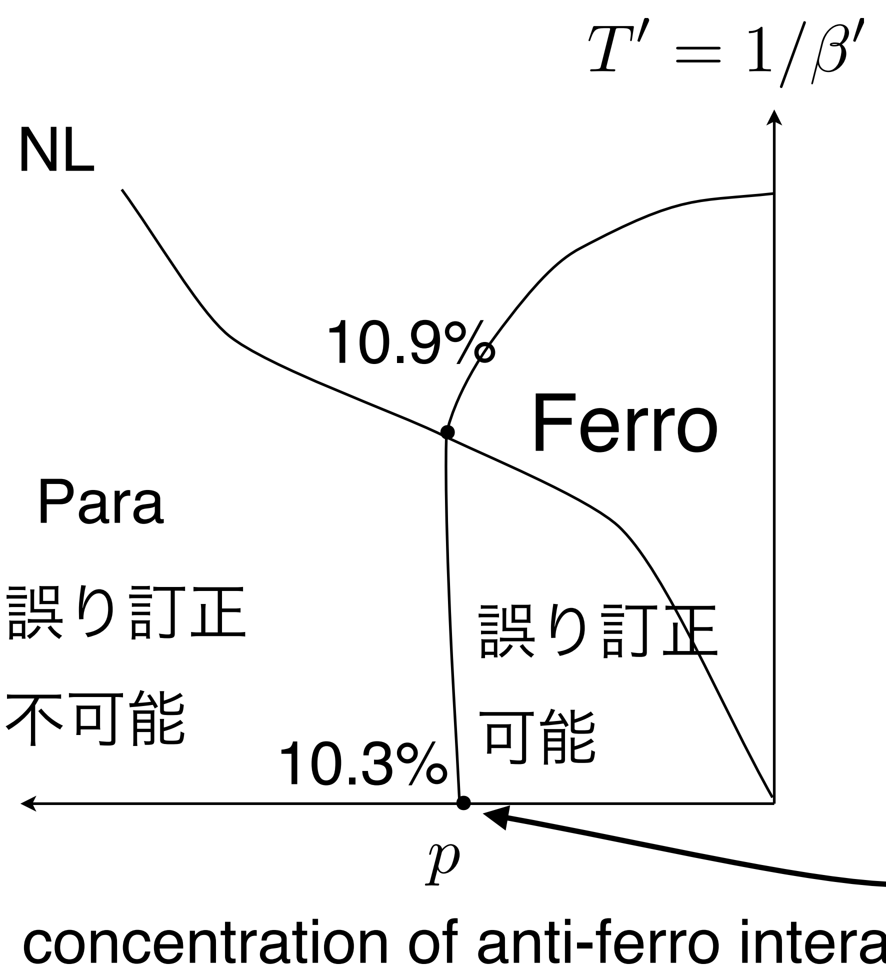
誤り訂正が成功する確率は，全ての trivial cycleでの確率の和なので：

$$p_{\rm suc} \propto \sum_{\{\sigma\}} \mathcal{N} \exp \left[\beta' \sum_{\langle ij \rangle} u_l^e \sigma_i \sigma_j \right]$$

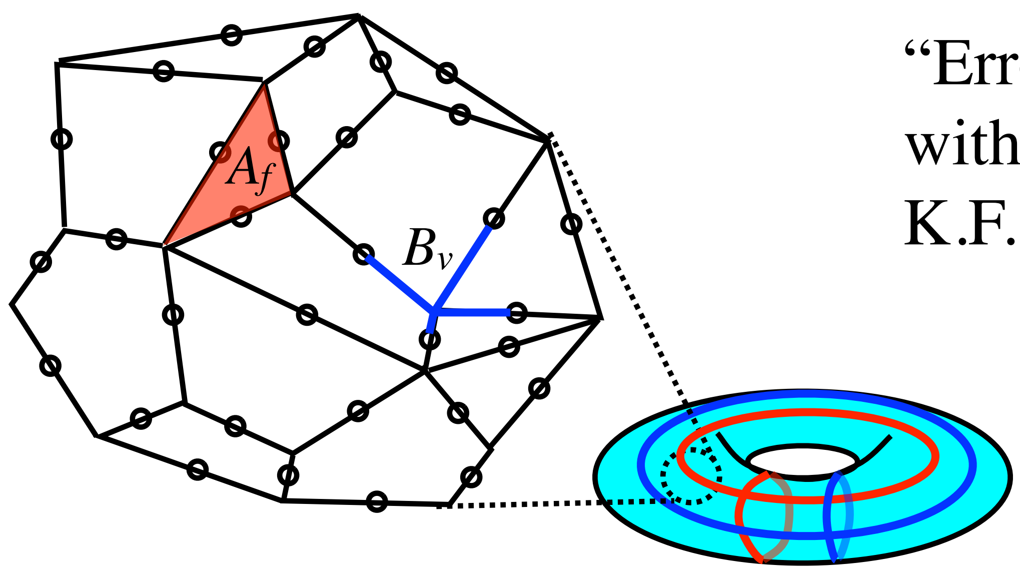
→RBIMの分配関数.

誤り訂正可能→ $p_{\rm suc} \to 1$

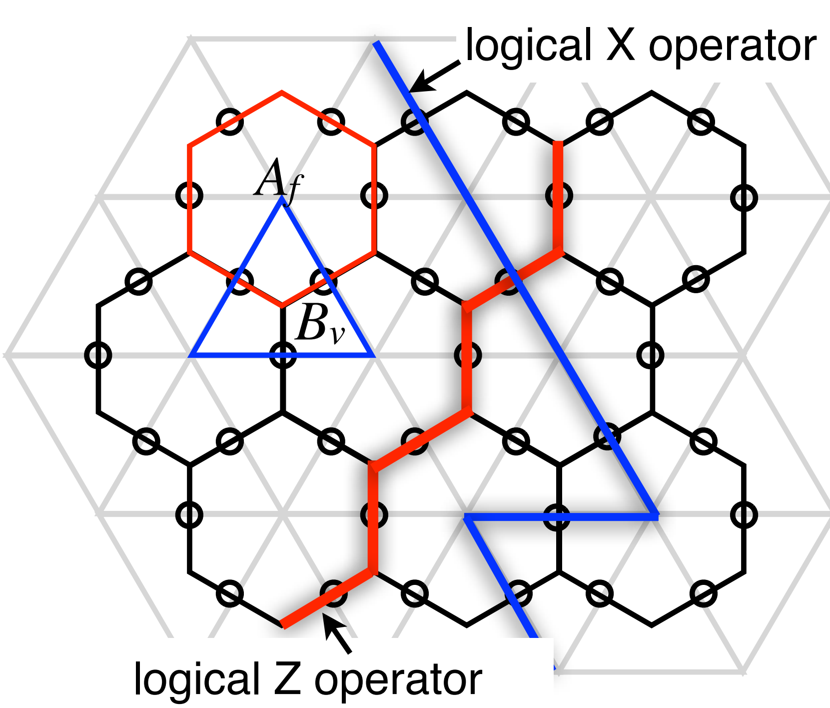
誤り訂正不可能→ $p_{\rm suc} \to \text{const.} < 1$

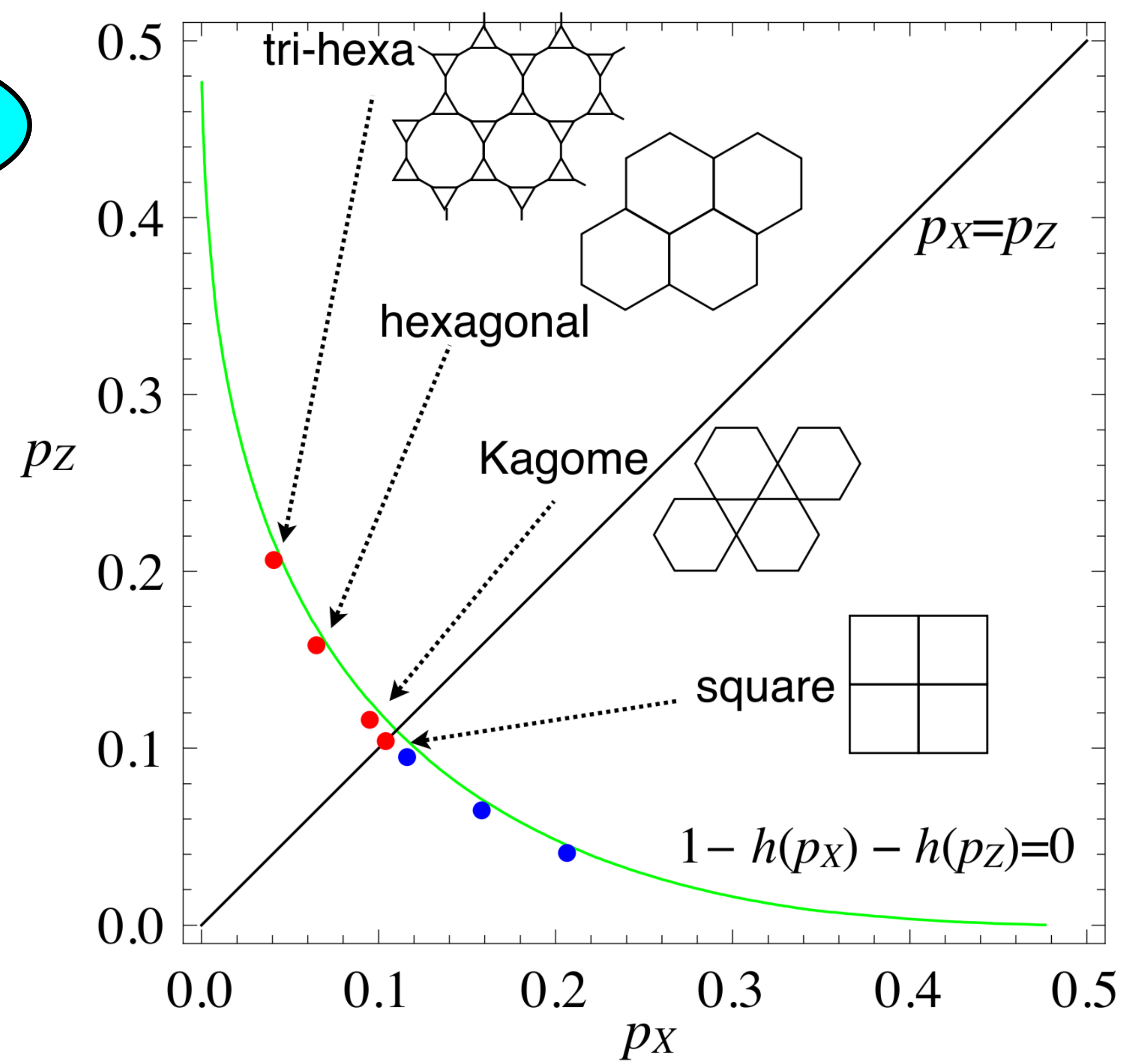


一般格子上のsurface code



"Error- and Loss-Tolerances of the surface codes with general lattice structures"
K.F. Y. Tokunaga, PRA **86**, 020303R (2012)





第五章: トポロジカル量子計算（回路モデルの視点から）

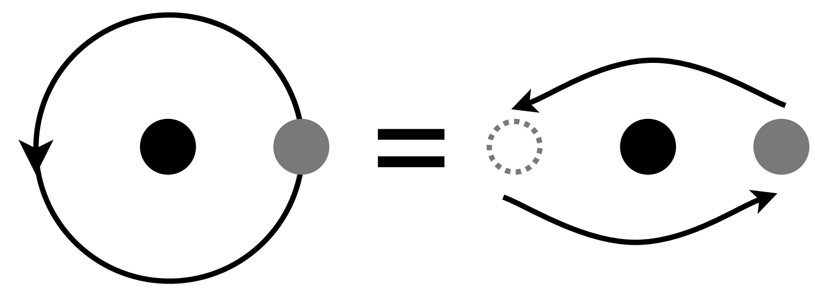
勉強する内容：

トポロジカル量子計算 = surface code 上で量子計算

トポロジカル量子計算とは？

粒子を”Braid”して量子計算する.

braiding：

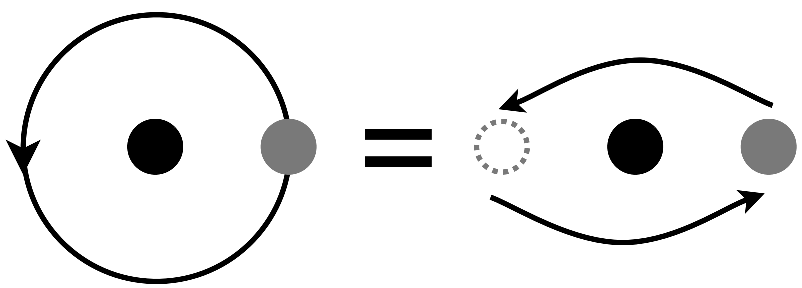


braiding = 2 × swap

→ボソンとフェルミオンだと何も起らない.

トポロジカル量子計算とは？

粒子を”Braid”して量子計算する.

braiding：

braiding = 2 × swap

→ボソンとフェルミオンだと何も起らない.

ボソン・フェルミオンと異なる

統計性をもった粒子→**エニオン**

２つのアプローチ

トポロジカル量子計算

物性的

本当に準粒子を生

成してBraidする.

non-Abelianエニオン

Fibonacciエニオン

２つのアプローチ

トポロジカル量子計算

量子情報的

qubitでsurfaceを構成して

擬似的に作った欠陥を

Braidする.

→一種の量子誤り訂正符号

Abelianエニオン

物性的

本当に準粒子を生

成してBraidする.

non-Abelianエニオン

Fibonacciエニオン

２つのアプローチ

トポロジカル量子計算

量子情報的

qubitでsurfaceを構成して

擬似的に作った欠陥を

Braidする.

→一種の量子誤り訂正符号

↓

Abelianエニオン

物性的

本当に準粒子を生

成してBraidする.

↓

non-Abelianエニオン

Fibonacciエニオン

２つのアプローチ

トポロジカル量子計算

量子情報的

qubitでsurfaceを構成して

擬似的に作った欠

Braidする.

→一種の量子誤り訂正符号

Abelianエニオン

Kitaev,
Annals Phys. **303**, 2 (2003);
arXiv:quant-ph/9707021.

物性的

本当に準粒子を生

Braidする.

non-Abelianエニオン

Fibonacciエニオン

２つのアプローチ

トポロジカル量子計算

量子情報的

qubitでsurfaceを構成して

擬似的に作った欠陥

Braidする.

→一種の量子誤

Abelianエニオン

Kitaev,
Annals Phys. **303**, 2 (2003);
arXiv:quant-ph/9707021.

Raussendorf-Harrington-Goyal,
Annals Phys. **321**, 2242 (2006);
NJP **9**, 199 (2007).

物性的

本当に準粒子を生

Braidする.

belianエニオン

Fibonacciエニオン

非トポロジカルとの違い？

非トポロジカル（回路的）

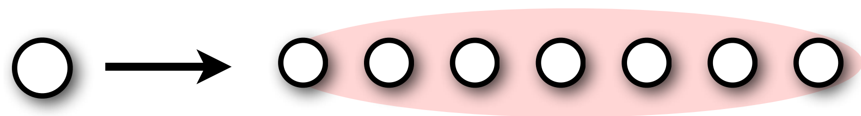
トポロジカル（物理的）

非トポロジカルとの違い？

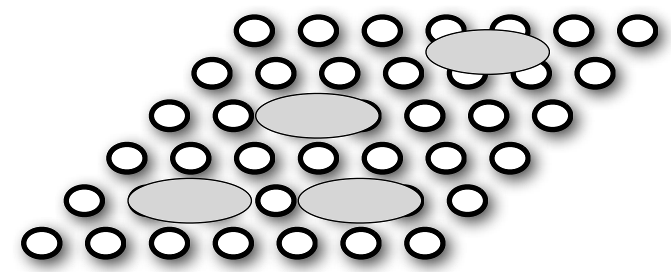
非トポロジカル（回路的）

physical qubit　　logical qubit

量子誤り訂正符号:



トポロジカル（物理的）



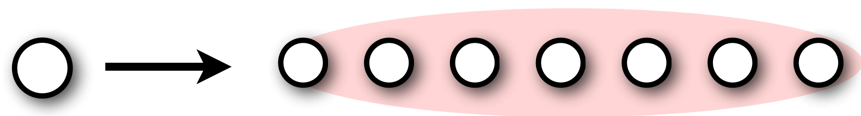
一つの巨大な符号の中に
たくさんのlogical qubitが

非トポロジカルとの違い？

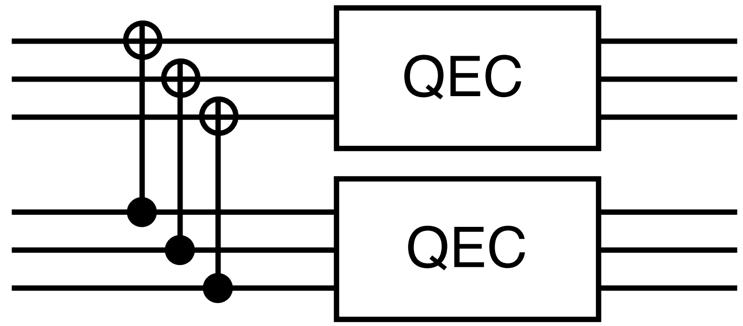
非トポロジカル（回路的）

physical qubit　　logical qubit

量子誤り訂正符号:

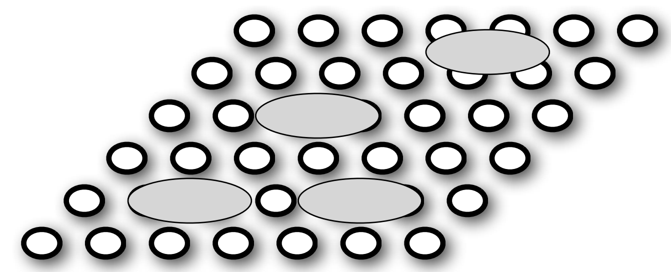


論理演算:

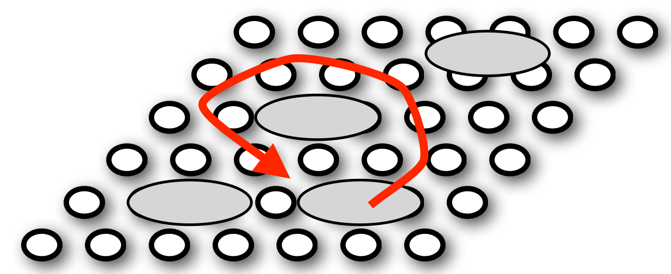


level-1 gate

トポロジカル（物理的）



一つの巨大な符号の中に
たくさんのlogical qubitが



符号空間を変形することにとって
てlogical operation←Braiding

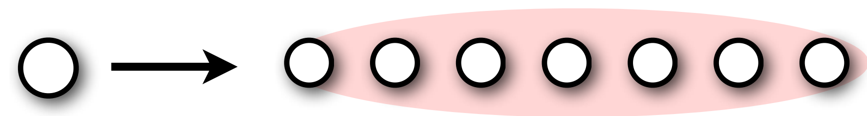
非トポロジカルとの違い？

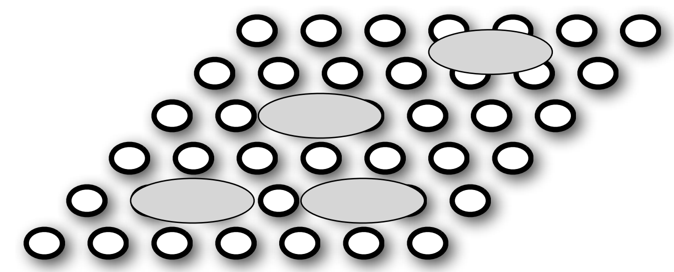
非トポロジカル（回路的）

トポロジカル（物理的）

量子誤り訂正符号:

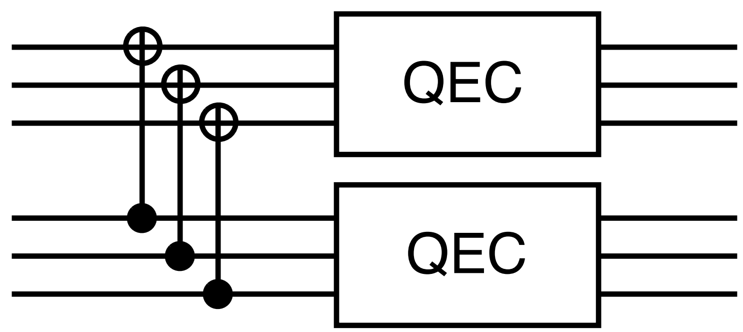
physical qubit　　logical qubit



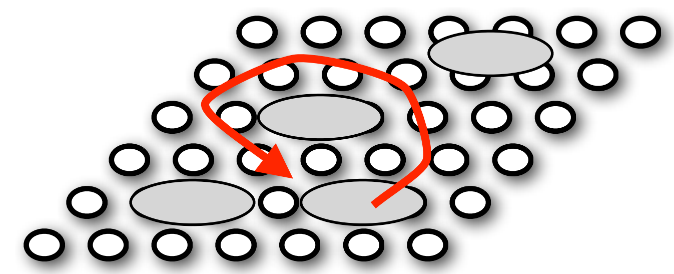


一つの巨大な符号の中に
たくさんのlogical qubitが

論理演算:

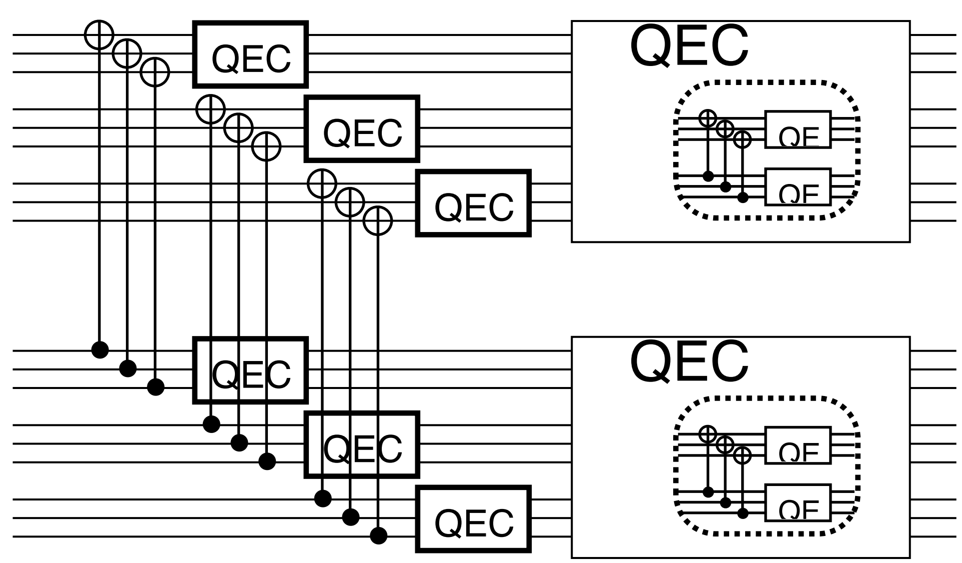


level-1 gate

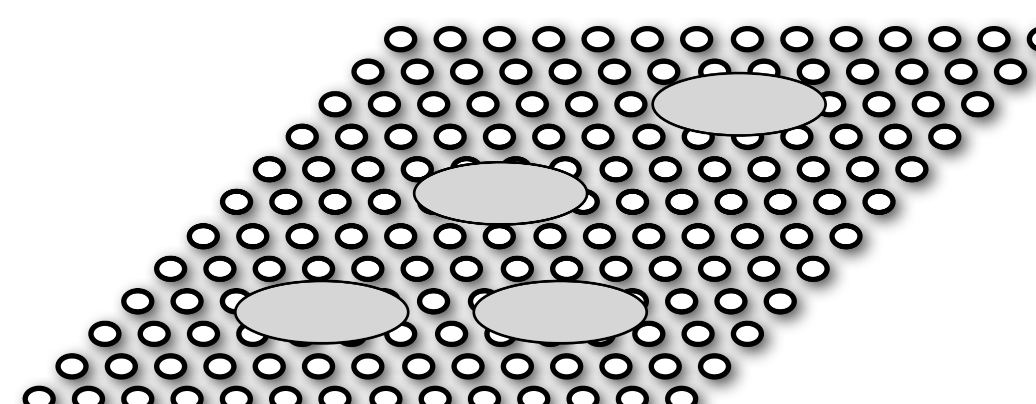


符号空間を変形することにとっ
てlogical operation←Braiding

フォールトト
レランス:



level-2 gate



ただサイズを大きくする.

特徴

１）フォールトトレラント量子計算が隣接する量子ビット間の演算だけでOK.

特徴

１）フォールトトレラント量子計算が隣接する量子ビット間の演算だけでOK.

２）にも関わらず誤り耐性は強い.

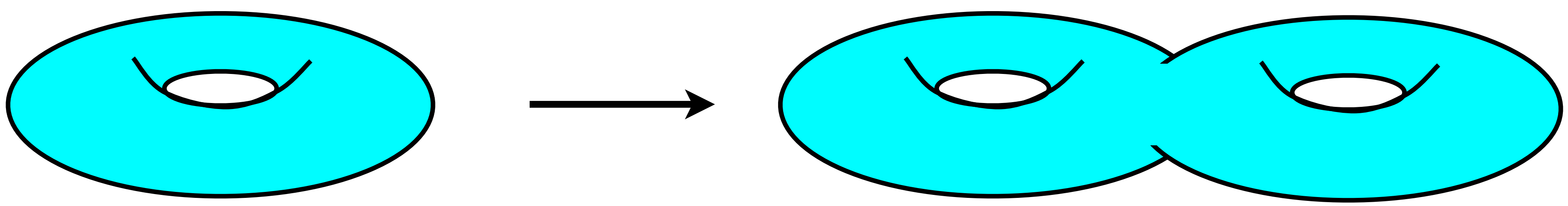
特徴

１）フォールトトレラント量子計算が隣接する量子ビット間の演算だけでOK.

２）にも関わらず誤り耐性は強い.

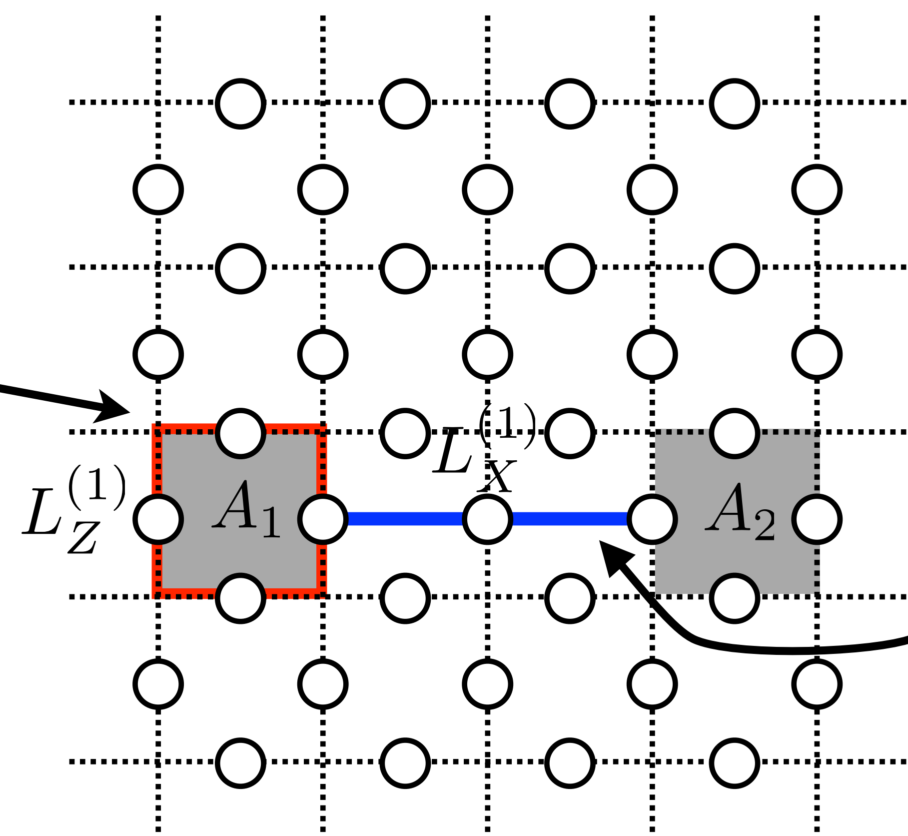
３）いわゆる情報理論っぽい構成とは全く異なって，むしろ物理的.

Defect pair = logical qubit



複雑になる．．．．

縮退をとく演算子
＝論理Pauli Z演算子
$L_Z^{(1)}$



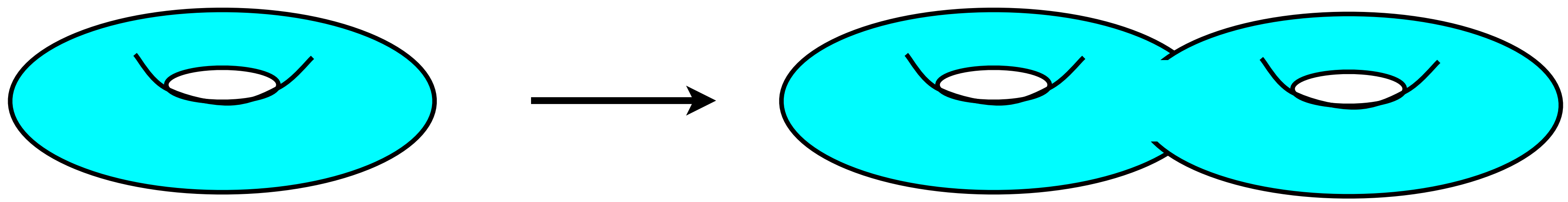
縮退をとく演算子
＝論理Pauli X演算子
$L_X^{(1)}$

表面に穴（defect）をあけよう！

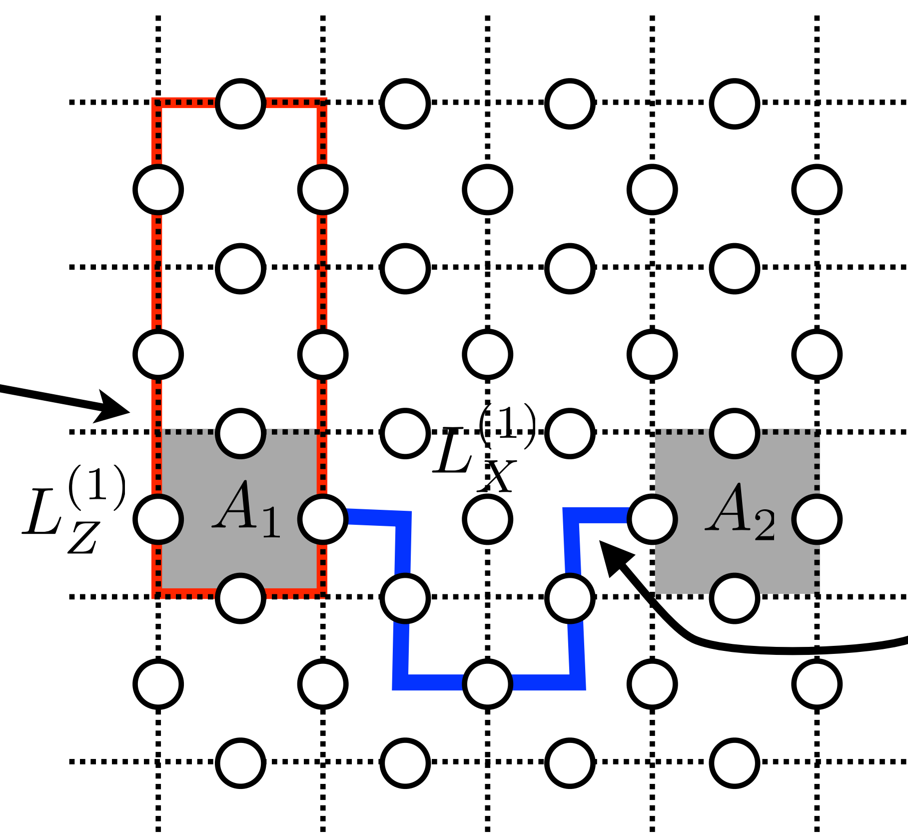
A_1とA_2をスタビライザーから取りのぞこう．

（ただし，A_1A_2の積は残したまま）→縮退が1qubit分増える

Defect pair = logical qubit



複雑になる．．．．

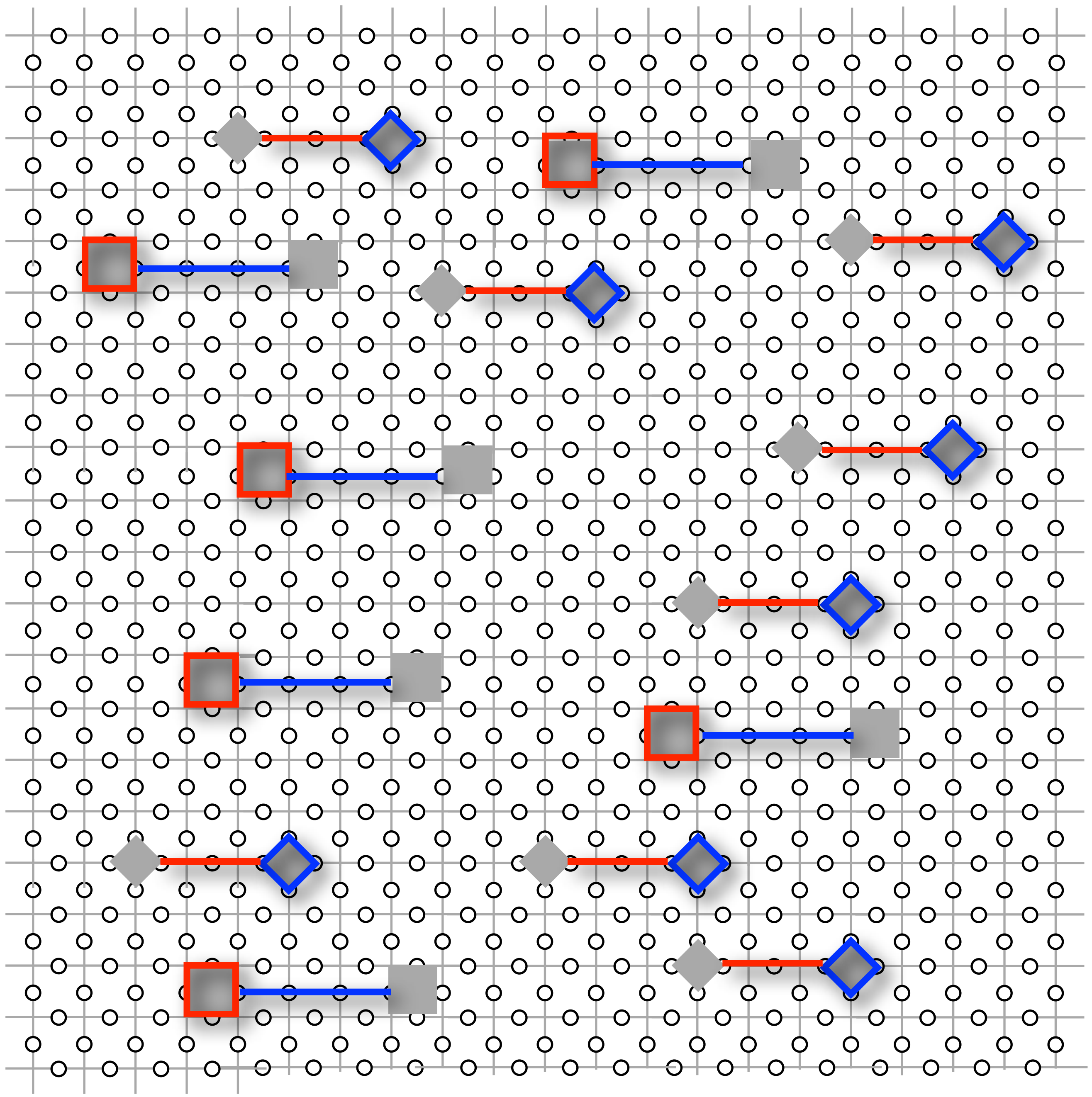


縮退をとく演算子
＝論理Pauli Z演算子
$L_Z^{(1)}$

縮退をとく演算子
＝論理Pauli X演算子
$L_X^{(1)}$

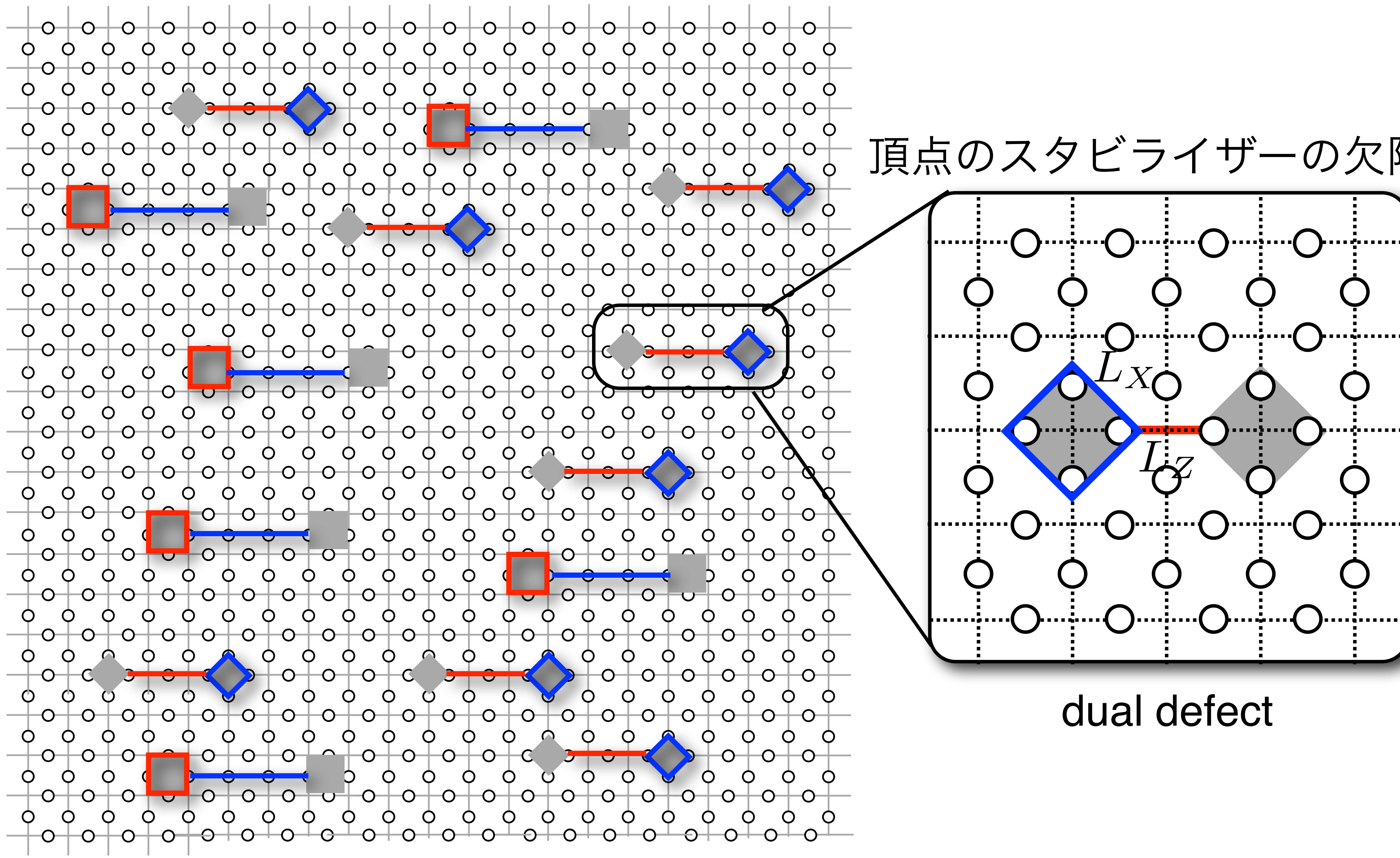
論理Pauli演算子は，トポロジーさえ同じであれば，
どの演算子でも等価（=スタビライザーを掛けると変形
できる）．

Defectが埋め込まれたsurface



一つの surface code にたくさんの論理qubitが埋め込まれている.

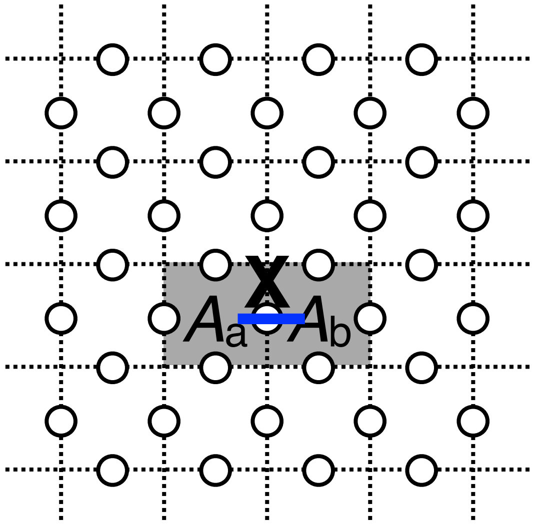
Defectが埋め込まれたsurface



一つの surface code にたくさんの論理qubitが埋め込まれている.

Defectの生成と移動

defect の生成 = 論理 qubit $\left|+\right\rangle_L^p$ の準備：

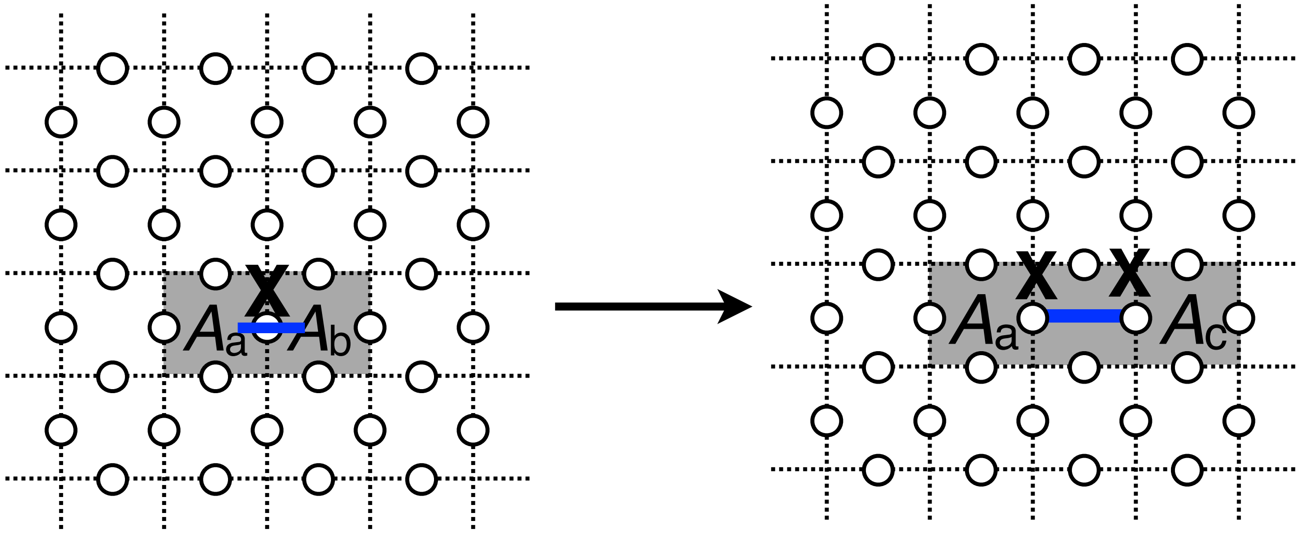


X基底で測定

defectの移動：

Defectの生成と移動

defect の生成 = 論理 qubit $\left|+\right\rangle_L^p$ の準備：



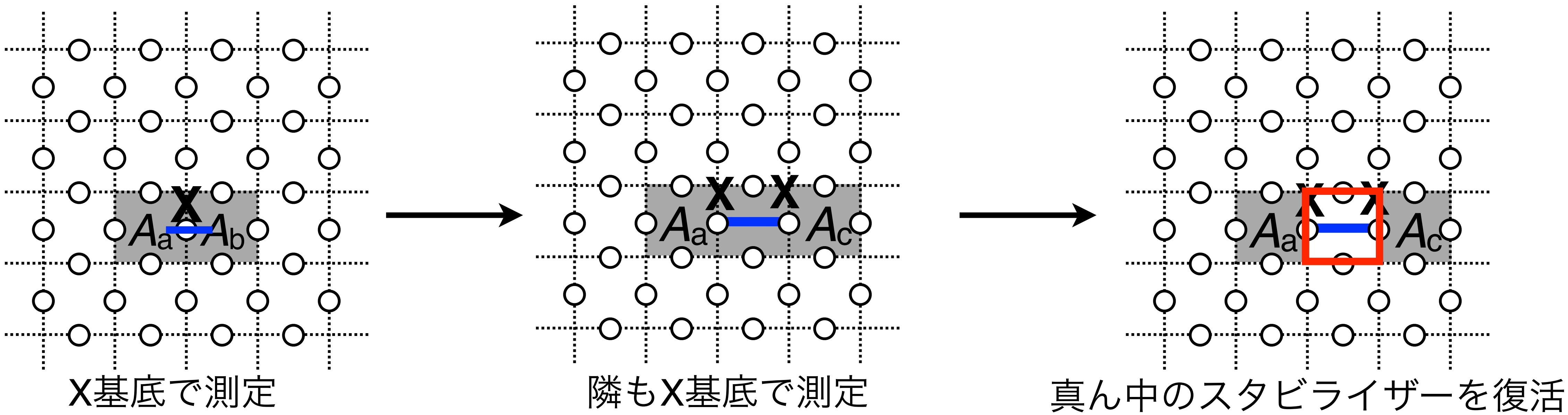
X基底で測定

隣もX基底で測定

defectの移動：

Defectの生成と移動

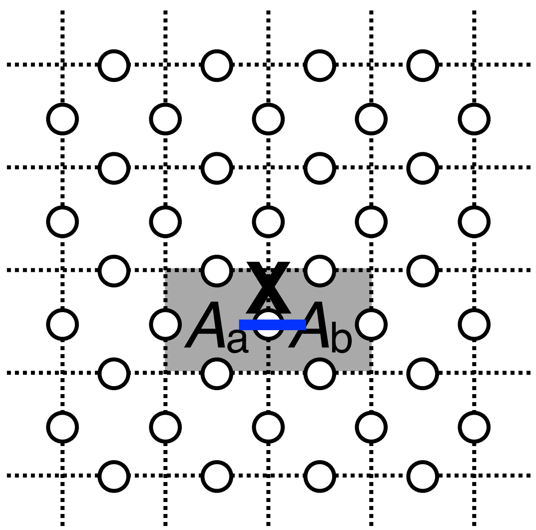
defect の生成 = 論理 qubit $\left|+\right\rangle_L^p$ の準備：



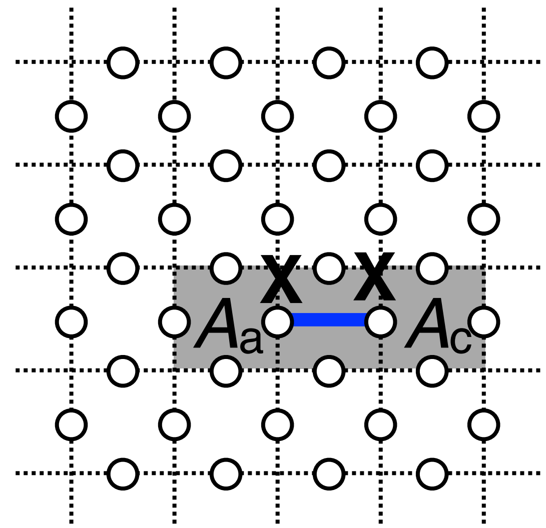
defectの移動：

Defectの生成と移動

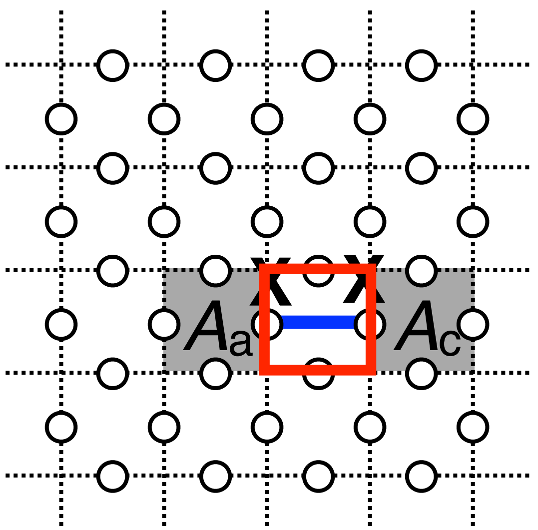
defect の生成 = 論理 qubit $|+\rangle_L^p$ の準備：



X基底で測定

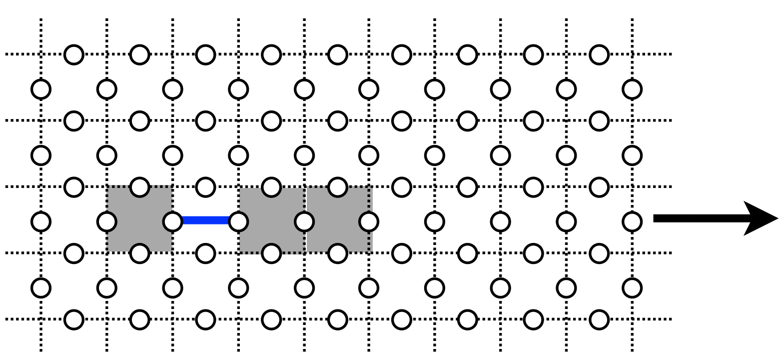


隣もX基底で測定



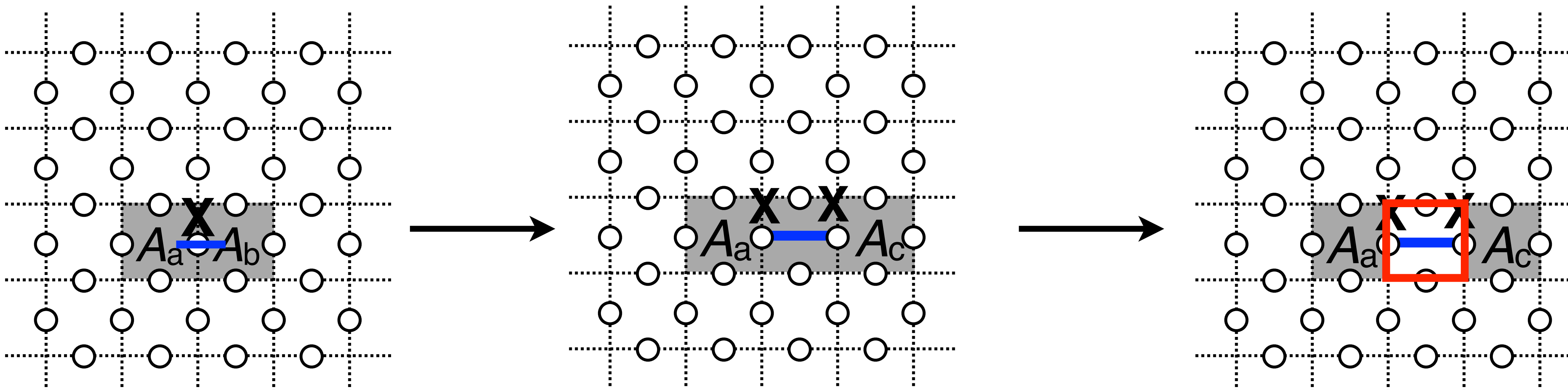
真ん中のスタビライザーを復活

defectの移動：



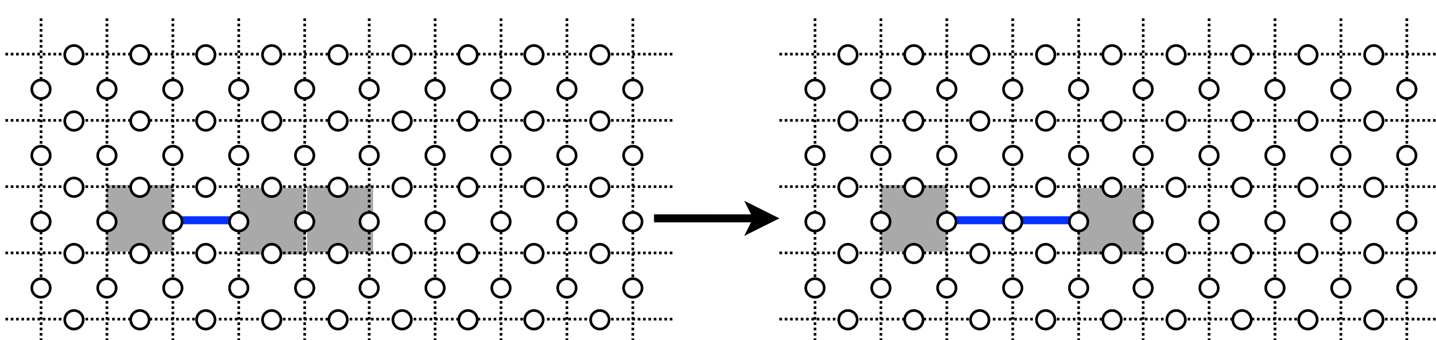
Defectの生成と移動

defect の生成 = 論理 qubit $\left|+\right\rangle_L^p$ の準備：



X基底で測定　　隣もX基底で測定　　真ん中のスタビライザーを復活

defectの移動：

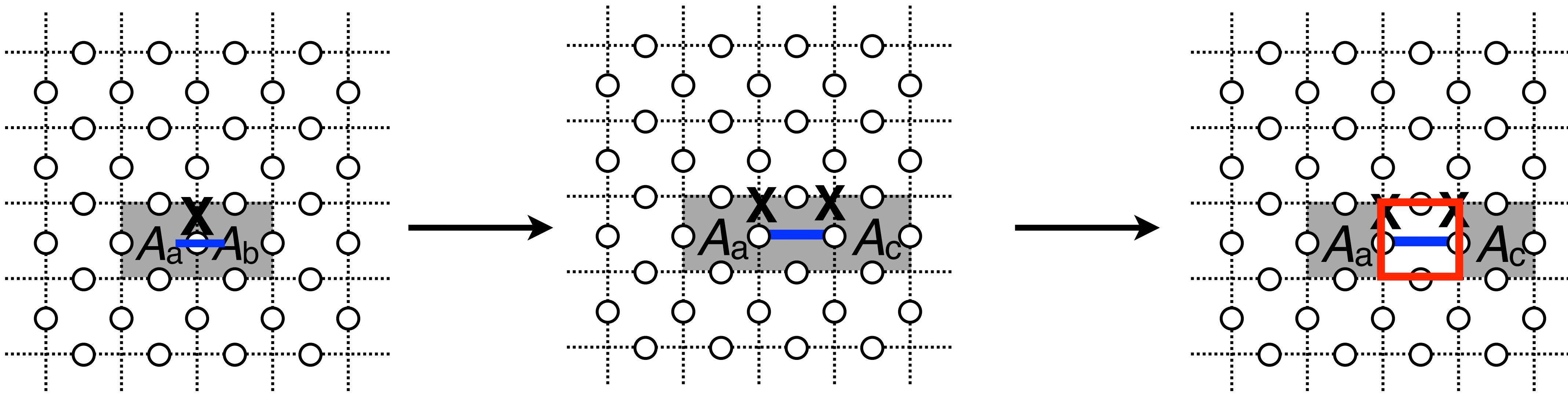


繰り返すと

ひとつ右にずれる

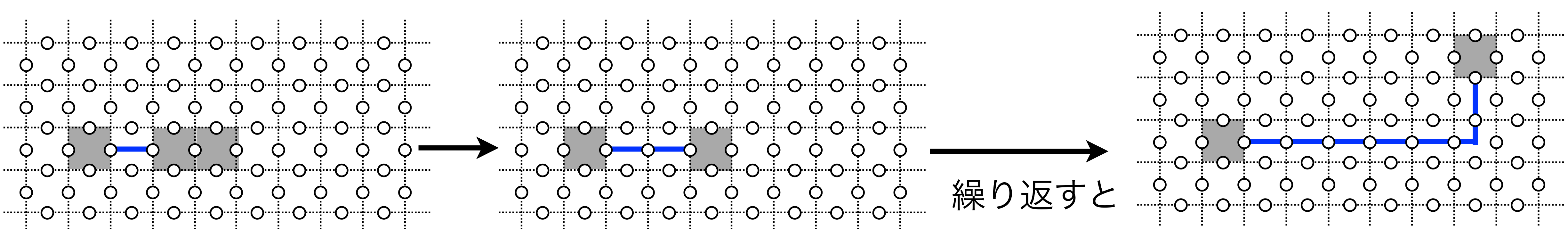
Defectの生成と移動

defect の生成 = 論理 qubit $\left|+\right\rangle_L^p$ の準備：



X基底で測定　　隣もX基底で測定　　真ん中のスタビライザーを復活

defectの移動：



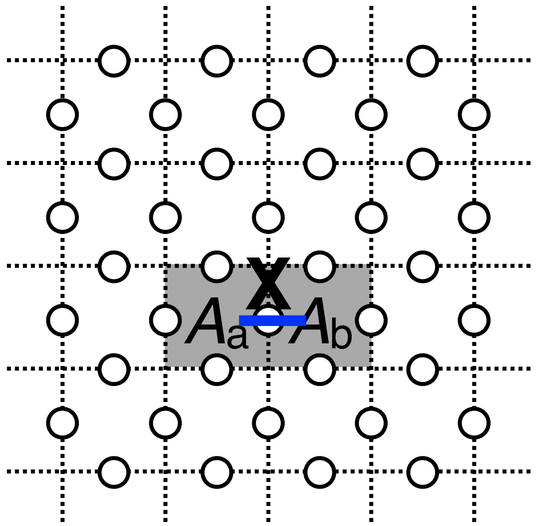
ひとつ右にずれる　　defectが動いた！

Logical qubit: defect pair creation

論理Pauli X演算子 $L_X^{(1)}$ の固有状態 $\left|+\right\rangle_L^p$

Logical qubit: defect pair creation

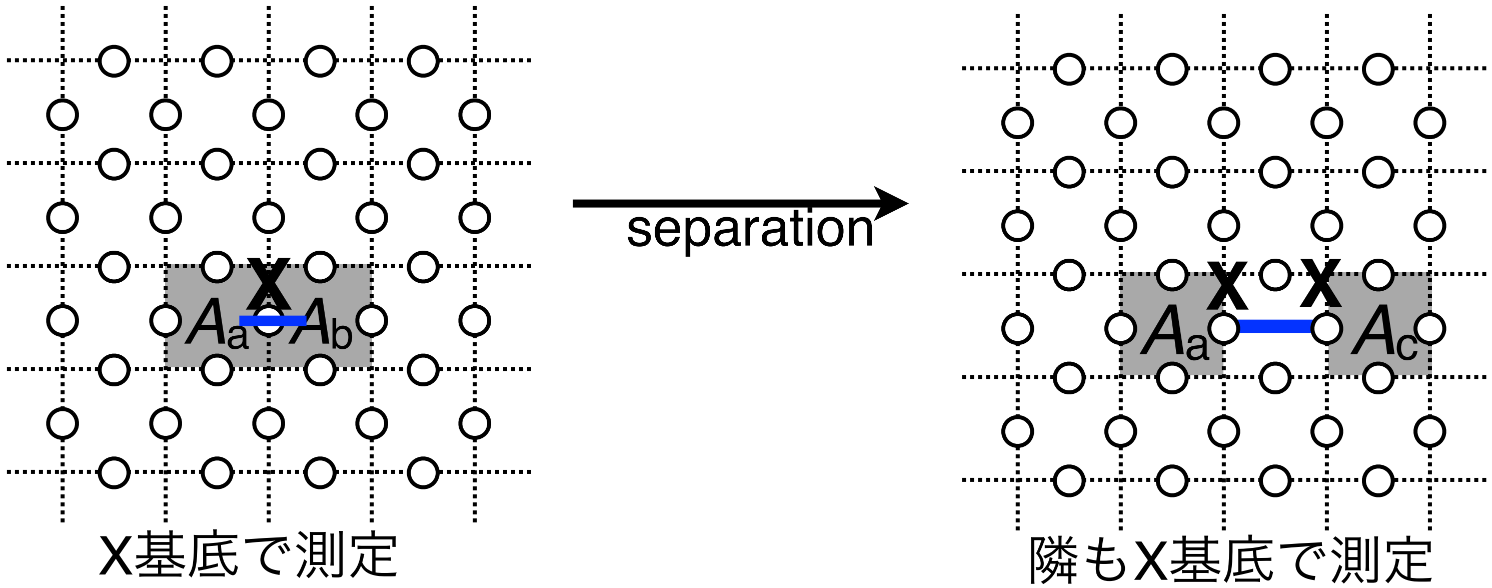
論理Pauli X演算子 $L_X^{(1)}$ の固有状態 $\left|+\right\rangle_L^p$



X基底で測定

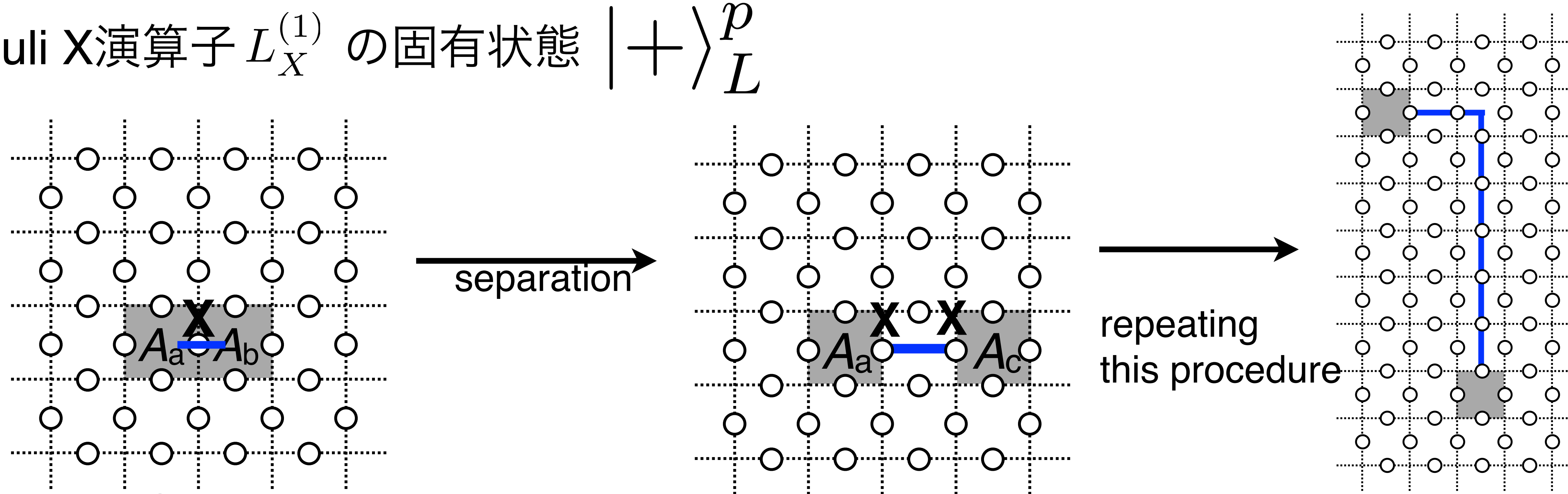
Logical qubit: defect pair creation

論理Pauli X演算子 $L_X^{(1)}$ の固有状態 $\left|+\right\rangle_L^p$



Logical qubit: defect pair creation

論理Pauli X演算子 $L_X^{(1)}$ の固有状態 $\left|+\right\rangle_L^p$



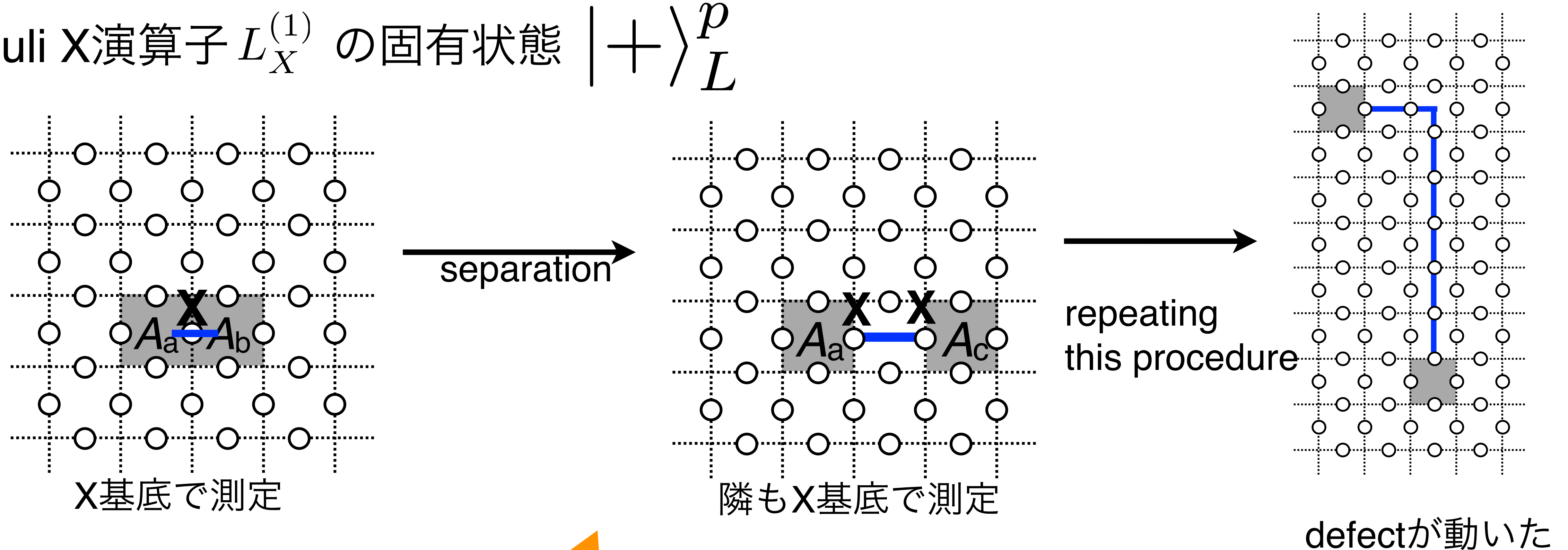
X基底で測定

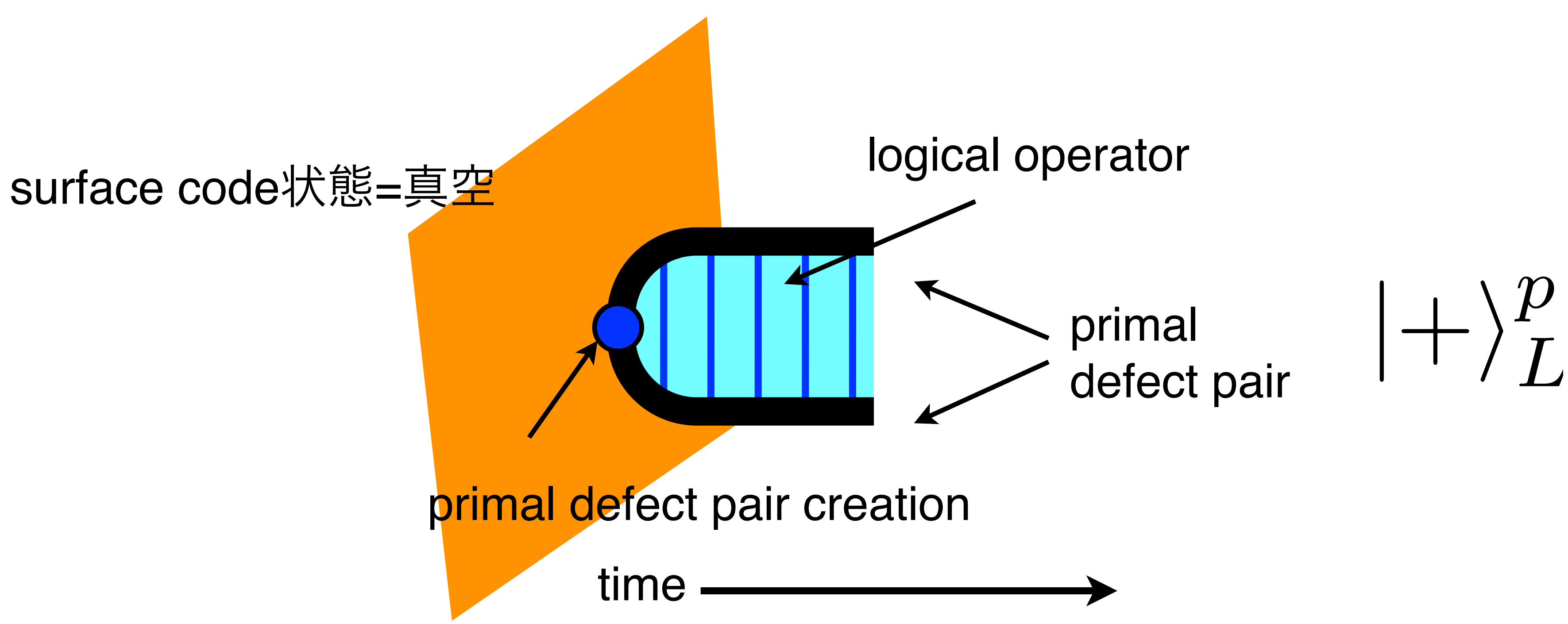
隣もX基底で測定

defectが動いた！

Logical qubit: defect pair creation

論理Pauli X演算子 $L_X^{(1)}$ の固有状態 $\left|+\right\rangle_L^p$



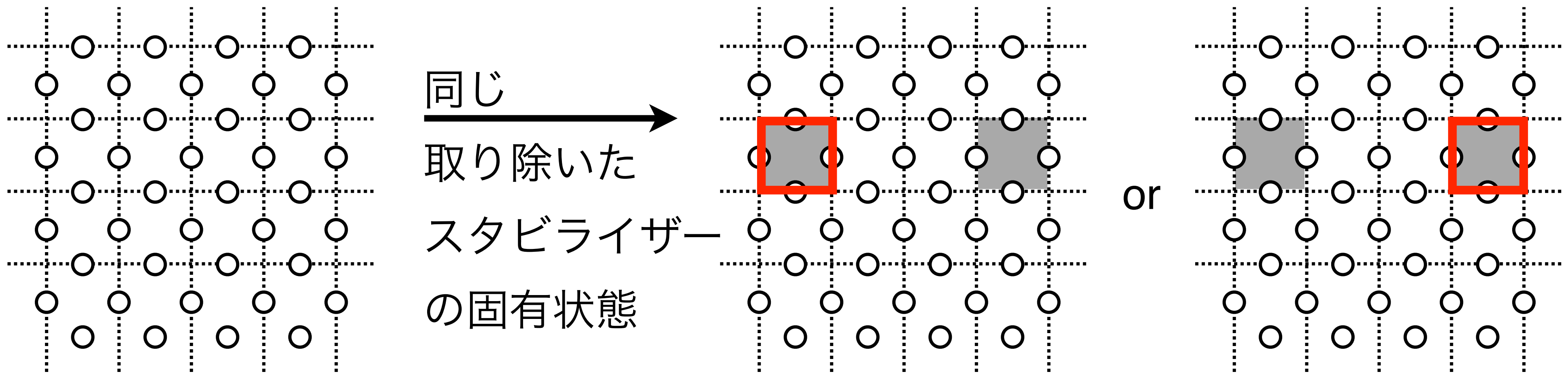


Logical qubit: defect pair creation

論理Pauli Z演算子 $L_Z^{(1)}$ の固有状態 $|0\rangle_L^p$

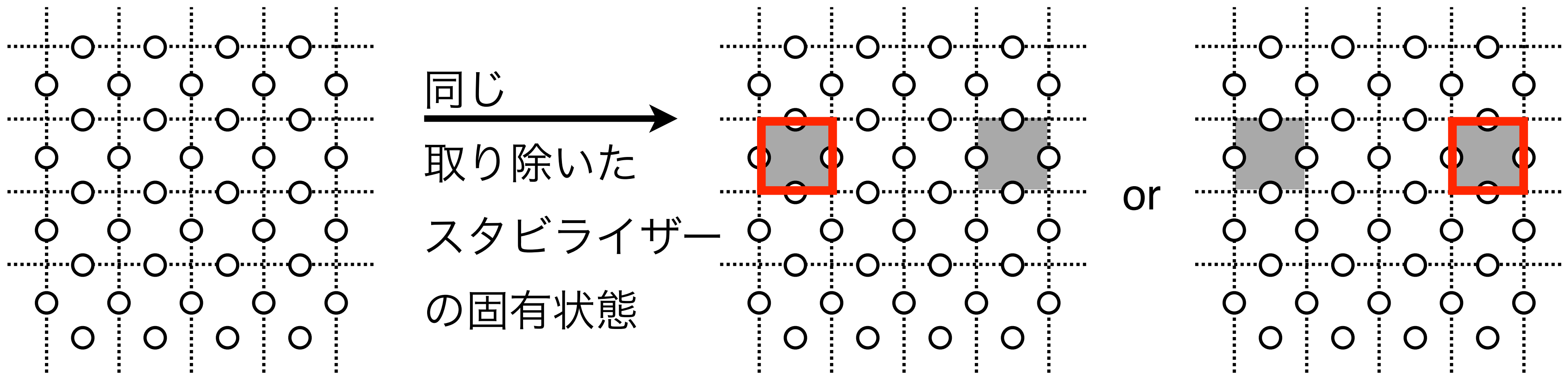
Logical qubit: defect pair creation

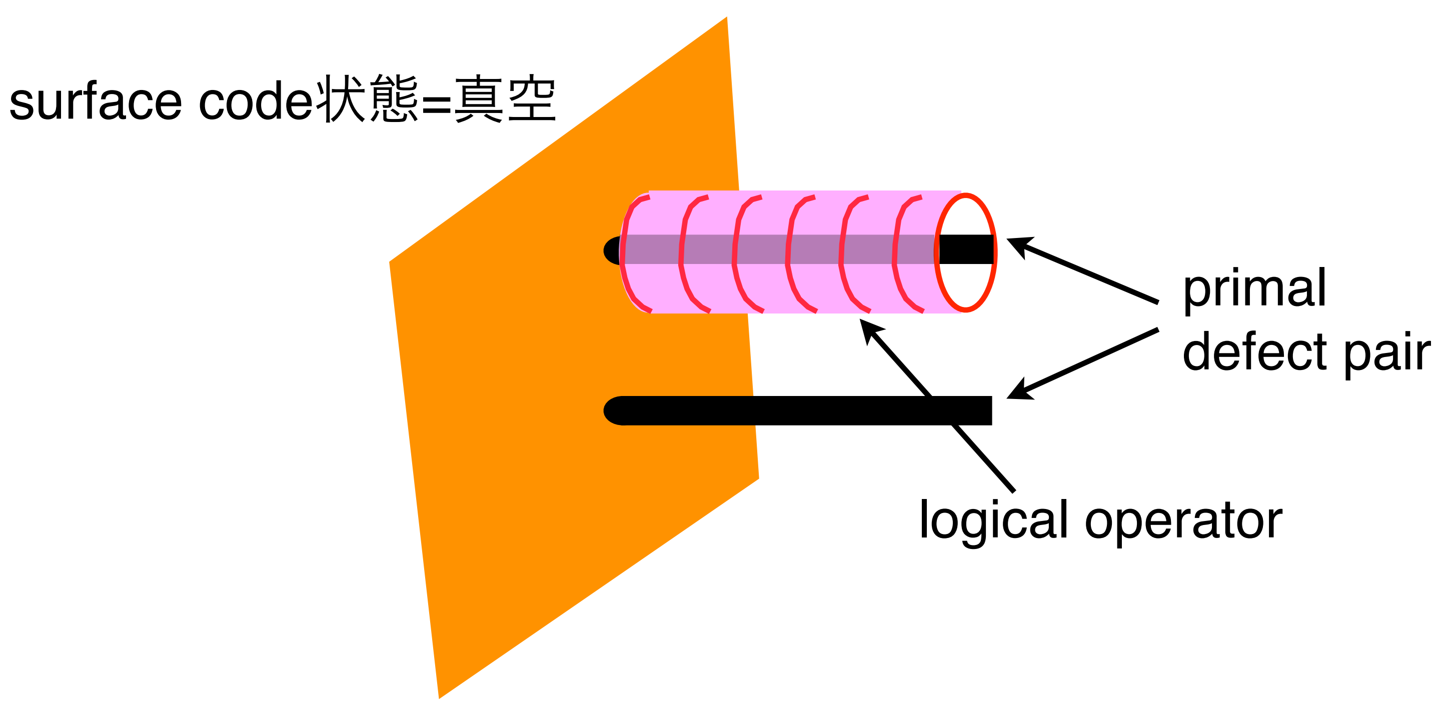
論理Pauli Z演算子 $L_Z^{(1)}$ の固有状態 $|0\rangle_L^p$



Logical qubit: defect pair creation

論理Pauli Z演算子 $L_Z^{(1)}$ の固有状態　$|0\rangle_L^p$

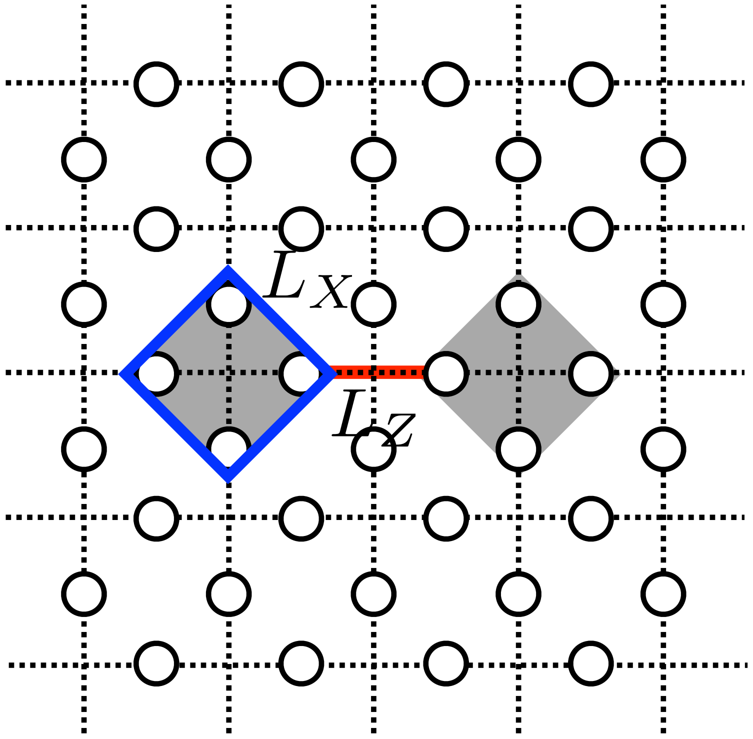




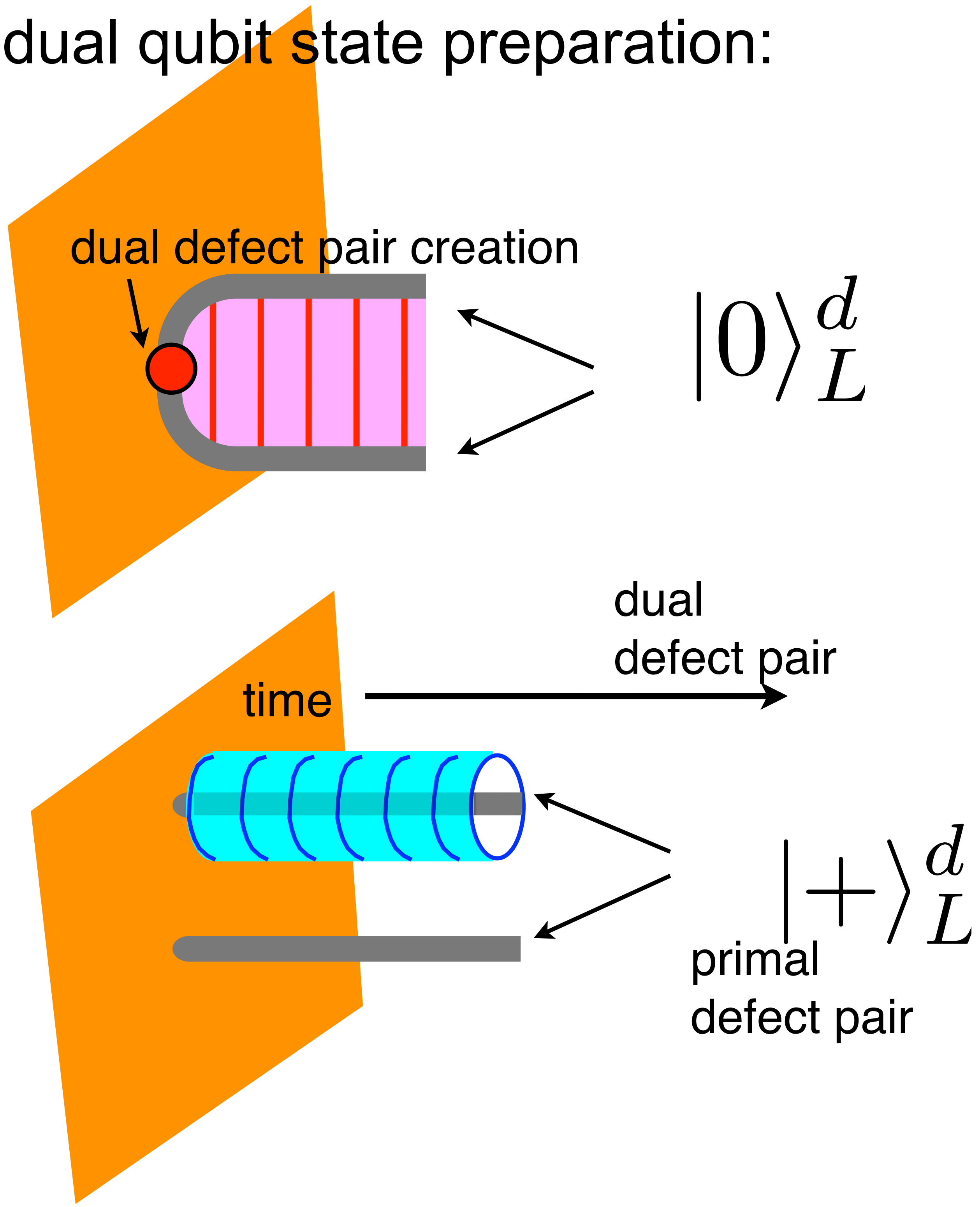
$$|0\rangle_L^p$$

Logical qubit: dual defect pair creation

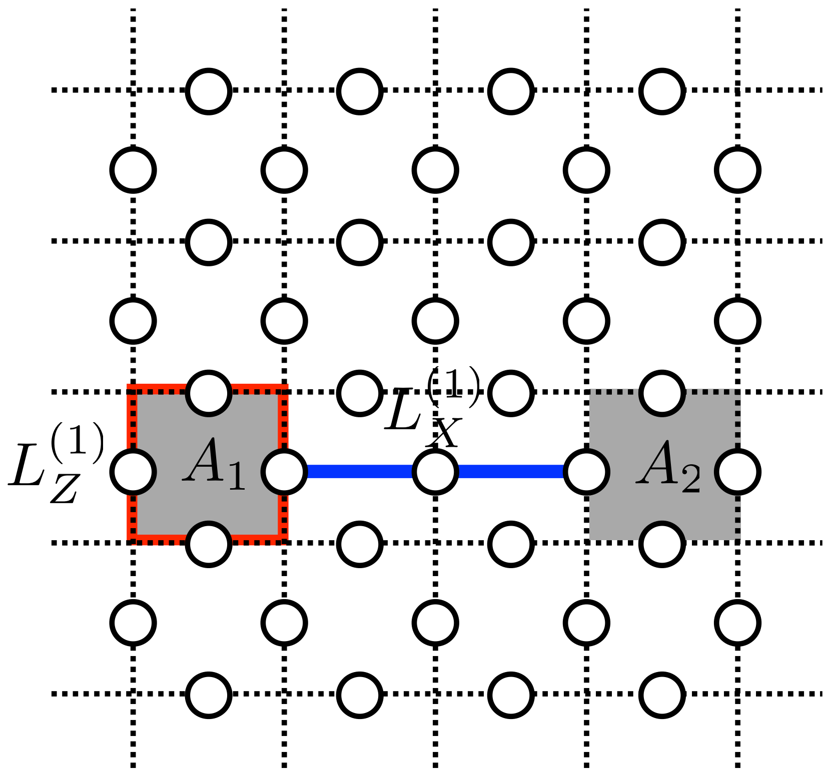
▶dual defect pair:



▶dual qubit state preparation:

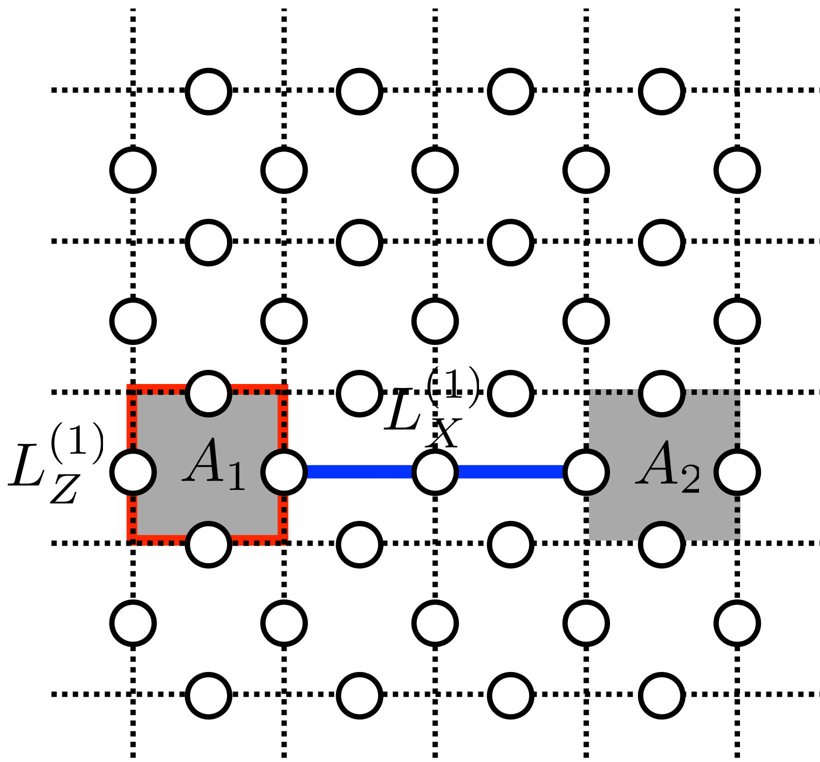


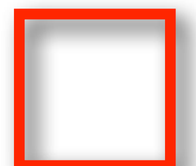
Logical qubitの測定

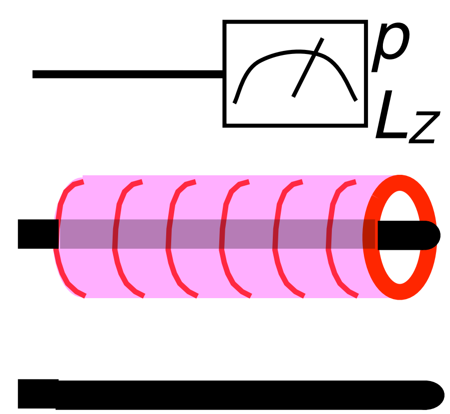


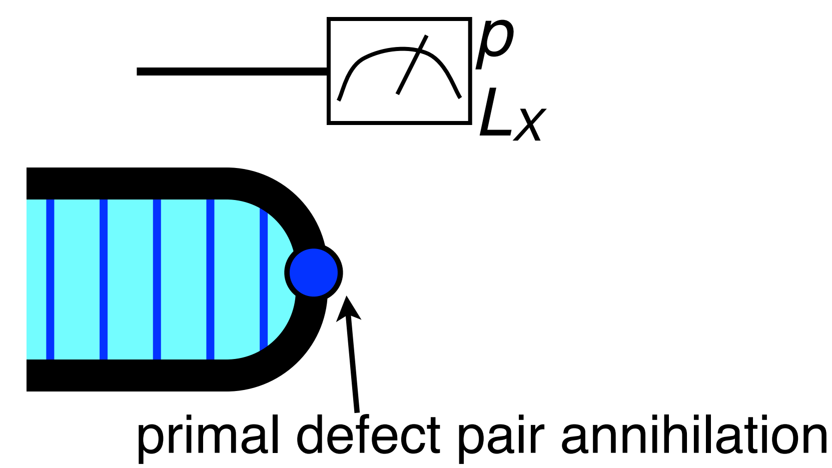
や を測定する.

Logical qubitの測定

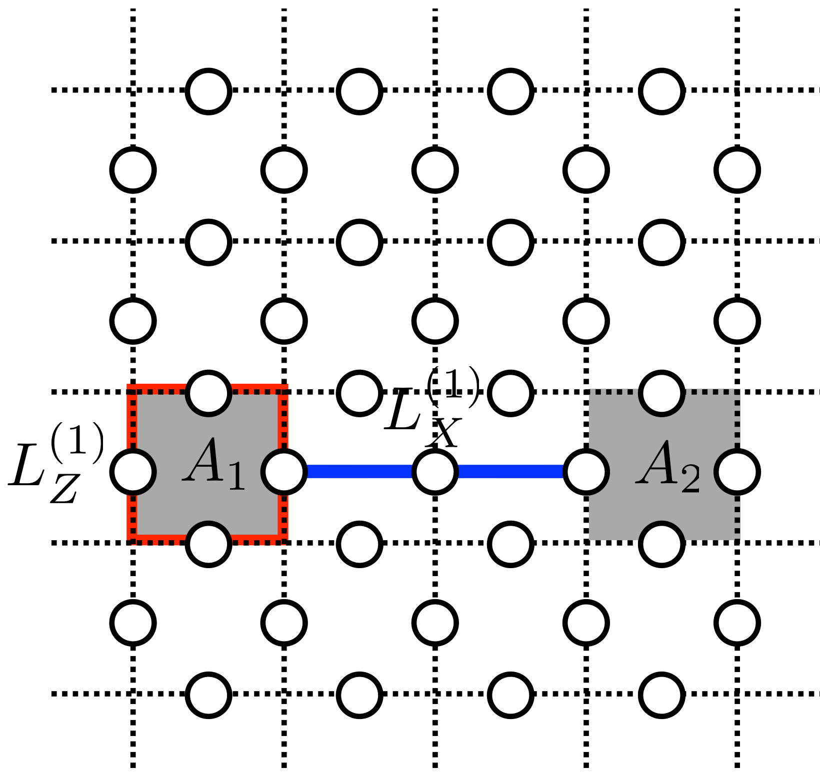


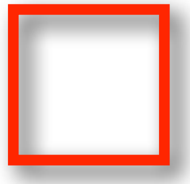
 や を測定する.

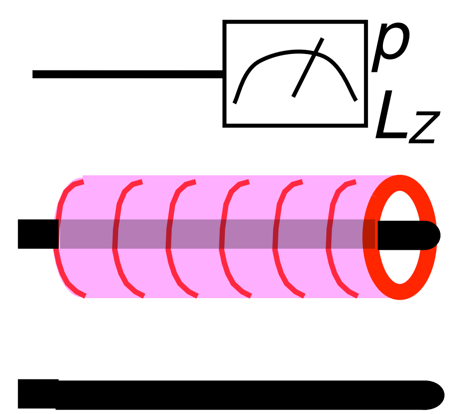


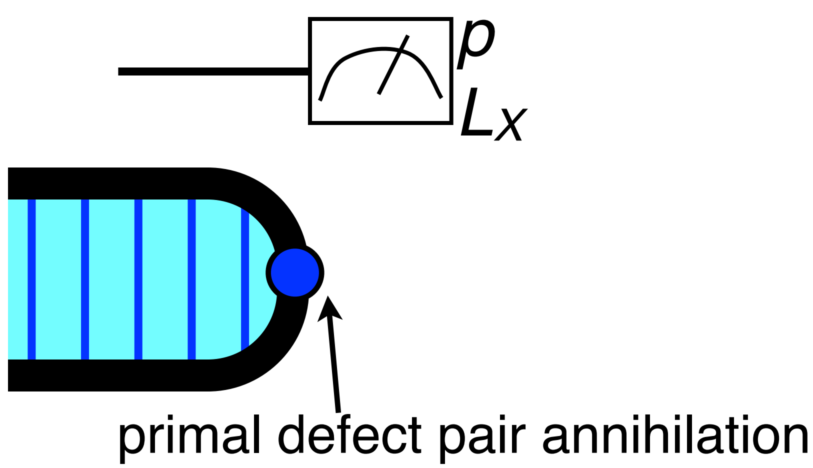


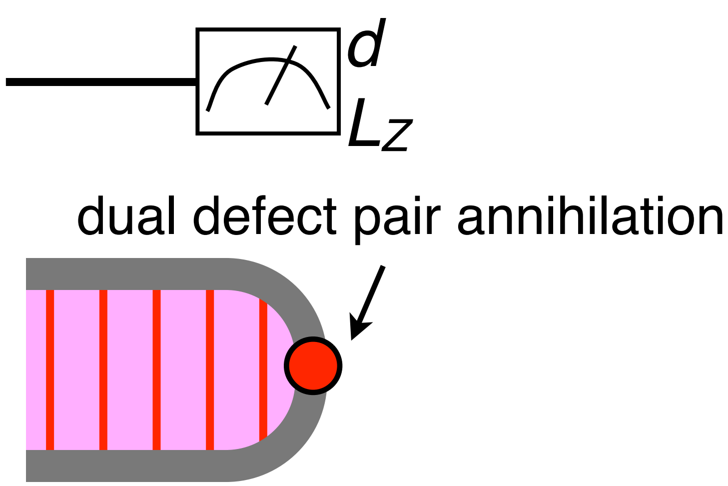
Logical qubitの測定

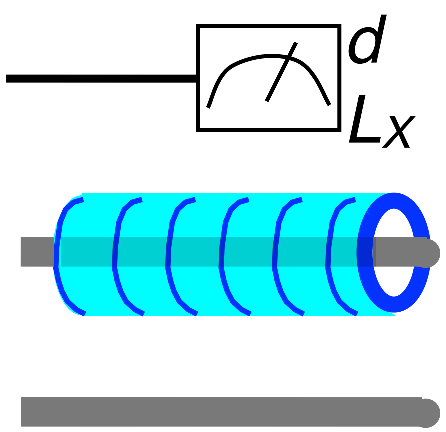


 や  を測定する.







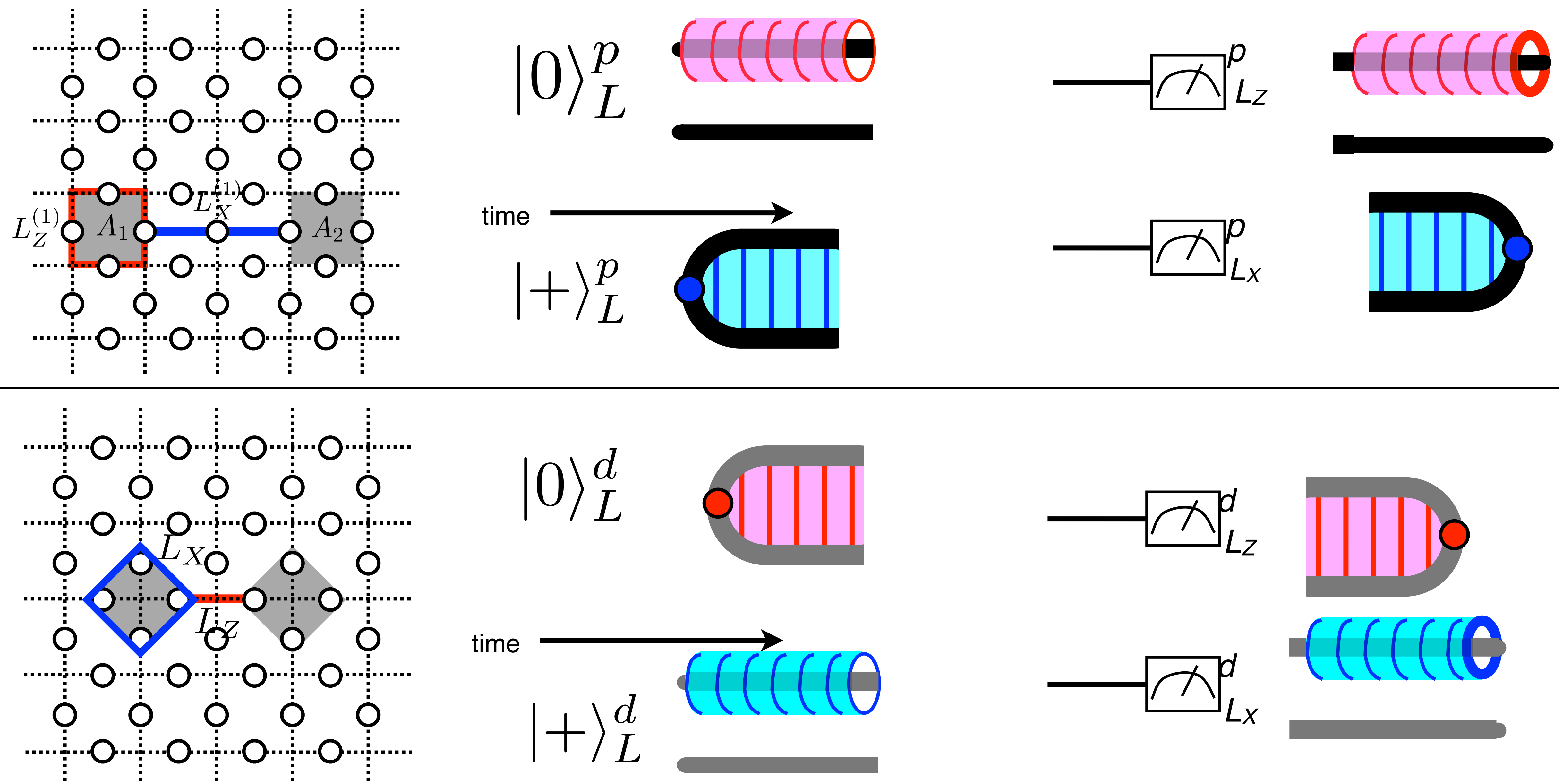


整理

演算子の固有状態として状態を定義→スタビライザー状態

スタビライザーを取り除くことによって縮退が増える（人工的な穴を開ける）.

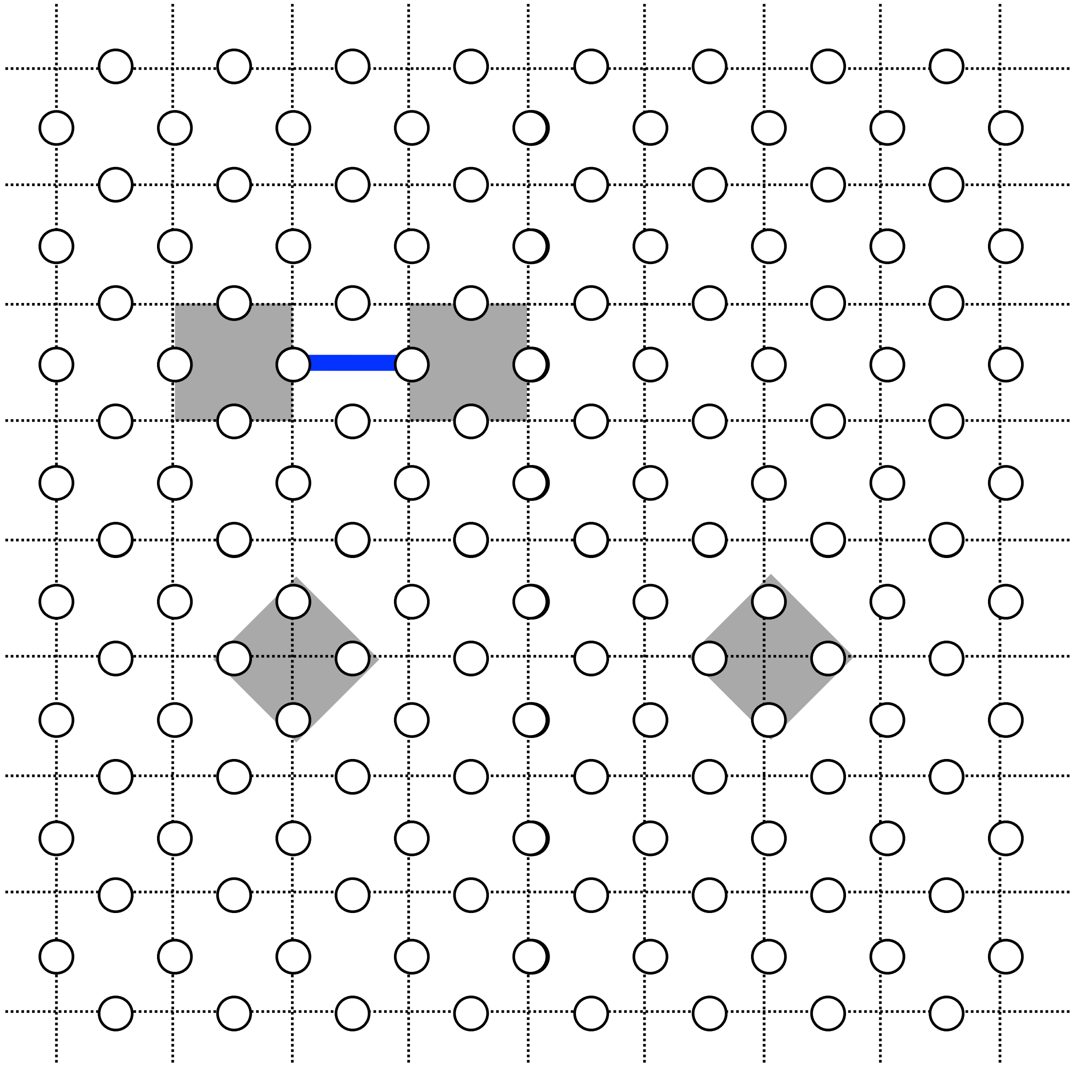
primalとdualの２種類のdefect = 論理qubitがいる.



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

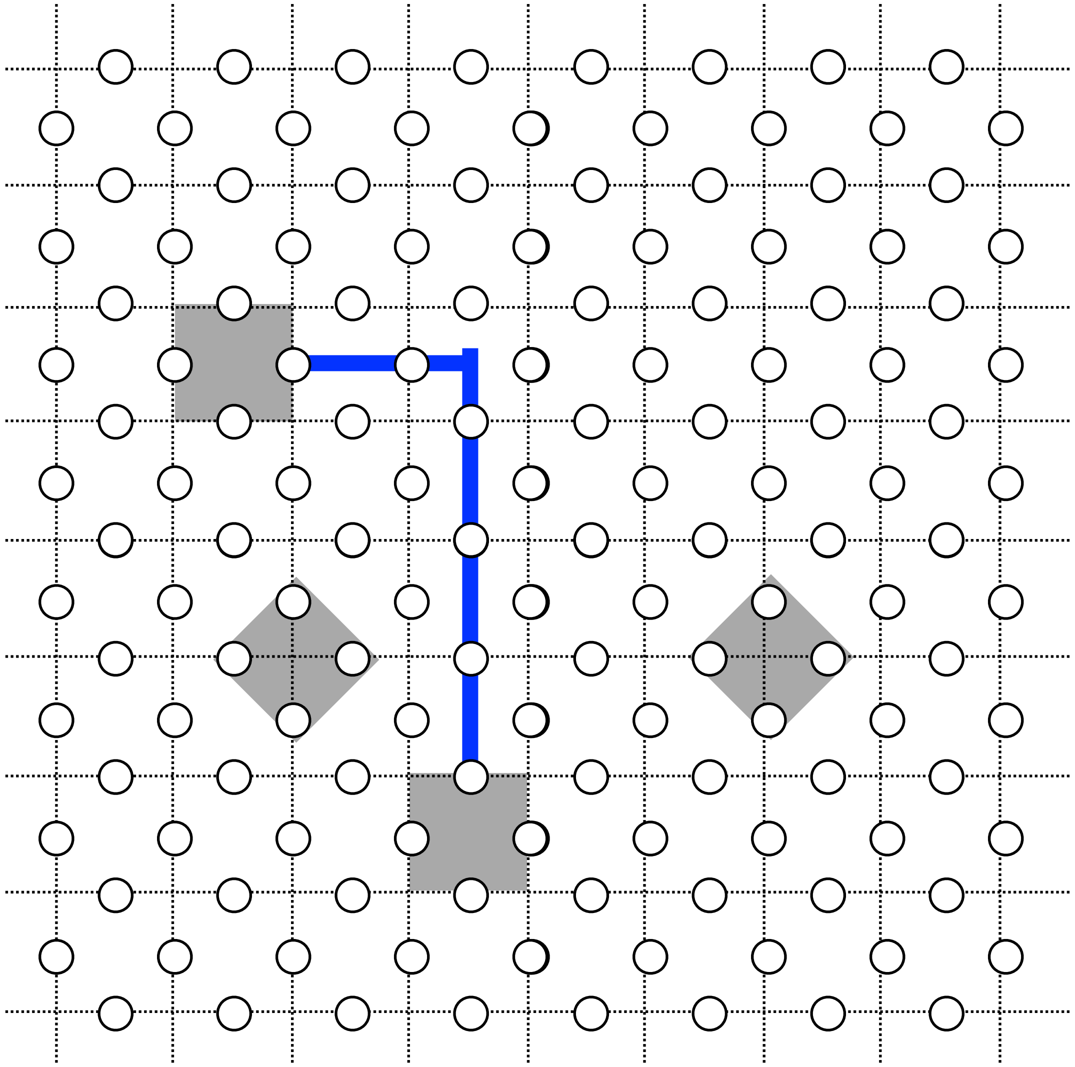
primal qubitのPauli X演算子はどのように発展するか？



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

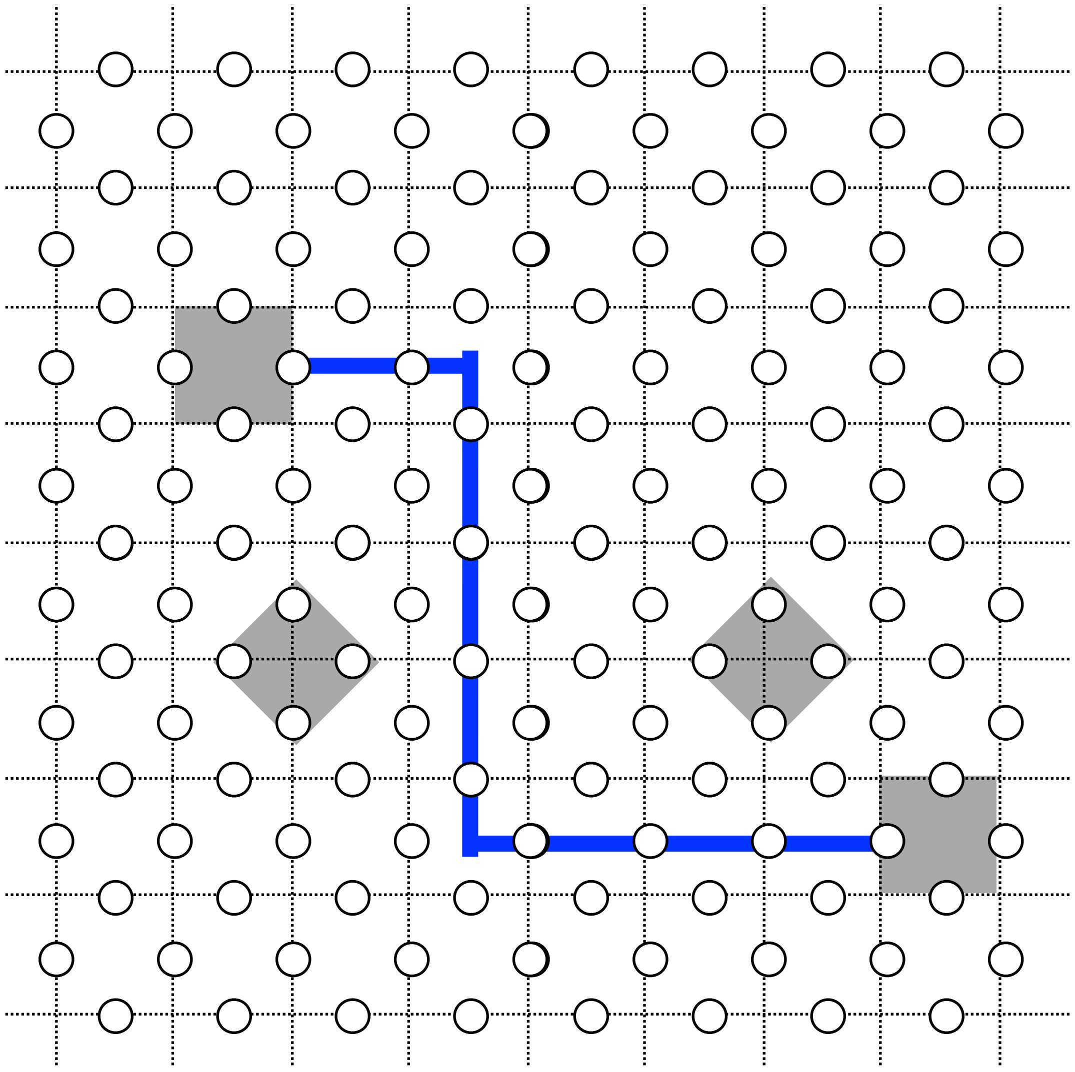
primal qubitのPauli X演算子はどのように発展するか？



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

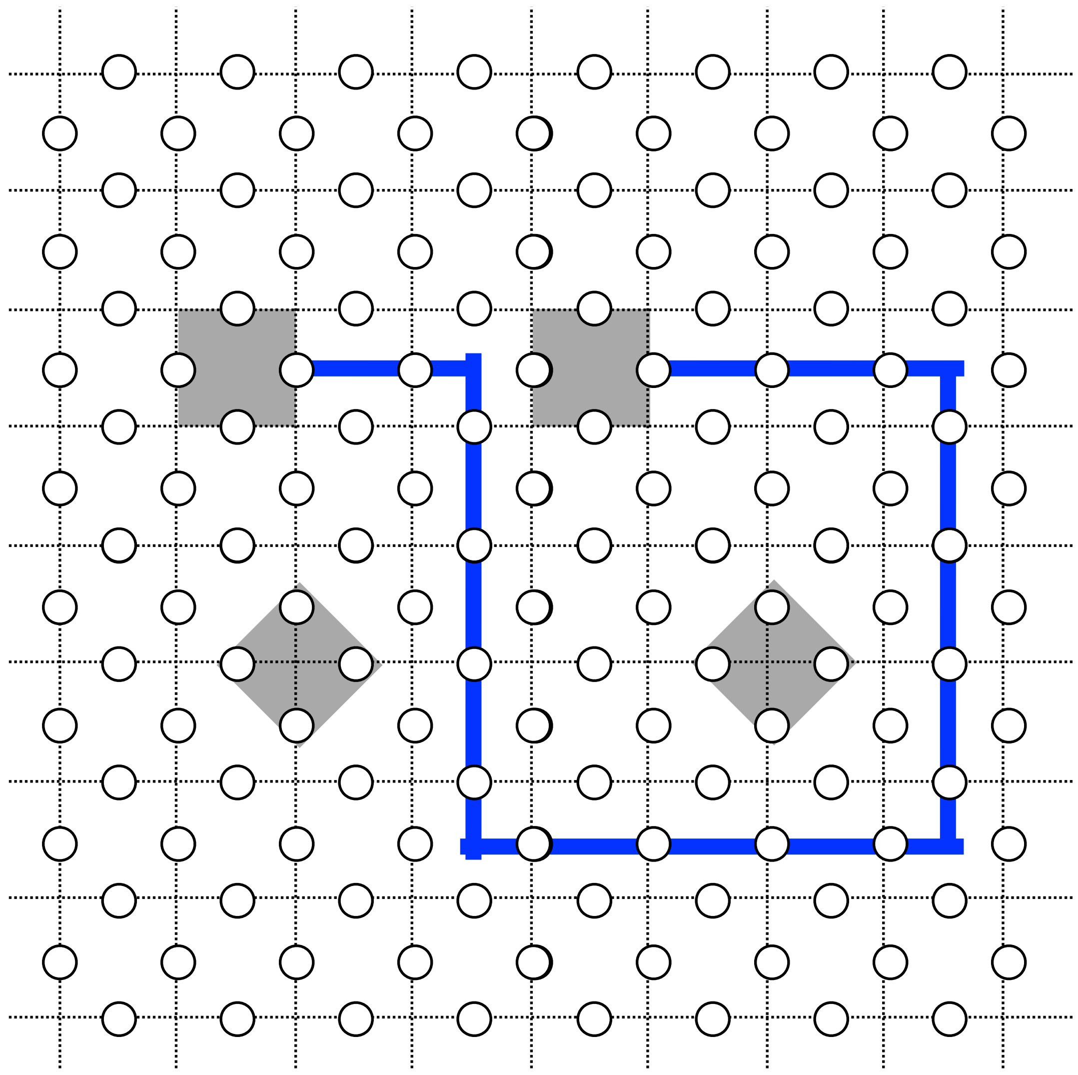
primal qubitのPauli X演算子はどのように発展するか？



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

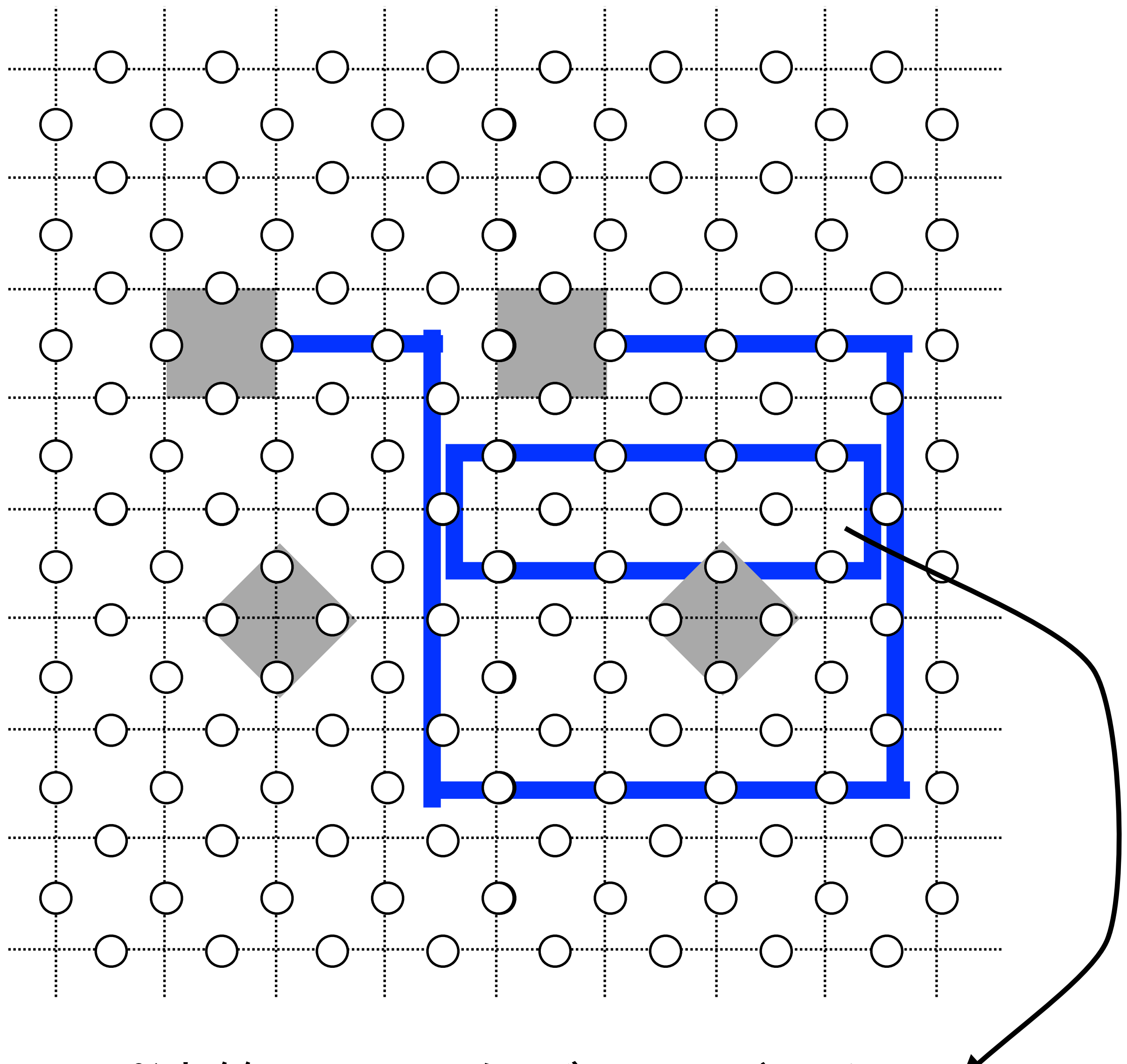
primal qubitのPauli X演算子はどのように発展するか？



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

primal qubitのPauli X演算子はどのように発展するか？



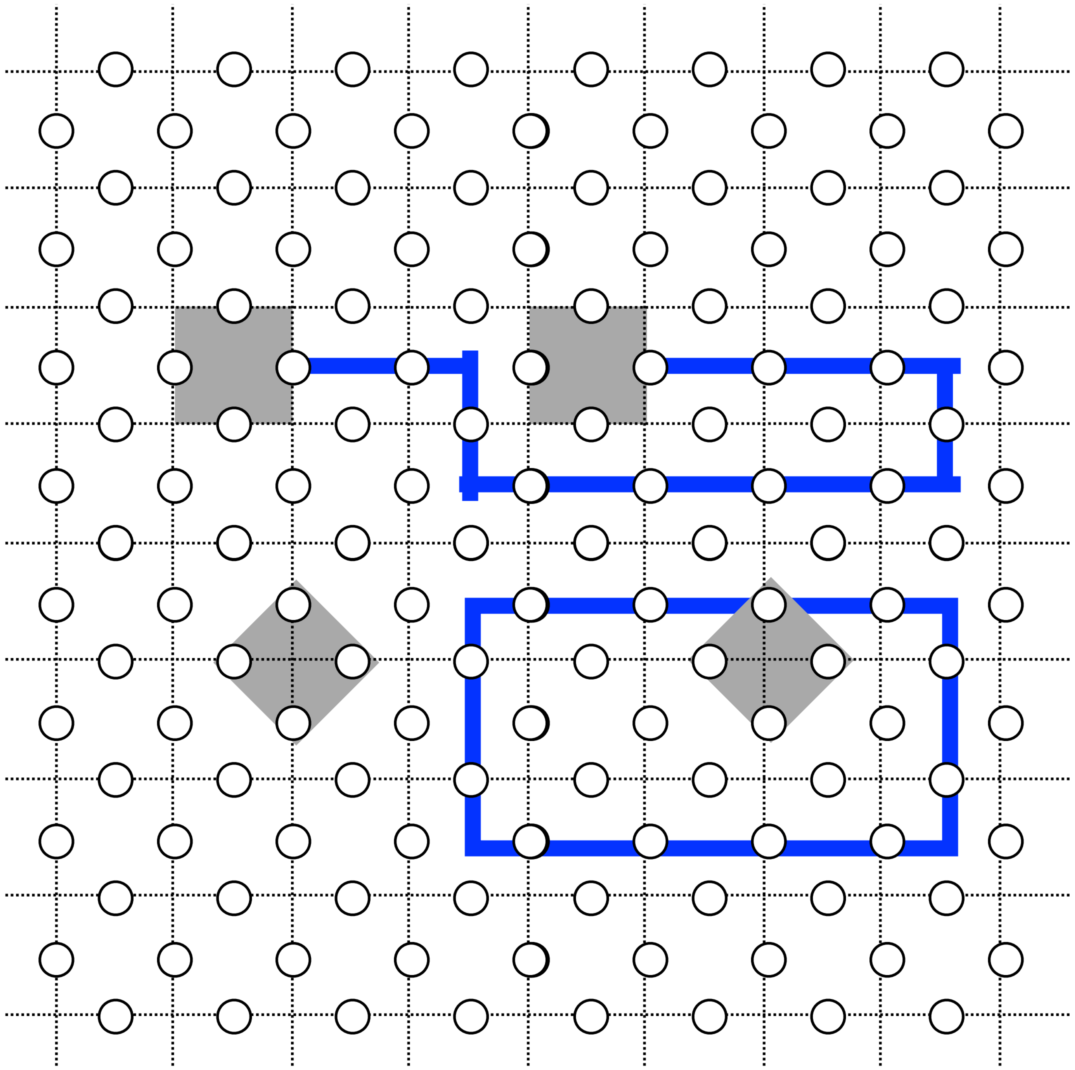
自明なループ演算子はスタビライザーなので,

掛けても何ら状態は変化しない.

CNOT by braiding

primal defect を dual defectの周りにBraidすると,

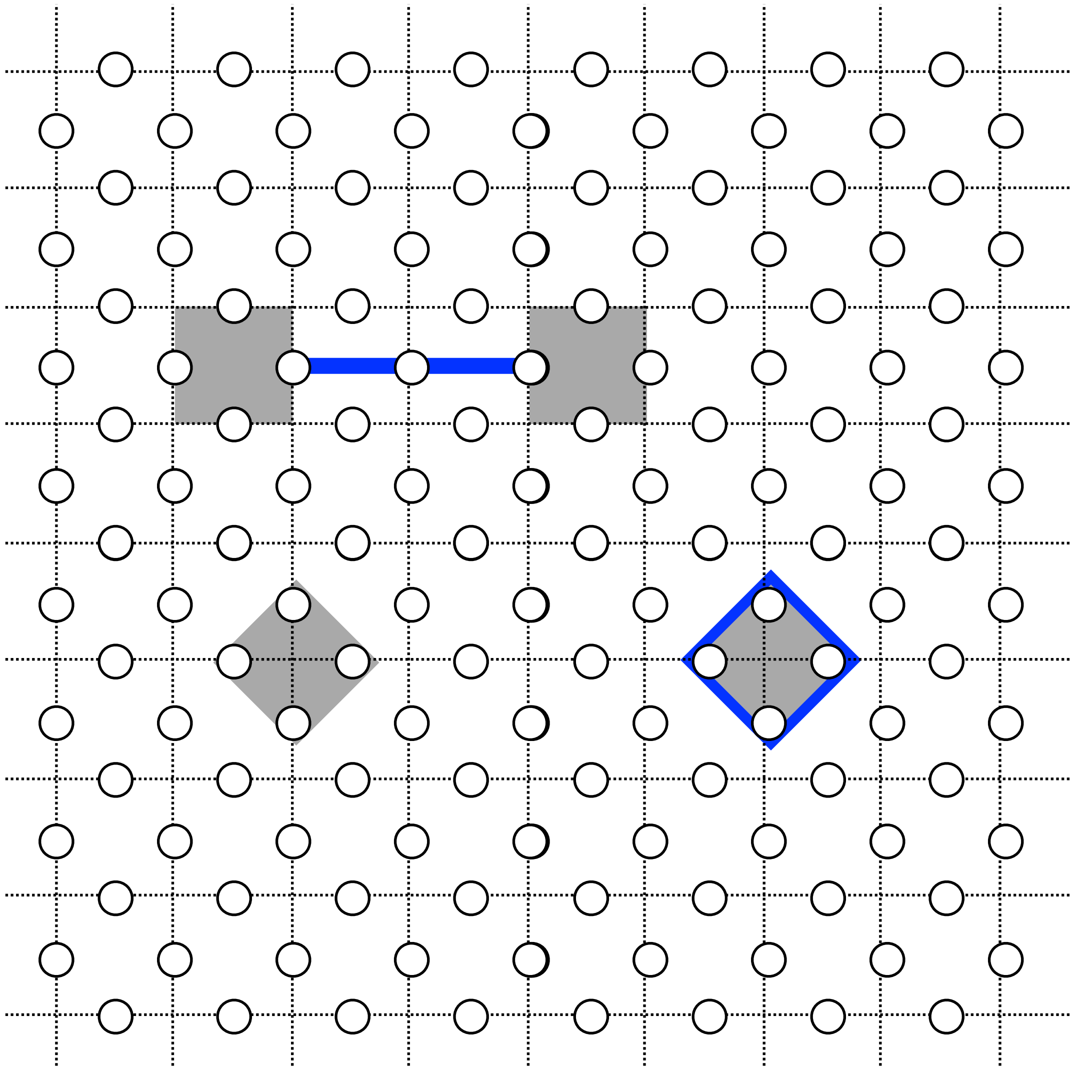
primal qubitのPauli X演算子はどのように発展するか？



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

primal qubitのPauli X演算子はどのように発展するか？



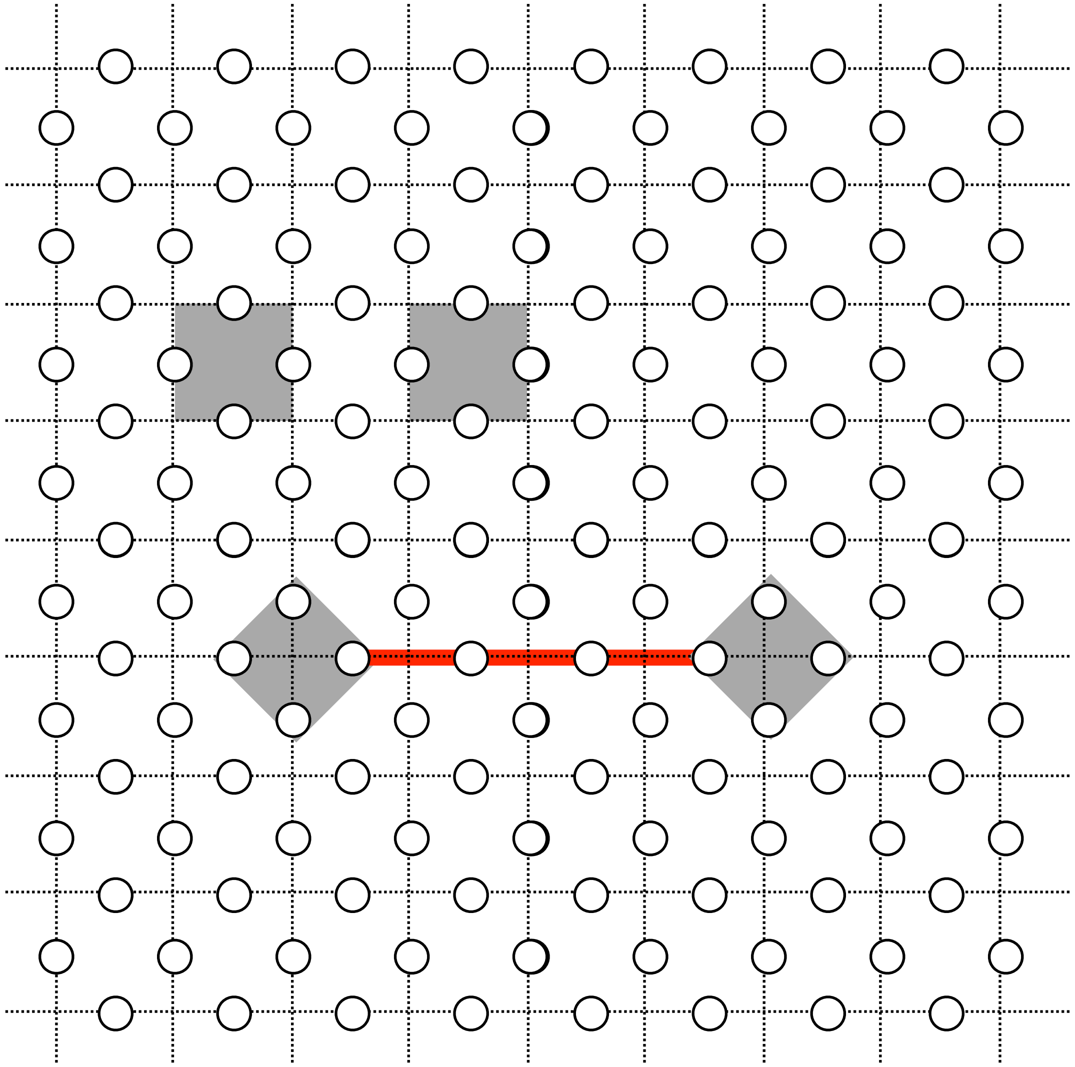
収縮させても,
トポロジーが同じなら同じ演算子.

$L_X^p \otimes I^d$ が $L_X^p \otimes L_X^d$ に変換された！

CNOT by braiding

primal defect を dual defectの周りにBraidすると,

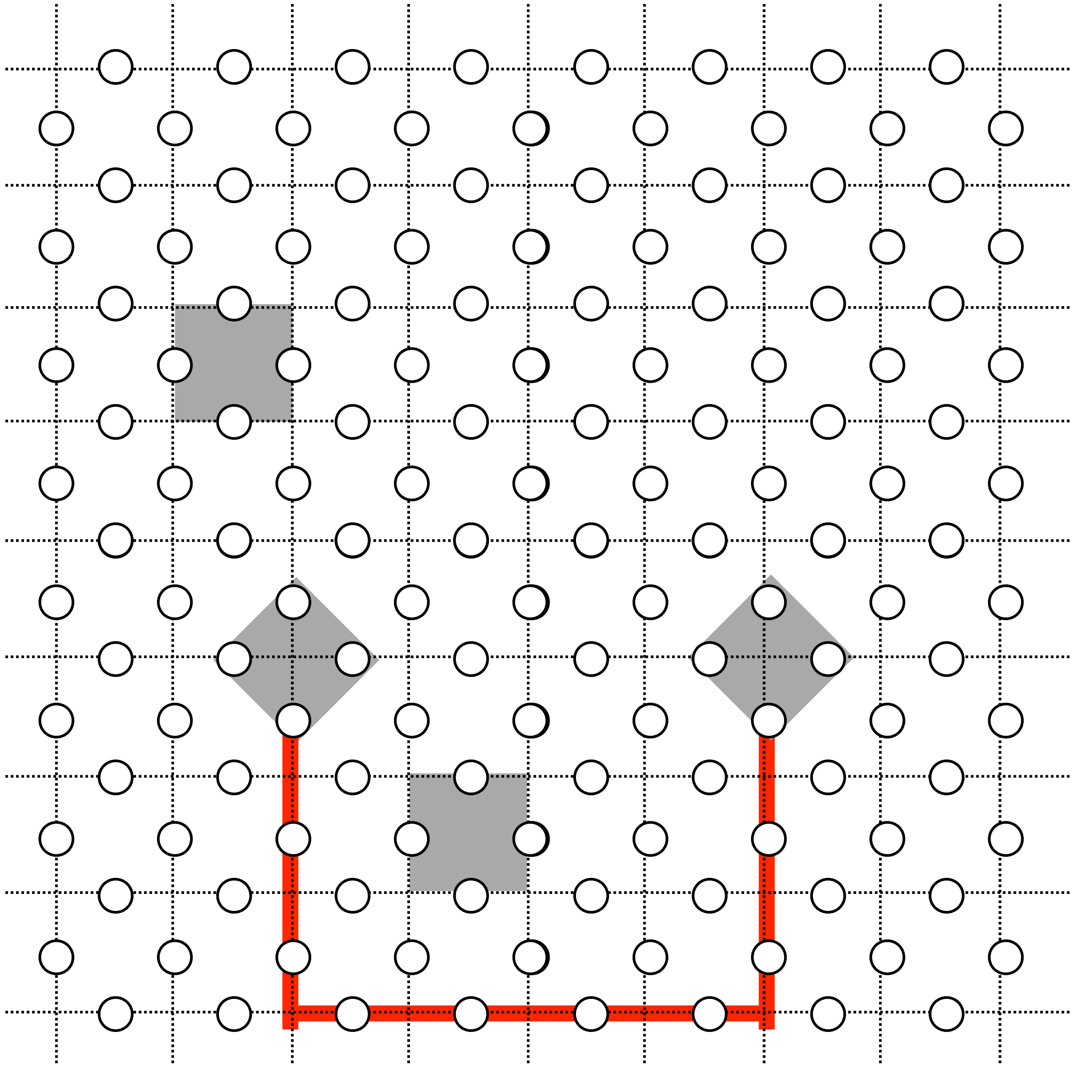
dual qubitのPauli Z演算子はどのように発展するか？



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

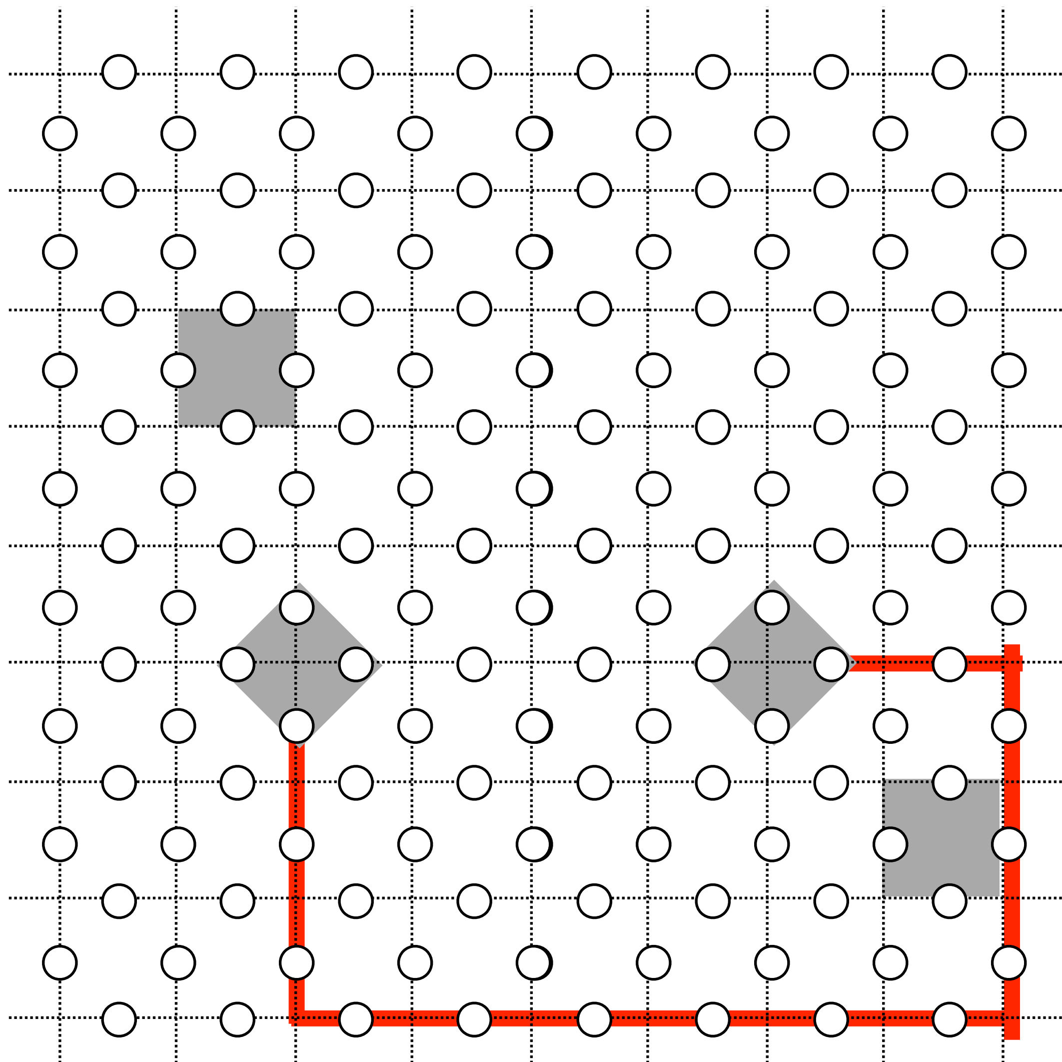
dual qubitのPauli Z演算子はどのように発展するか？



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

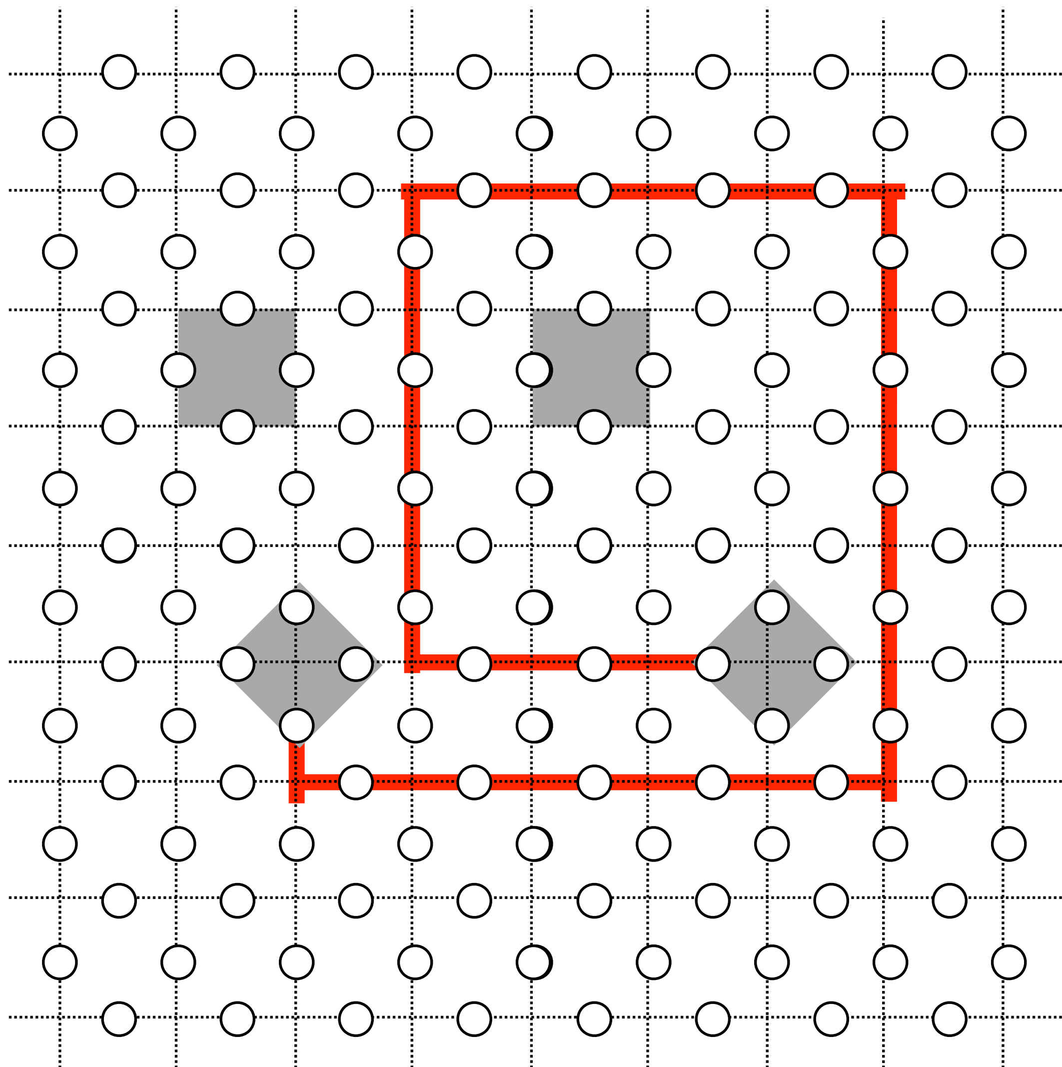
dual qubitのPauli Z演算子はどのように発展するか？



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

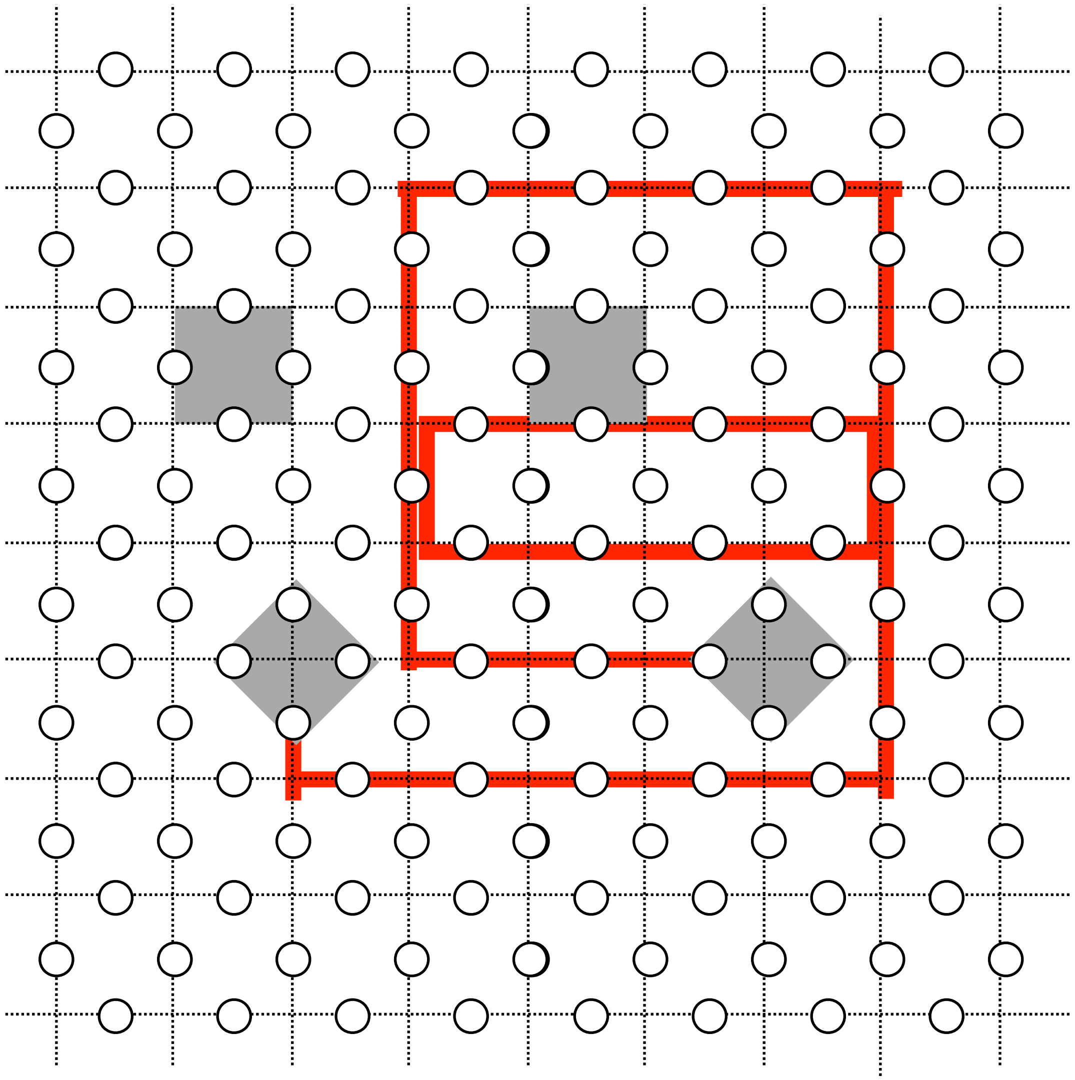
dual qubitのPauli Z演算子はどのように発展するか？



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

dual qubitのPauli Z演算子はどのように発展するか？



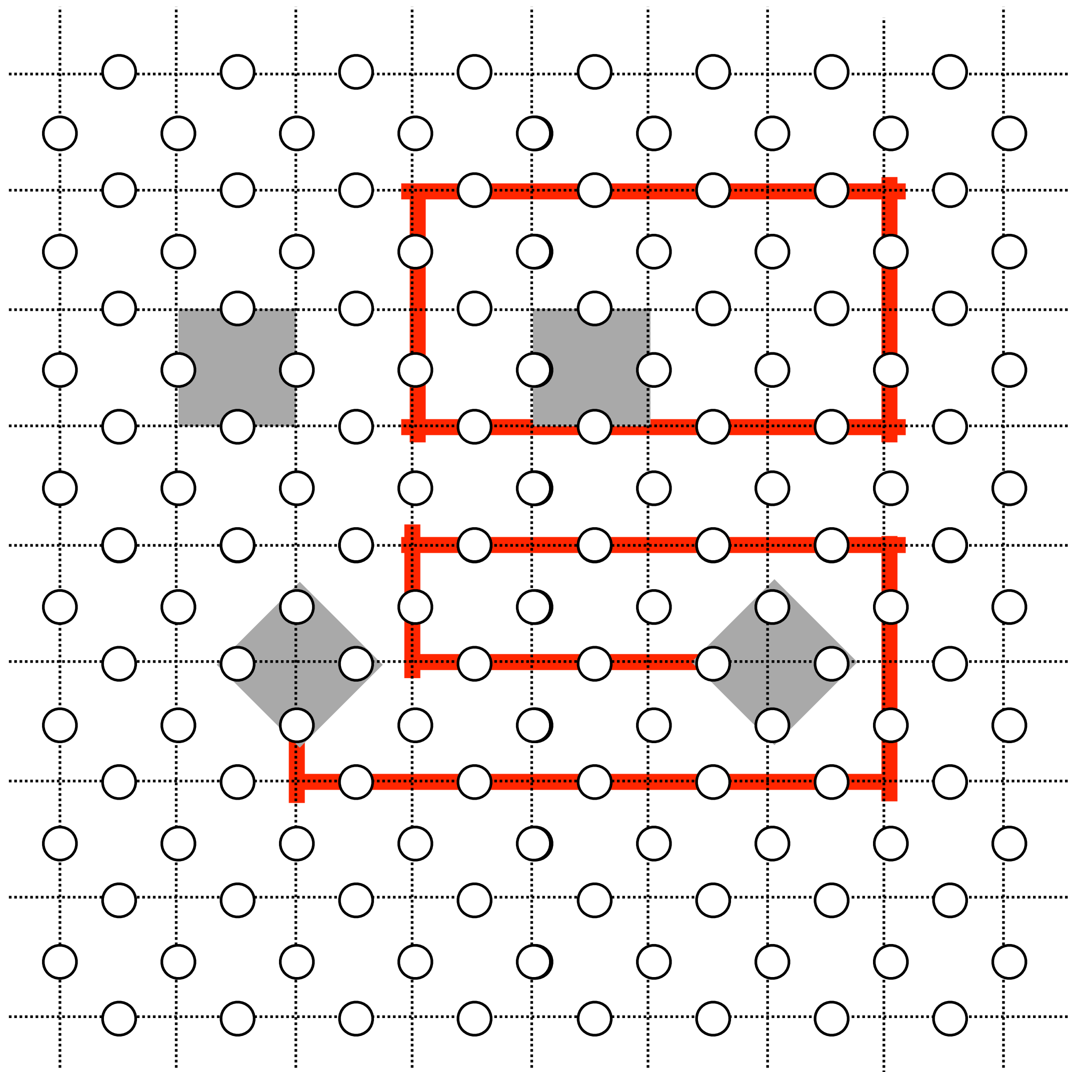
自明なループ演算子はスタビライザーなので,

掛けても何ら状態は変化しない.

CNOT by braiding

primal defect を dual defectの周りにBraidすると,

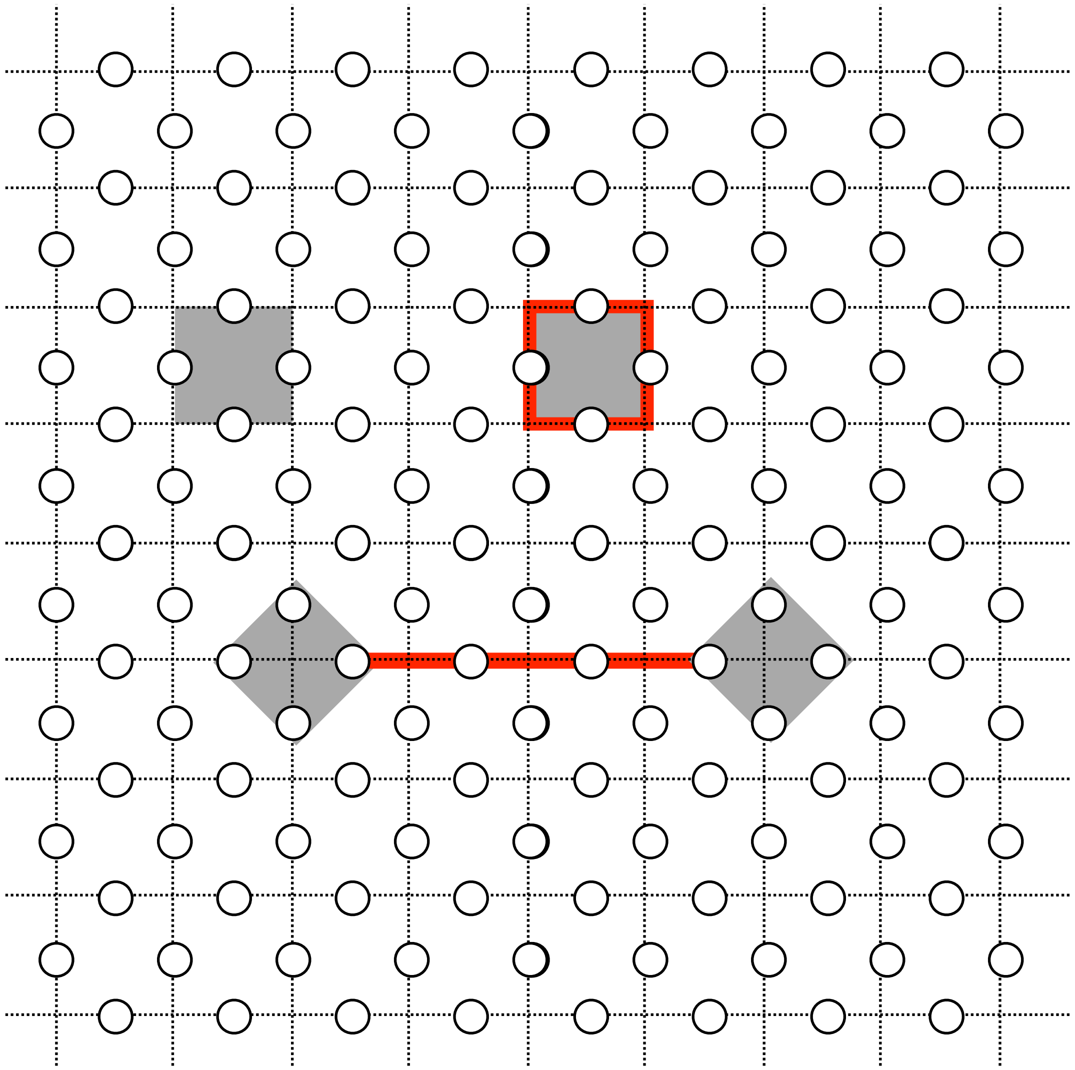
dual qubitのPauli Z演算子はどのように発展するか？



CNOT by braiding

primal defect を dual defectの周りにBraidすると,

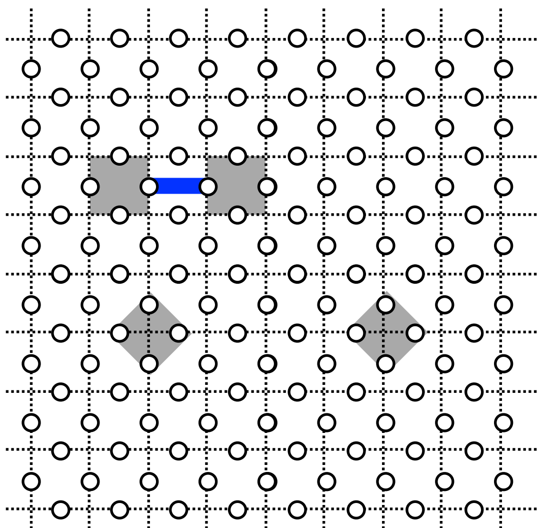
dual qubitのPauli Z演算子はどのように発展するか？



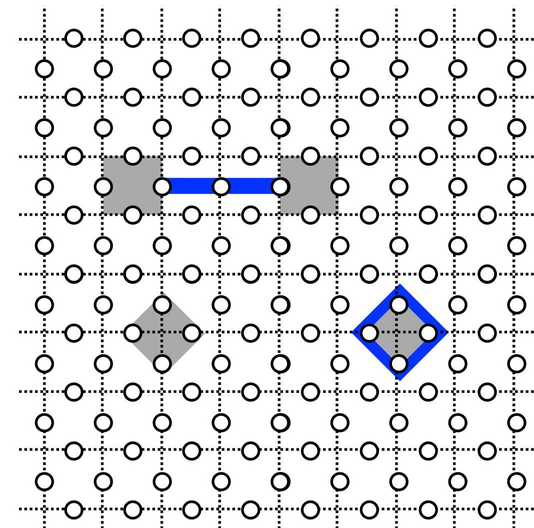
$I^p \otimes L_Z^d$ が $L_Z^p \otimes L_Z^d$ に変換された！

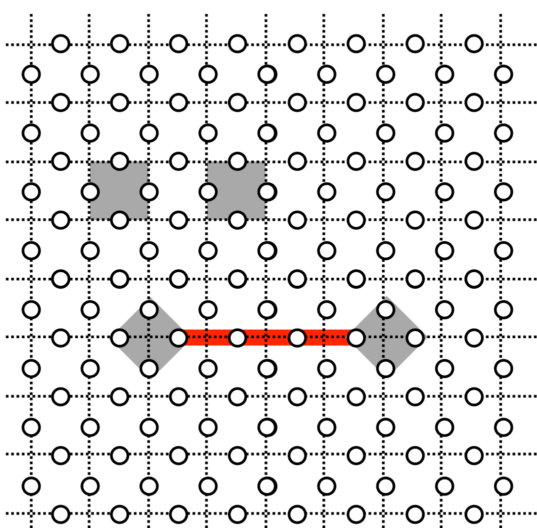
CNOT by braiding

まとめると

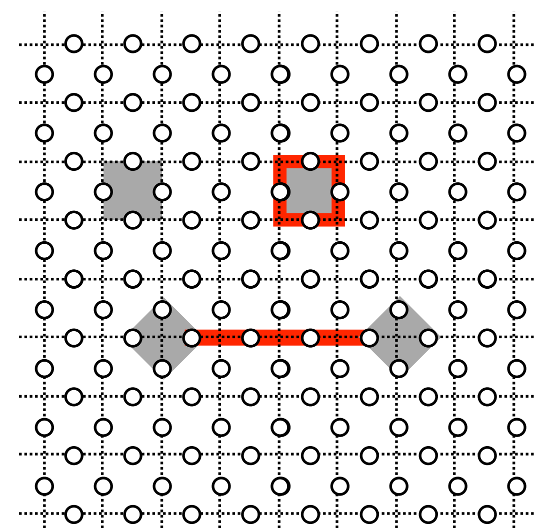


$$L_X^p \otimes I^d \xrightarrow{\text{Braiding}} L_X^p \otimes L_X^d$$



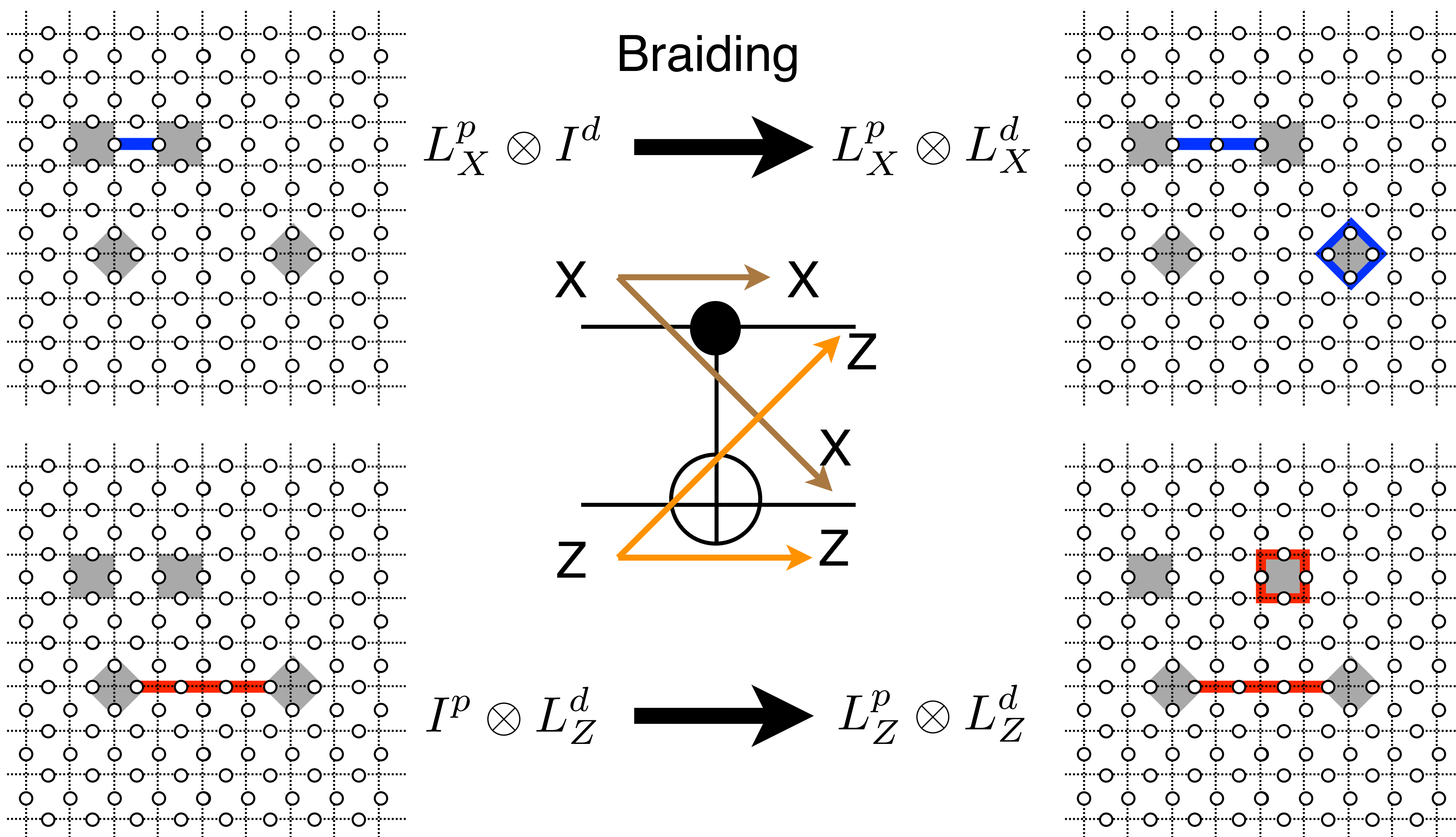


$$I^p \otimes L_Z^d \longrightarrow L_Z^p \otimes L_Z^d$$



CNOT by braiding

まとめると

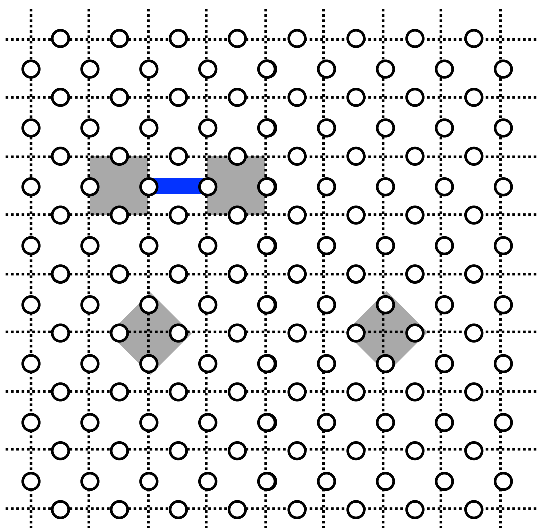


つまり，BraidingはprimalからdualへのフCNOT演算

←符号空間を変形させることによって、論理演算ができる.

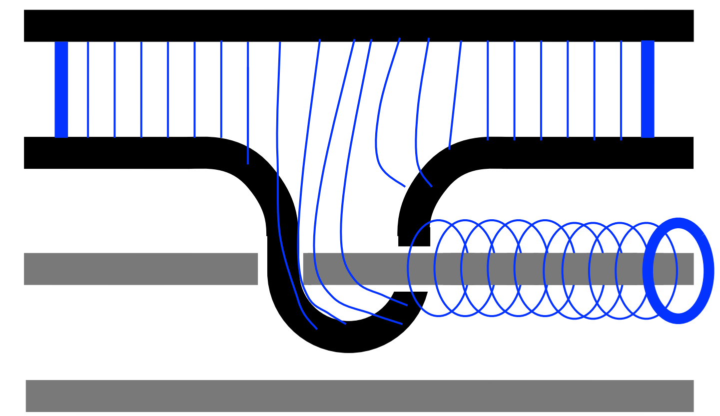
CNOT by braiding

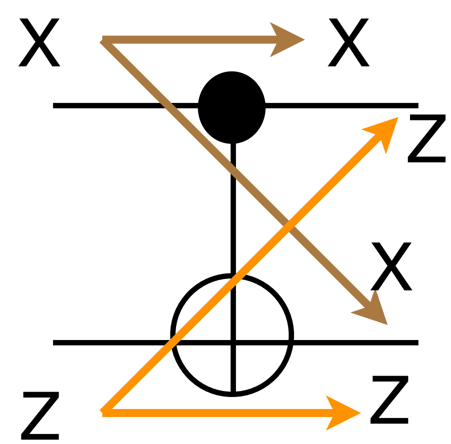
まとめると

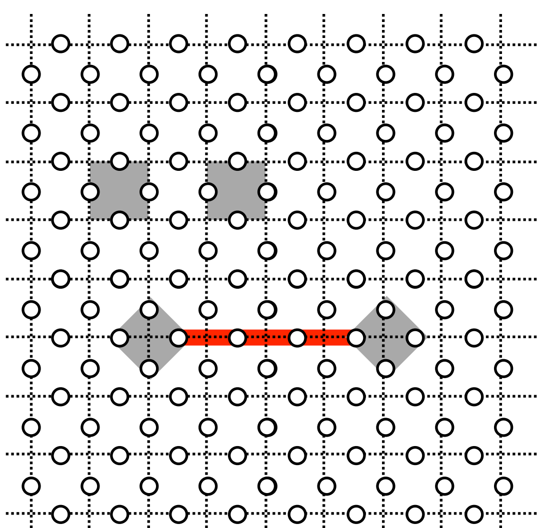


Braiding

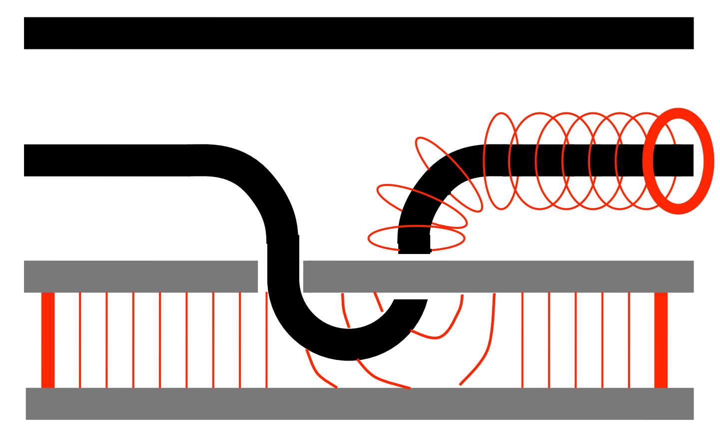
$$L_X^p \otimes I^d \longrightarrow L_X^p \otimes L_X^d$$







$$I^p \otimes L_Z^d \longrightarrow L_Z^p \otimes L_Z^d$$

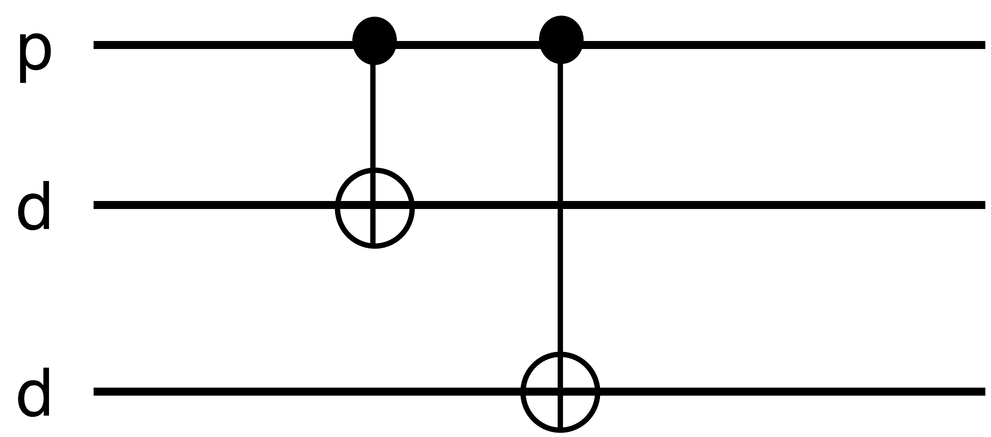


つまり，Braidingはprimalからdualへの CNOT演算

←符号空間を変形させることによって、論理演算ができる.

CNOT by braiding

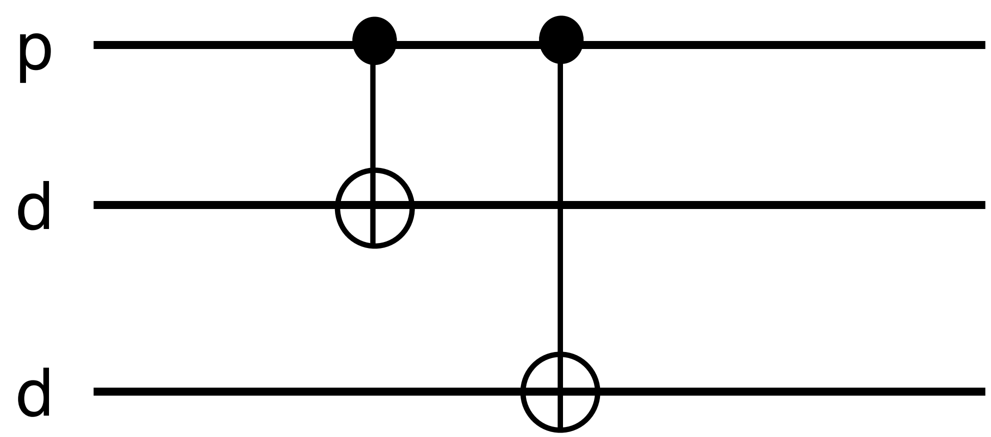
CNOTは必ず primal （control）から dual (target)へ



全部可換！→defect は Abelian (可換な)エニオン

CNOT by braiding

CNOTは必ず primal （control）から dual (target)へ

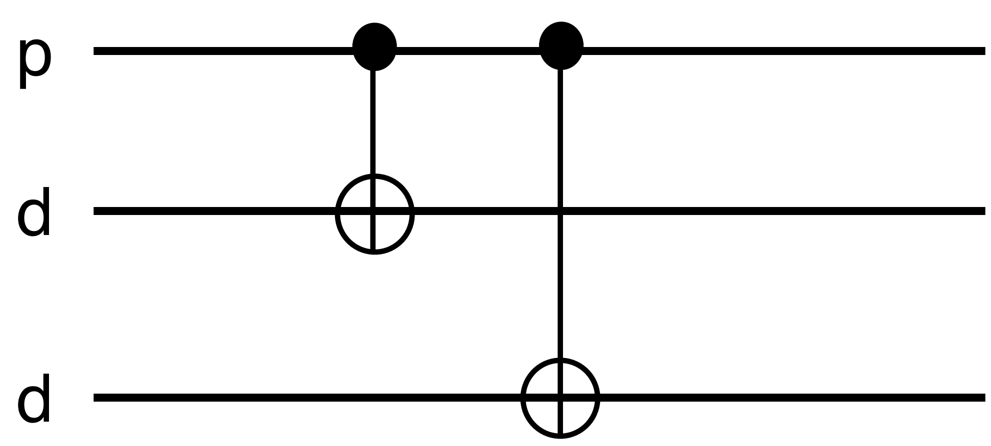


全部可換！→defect は Abelian (可換な)エニオン

だったら量子計算につかえないのでは？

CNOT by braiding

CNOTは必ず primal （control）から dual (target)へ



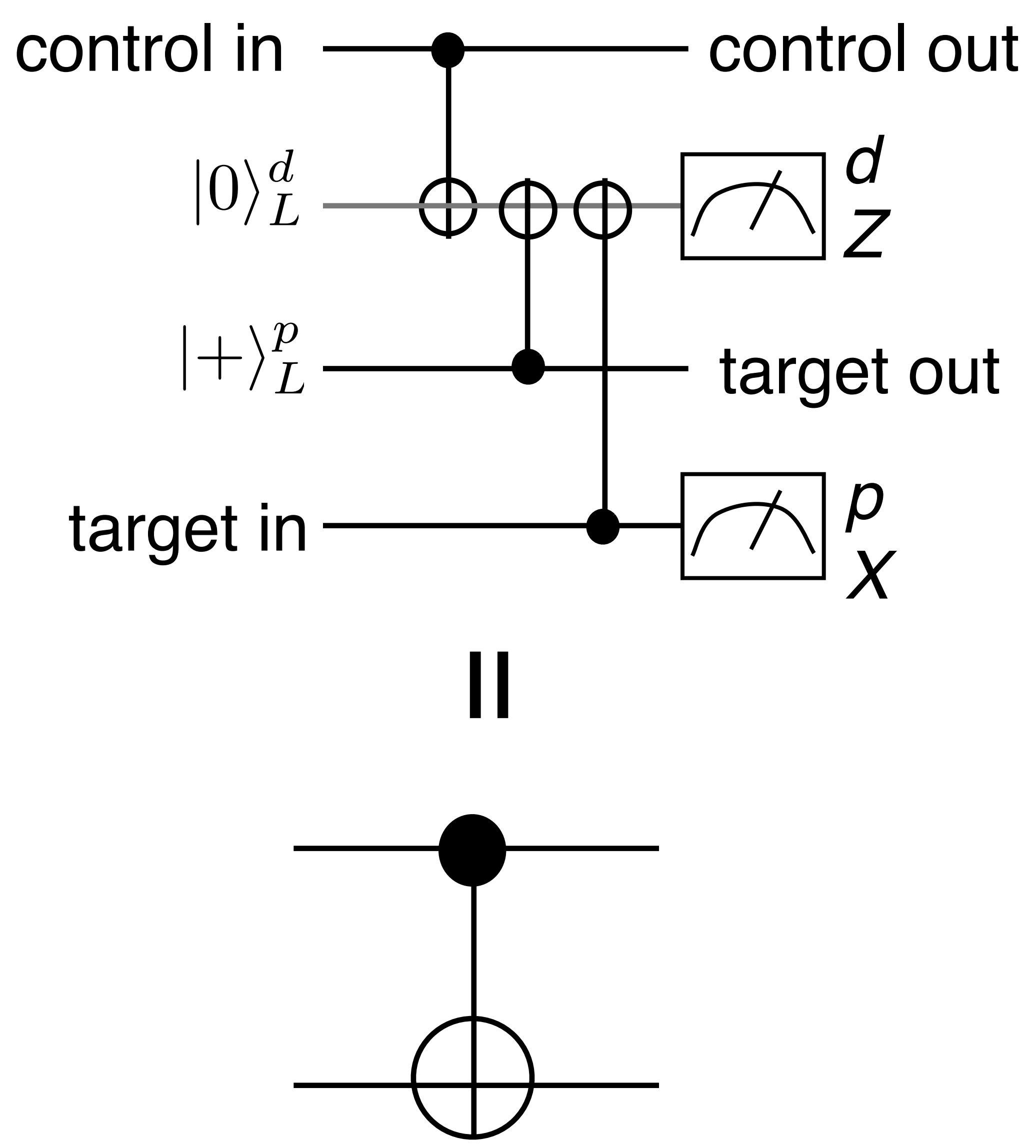
全部可換！→defect は Abelian (可換な)エニオン

だったら量子計算につかえないのでは？

Raussendorf 曰く，トポロジーを時々刻々と変化させれば良いのだよ．

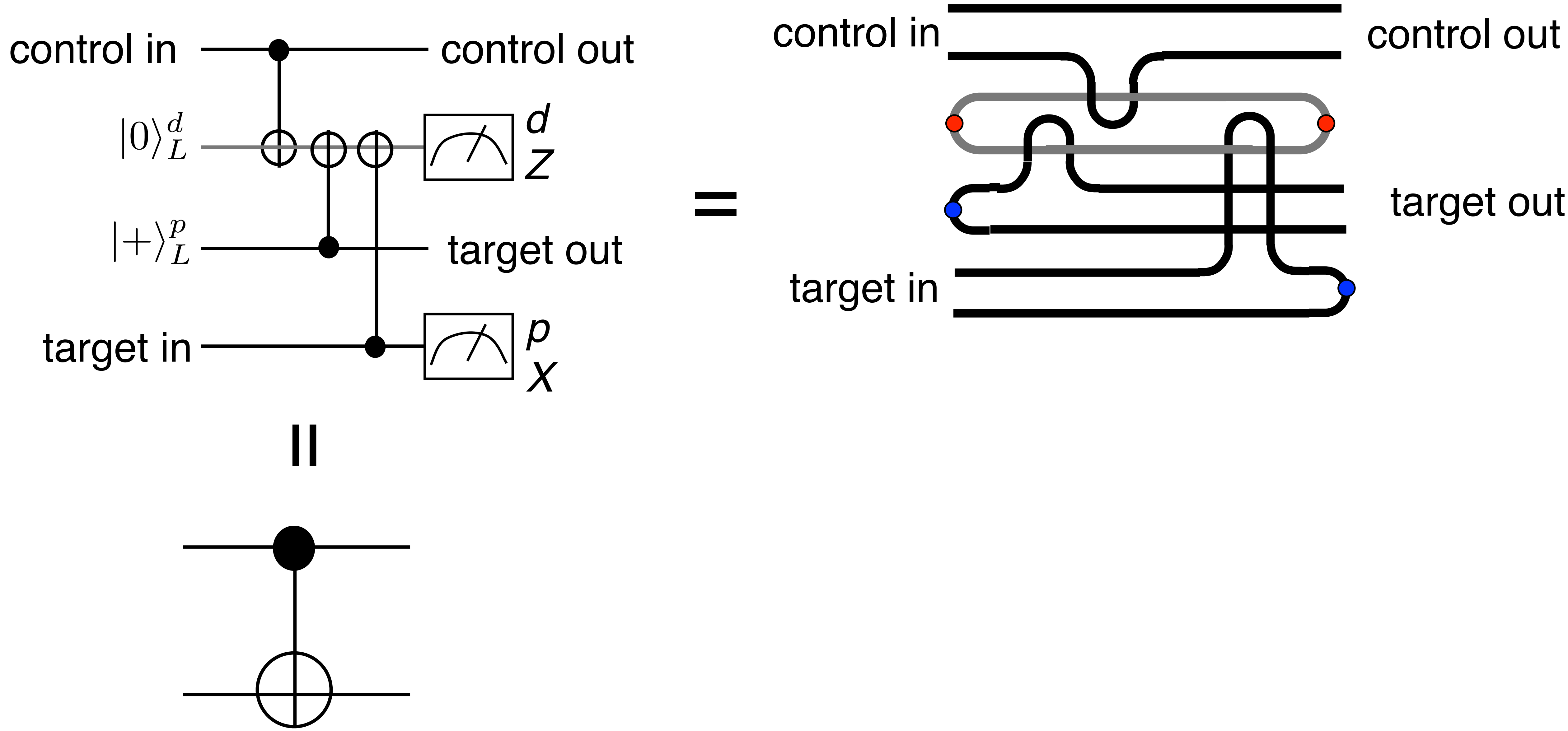
CNOT by braiding

Abelian CNOTでも測定を使えば measurement-based で primal から primal への CNOTが可能.



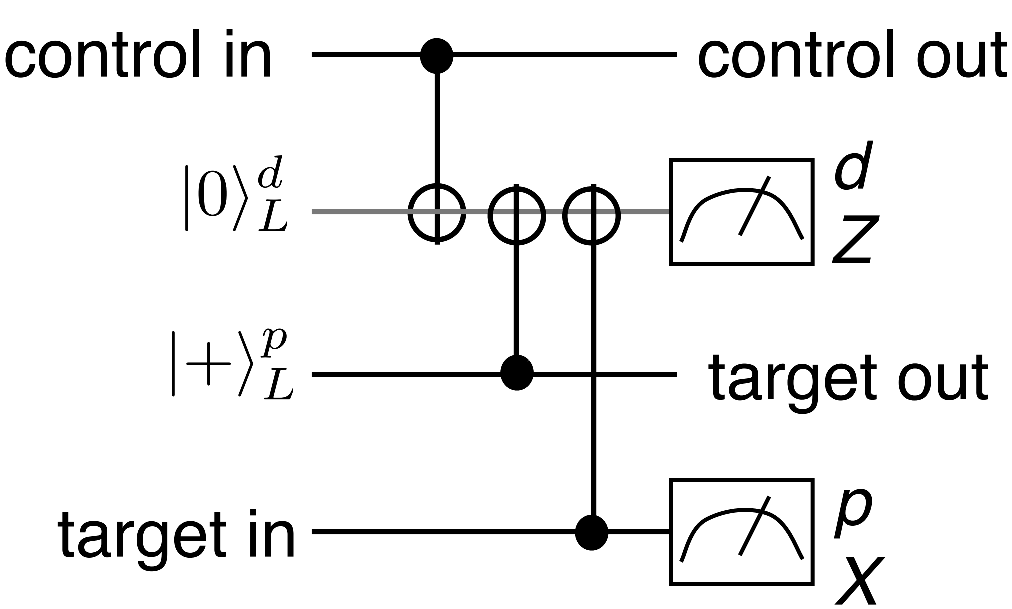
CNOT by braiding

Abelian CNOTでも測定を使えば measurement-based で primal から primal への CNOTが可能.

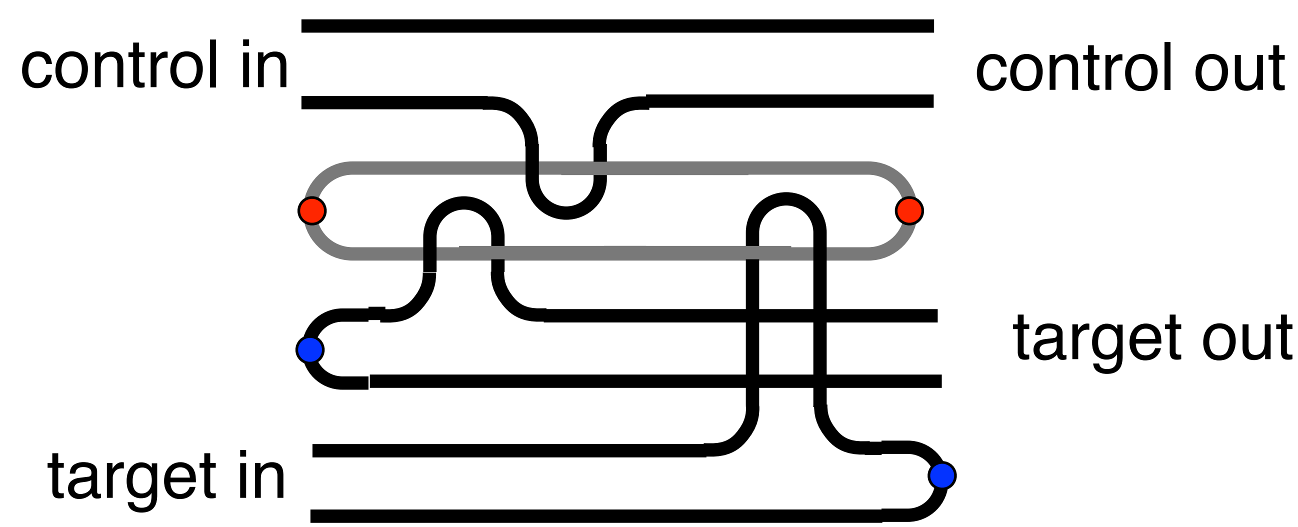


CNOT by braiding

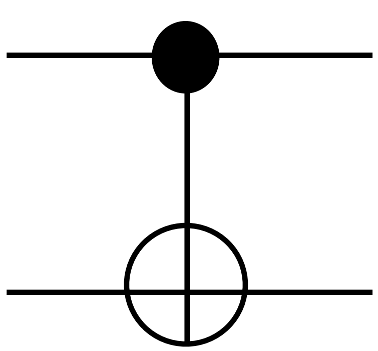
Abelian CNOTでも測定を使えば measurement-based で primal から primal への CNOTが可能.



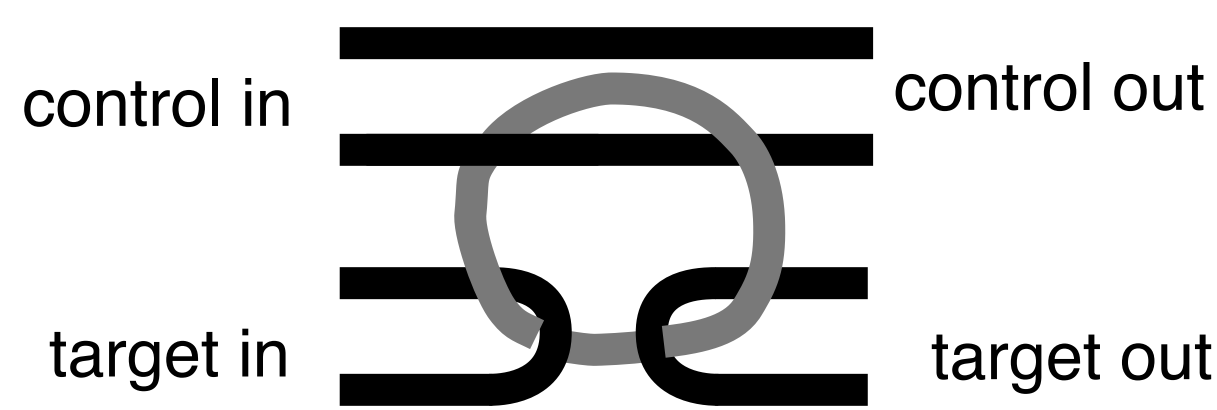
=



||

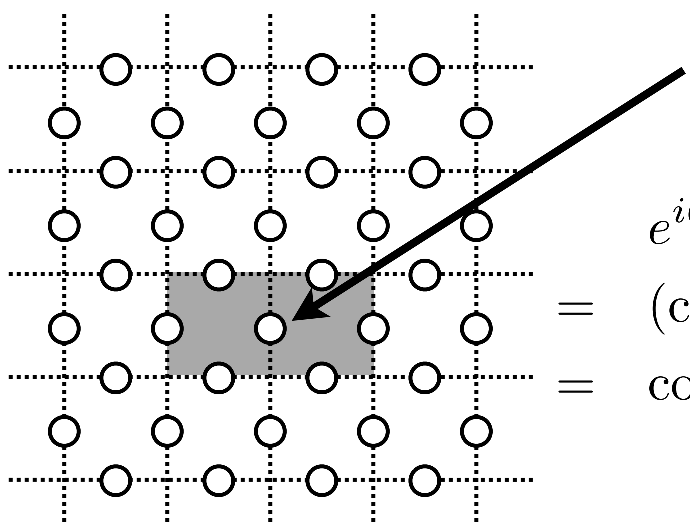


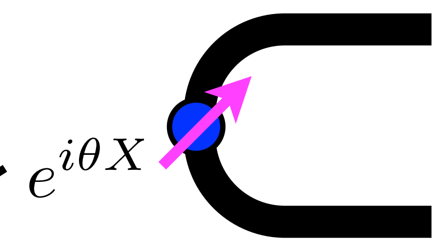
|| トポロジーが同じだったら同じ演算



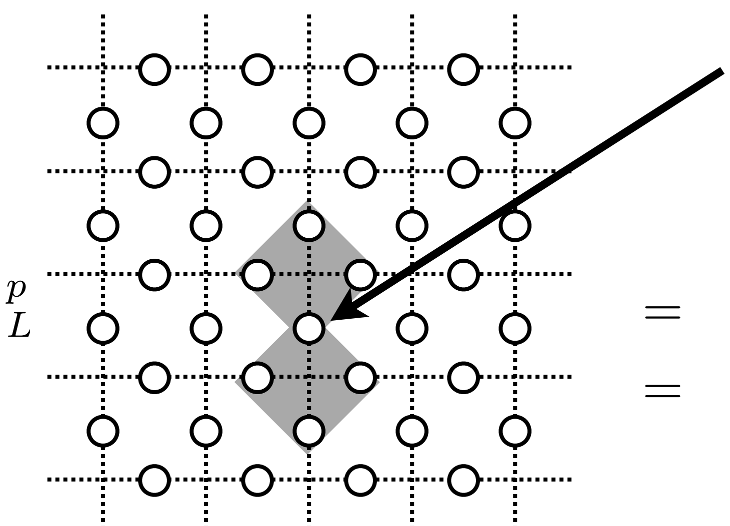
Universal QC

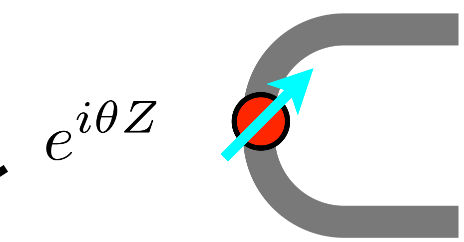
▶ State injection:





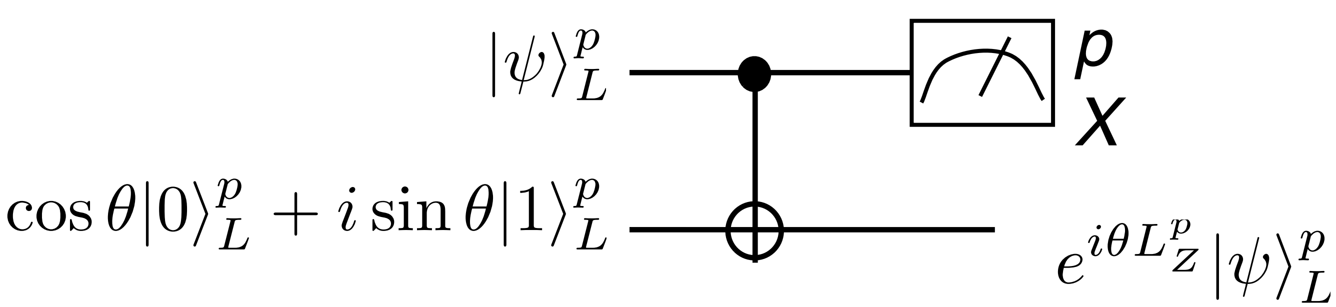
$$
\begin{aligned}
& e^{i\theta X}|\Psi_{\mathrm{vac}}\rangle \\
= \; & (\cos\theta I + i\sin\theta L_X^p)|0\rangle_L^p \\
= \; & \cos\theta|0\rangle_L^p + i\sin\theta|1\rangle_L^p
\end{aligned}
$$

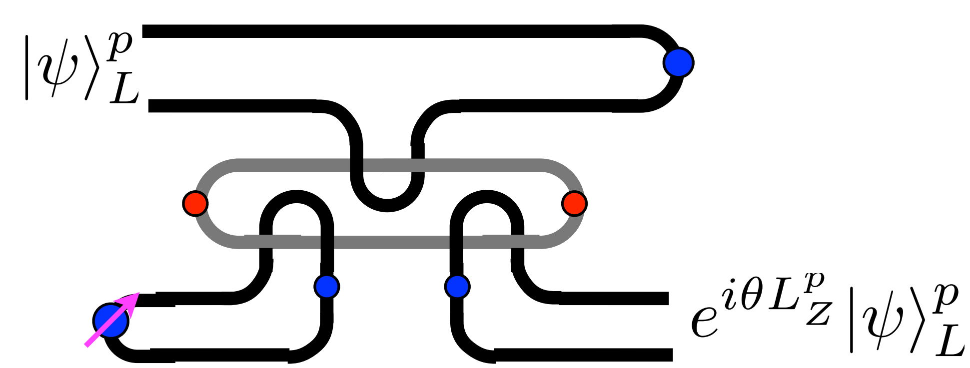




$$
\begin{aligned}
& e^{iZ\theta}|\Psi_{\mathrm{vac}}\rangle \\
= \; & \cos\theta|+\rangle_L^d + i\sin\theta|-\rangle_L^d \\
= \; & e^{i\theta}(|0\rangle_L^d + e^{-i2\theta}|1\rangle_L^d)
\end{aligned}
$$

▶ One-bit teleportation for non-Clifford gate



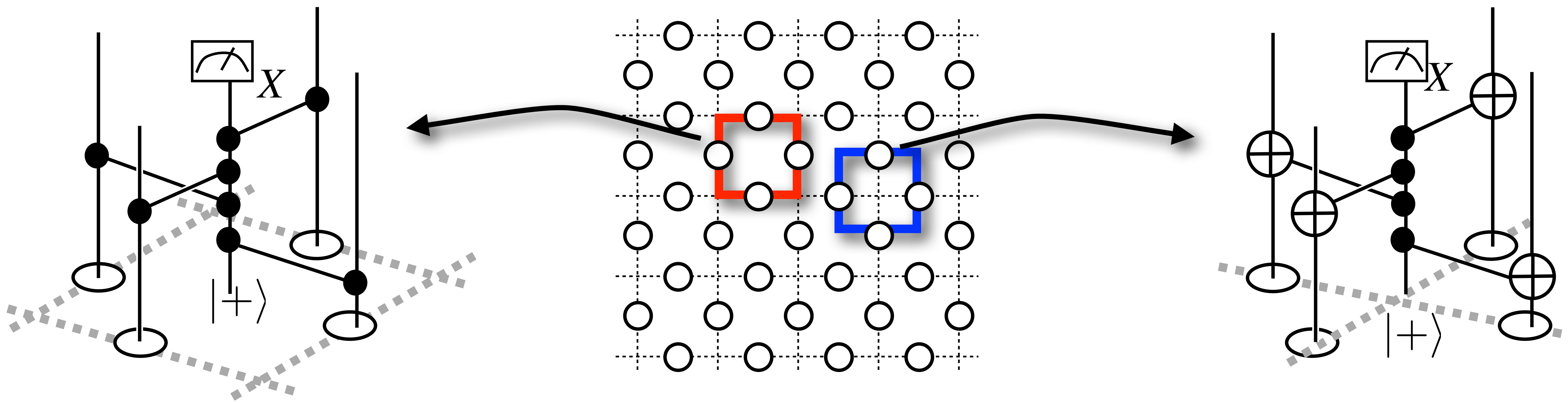


第六章:トポロジカル量子計算(MBQCの視点から)

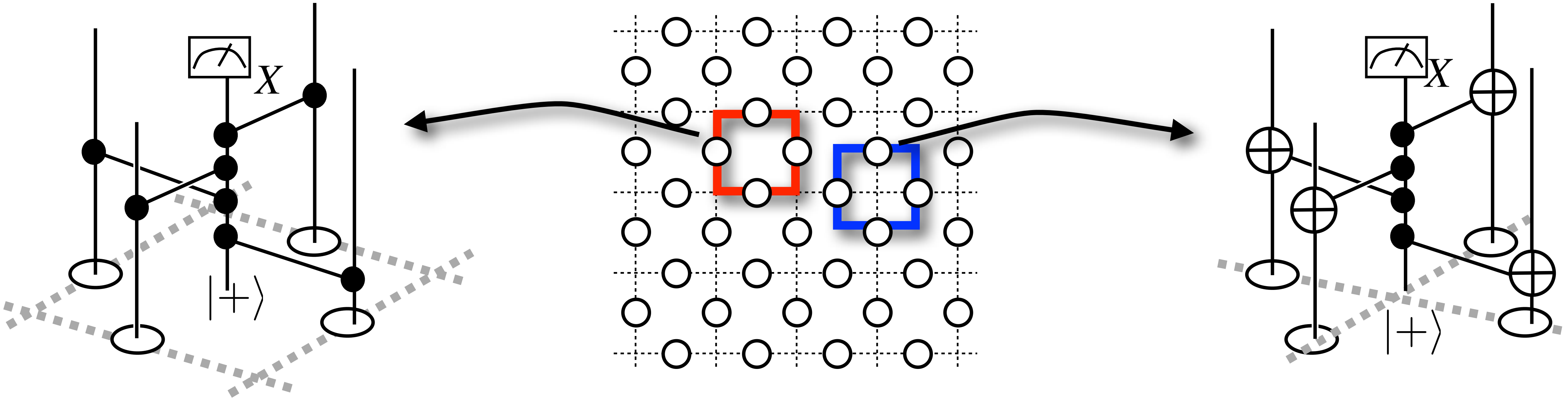
ここで勉強する内容：

第五章の内容をMBQCの文脈で理解する.

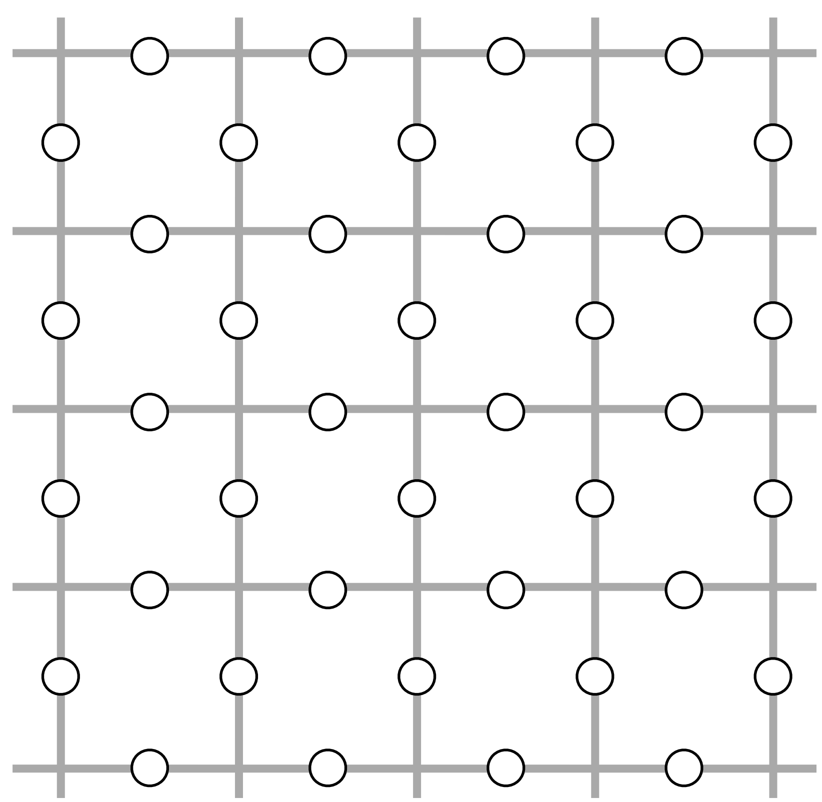
Syndrome measurement in 2D



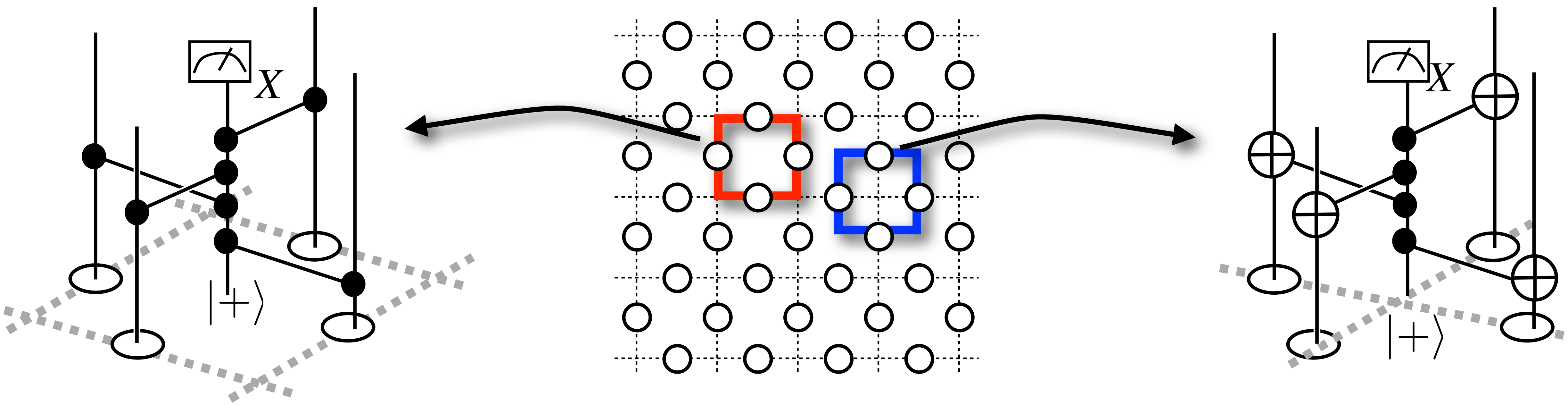
Syndrome measurement in 2D



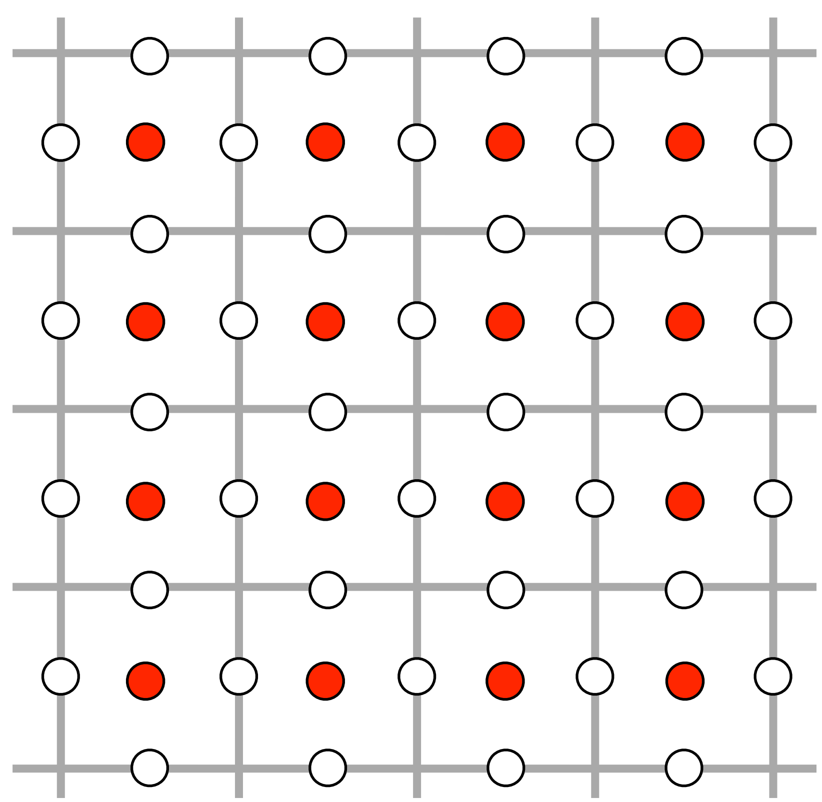
奇数タイムステップ



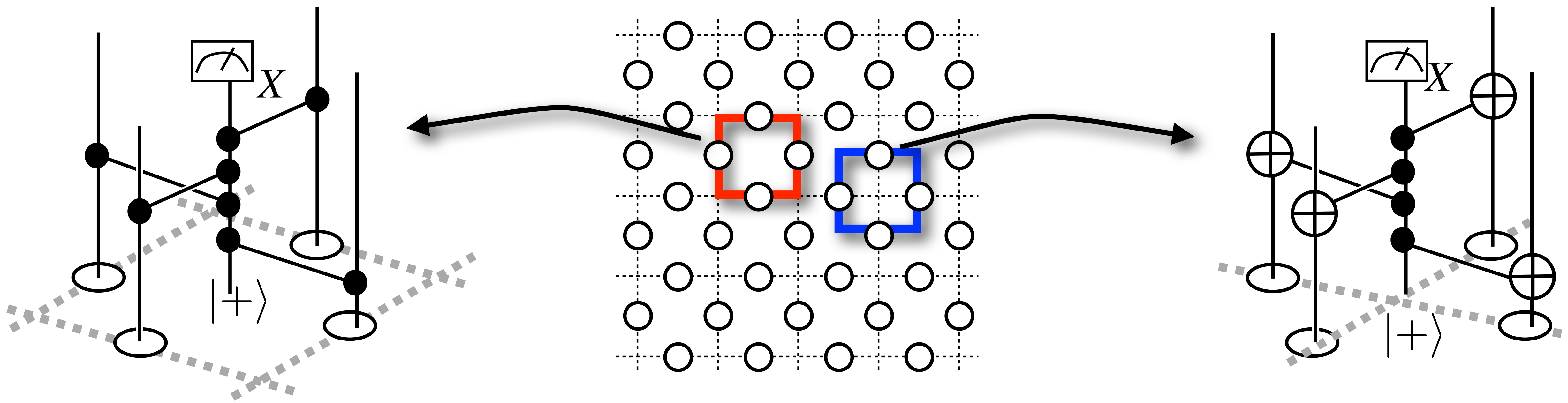
Syndrome measurement in 2D



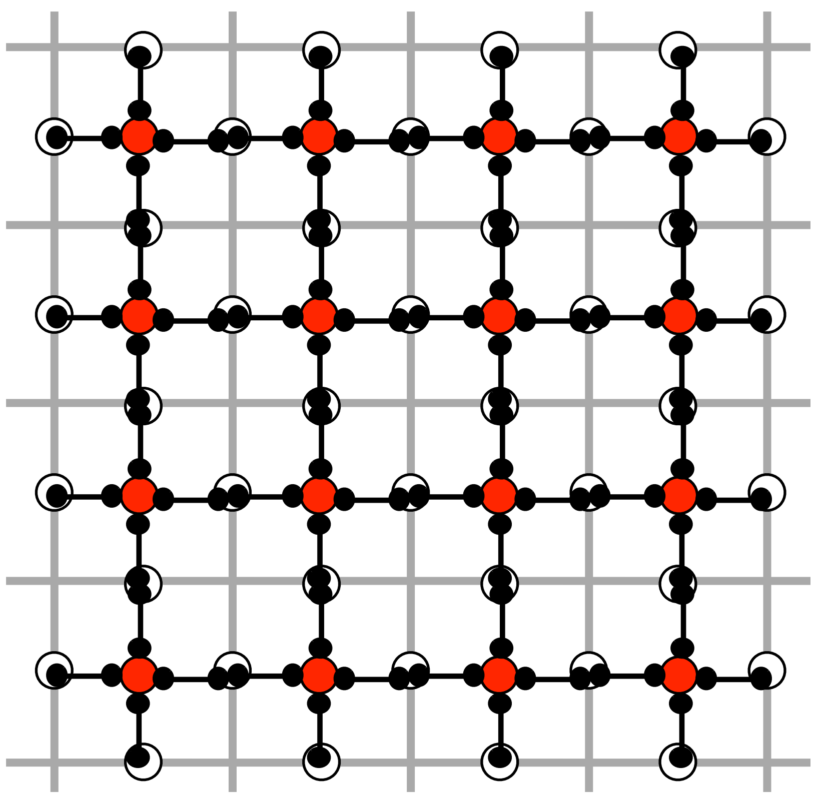
奇数タイムステップ



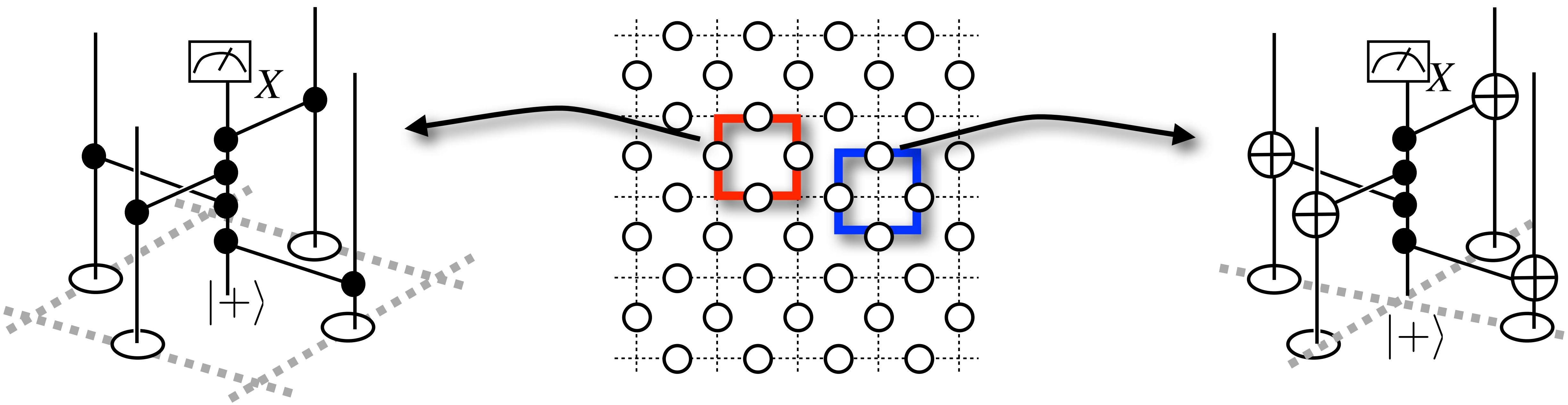
Syndrome measurement in 2D



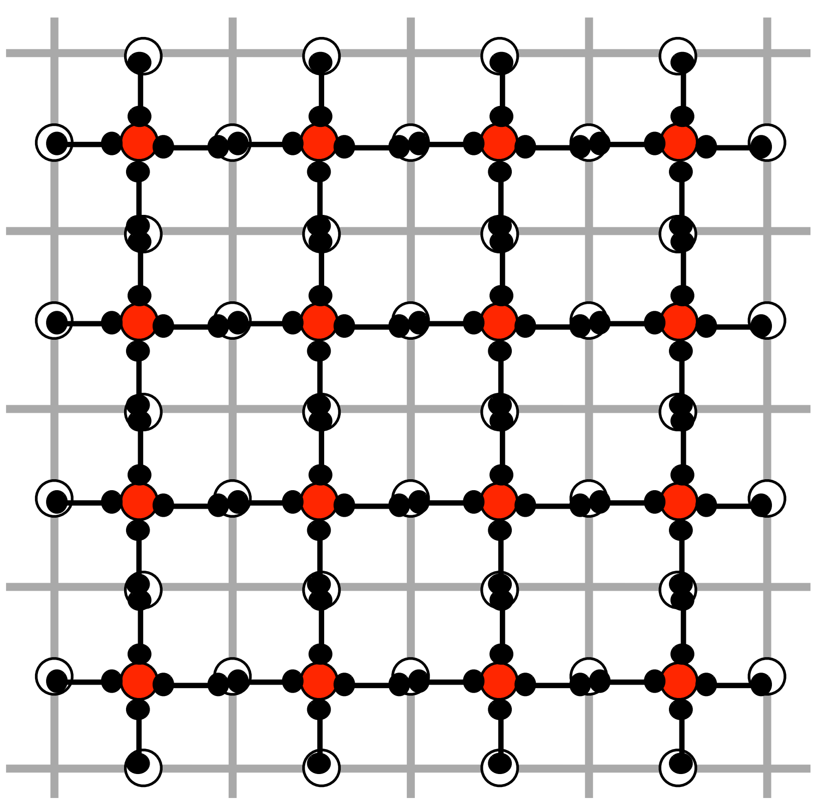
奇数タイムステップ



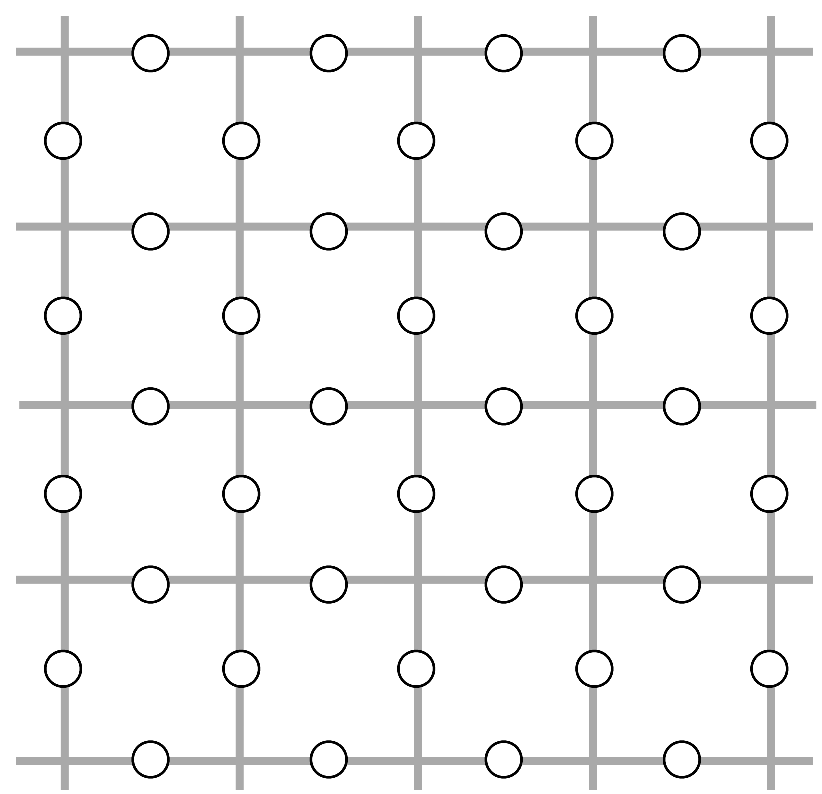
Syndrome measurement in 2D



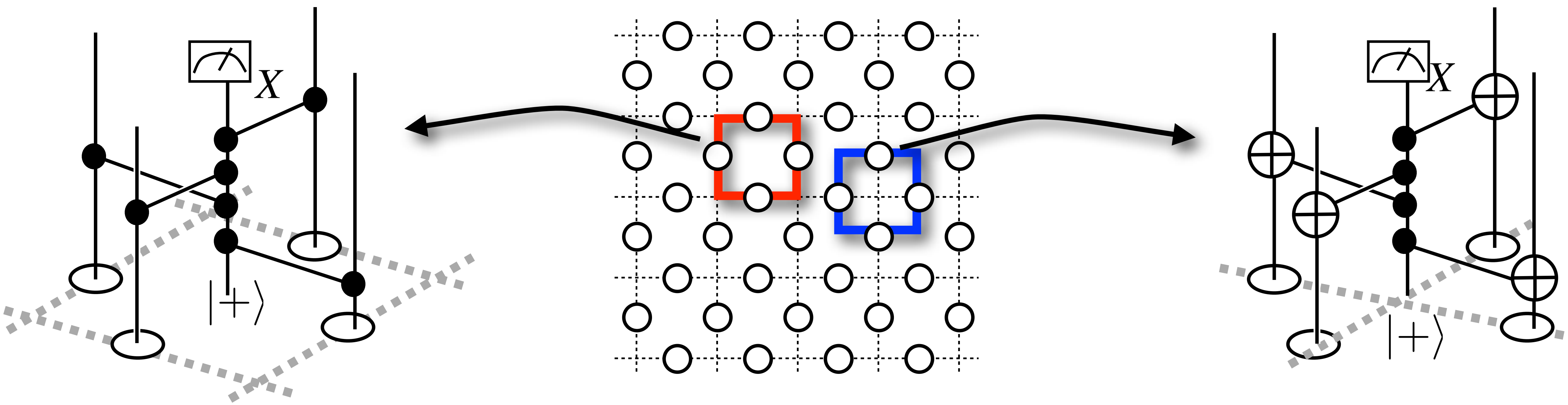
奇数タイムステップ



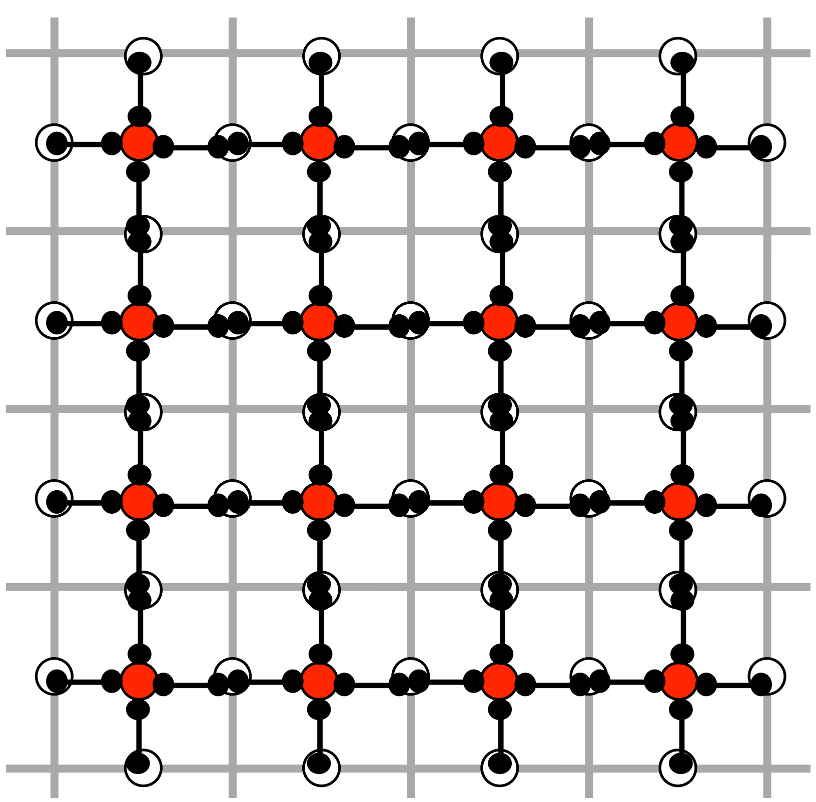
偶数タイムステップ



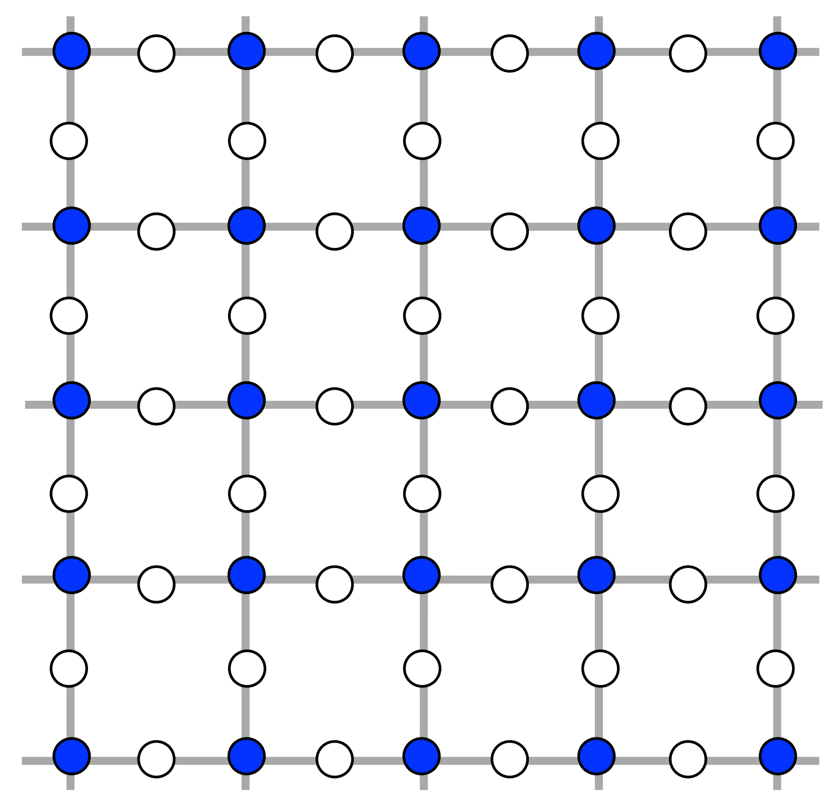
Syndrome measurement in 2D



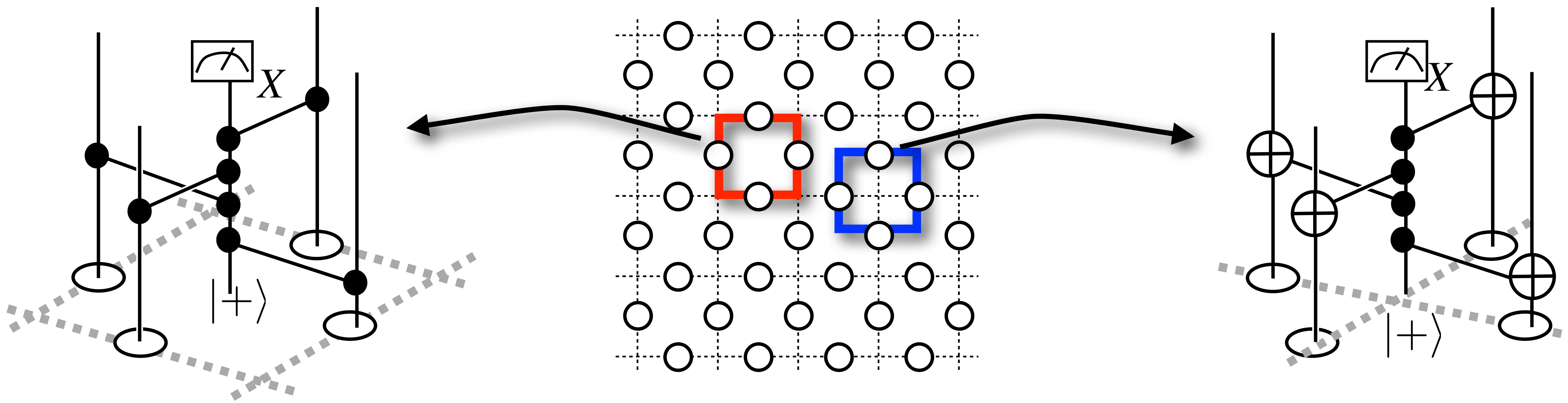
奇数タイムステップ

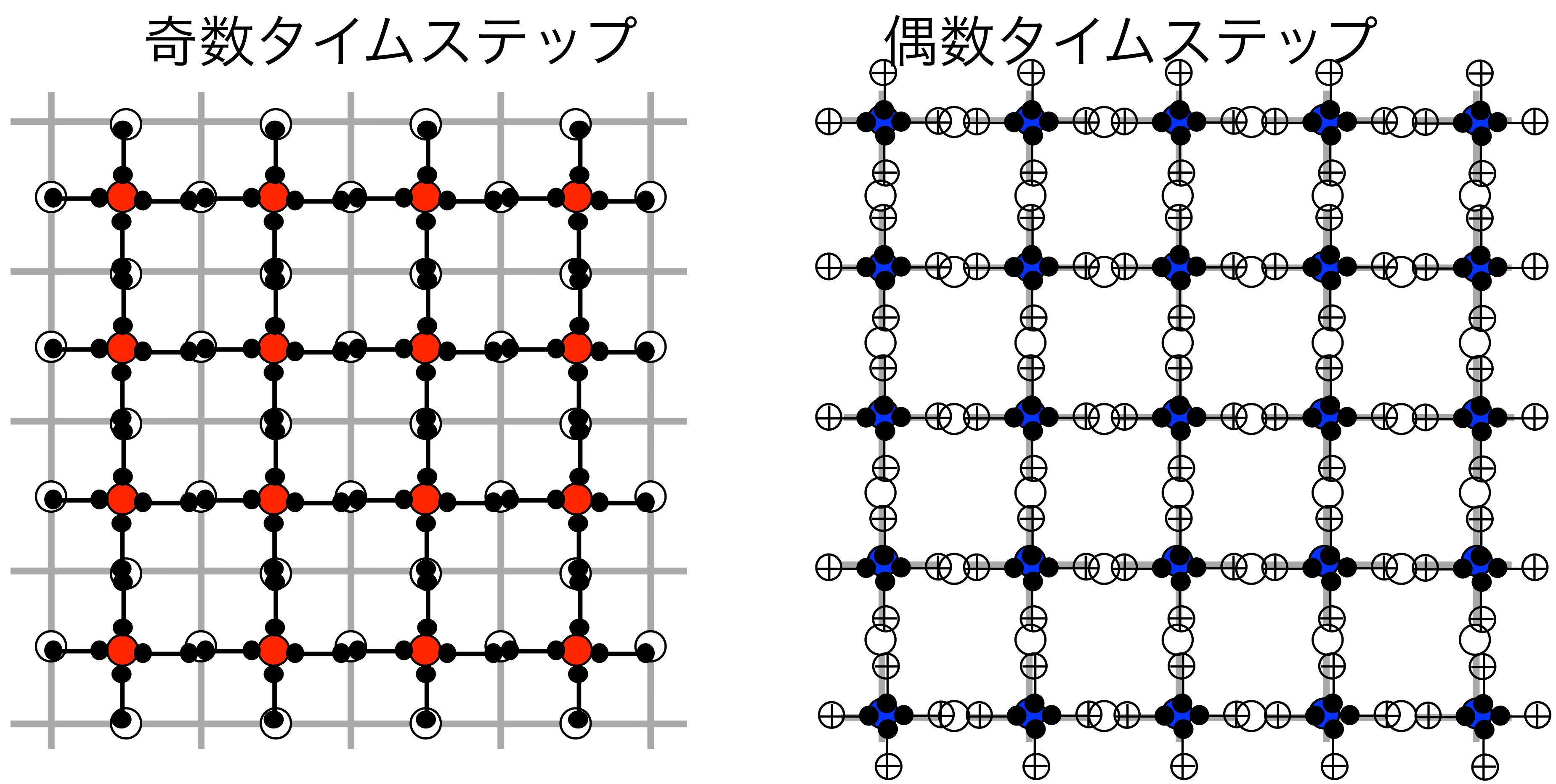


偶数タイムステップ

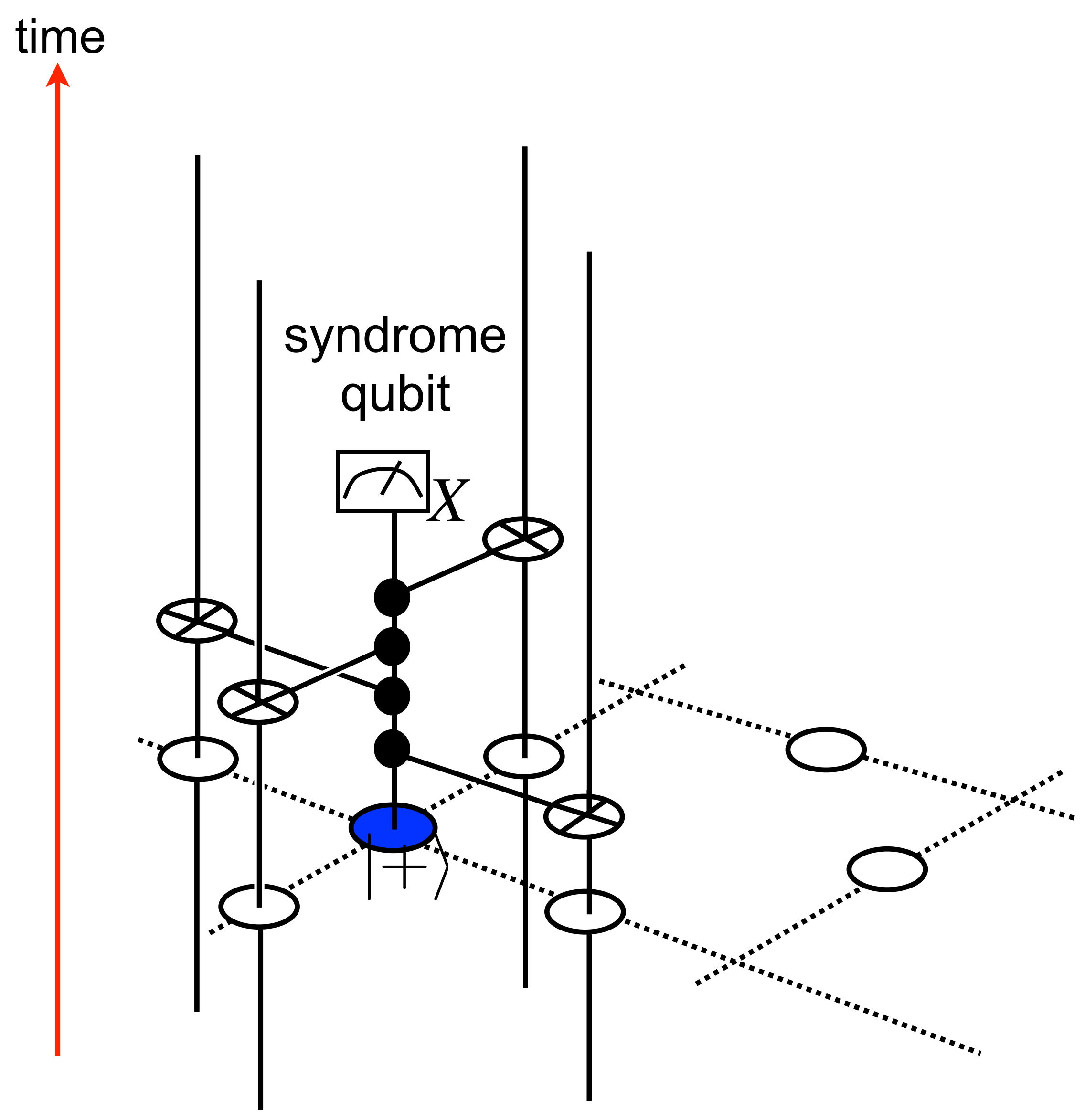


Syndrome measurement in 2D

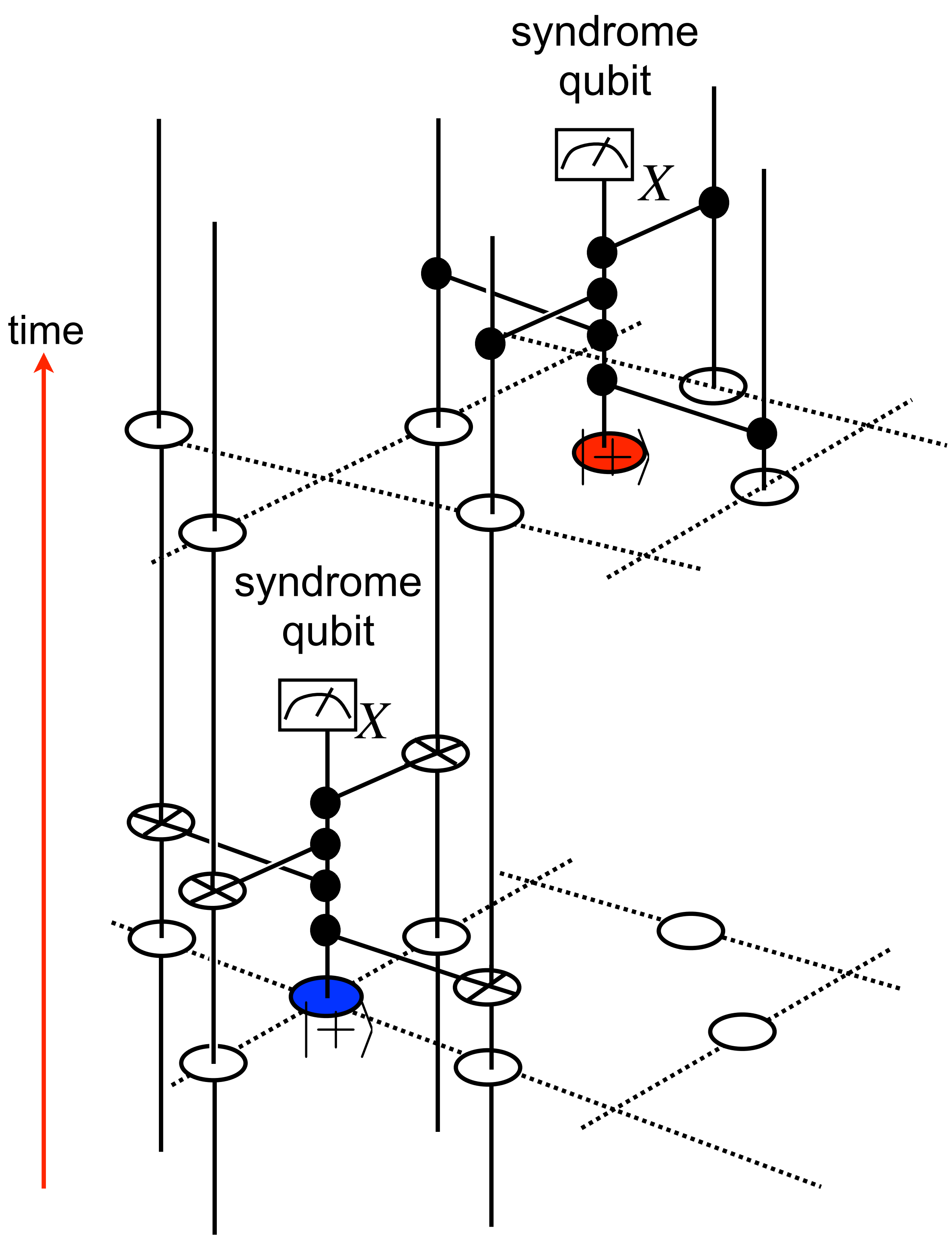




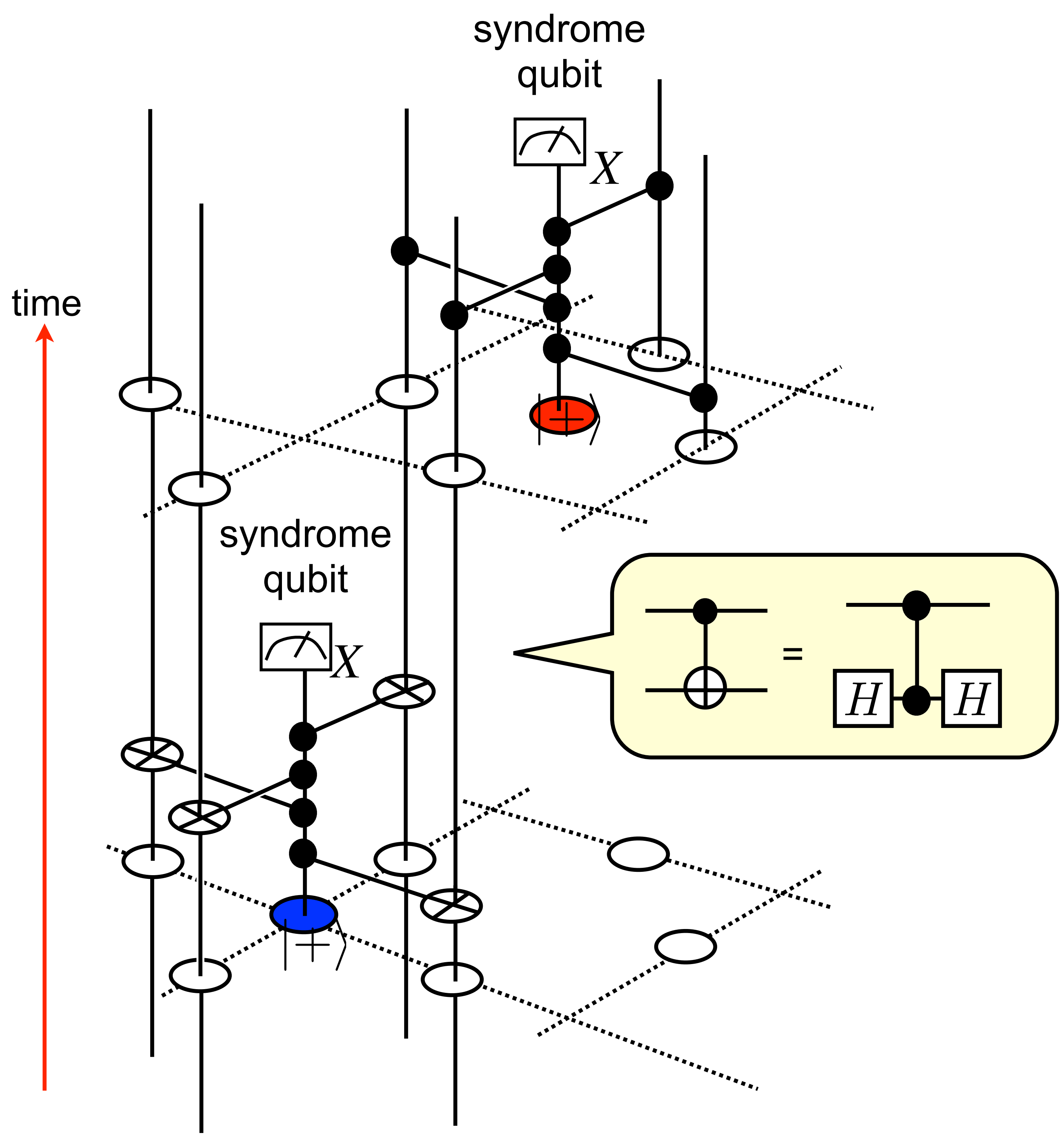
Topological MBQC in 3D



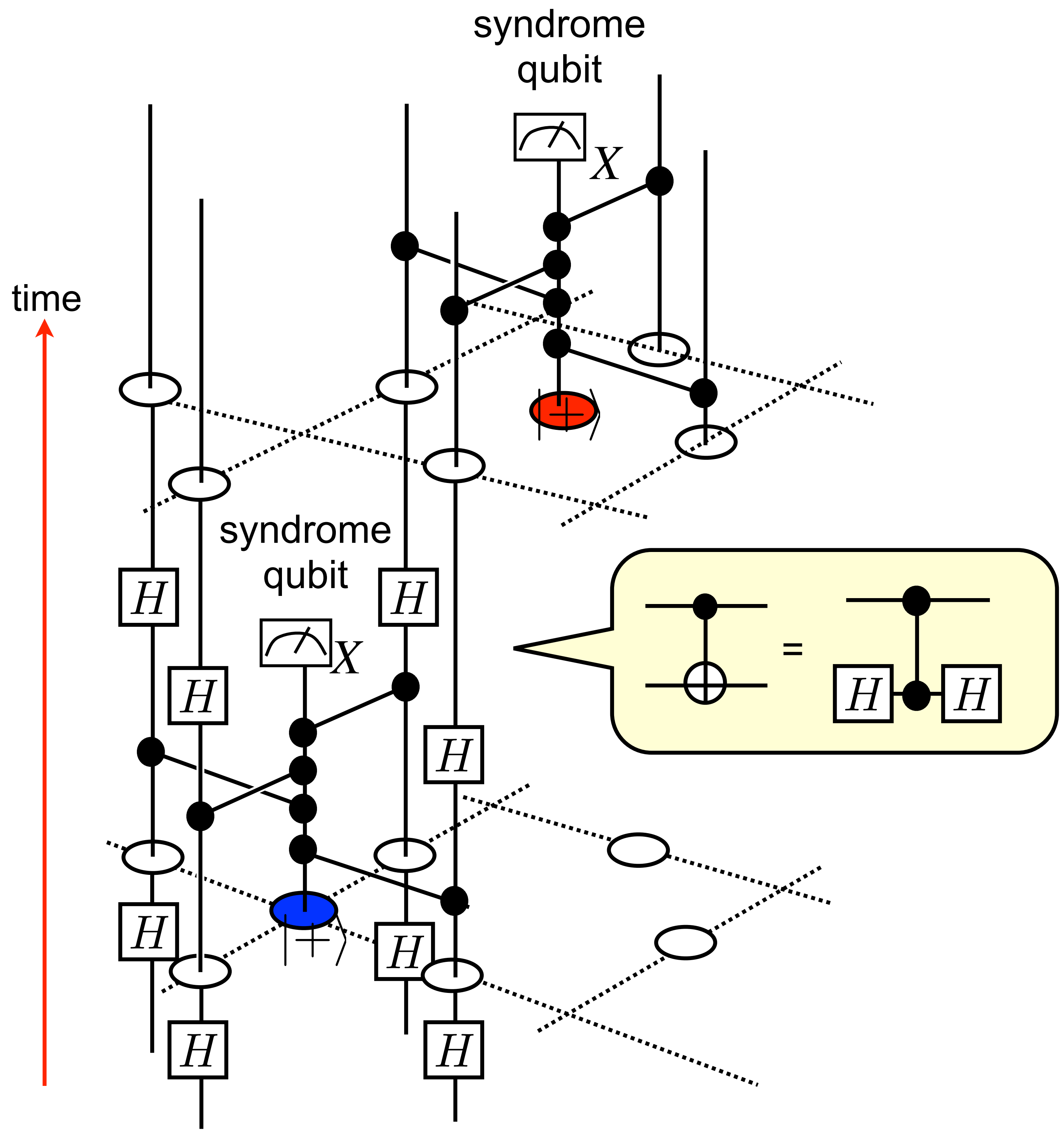
Topological MBQC in 3D



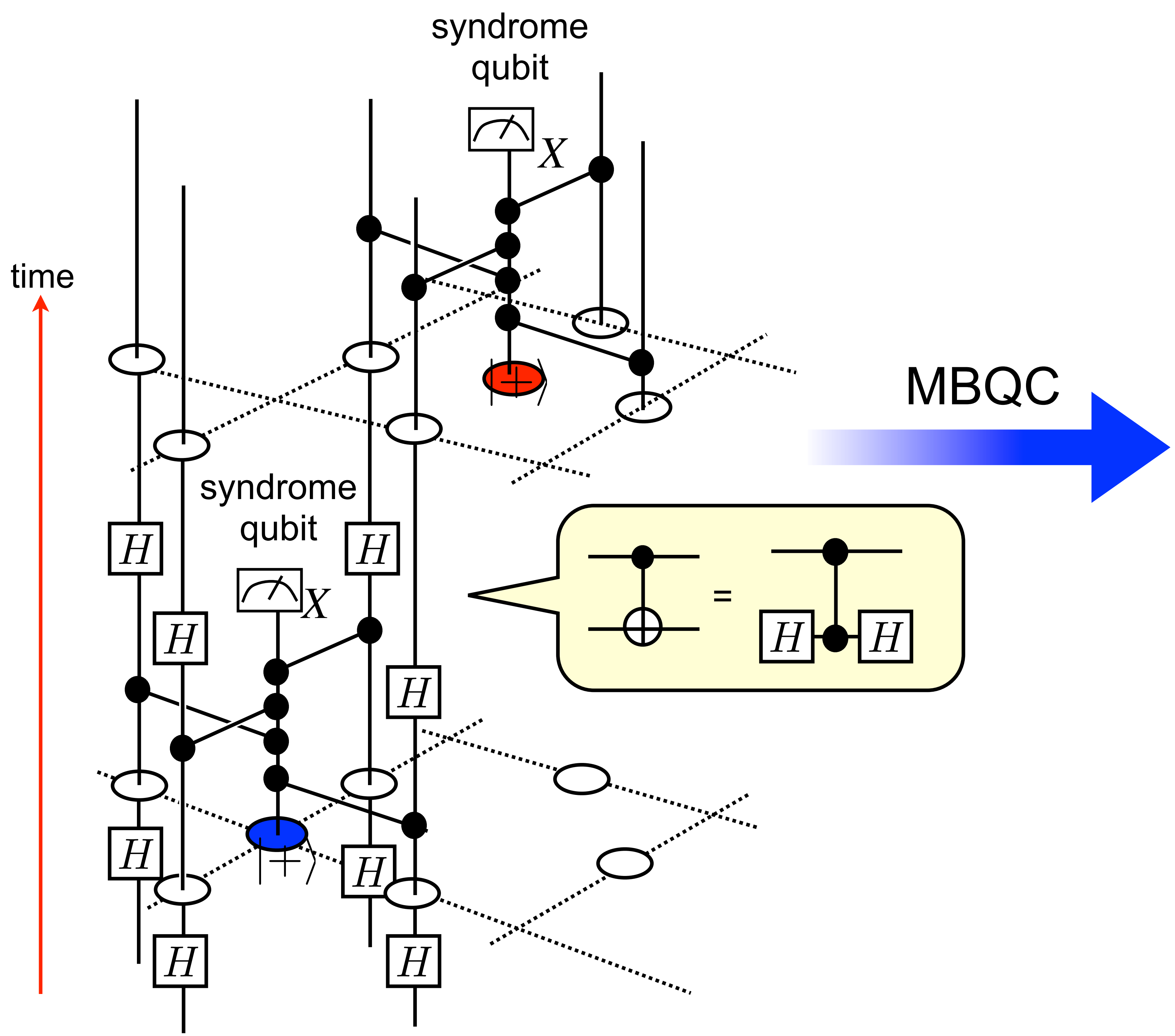
Topological MBQC in 3D



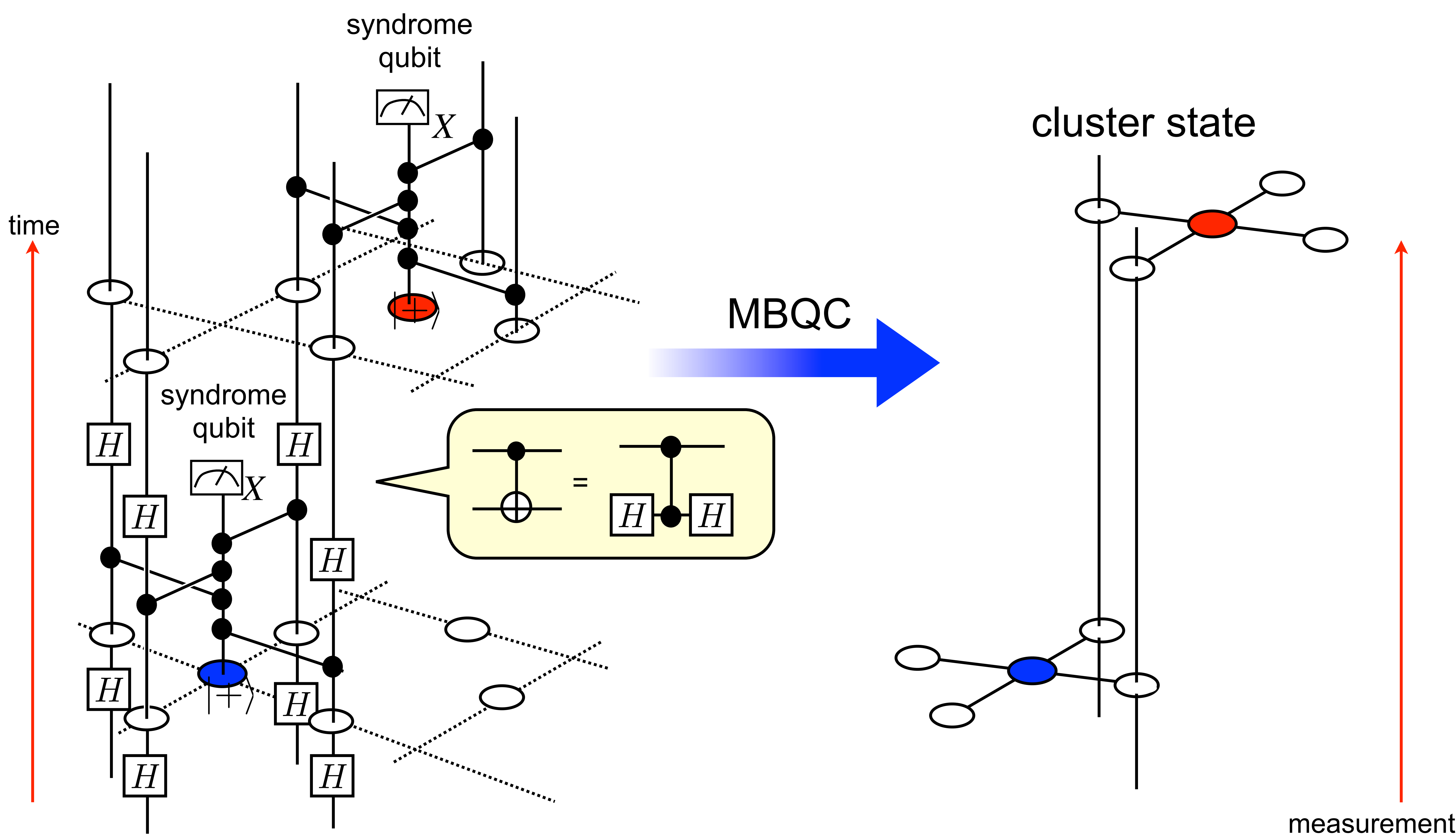
Topological MBQC in 3D



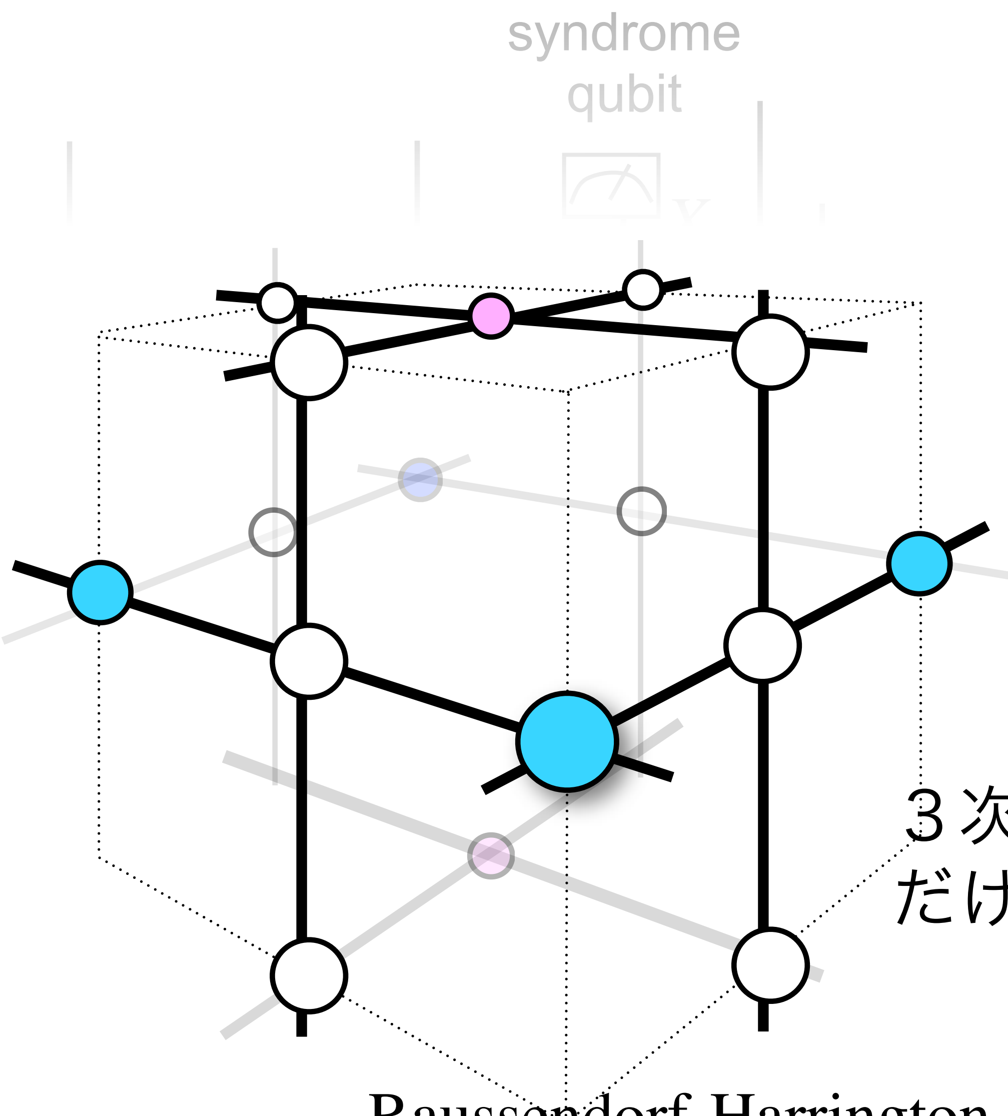
Topological MBQC in 3D



Topological MBQC in 3D



Topological MBQC in 3D

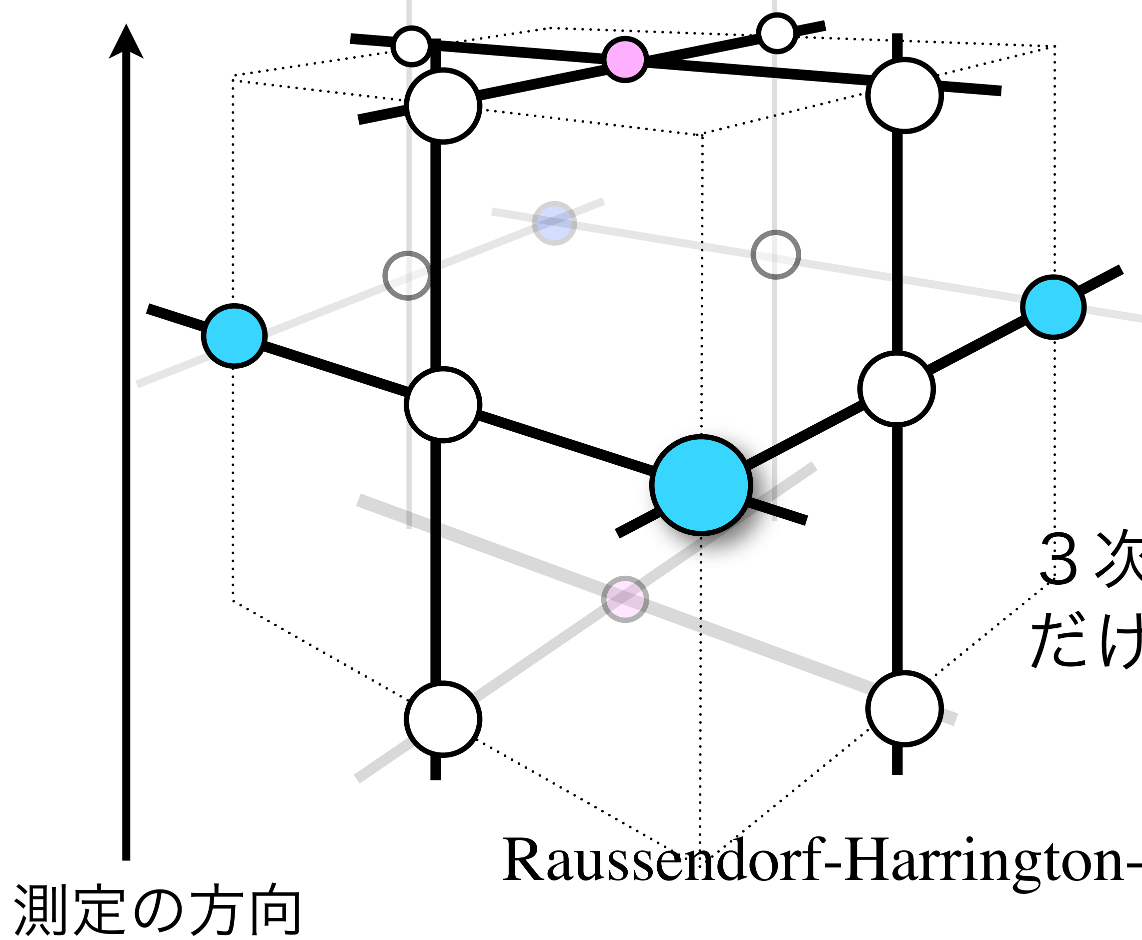


３次元のクラスター状態
（Raussendorf-Harrington-Goyal 格子）

３次元クラスター状態に対してX基底で測定をする
だけで誤り訂正が出来る
→**Topologically protected MBQC**

Raussendorf-Harrington-Goyal,Annals Phys. **321**, 2242 (2006);NJP **9**, 199 (2007).

Topological MBQC in 3D

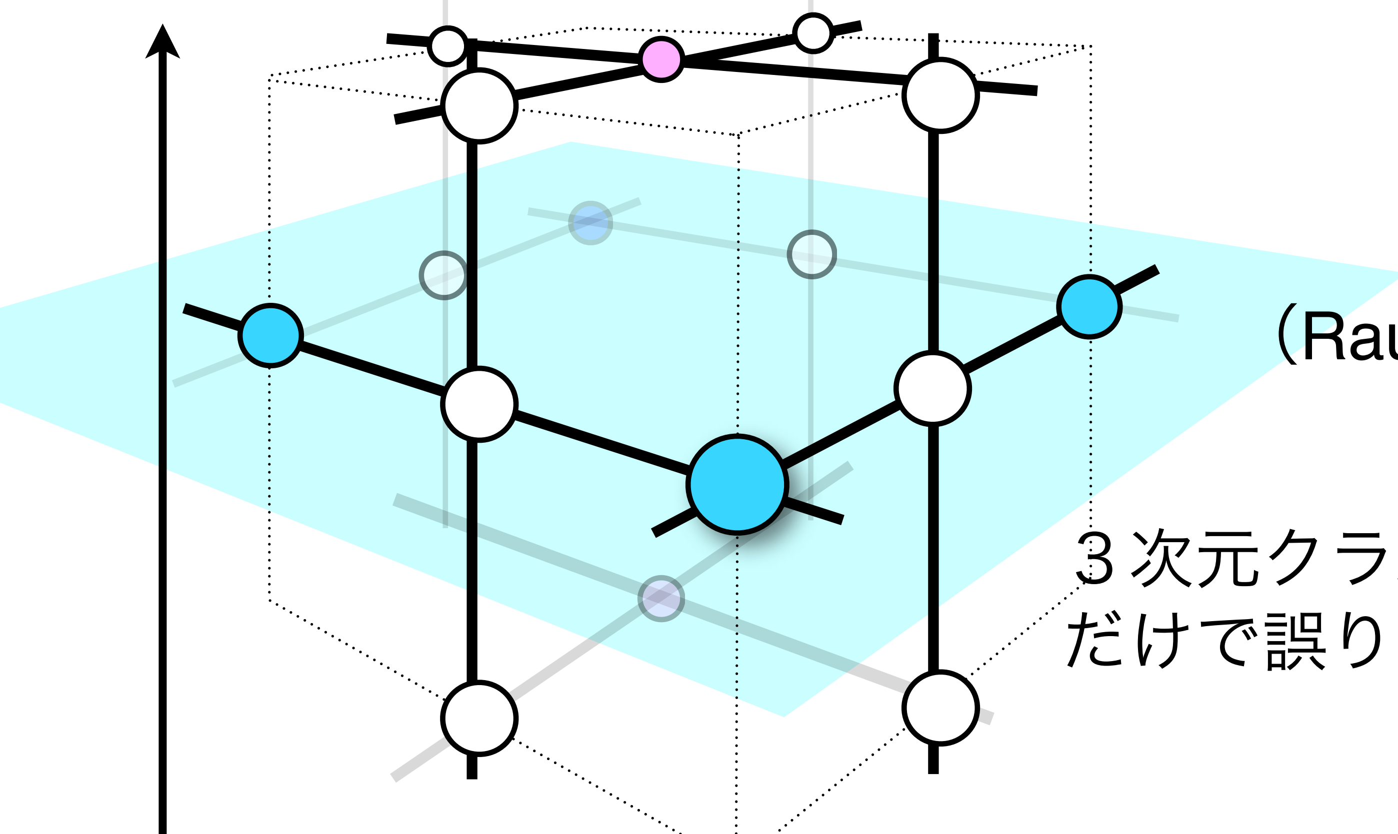


３次元のクラスター状態
（Raussendorf-Harrington-Goyal 格子）

３次元クラスター状態に対してX基底で測定をする
だけで誤り訂正が出来る
→**Topologically protected MBQC**

Raussendorf-Harrington-Goyal,Annals Phys. **321**, 2242 (2006);NJP **9**, 199 (2007).

Topological MBQC in 3D



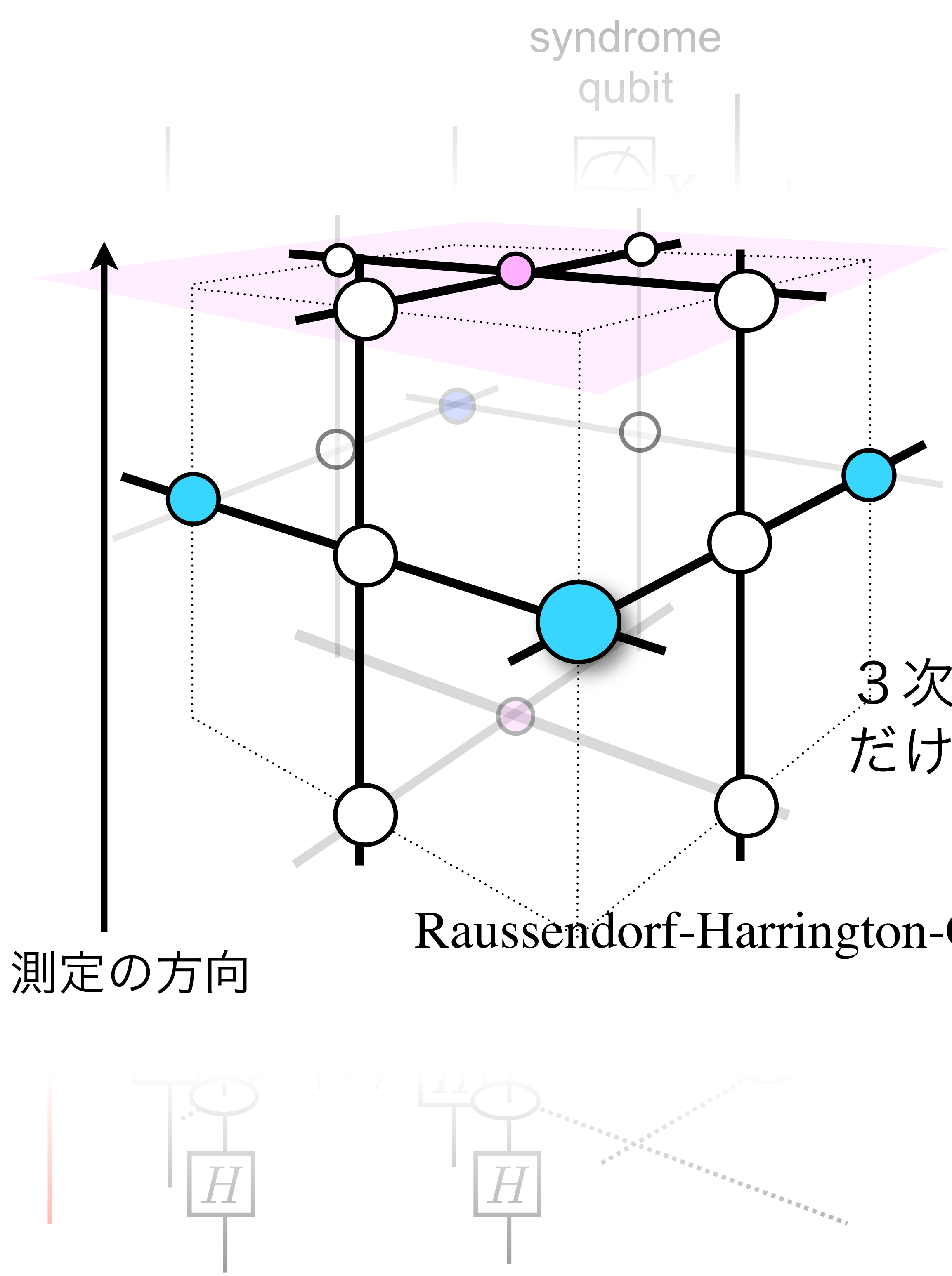
測定の方向

３次元のクラスター状態
（Raussendorf-Harrington-Goyal 格子）

３次元クラスター状態に対してX基底で測定をするだけで誤り訂正が出来る
→**Topologically protected MBQC**

Raussendorf-Harrington-Goyal,Annals Phys. **321**, 2242 (2006);NJP **9**, 199 (2007).

Topological MBQC in 3D



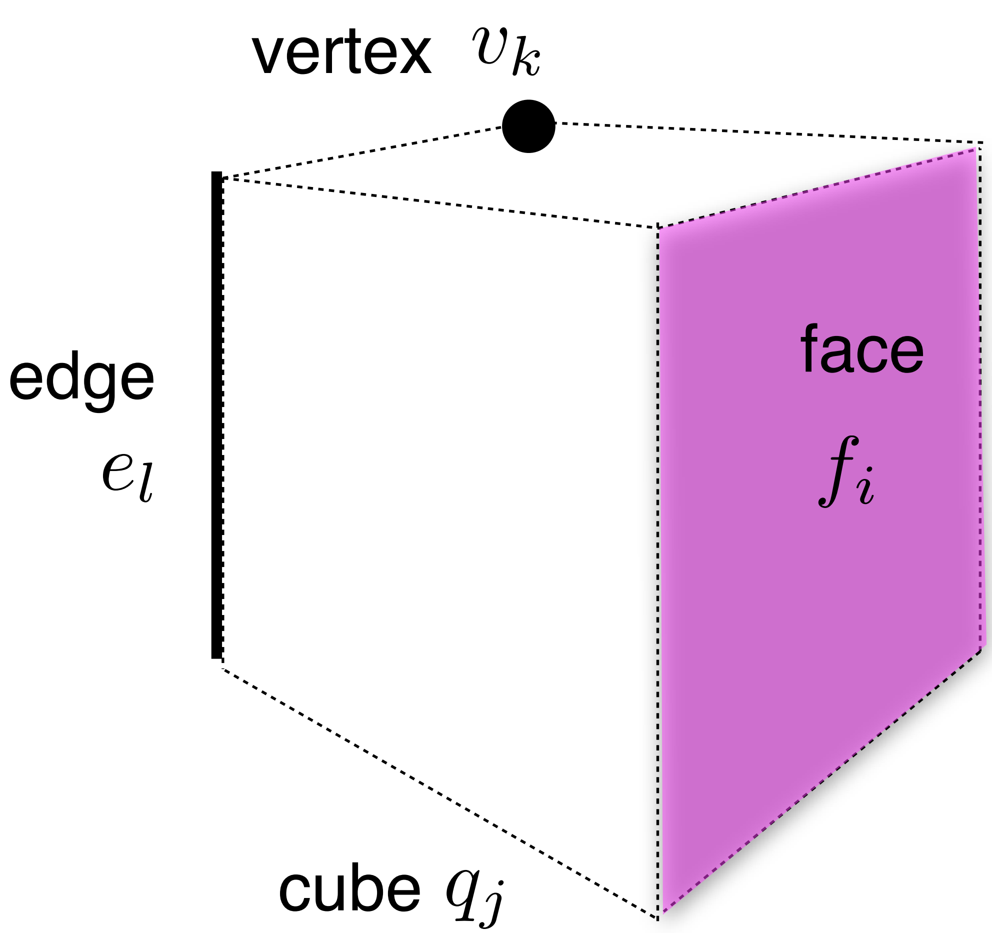
３次元のクラスター状態
（Raussendorf-Harrington-Goyal 格子）

３次元クラスター状態に対してX基底で測定をするだけで誤り訂正が出来る
→**Topologically protected MBQC**

Raussendorf-Harrington-Goyal,Annals Phys. **321**, 2242 (2006);NJP **9**, 199 (2007).

measurement

Chain complex in 3D

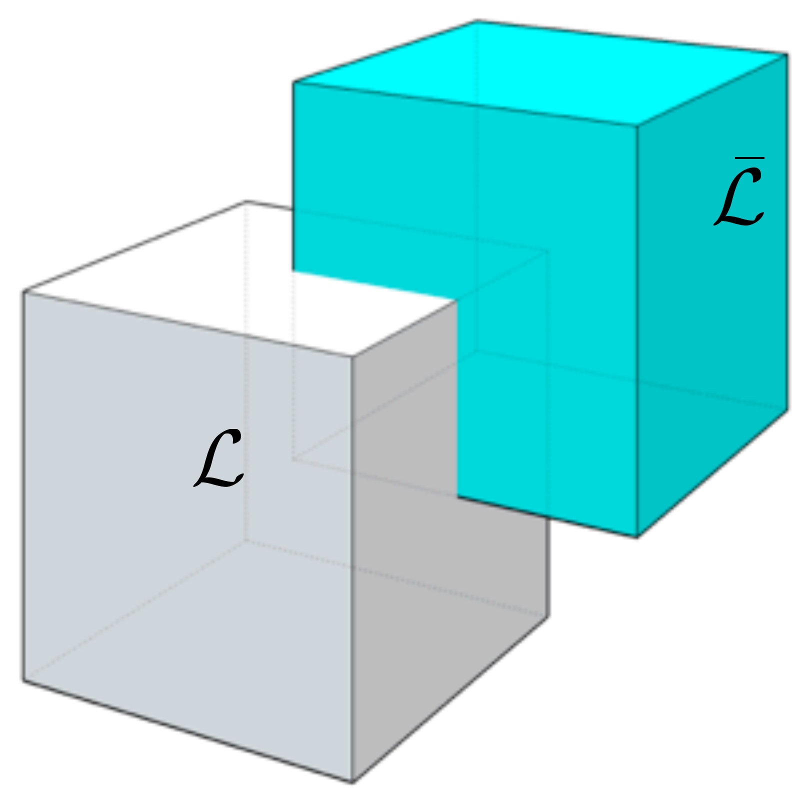


primal lattice $\mathcal{L}$	dual lattice $\bar{\mathcal{L}}$	
primal cube	dual cube	0
primal vertex	dual vertex	1
primal edge	dual edge	2
primal face	dual face	3

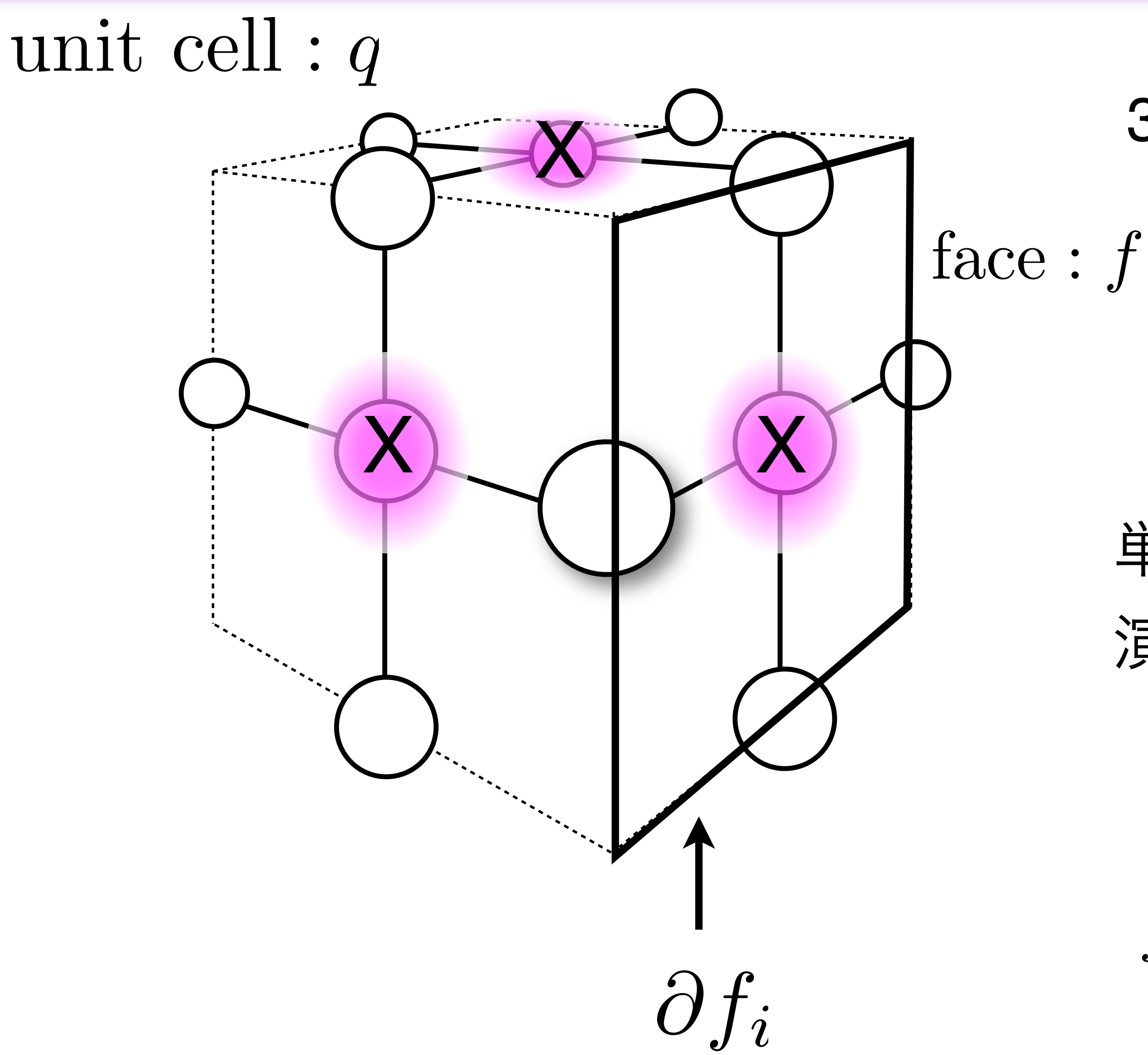
3-chain:

$$c_3 = \sum_j z_j q_j$$

∂c_3 は多面体の表面



Topological protection in *V*



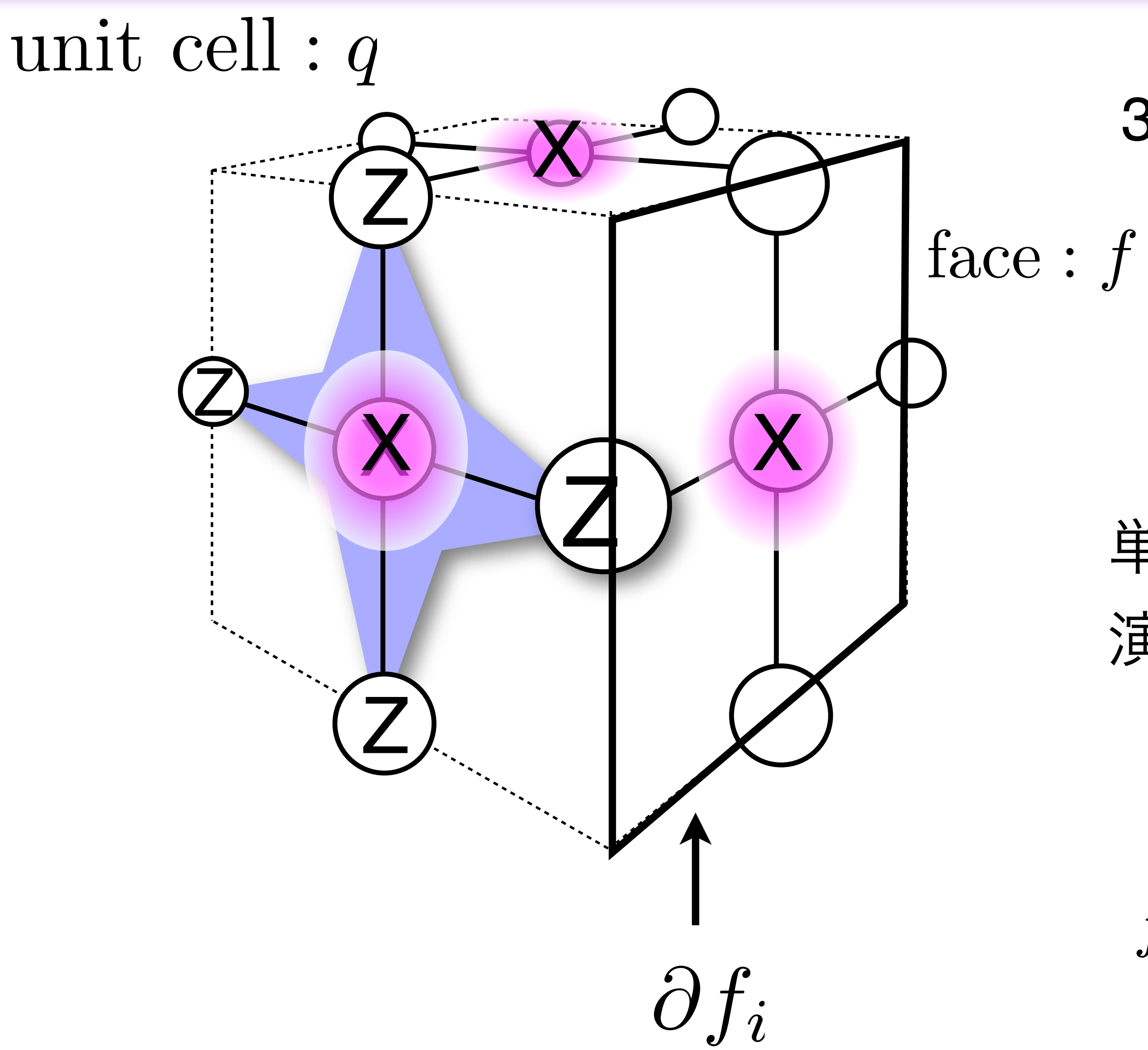
3D clusterのstabilizer演算子：

$$K(f_i) = X_{f_i} \prod_{e_l \in \partial f_i} Z_{e_l}$$

単位立方上の面qubitに対するスタビライザー演算子の積をとると、

$$\prod_{f_i \in \partial q_j} K(f_i) = \prod_{f_i \in \partial q_j} X_{f_i} \equiv S(q_j)$$

Topological protection in *V*



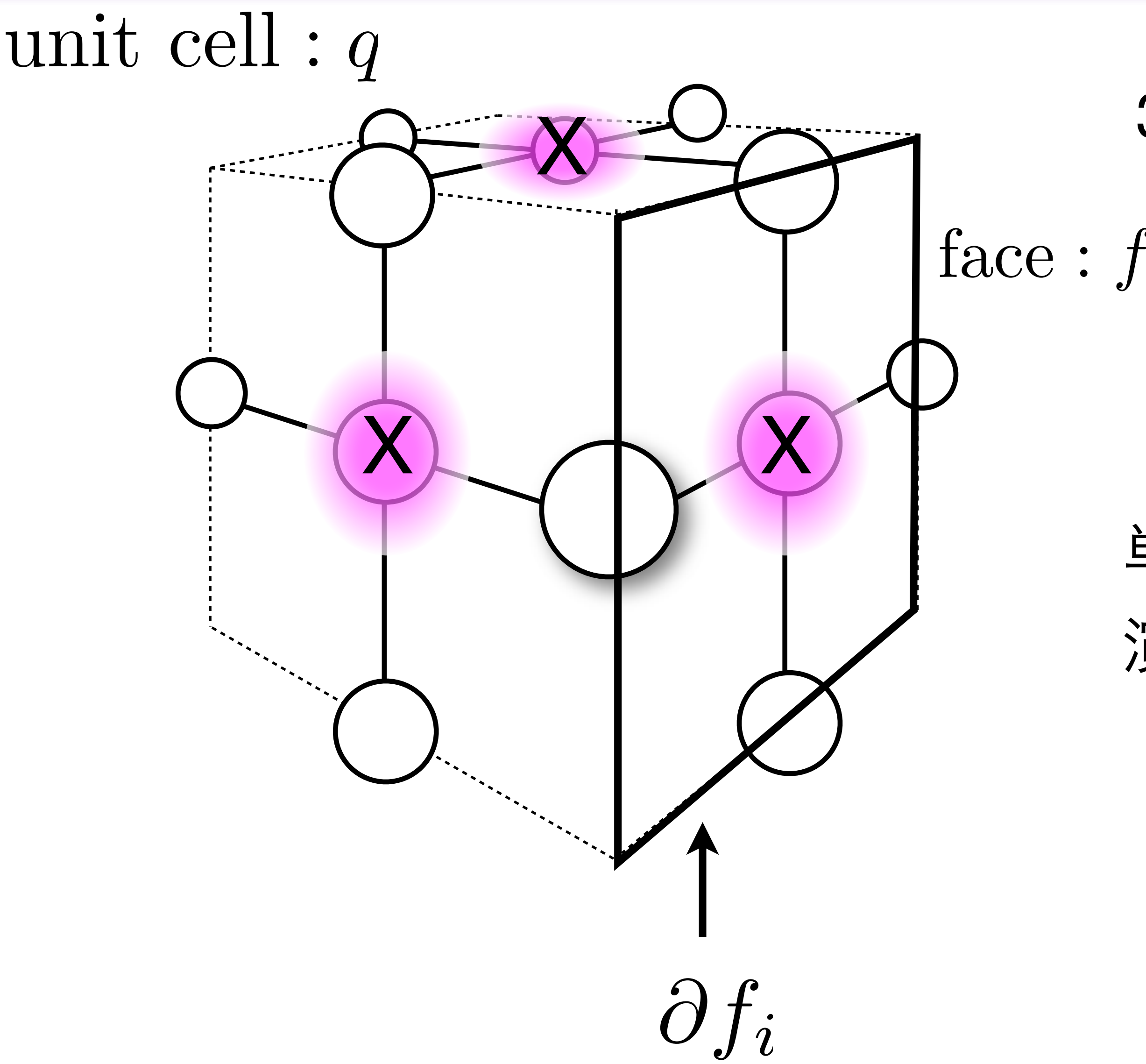
3D clusterのstabilizer演算子：

$$K(f_i) = X_{f_i} \prod_{e_l \in \partial f_i} Z_{e_l}$$

単位立方上の面qubitに対するスタビライザー演算子の積をとると、

$$\prod_{f_i \in \partial q_j} K(f_i) = \prod_{f_i \in \partial q_j} X_{f_i} \equiv S(q_j)$$

Topological protection in *V*



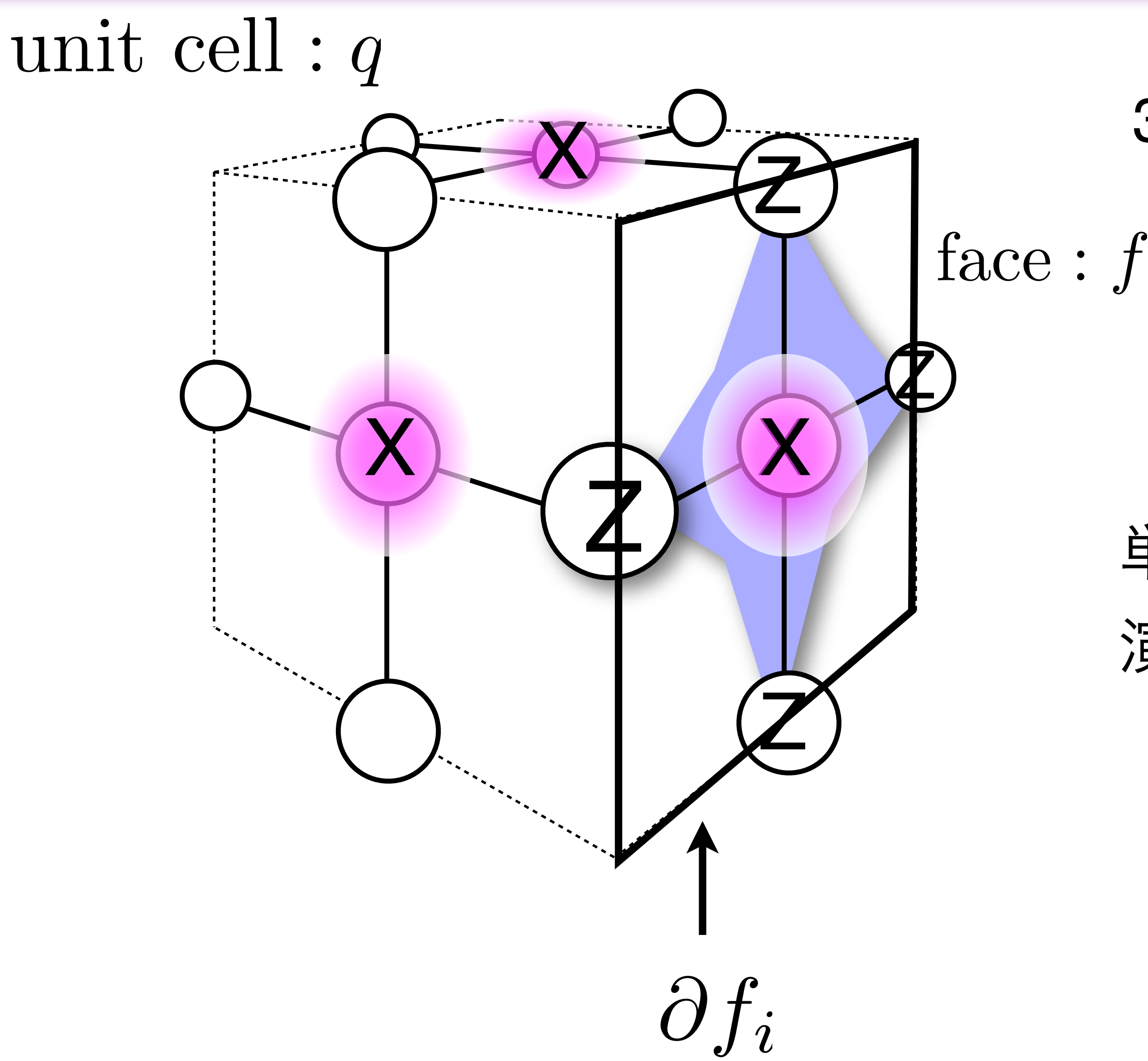
3D clusterのstabilizer演算子：

$$K(f_i) = X_{f_i} \prod_{e_l \in \partial f_i} Z_{e_l}$$

単位立方上の面qubitに対するスタビライザー演算子の積をとると、

$$\prod_{f_i \in \partial q_j} K(f_i) = \prod_{f_i \in \partial q_j} X_{f_i} \equiv S(q_j)$$

Topological protection in *V*



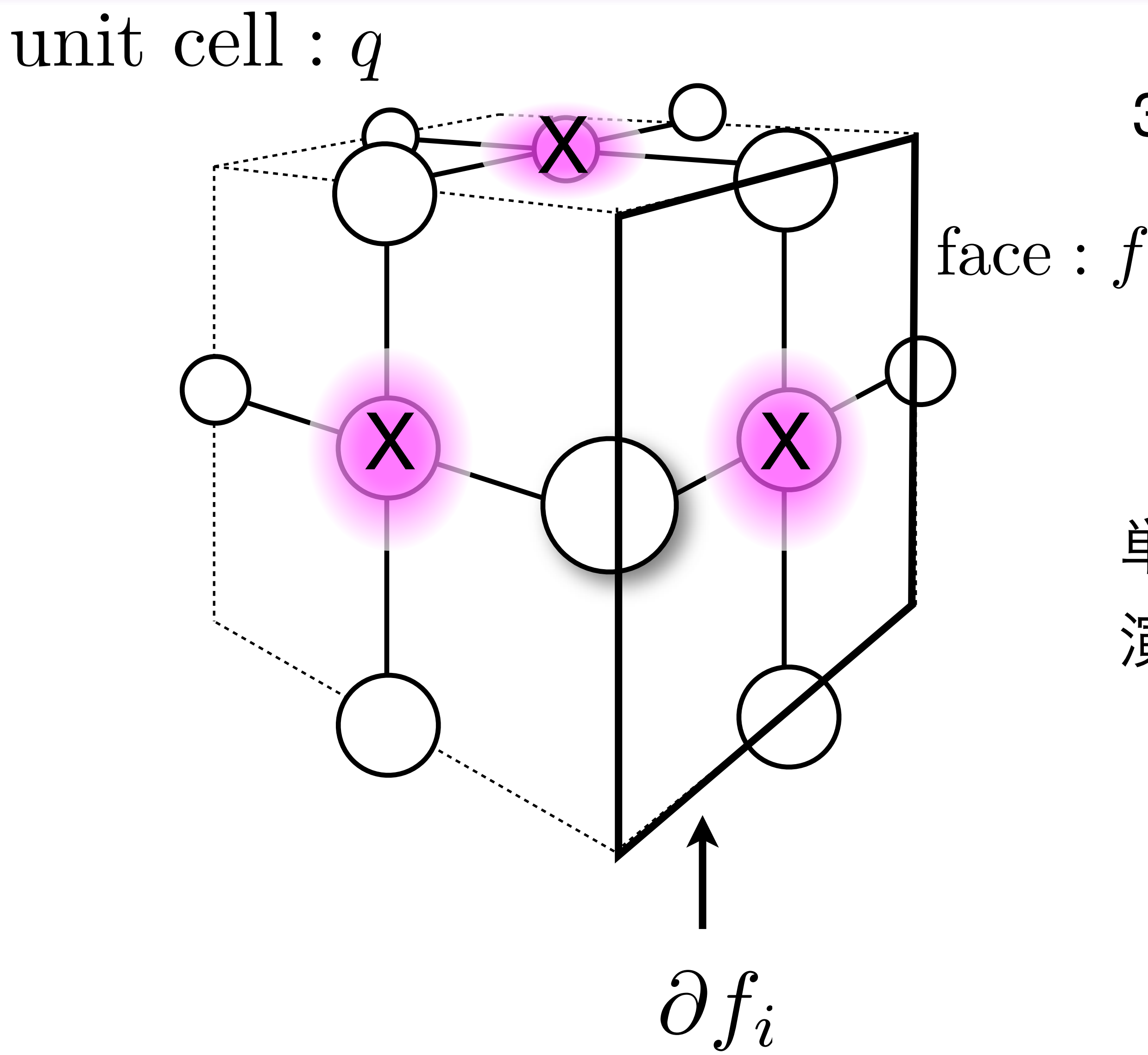
3D clusterのstabilizer演算子：

$$K(f_i) = X_{f_i} \prod_{e_l \in \partial f_i} Z_{e_l}$$

単位立方上の面qubitに対するスタビライザー演算子の積をとると、

$$\prod_{f_i \in \partial q_j} K(f_i) = \prod_{f_i \in \partial q_j} X_{f_i} \equiv S(q_j)$$

Topological protection in *V*



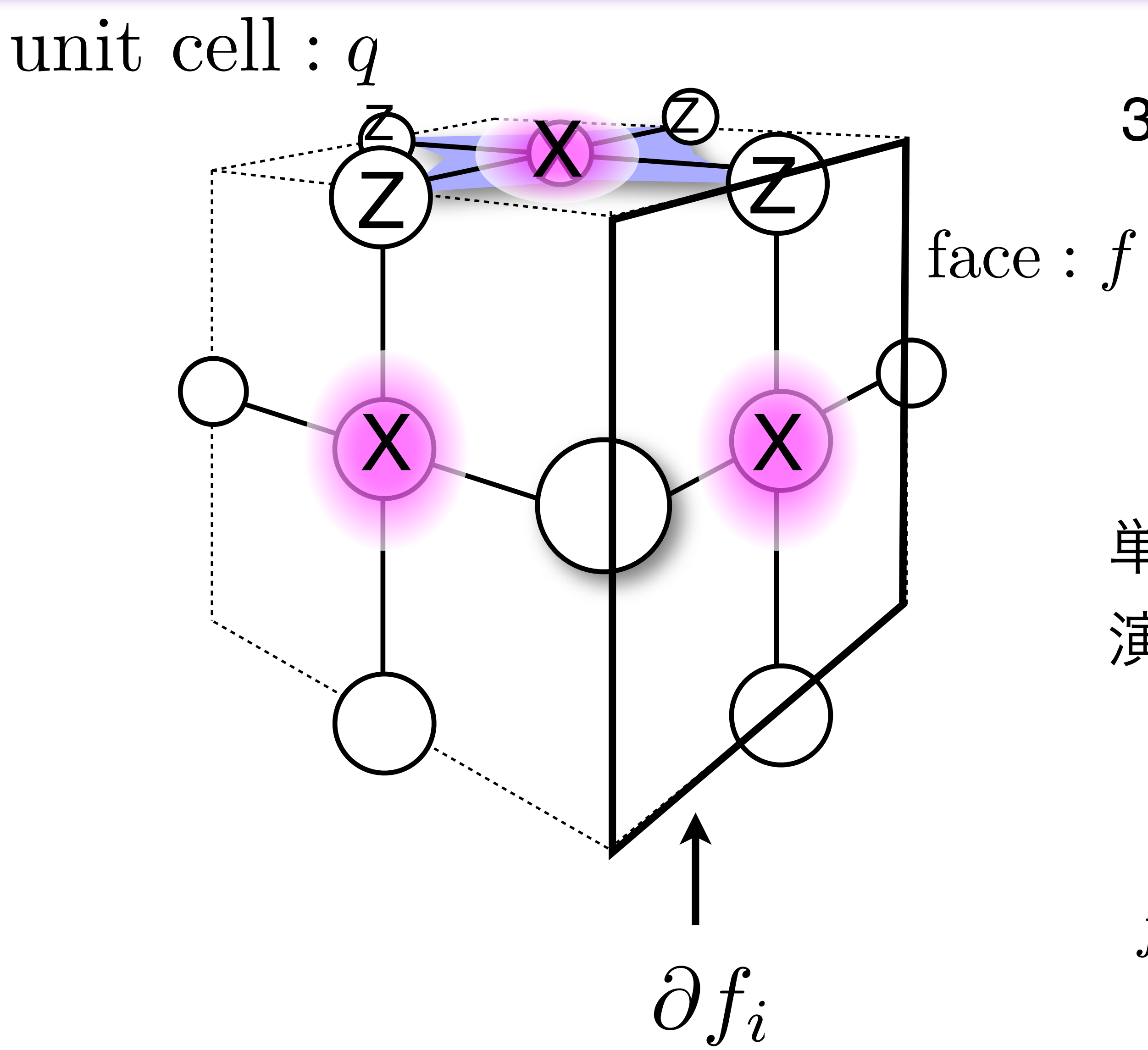
3D clusterのstabilizer演算子：

$$K(f_i) = X_{f_i} \prod_{e_l \in \partial f_i} Z_{e_l}$$

単位立方上の面qubitに対するスタビライザー演算子の積をとると、

$$\prod_{f_i \in \partial q_j} K(f_i) = \prod_{f_i \in \partial q_j} X_{f_i} \equiv S(q_j)$$

Topological protection in *V*



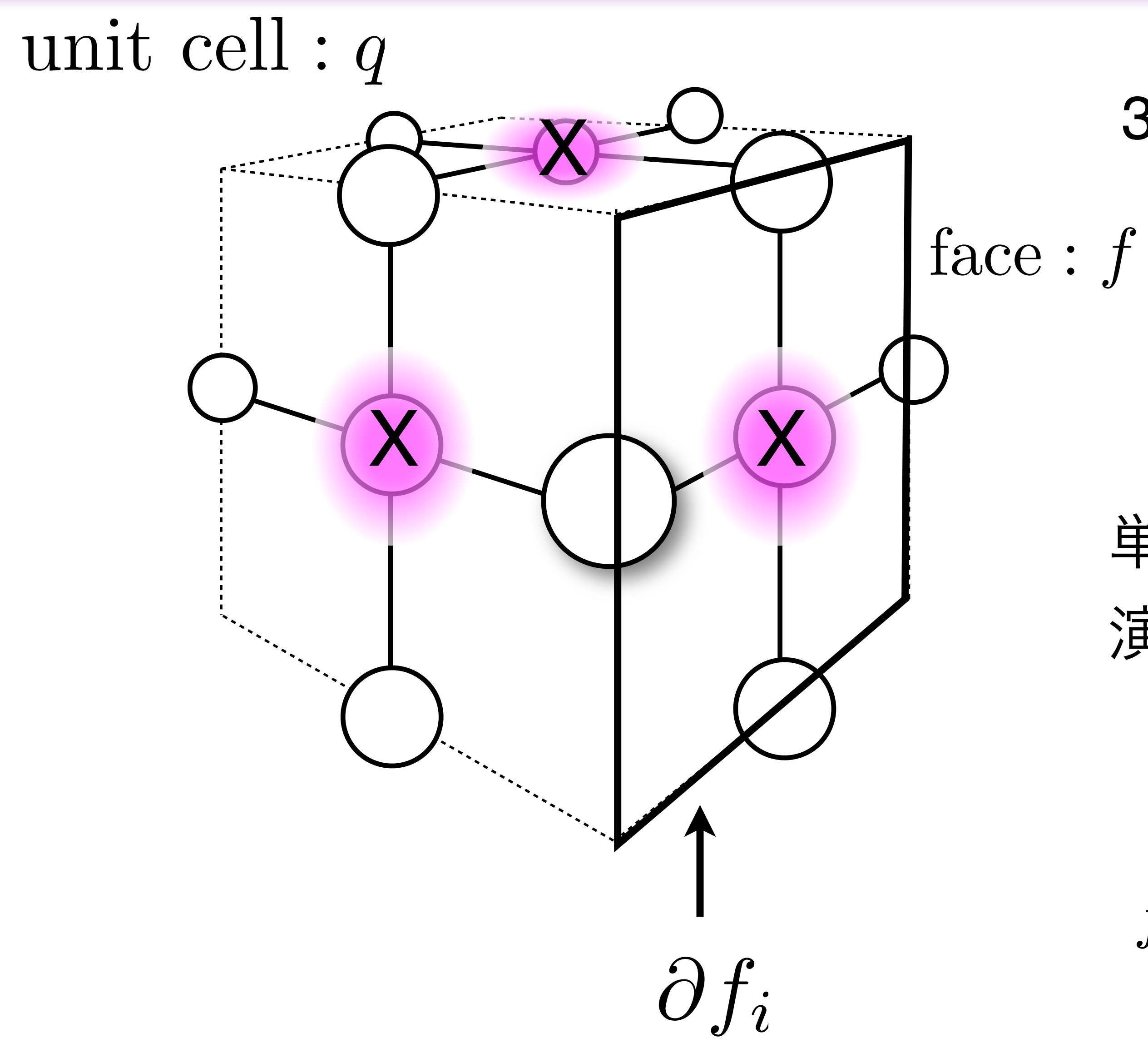
3D clusterのstabilizer演算子：

$$K(f_i) = X_{f_i} \prod_{e_l \in \partial f_i} Z_{e_l}$$

単位立方上の面qubitに対するスタビライザー演算子の積をとると、

$$\prod_{f_i \in \partial q_j} K(f_i) = \prod_{f_i \in \partial q_j} X_{f_i} \equiv S(q_j)$$

Topological protection in *V*



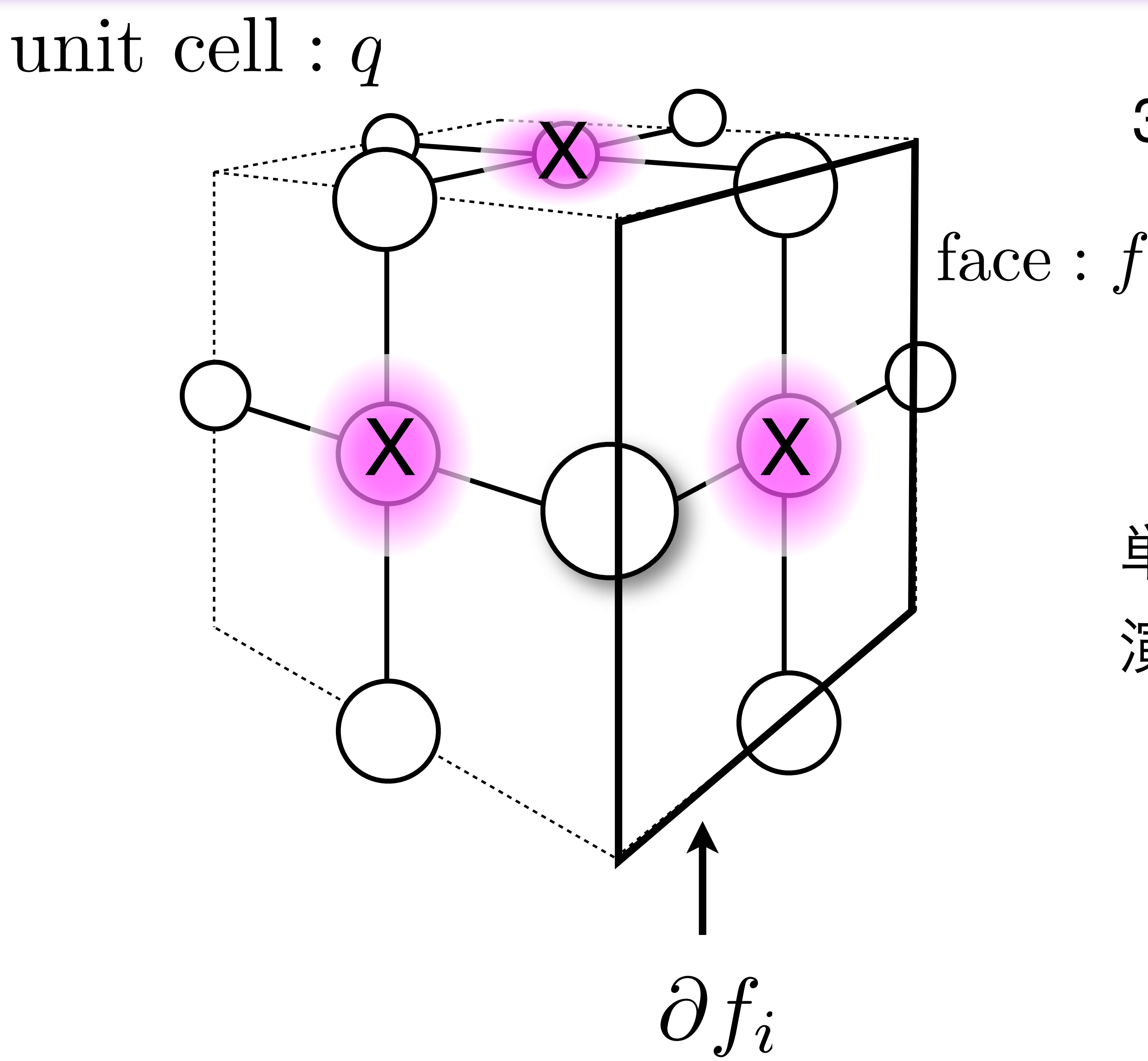
3D clusterのstabilizer演算子：

$$K(f_i) = X_{f_i} \prod_{e_l \in \partial f_i} Z_{e_l}$$

単位立方上の面qubitに対するスタビライザー演算子の積をとると、

$$\prod_{f_i \in \partial q_j} K(f_i) = \prod_{f_i \in \partial q_j} X_{f_i} \equiv S(q_j)$$

Topological protection in *V*



3D clusterのstabilizer演算子：

$$K(f_i) = X_{f_i} \prod_{e_l \in \partial f_i} Z_{e_l}$$

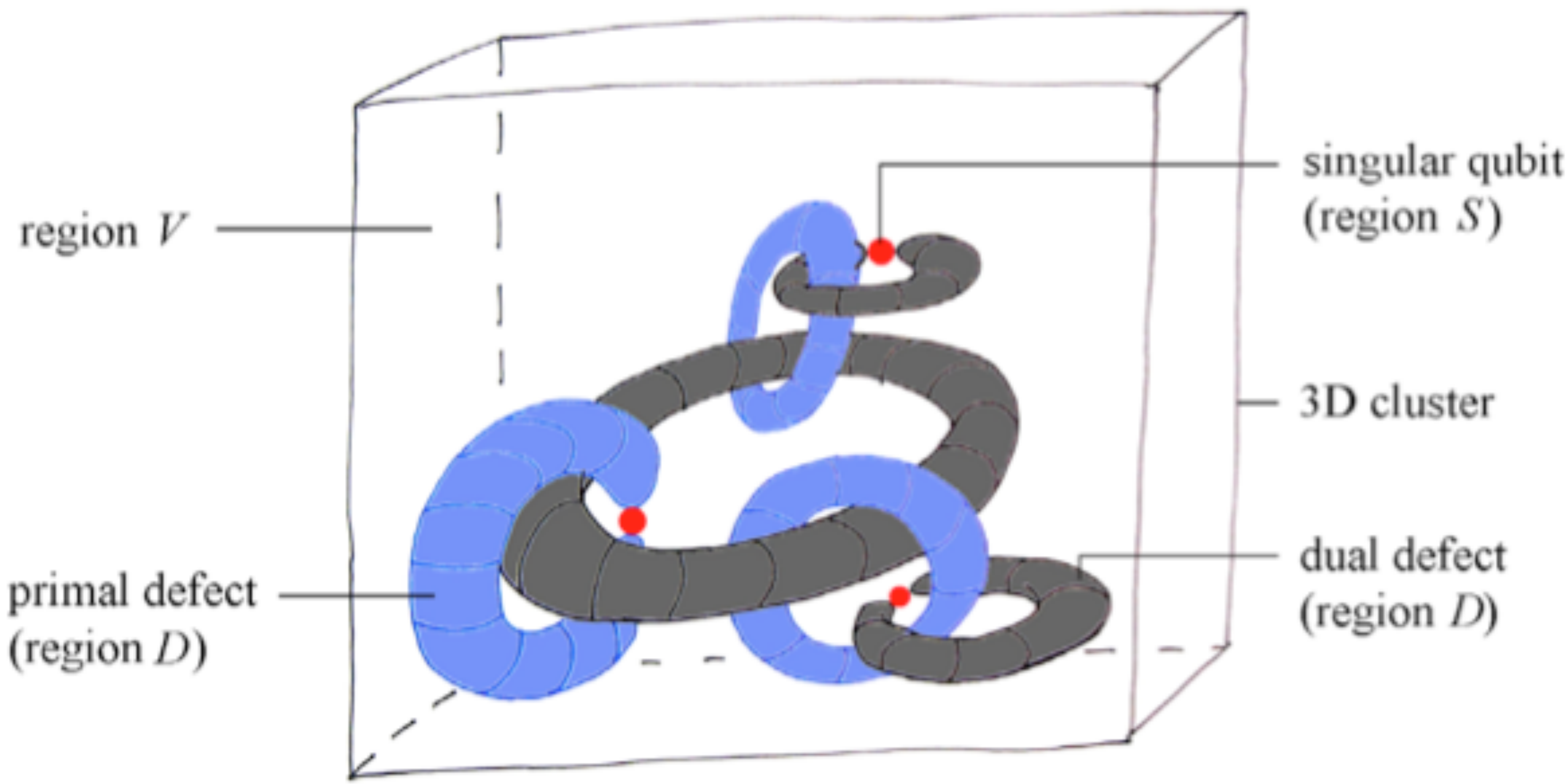
単位立方上の面qubitに対するスタビライザー演算子の積をとると、

$$\prod_{f_i \in \partial q_j} K(f_i) = \prod_{f_i \in \partial q_j} X_{f_i} \equiv S(q_j)$$

→RHGクラスター状態に対するX基底での測定結果の単位立方上での積は+1

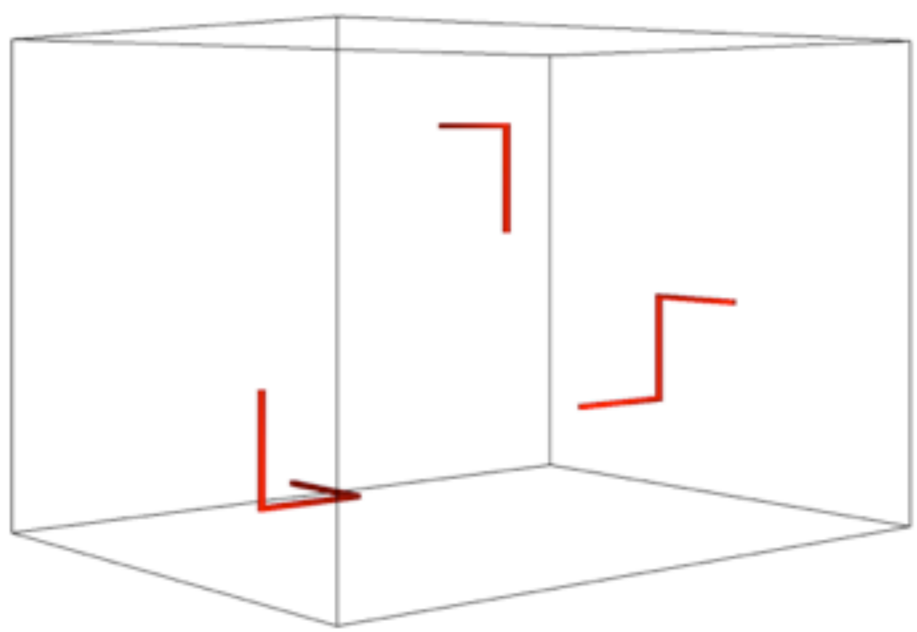
Defect and singular qubit injection

（お絵描きする）



Raussendorf-Harrington-Goyal,
Annals Phys. **321**, 2242 (2006).

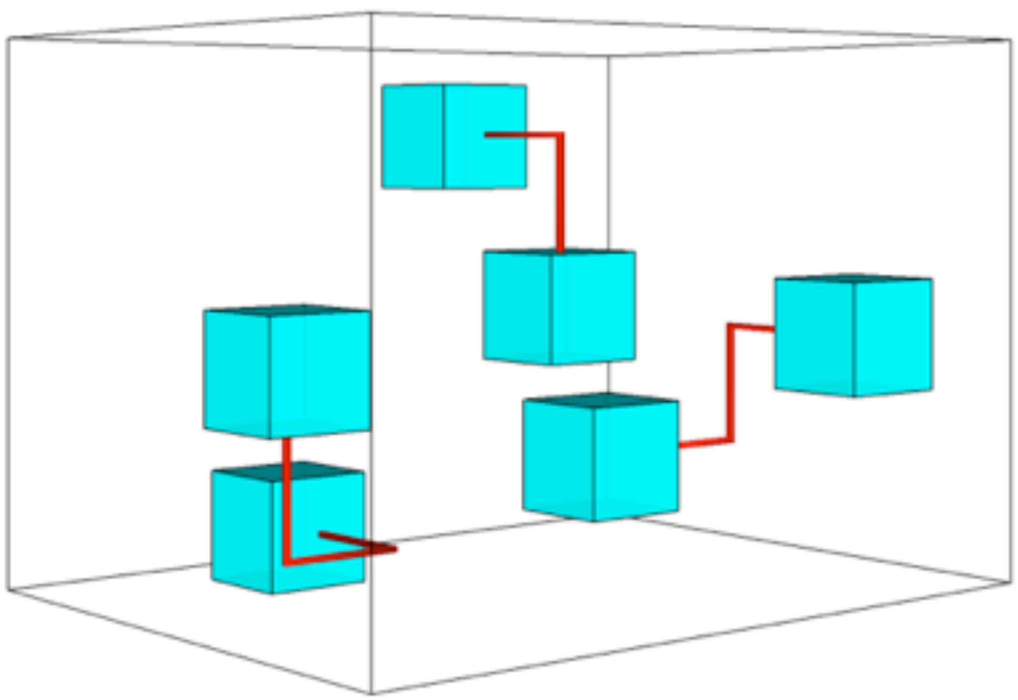
Error syndrome



エラーは primal , dual lattice上の 1-chain

→ $c_1^e, \bar{c}_1^e$

Error syndrome



エラーは primal , dual lattice上の 1-chain

→ $c_1^e, \bar{c}_1^e$

エラー鎖の端点となる単位立方上でエラーが検出される.

→ $\partial \bar{c}_1^e = \bar{c}_0^e = c_3^e$

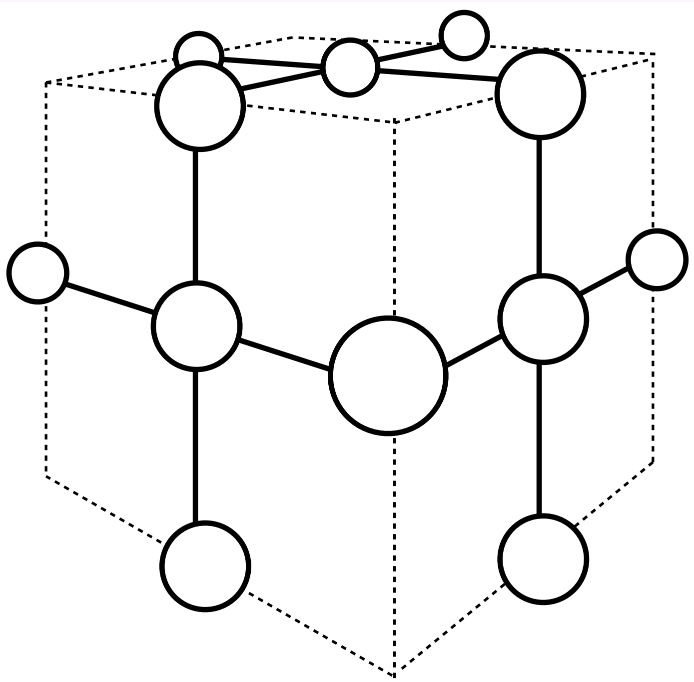
$$\text{for all } q_j \in c_3^e S(q), \quad [S(q_j), Z(\bar{c}_1^e)] \neq 0$$

3D lattice上の MWPM で誤り訂正できる.

Noise model & threshold

surface code 状態 (独立X,Zエラー)	10.3 % (minimum weight-perfect match) 10.9 % (random bond Ising model)	Dennis *et al*., J. Math. Phys. **49**, 4452 (2002). Ohzeki, PRE **79**, 021129 (2009).
(depolarizing error)	18.9 % (random bond Ising model)	Bombin *et al*., PRX **2**, 021004 (2012).
現象論的のノイズモデル (data qubit と syndrome qubit の独立にエラーが生じる)	2.9% (minimum weight-perfect match) 3.3% (random lattice gauge theory)	Wang-Harrington-Preskill, Ann. Phys. **303**, 31 (2003). Ohno *et al*., Nuc. Phys. B **697**, 462 (2004).
量子回路的ノイズモデル (それぞれの操作（演算）ごとにエラーが生じる)	2D nearest-neighbor gate: 0.75% (minimum weight-perfect match) 3D cluster state: 0.67% (minimum weight-perfect match)	Raussendorf-Harrington-Goyal, NJP **9**, 199 (2007). Raussendorf-Harrington-Goyal, Ann. Phys. **321**, 2242 (2006).

誤り訂正限界とランダムゲージ模型

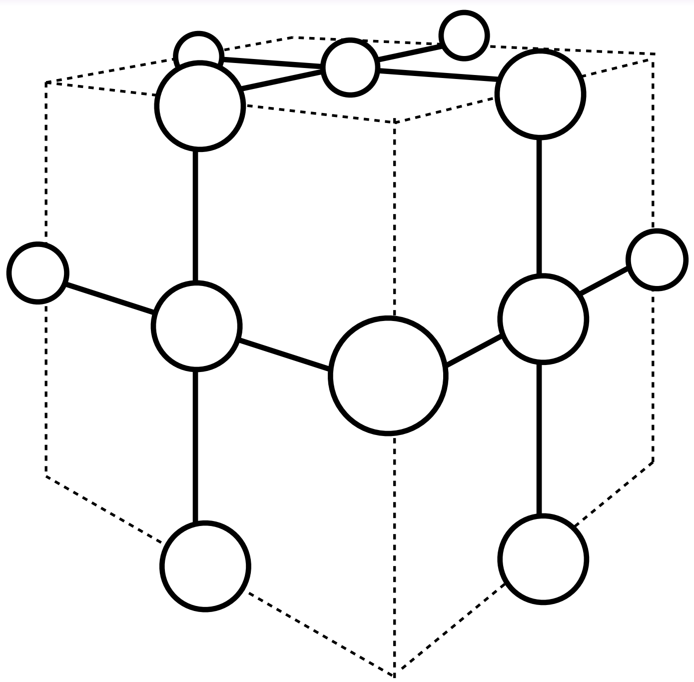


面qubitと辺qubit上のエラーは相関をもたないので、前者だけを考える.

Vにおける topological protection の限界を調べる：

エラー鎖 C（dual 1-chain）が発生したとする.

誤り訂正限界とランダムゲージ模型



面qubitと辺qubit上のエラーは相関をもたないので、前者だけを考える.

Vにおける topological protection の限界を調べる：

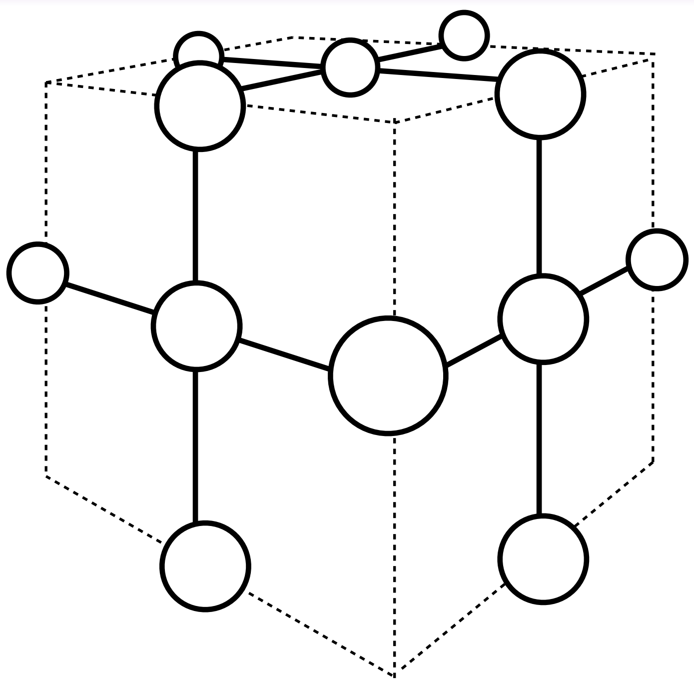
エラー鎖 C（dual 1-chain）が発生したとする.

エラー鎖 C と同じsyndrome を持つエラー鎖 C',

$$C' = \sum_l z_l^{C'} \bar{e}_l$$

（$\partial C = \partial C'$）を推定する.

誤り訂正限界とランダムゲージ模型



面qubitと辺qubit上のエラーは相関をもたないので、前者だけを考える.

Vにおける topological protection の限界を調べる：

エラー鎖 C（dual 1-chain）が発生したとする.

エラー鎖 C と同じsyndrome を持つエラー鎖 C',

$$C' = \sum_l z_l^{C'} \bar{e}_l$$

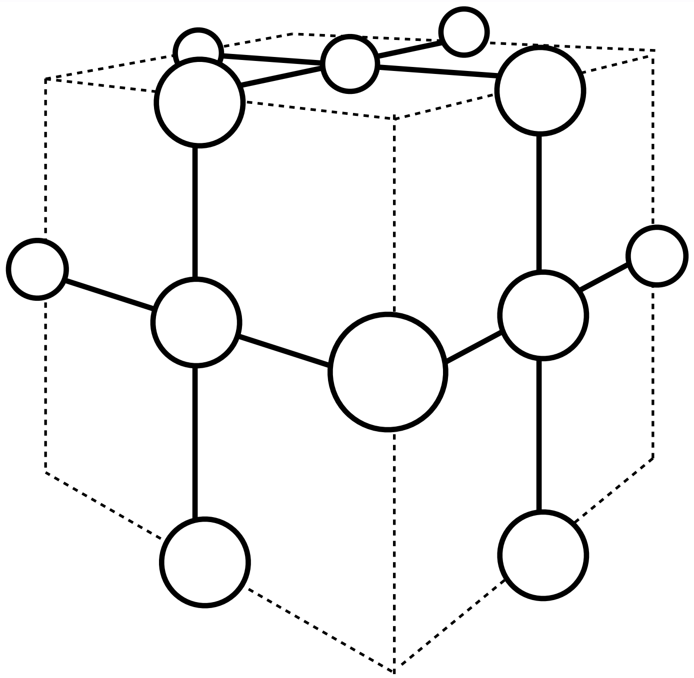
（$\partial C = \partial C'$）を推定する.

その出現確率は,

$$\begin{aligned} p(C'|\partial C) &= (1-p)^{|\bar{E}|} \prod_l \left(\frac{p}{1-p} \right)^{z_l^{C'}} \Big|_{\partial C = \partial C'} \\ &= \mathcal{N} \exp \left[\beta' J \sum_l u_l^{C'} \right] \Big|_{\partial C = \partial C'} \end{aligned}$$

ここで $e^{-\beta' J} = \sqrt{\dfrac{p'}{1-p'}}$ ←推定のためのパラメータ ， $u_l^{C'} = (-1)^{z_l^{C'}}$ ←スピン変数と見る

誤り訂正限界とランダムゲージ模型

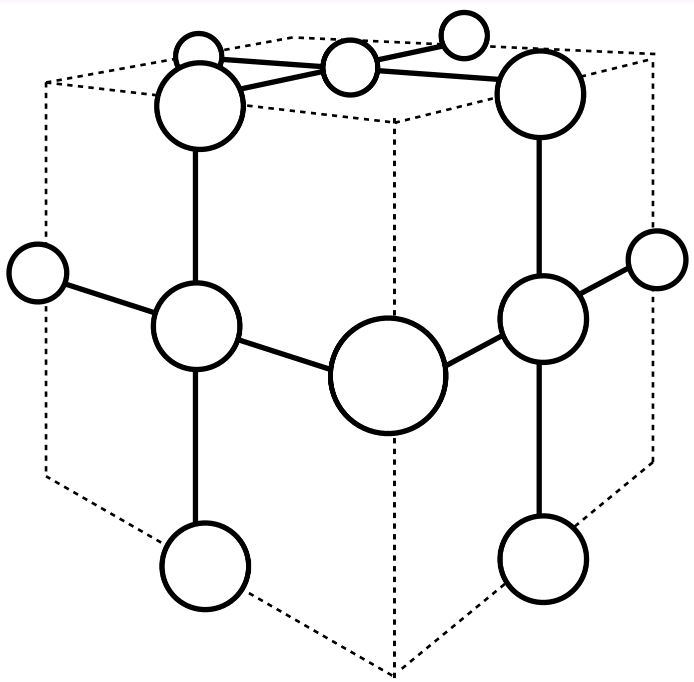


面qubitと辺qubit上のエラーは相関をもたないので、前者だけを考える.

エラー鎖Cとシンドロームを共有するエラー鎖C'の出現確率：

$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^{C'}} \Big|_{\partial C = \partial C'},$$

誤り訂正限界とランダムゲージ模型



面qubitと辺qubit上のエラーは相関をもたないので、前者だけを考える.

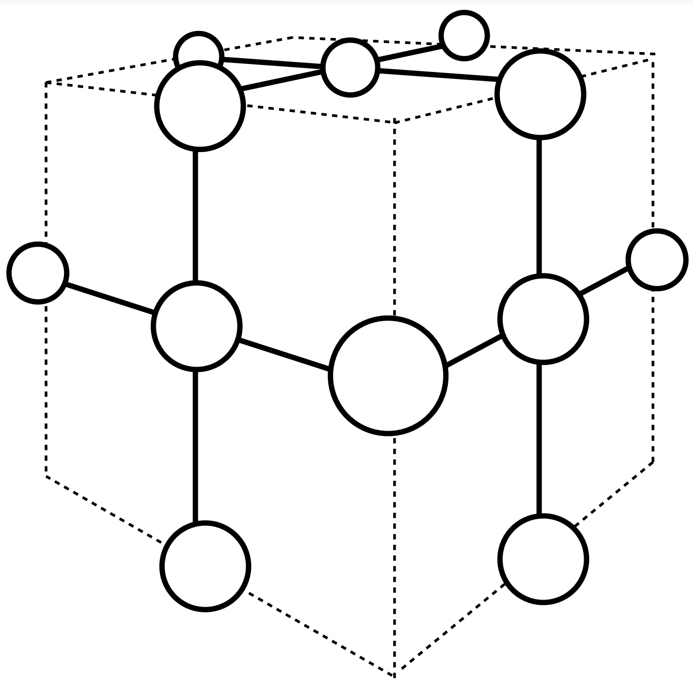
エラー鎖Cとシンドロームを共有するエラー鎖C'の出現確率：

$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^{C'}} \Big|_{\boxed{\partial C = \partial C'}},$$

$$C + C' = L$$

ループ(trivial cycle)

誤り訂正限界とランダムゲージ模型



面qubitと辺qubit上のエラーは相関をもたないので、前者だけを考える.

エラー鎖Cとシンドロームを共有するエラー鎖C’の出現確率：

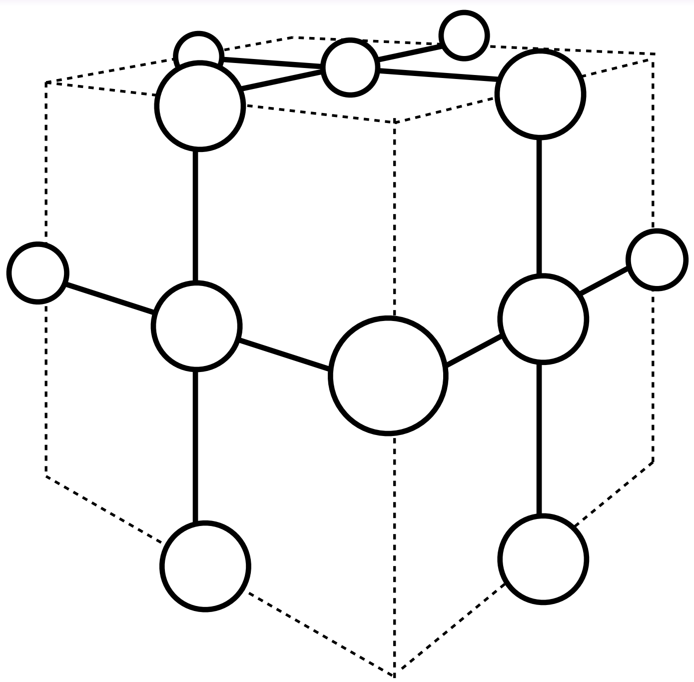
$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^{C'}} \Big|_{\boxed{\partial C = \partial C'}},$$

$$C + C' = L$$

ループ(trivial cycle)

$$u_f^{C} u_f^{C'} = u_f^{L}$$

誤り訂正限界とランダムゲージ模型



面qubitと辺qubit上のエラーは相関をもたないので、前者だけを考える.

エラー鎖Cとシンドロームを共有するエラー鎖C'の出現確率：

$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^{C'}} \Big|_{\boxed{\partial C = \partial C'}},$$

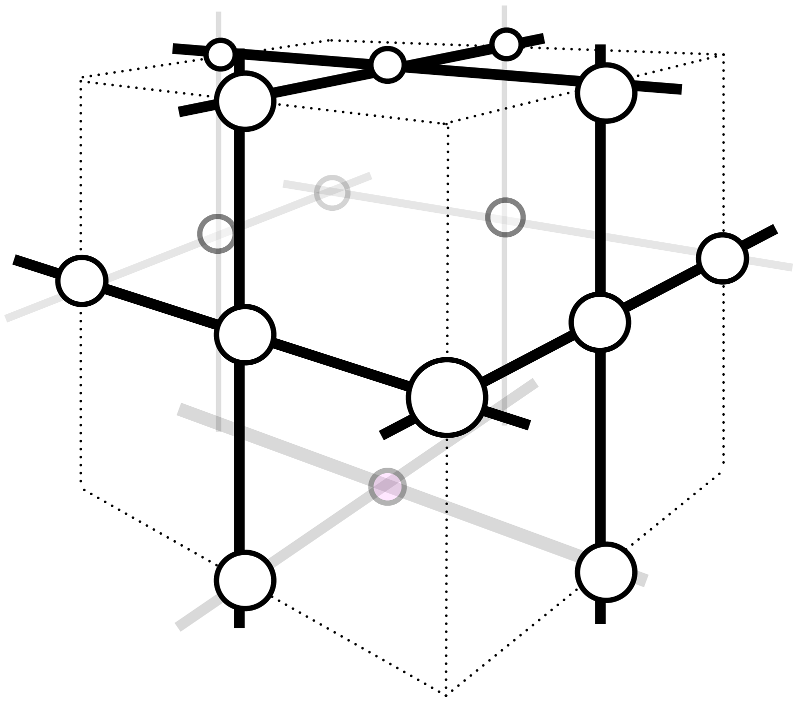
$$C + C' = L$$

ループ(trivial cycle)

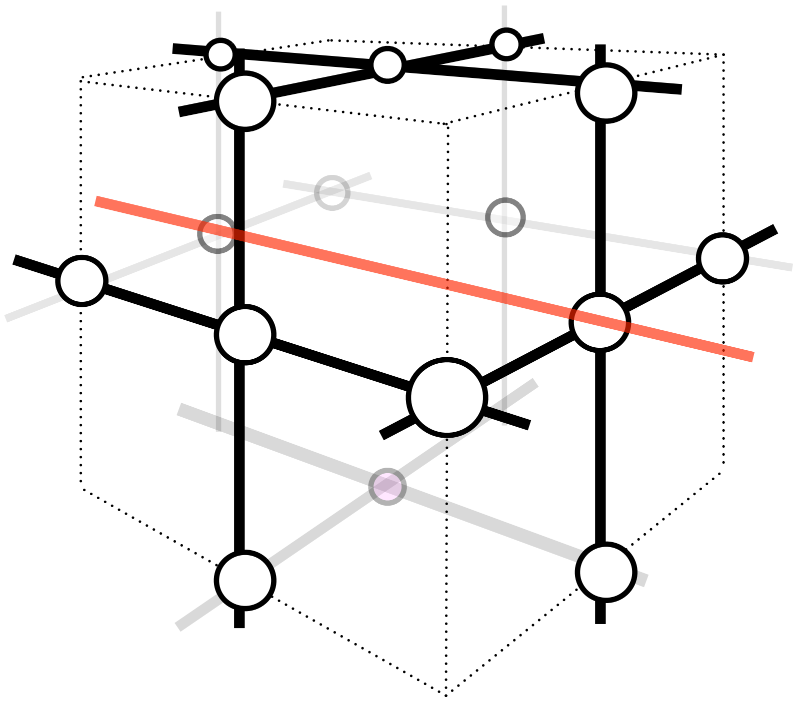
$$u_f^{C} u_f^{C'} = u_f^{L}$$

$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^{C} u_f^{L}} \Big|_{\partial L = 0},$$

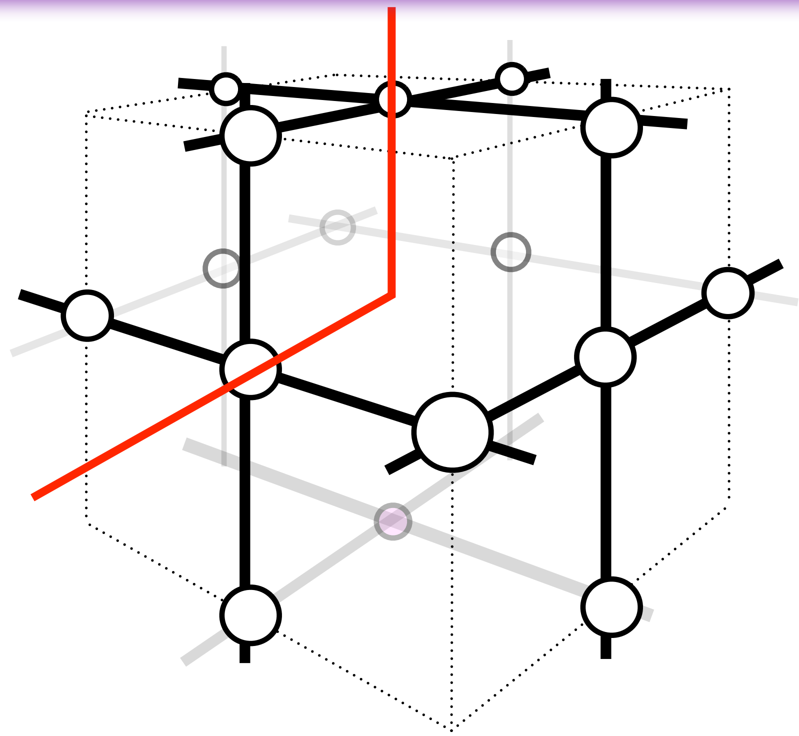
誤り訂正限界とランダムゲージ模型



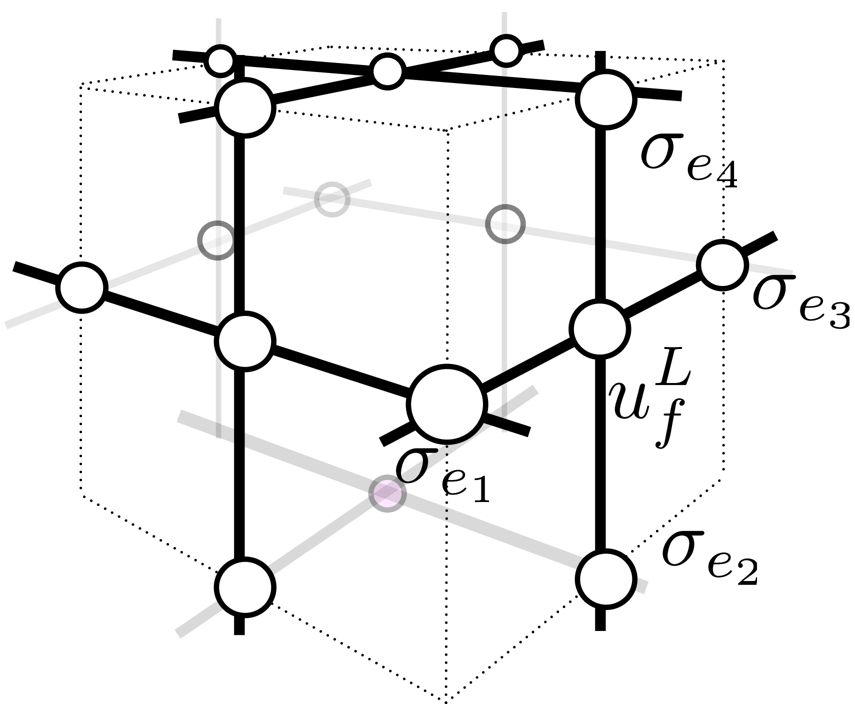
誤り訂正限界とランダムゲージ模型



誤り訂正限界とランダムゲージ模型



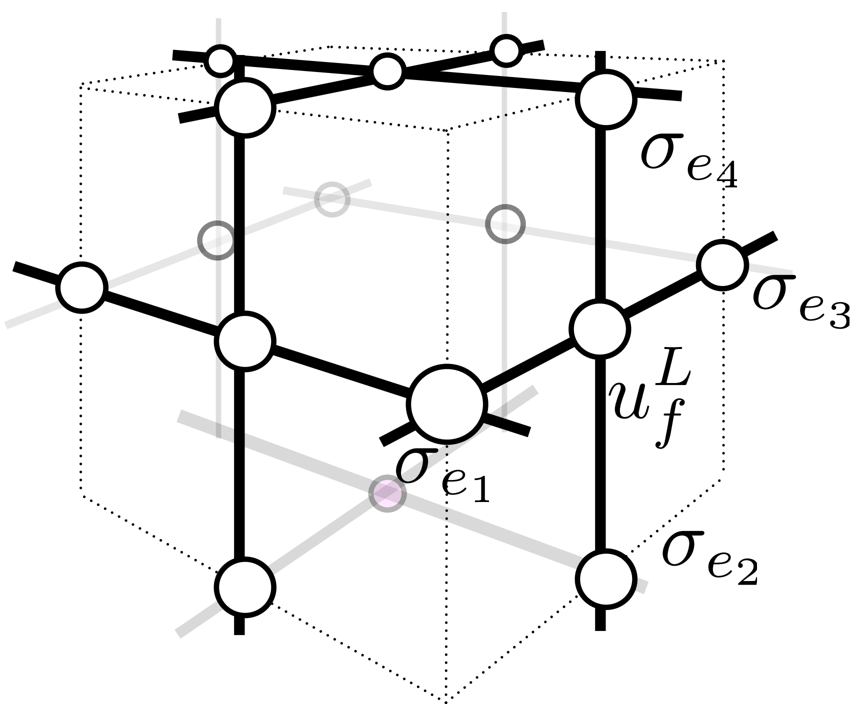
誤り訂正限界とランダムゲージ模型



ゲージスピンの導入：$u_f^L = \prod_{e \in f} \sigma_e \equiv P_f$

→自動的に$\partial L = 0$ つまり$\prod_{f \in q} P_f = 1$ が満たされる.

誤り訂正限界とランダムゲージ模型



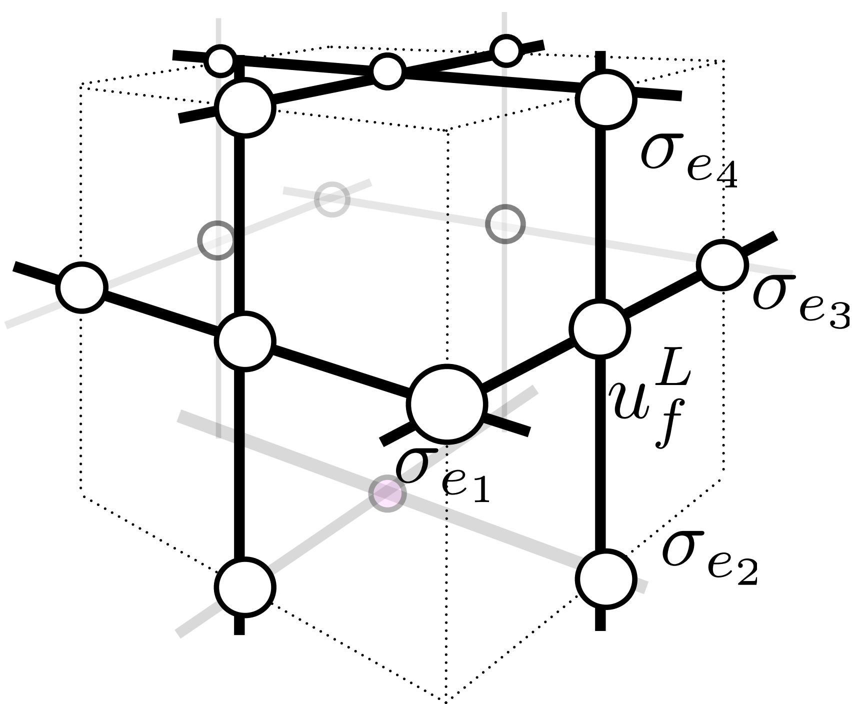
ゲージスピンの導入：$u_f^L = \prod_{e \in f} \sigma_e \equiv P_f$

→自動的に$\partial L = 0$ つまり$\prod_{f \in q} P_f = 1$ が満たされる.

エラー鎖 C'=C+L の出現確率：

$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^C P_f},$$

誤り訂正限界とランダムゲージ模型



ゲージスピンの導入：$u_f^L = \prod_{e \in f} \sigma_e \equiv P_f$

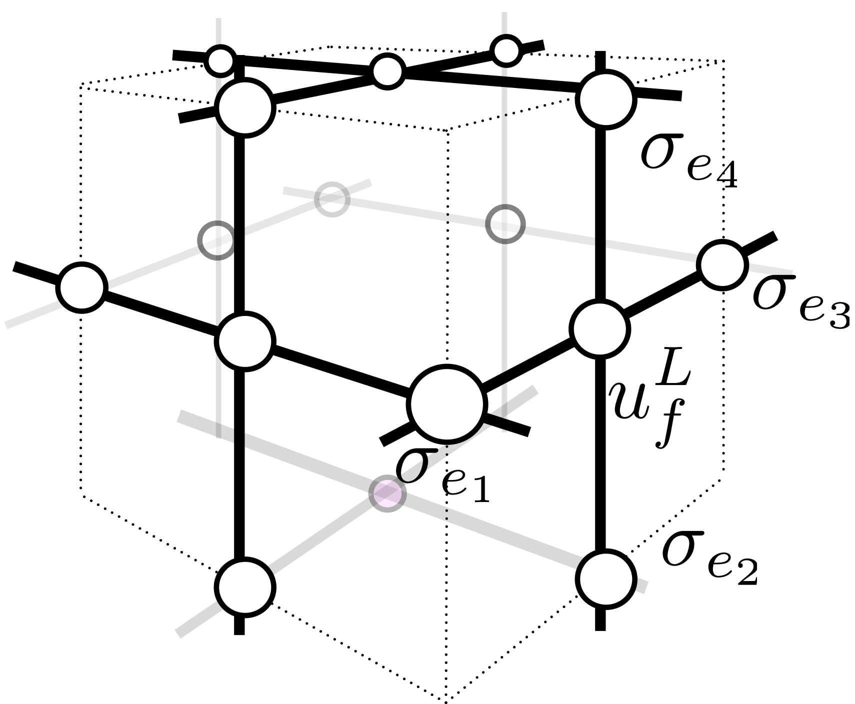
→自動的に$\partial L = 0$ つまり$\prod_{f \in q} P_f = 1$ が満たされる.

エラー鎖 C'=C+L の出現確率：

$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^C P_f},$$

→random plaquette Z2 gauge model (RPGM)

誤り訂正限界とランダムゲージ模型



ゲージスピンの導入：$u_f^L = \prod_{e \in f} \sigma_e \equiv P_f$

→自動的に$\partial L = 0$ つまり$\prod_{f \in q} P_f = 1$ が満たされる.

エラー鎖 C'=C+L の出現確率：

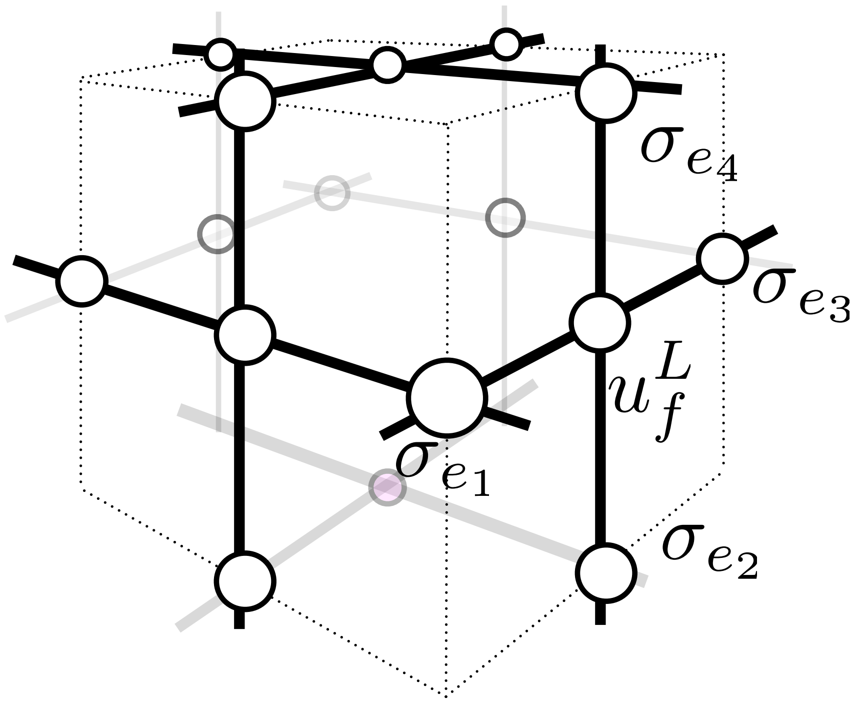
$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^C P_f},$$

→random plaquette Z2 gauge model (RPGM)

→ $p = p'$（西森線）の場合が最適誤り訂正.

H. Nishimori, Prog. Theor. Phys. **66**, 1169 (1981)

誤り訂正限界とランダムゲージ模型



ゲージスピンの導入：$u_f^L = \prod_{e \in f} \sigma_e \equiv P_f$

→自動的に$\partial L = 0$ つまり$\prod_{f \in q} P_f = 1$ が満たされる.

エラー鎖 C'=C+L の出現確率：

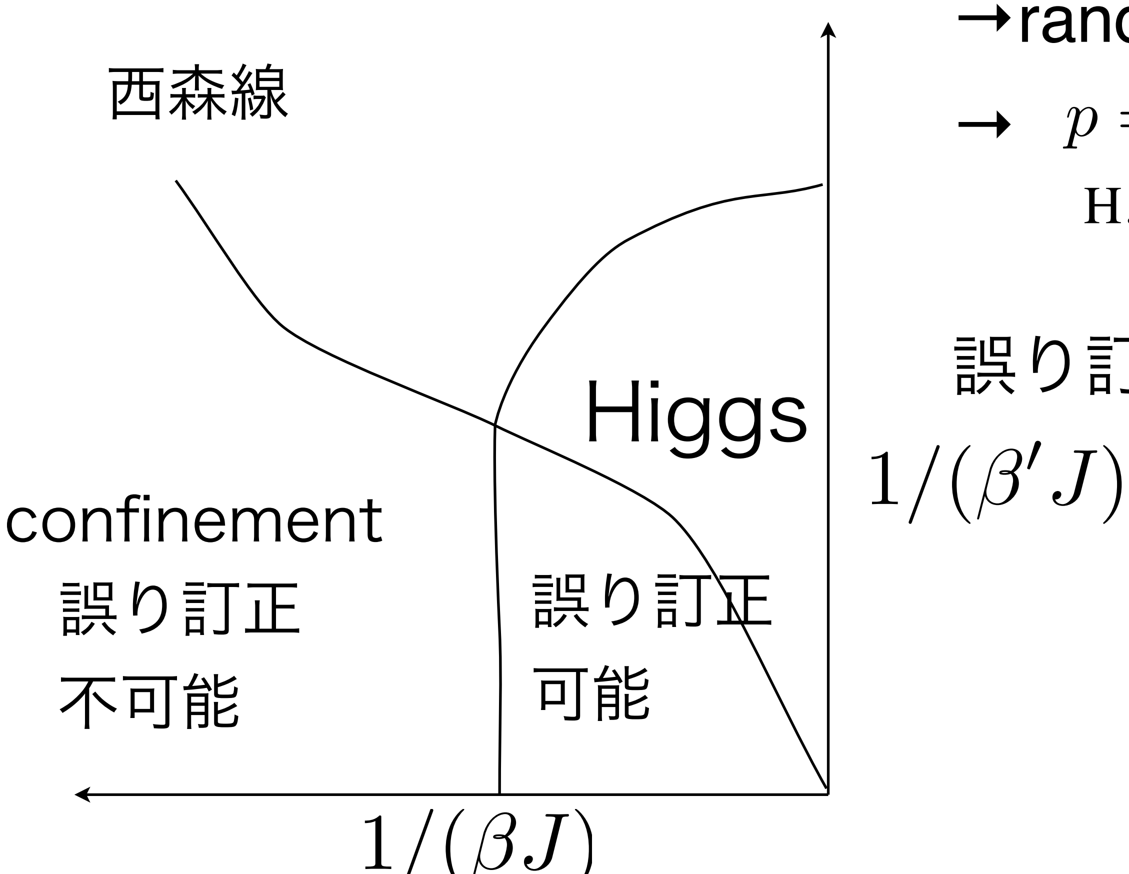
$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^C P_f},$$

→random plaquette Z2 gauge model (RPGM)

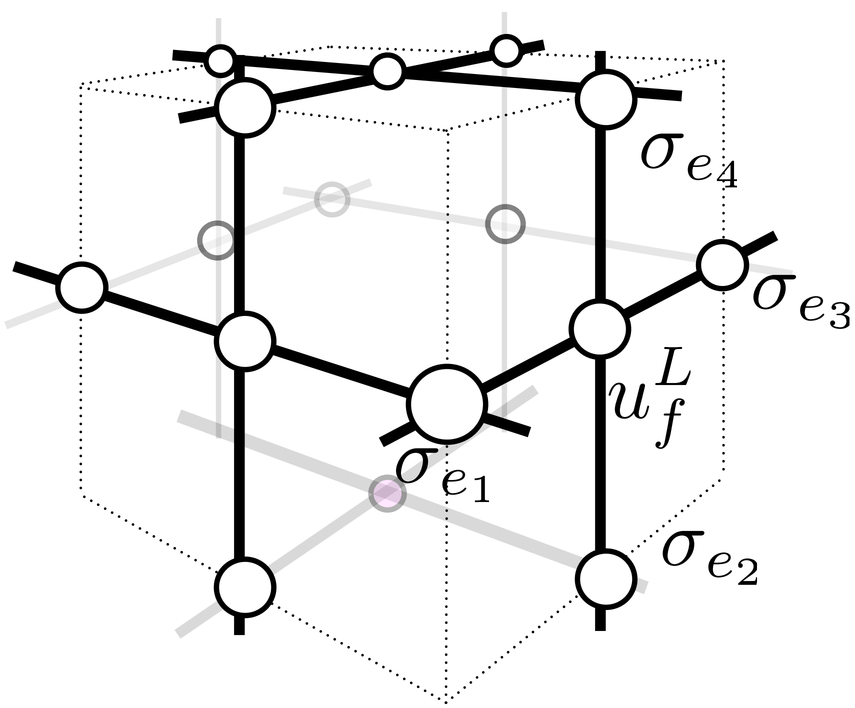
→ $p = p'$（西森線）の場合が最適誤り訂正.

H. Nishimori, Prog. Theor. Phys. **66**, 1169 (1981)

誤り訂正限界は ヒッグス-閉じ込め転移点：



誤り訂正限界とランダムゲージ模型



ゲージスピンの導入：$u_f^L = \prod_{e \in f} \sigma_e \equiv P_f$

→自動的に$\partial L = 0$ つまり$\prod_{f \in q} P_f = 1$ が満たされる.

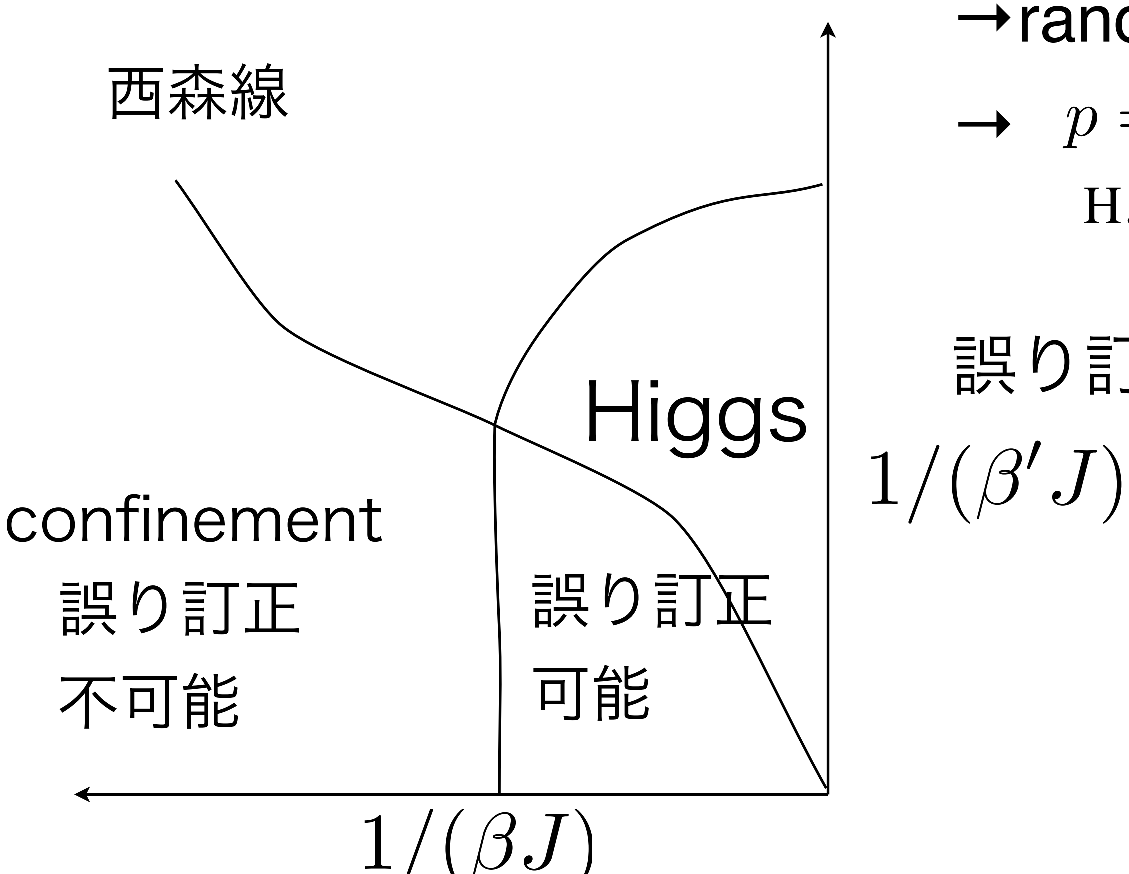
エラー鎖 C'=C+L の出現確率：

$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^C P_f},$$

→random plaquette Z2 gauge model (RPGM)

→ $p = p'$（西森線）の場合が最適誤り訂正.

H. Nishimori, Prog. Theor. Phys. **66**, 1169 (1981)

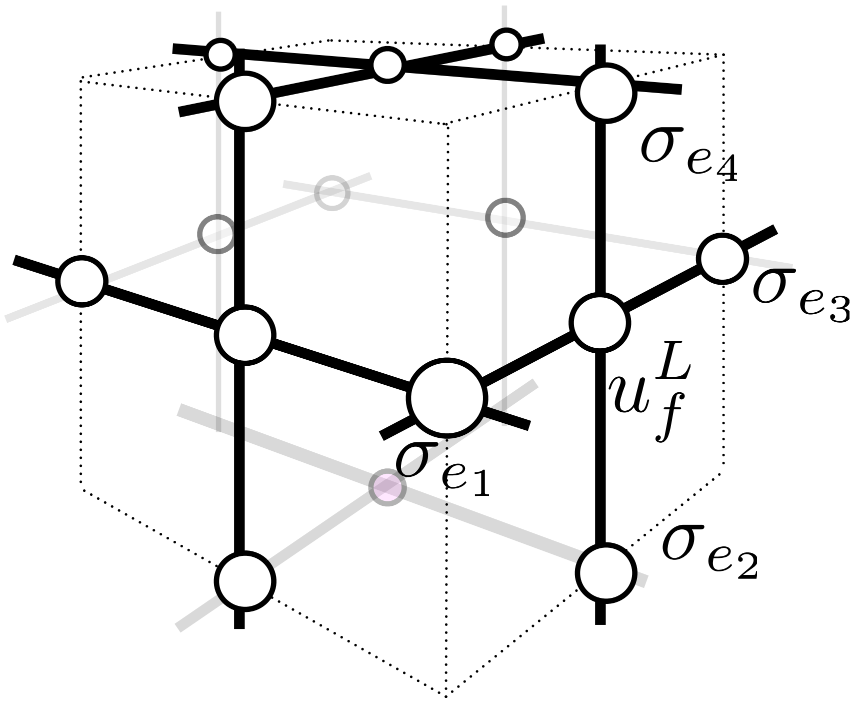


誤り訂正限界は ヒッグス-閉じ込め転移点：

ゼロ温度：0.57 (2.9%)

Wang-Harrington-Preskill, Ann. Phys. **303**, 31 (2003).

誤り訂正限界とランダムゲージ模型



ゲージスピンの導入：$u_f^L = \prod_{e\in f} \sigma_e \equiv P_f$

→自動的に$\partial L = 0$ つまり$\prod_{f\in q} P_f = 1$ が満たされる.

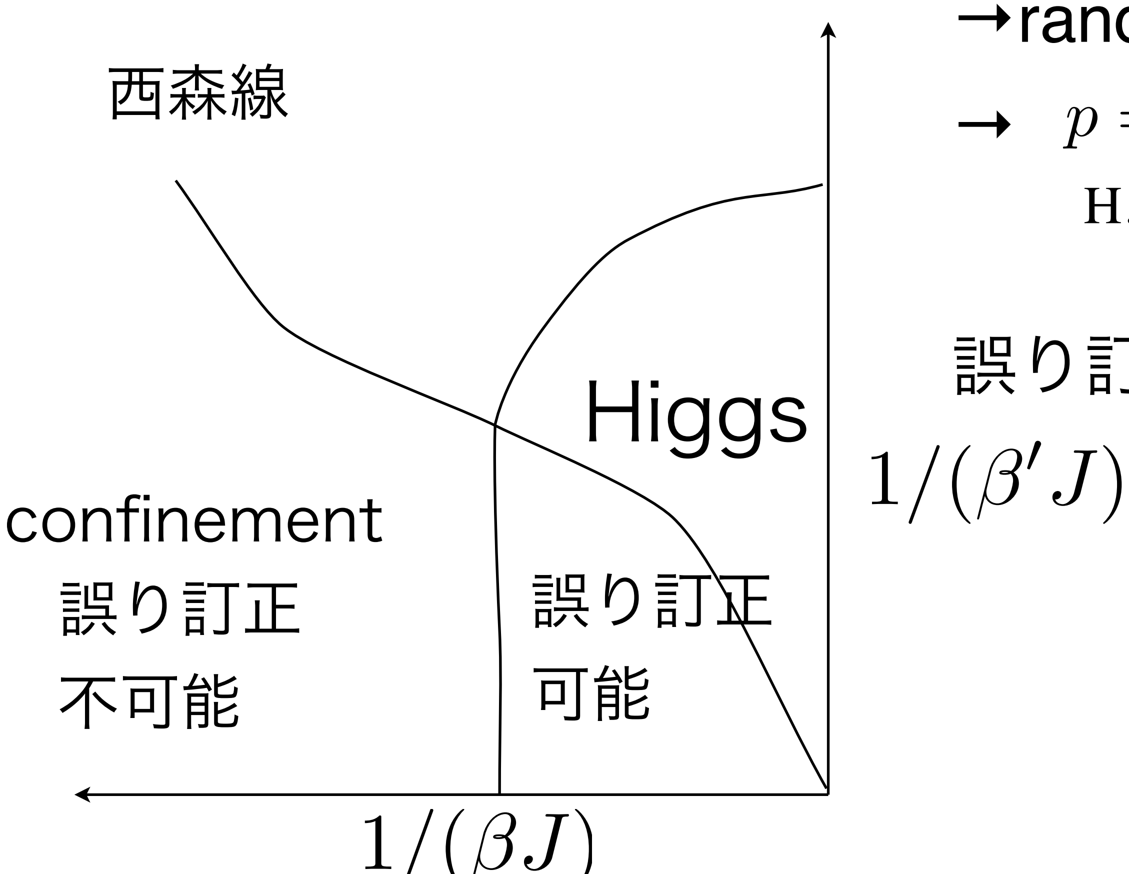
エラー鎖 C'=C+L の出現確率：

$$p(C'|\partial C) = \mathcal{N} e^{\beta' J \sum_f u_f^C P_f},$$

→random plaquette Z2 gauge model (RPGM)

→ $p = p'$（西森線）の場合が最適誤り訂正.

H. Nishimori, Prog. Theor. Phys. **66**, 1169 (1981)



誤り訂正限界は ヒッグス-閉じ込め転移点：

ゼロ温度：0.57 (2.9%)

Wang-Harrington-Preskill, Ann. Phys. **303**, 31 (2003).

多重臨界点：0.59 (3.3%)

Ohno *et al*., Nuc. Phys. B **697**, 462 (2004).

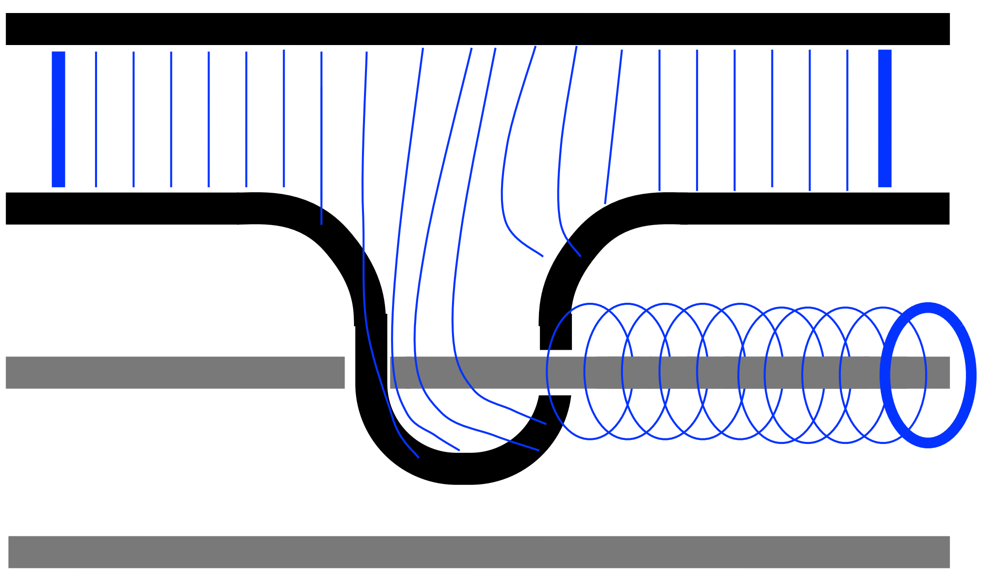
目次

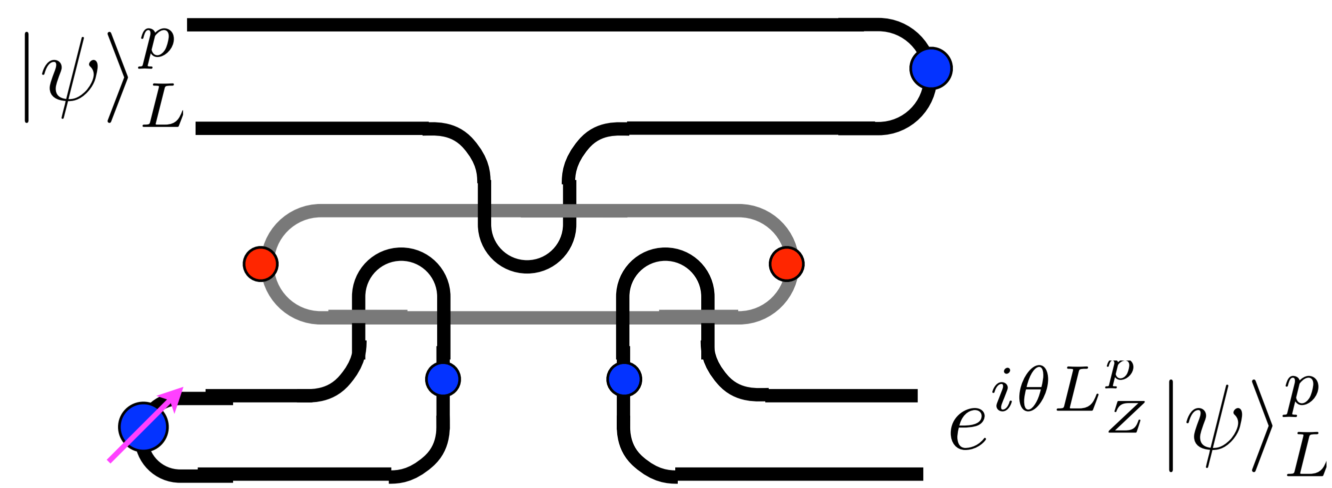
- ✓ 第一章：量子系の取り扱い
- ✓ 第二章：グラフ状態とMBQC
- ✓ 第三章：量子誤り訂正とフォールトトレランス
- ✓ 第四章：Surface符号上での量子誤り訂正
- ✓ 第五章：トポロジカル量子計算（回路モデルの視点から）
- ✓ 第六章：トポロジカル量子計算（MBQCの視点から）

目次

✔ 第一章：量子系の取り扱い

✔ 第二章：グラフ状態とMBQC





✔ 第六章：トポロジカル量子計算（MBQCの視点から）

Noise model & threshold

量子誤り訂正符号

- H. Bombin & M.Delgado, PRL '06 '07
- T. Stace, S. Barret PRL '09 '10
- KF &Y. Tokunaga, PRA '12

Condensed matter physics

トポロジカル秩序

- B. Yoshida Ann. Phys. '11
- S. Bravyi & J. Haah, arXiv '11
- J. Haah, arXiv '11

スピングラス理論

RBIM, RPGT

- M. Ohzeki PRE '09
- M. Ohzeki arXiv '12
- H. Bombin *et al*. PRX '12

Surface code by **Kitaev '97**

トポロジカルフォールトトレラント量子計算

by **Raussendorf et al. '06**

熱平衡状態を用いたMBQC

- Raussendorf *et al*., PRA '05
- Barrett *et al*. PRA '09
- Y. Li *et al*., PRL '11
- KF & T. Morimae PRA '12
- KF, Y. Nakata, M. Ohzeki, M. Mio, arXiv '12

ブラインド量子計算

- T. Morimae & KF NatComm '11
- J. F. Fitzsimons & E. Kashefi '12

分散型量子計算

- Y. Li *et al*., PRL '10
- KF & Y. Tokunaga, PRL '10
- KF *et al*., arXiv '12
- Y. Li & S. Benjamin, arXiv '12

Noise model & threshold

Condensed matter physics

トポロジカル秩序

- B. Yoshida Ann. Phys. '11
- S. Bravyi & J. Haah, arXiv '11
- J. Haah, arXiv '11

量子誤り訂正符号

- H. Bombin & M.Delgado, PRL '06 '07
- T. Stace, S. Barret PRL '09 '10
- KF &Y. Tokunaga, PRA '12

スピングラス理論

RBIM, RPGT

- M. Ohzeki PRE '09
- M. Ohzeki arXiv '12
- H. Bombin *et al*. PRX '12

Surface code by **Kitaev '97**

トポロジカルフォールトトレラント量子計算

by **Raussendorf et al. '06**

ブラインド量子計算

- T. Morimae & KF NatComm '11
- J. F. Fitzsimons & E. Kashefi '12

分散型量子計算

- Y. Li *et al*., PRL '10
- KF & Y. Tokunaga, PRL '10
- KF *et al*., arXiv '12
- Y. Li & S. Benjamin, arXiv '12

熱平衡状態を用いたMBQC

- Raussendorf *et al*., PRA '05
- Barrett *et al*. PRA '09
- Y. Li *et al*., PRL '11
- KF & T. Morimae PRA '12
- KF, Y. Nakata, M. Ohzeki, M. Mio, arXiv '12

ご清聴ありがとうございました。