# The Pauli and Clifford Groups

- The Pauli group is the group of all tensor products of single-qubit Pauli operators. The elements of the Pauli group are called (multi-qubit) Pauli operators.
- The Clifford group is the group of all unitary operators that leave the Pauli group invariant under conjugation. The elements of the Clifford group are called Clifford operators.

---

Make sure that the Q3 package is loaded.

In[•]:=
```
<< Q3`
```

---

Throughout this document, symbol `S` will be used to denote qubits and Pauli operators on them.

In[•]:=
```
Let[Qubit, S]
Let[Complex, c]
```

---



---

# The Pauli Group

## The Full Pauli Group

**The full Pauli group on a single qubit** consists of the single-qubit Pauli operators. Enlisting the elements, it consists of 16 elements, $\tilde{\mathcal{P}}(1) = \{\pm I, \pm i\, I,\ \pm X, \pm i\, X,\ \pm Y, \pm i\, Y,\ \pm Z, \pm i\, Z\}$, with $I$, $X$, $Y$, and $Z$ associated with the 2×2 matrices

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

```
In[•]:= grp = FullPauliGroup[S]
        elm = GroupElements[grp];
        elm // PauliForm
Out[•]= FullPauliGroup[{S}]
Out[•]= {I, X, Z, Y, -I, -X, -Z, -Y, ⅈ I, ⅈ X, ⅈ Z, ⅈ Y, -ⅈ I, -ⅈ X, -ⅈ Z, -ⅈ Y}
```

The additional phase factors $\pm 1$ and $\pm i$ are required because the product of two Pauli operators gives another Pauli operator accompanied by factor $\pm i$. Otherwise, the underlying set is not closed under the group multiplication.

**The full Pauli group $\tilde{\mathcal{P}}(n)$ on $n$ qubits** is given by

$$\tilde{\mathcal{P}}(n) = \{\pm 1, \pm i\} \times \{I, X, Y, Z\}^{\otimes n}.$$

Elements of the full Pauli group are called (multi-qubit) **Pauli operators**.

```
In[•]:= grp = FullPauliGroup[S@{1, 2}]
        elm = GroupElements[grp];
        elm // PauliForm
Out[•]= FullPauliGroup[{S1, S2}]
Out[•]= {I⊗I, X⊗I, Z⊗I, Y⊗I, I⊗X, X⊗X, Z⊗X, Y⊗X, I⊗Z, X⊗Z, Z⊗Z, Y⊗Z,
         I⊗Y, X⊗Y, Z⊗Y, Y⊗Y, -(I⊗I), -(X⊗I), -(Z⊗I), -(Y⊗I), -(I⊗X),
         -(X⊗X), -(Z⊗X), -(Y⊗X), -(I⊗Z), -(X⊗Z), -(Z⊗Z), -(Y⊗Z), -(I⊗Y),
         -(X⊗Y), -(Z⊗Y), -(Y⊗Y), ⅈ I⊗I, ⅈ X⊗I, ⅈ Z⊗I, ⅈ Y⊗I, ⅈ I⊗X, ⅈ X⊗X,
         ⅈ Z⊗X, ⅈ Y⊗X, ⅈ I⊗Z, ⅈ X⊗Z, ⅈ Z⊗Z, ⅈ Y⊗Z, ⅈ I⊗Y, ⅈ X⊗Y, ⅈ Z⊗Y, ⅈ Y⊗Y,
         -ⅈ I⊗I, -ⅈ X⊗I, -ⅈ Z⊗I, -ⅈ Y⊗I, -ⅈ I⊗X, -ⅈ X⊗X, -ⅈ Z⊗X, -ⅈ Y⊗X,
         -ⅈ I⊗Z, -ⅈ X⊗Z, -ⅈ Z⊗Z, -ⅈ Y⊗Z, -ⅈ I⊗Y, -ⅈ X⊗Y, -ⅈ Z⊗Y, -ⅈ Y⊗Y}

In[•]:= GroupOrder[grp]
Out[•]= 64
```

The number of elements quickly increases with the number of qubits, and it is more convenient to describe the full Pauli group in terms of **generators**. The Pauli group on $n$ qubits is generated by tensor products of the Pauli operators acting non-trivially on one and only one qubit. For example, the full Pauli group on two qubits is given by

$$\tilde{\mathcal{P}}(n) = \langle\{X \otimes I,\ Y \otimes I,\ Z \otimes I,\ I \otimes X,\ I \otimes Y,\ I \otimes Z\}\rangle.$$

```
In[•]:= gnr = GroupGenerators[grp];
        PauliForm[gnr]
Out[•]= {X⊗I, Y⊗I, Z⊗I, I⊗X, I⊗Y, I⊗Z}
```

## Properties of Pauli Operators

Pauli operators (i.e., elements in the full Pauli group) have two extremely useful properties that **eventually determine the group-theoretical structure of the stabilizers and stabilizer codes**.

1. Any two elements of the full Pauli group either commute or anti-commute with each other.

$$\sigma^{\mu}\,\sigma^{\nu} = i\,\epsilon_{\mu\nu\lambda}\,\sigma^{\lambda}$$

$$\{\sigma^\mu, \sigma^\nu\} = 2\,\delta_{\mu\nu}\,I$$

2. Every element of the Pauli group squares to $\pm I$.
   The elements that square to $I$ have eigenvalues $\pm 1$ whereas those that square to $-I$ have eigenvalues $\pm i$.

### Demonstration: Property 1

### Demonstration: Property 2

## The Pauli Group

In most applications of the Pauli group, one can **ignore the phase factors ±1 and ±i** in the elements of the full Pauli group. Mathematically, this can be achieved by considering the ***quotient group***

$$\mathcal{P}(n) := \tilde{\mathcal{P}}(n) \big/ \mathcal{Z}(n)$$

where $\mathcal{Z}(n) := \{\pm I^{\otimes n},\ \pm i\, I^{\otimes n}\}$ is the center (the largest Abelian invariant subgroup) of $\tilde{\mathcal{P}}(n)$.

This quotient group is called the ***Pauli group***. As a quotient group, the Pauli group is a group of ***cosets***, and the elements of the same meaning belong to the same coset.

For example, in the single-qubit Pauli group, there are four cosets

$$\mathcal{P}(1) = \{\{\pm I, \pm i\, I\},\ \{\pm X, \pm i\, X\},\ \{\pm Y, \pm i\, Y\},\ \{\pm Z, \pm i\, Z\}\}.$$

Belonging to the same coset $X\mathcal{Z}$, $\pm X$ and $\pm i\, X$ are not distinguished any longer.

In Q3, cosets in the Pauli group are denoted by their representatives.

```
In[•]:= grp = PauliGroup[S@{1, 2}]
Out[•]= PauliGroup[{S1, S2}]

In[•]:= GroupOrder[grp]
Out[•]= 16

In[•]:= elm = GroupElements[grp];
        elm // PauliForm
Out[•]= {I⊗I, X⊗I, Z⊗I, Y⊗I, I⊗X, X⊗X, Z⊗X,
         Y⊗X, I⊗Z, X⊗Z, Z⊗Z, Y⊗Z, I⊗Y, X⊗Y, Z⊗Y, Y⊗Y}

In[•]:= gnr = GroupGenerators[grp];
        gnr // PauliForm
Out[•]= {X⊗I, Y⊗I, Z⊗I, I⊗X, I⊗Y, I⊗Z}
```

Ignoring the phase factors leads to a drastically simple structure of $\mathcal{P}(n)$, compared to the full Pauli group $\tilde{\mathcal{P}}(n)$. For example, $X\,Y$ is now equal to $Y\,X$, that is,

$$(X\mathcal{Z})\,(Y\mathcal{Z}) = (X\,Y)\,\mathcal{Z} = (Y\,X)\,\mathcal{Z}.$$

1. This means that unlike the full Pauli group $\tilde{\mathcal{P}}(n)$, **the Pauli group $\mathcal{P}(n)$ is *Abelian***.
2. Pauli group $\mathcal{P}(n)$ is isomorphic to $\mathbb{Z}_2^{2n}$ with group multiplication given by addition modulo 2.

$$X\,Z = Z\,X = Y$$

## Gottesman Vectors

Let us examine the above mentioned **isomorphism $\mathcal{P}(n) \simeq \mathbb{Z}_2^{2n}$** between the Pauli group $\mathcal{P}(n)$ and the binary Abelian group $\mathbb{Z}_2^{2n}$ more closely. Consider the isomorphism defined by

$$X \leftrightarrow (1, 0),\ Z \leftrightarrow (0, 1)$$

The correspondence

$$Y \leftrightarrow (1, 1)$$

follows automatically from the group multiplication (between cosets).

Equivalently, a ***Pauli operator*** of the form

$$P = (X^{x_1} Z^{z_1}) \otimes (X^{x_2} Z^{z_2}) \otimes \ldots \otimes (X^{x_n} Z^{z_n}) \in \mathcal{P}(n)$$

corresponds to the ***Gottesman vector***

$$v = (x_1, z_1, x_2, z_2, \ldots, \ldots, x_n, z_n) \in \mathbb{Z}_2^{2n}.$$

---

For example, consider the following three-qubit Pauli operator.

In[•]:=
```
op = S[1, 1] ** S[2, 3] ** S[3, 2];
op // PauliForm
```

Out[•]=
```
X ⊗ Z ⊗ Y
```

Get the Gottesman vector corresponding to `op`.

In[•]:=
```
vec = GottesmanVector[op, S@{1, 2, 3}]
```

Out[•]=
```
{1, 0, 0, 1, 1, 1}
```

Here is an exhaustive conversion table of the whole Pauli group on two qubits.

In[•]:=
```
grp = PauliGroup[S@{1, 2}]
elm = GroupElements[grp];
elm // PauliForm
```

Out[•]=
```
PauliGroup[{S₁, S₂}]
```

Out[•]=
```
{I ⊗ I, X ⊗ I, Z ⊗ I, Y ⊗ I, I ⊗ X, X ⊗ X, Z ⊗ X,
 Y ⊗ X, I ⊗ Z, X ⊗ Z, Z ⊗ Z, Y ⊗ Z, I ⊗ Y, X ⊗ Y, Z ⊗ Y, Y ⊗ Y}
```

```
In[•]:= vec = GottesmanVector[#, S@{1, 2}] & /@ elm;
        Thread[PauliForm[elm] ↔ vec] // TableForm
```

```
Out[•]//TableForm=
I ⊗ I ↔ {0, 0, 0, 0}
X ⊗ I ↔ {1, 0, 0, 0}
Z ⊗ I ↔ {0, 1, 0, 0}
Y ⊗ I ↔ {1, 1, 0, 0}
I ⊗ X ↔ {0, 0, 1, 0}
X ⊗ X ↔ {1, 0, 1, 0}
Z ⊗ X ↔ {0, 1, 1, 0}
Y ⊗ X ↔ {1, 1, 1, 0}
I ⊗ Z ↔ {0, 0, 0, 1}
X ⊗ Z ↔ {1, 0, 0, 1}
Z ⊗ Z ↔ {0, 1, 0, 1}
Y ⊗ Z ↔ {1, 1, 0, 1}
I ⊗ Y ↔ {0, 0, 1, 1}
X ⊗ Y ↔ {1, 0, 1, 1}
Z ⊗ Y ↔ {0, 1, 1, 1}
Y ⊗ Y ↔ {1, 1, 1, 1}
```

Since the group isomorphism between $\mathcal{P}(n)$ and $\mathbb{Z}_2^{2n}$ preserves group multiplication, the multiplication of Pauli operators corresponds to addition modulo 2 (as the group multiplication in $\mathbb{Z}_2^{2n}$).

```
In[•]:= S[1, 1] ** S[1, 2] /. {_?CommutativeQ * op_ :> op}
```

```
Out[•]= S_1^z
```

```
In[•]:= Mod[GottesmanVector[S[1, 1]] + GottesmanVector[S[1, 2]], 2]
```

```
Out[•]= {0, 1}
```

## Two Applications of Gottesman Vectors

1. This provides a **convenient way to check for the independence of Pauli operators**.

   Suppose that Pauli operators $P_1, P_2, \ldots, P_k \in \mathcal{P}(n)$ are independent, so that none of them can be obtained (up to a phase factor) by a multiplication of the others. Then, it follows that the corresponding Gottesman vectors $v(P_1), v(P_1), \ldots, v(P_k)$ are linearly independent of each other in $\mathbb{Z}_2^{2n}$. The converse is also true in that if the Gottesman vectors $v_1, v_2, \ldots, v_k$ are linearly independent of each other in $\mathbb{Z}_2^{2n}$, then the operators $P(v_1), P(v_1), \ldots, P(v_k)$ are independent (up to a phase factor) of each other in $\mathcal{P}(n)$.

2. **Theorem: Consider again a set of independent Pauli operators $P_1, P_2, \ldots, P_k \in \mathcal{P}(n)$. Then, there exits an operator $P$ in $\mathcal{P}(n)$ such that $P$ anti-commutes with one and only one of $P_1, P_2, \ldots, P_k$ but commutes with the rest**.

   One can use Gottesman vectors to prove the above property of the Pauli group (Problem 6.3 of the Quantum Workbook).

For example, consider the following two independent Pauli operators.

```
In[•]:= P1 = S[1, 3] ** S[2, 3]; PauliForm[P1, S@{1, 2, 3}]
        P2 = S[2, 3] ** S[3, 3];
        PauliForm[P2, S@{1, 2, 3}]
Out[•]= Z ⊗ Z ⊗ I
Out[•]= I ⊗ Z ⊗ Z
```

This Pauli operator anti-commutes with **P1** and but commutes with **P2**.

```
In[•]:= A = S[1, 1]; PauliForm[A, S@{1, 2, 3}]
        A ** P1 + P1 ** A
        A ** P2 - P2 ** A
Out[•]= X ⊗ I ⊗ I
Out[•]= 0
Out[•]= 0
```

## Relation to the Full Pauli Group

Interestingly, the correspondence between the Pauli group $\mathcal{P}(n)$ and the binary Abelian group $\mathbb{Z}_2^{2n}$ also carries the commutation relation of Pauli operators in the *full* Pauli group to a certain type of relation of the corresponding Gottesman vectors. Let $P_1$ and $P_2$ be $n$-qubit Pauli operators. It is straightforward to show by inspection that they commute with each other if and only if

$$\langle v(P_1),\, v(P_2)\rangle = 0,$$

where $\langle \cdot\,,\, \cdot\rangle$ is the *Gottesman inner product* in $\mathbb{Z}_2^{2n}$ defined by

$$\langle v,\, w\rangle := \sum_i \sum_j v_i\, J_{ij}\, w_j$$

with $2n \times 2n$ matrix

$$J := I_n \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Furthermore, the two Pauli operators anti-commute with each other if and only if

$$\langle v(P_1),\, v(P_2)\rangle = 1.$$

Here are examples.

```
In[•]:= P1 = S[1, 1] ** S[2, 1]; PauliForm[P1, S@{1, 2}]
        P2 = S[1, 3] ** S[2, 2];
        PauliForm[P2, S@{1, 2}]
Out[•]= X ⊗ X
Out[•]= Z ⊗ Y
```

```
In[•]:= Commutator[P1, P2]
Out[•]= 0

In[•]:= v1 = GottesmanVector[P1, S@{1, 2}]
        v2 = GottesmanVector[P2, S@{1, 2}]
        GottesmanInner[v1, v2]
Out[•]= {1, 0, 1, 0}

Out[•]= {0, 1, 1, 1}

Out[•]= 0

In[•]:= P1 = S[1, 1] ** S[2, 1]; PauliForm[P1, S@{1, 2}]
        P2 = S[1, 2] ** S[2, 0];
        PauliForm[P2, S@{1, 2}]
Out[•]= X ⊗ X

Out[•]= Y ⊗ I

In[•]:= Anticommutator[P1, P2]
Out[•]= 0

In[•]:= v1 = GottesmanVector[P1, S@{1, 2}]
        v2 = GottesmanVector[P2, S@{1, 2}]
        GottesmanInner[v1, v2]
Out[•]= {1, 0, 1, 0}

Out[•]= {1, 1, 0, 0}

Out[•]= 1
```

# The Clifford Group

## The Full Clifford Group

### Definition

The ***full Clifford group*** $\tilde{C}(n)$ on $n$ qubits is the group of unitary operators $U$ that leave the full Pauli group $\tilde{\mathcal{P}}(n)$ invariant under conjugation, that is,

$$\tilde{C}(n) := \{U : U\,P\,U^{\dagger} \in \tilde{\mathcal{P}}(n) \text{ for all } P \in \tilde{\mathcal{P}}(n)\}.$$

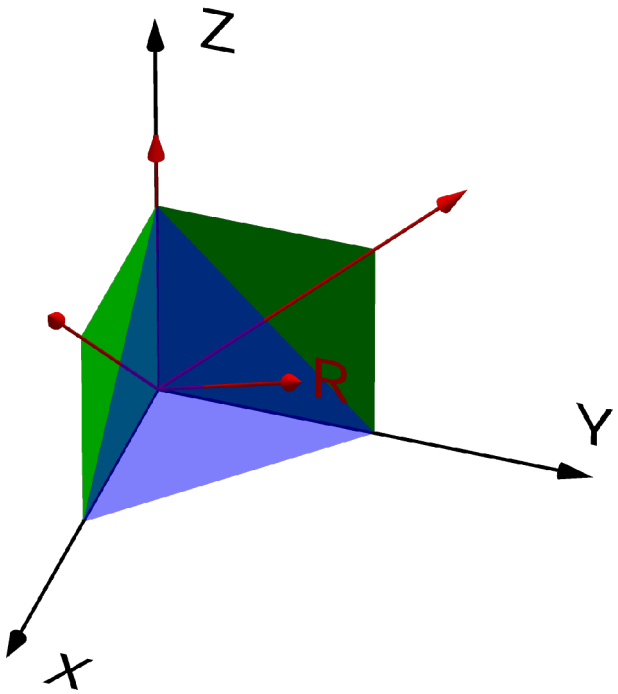
Mathematically, the full Clifford group is the normalizer of the full Pauli group in unitary group $\mathcal{U}(2\,n)$. The elements of the full Clifford group on $n$ qubits are called $n$-qubit Clifford operators.

**Example**: Hadamard operator

In[•]:=
```
op = S[1, 6];
spr = Supermap[op];
Thread[S[1, All] → spr[S[1, All]]] // Elaborate // PauliForm // TableForm
```

Out[•]//TableForm=

| | |
|---|---|
| X | → Z |
| Y | → − Y |
| Z | → X |

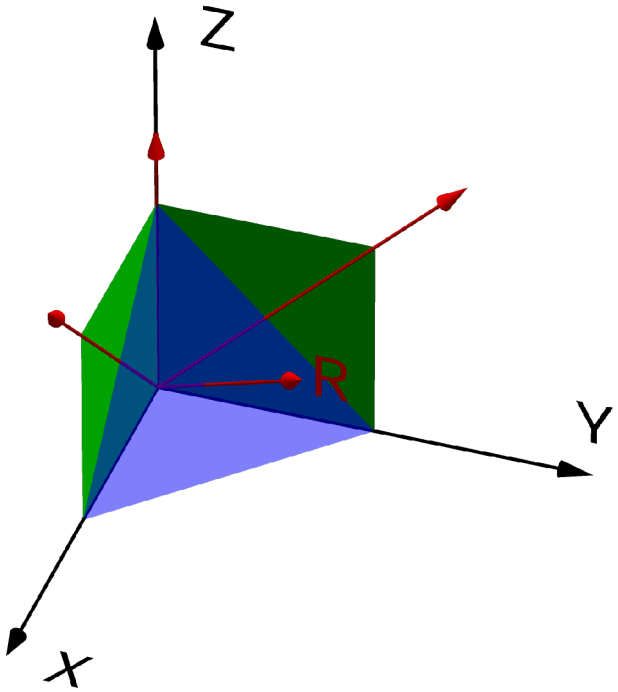



**Example**: Quadrant phase shift

In[•]:=
```
op = S[1, 7];
spr = Supermap[op];
Thread[S[1, All] → spr[S[1, All]]] // Elaborate // PauliForm // TableForm
```

Out[•]//TableForm=

| | |
|---|---|
| X | → Y |
| Y | → − X |
| Z | → Z |




**Example**: CNOT

In[•]:=
```
op = CNOT[S[1], S[2]];
spr = Supermap[op];
in = Join[S[{1, 2}, 1], S[{1, 2}, 3]];
out = spr[in];
Thread[in → out] // PauliForm // TableForm
```

Out[•]//TableForm=

$X \otimes I \to X \otimes X$
$I \otimes X \to I \otimes X$
$Z \otimes I \to Z \otimes I$
$I \otimes Z \to Z \otimes Z$

## Elementary Properties

1. The **full Clifford group** is huge.

In[•]:=
```
grp = FullCliffordGroup[S@{1, 2}]
GroupOrder[grp]
```

Out[•]= FullCliffordGroup[{$S_1$, $S_2$}]

Out[•]= 92 160

2. The full Clifford group $\tilde{C}(n)$ is generated by **Hadamard gates** and **quadrant phase gates** on individual qubits, and **CNOT** (or equivalently CZ) gates on all pairs of qubits.

$$H \doteq \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},\quad S \doteq \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},\quad T \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix},\quad \text{CNOT} \doteq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

```
In[•]:= grp = FullCliffordGroup[S@{1, 2, 3}]
        gnr = GroupGenerators[grp]
Out[•]= FullCliffordGroup[{S1, S2, S3}]
Out[•]= {S1^H, S1^S, S2^H, S2^S, S3^H, S3^S, CZ[{S1}, {S2}], CZ[{S1}, {S3}], CZ[{S2}, {S3}]}
```

Recall that the Hadamard and quadrant phase gates are represented by the following respective matrices.

```
In[•]:= S[1, 6] // Matrix // MatrixForm (*the Hadarmard gate*)
        S[1, 7] // Matrix // MatrixForm (*the quadrant phase gate*)
```

Out[•]//MatrixForm=

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Out[•]//MatrixForm=

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

For a proof that the full Clifford group is generated by the Hadamard, quadrant phase, and CNOT gates, we refer the readers to Gottesman's original paper (Gottesman, 1998), Nielsen and Chuang (2011), or the Quantum Workbook.

3. Since they generate the full Clifford group, the Hadamard, quadrant phase, and CNOT gates are called the *elementary Clifford gates* or simply Clifford gates. Quantum circuits consisting of elementary Clifford gates are called *Clifford circuits*.

## The Clifford Group

As in the full Pauli group, elements in the full Clifford group appear redundantly and differ only in global phase factors.

For example, this shows four elements of the Clifford group on two qubits. The last two elements are different from the first two only by a global phase factor $e^{i\pi/4}$.

```
In[•]:= grp = FullCliffordGroup[S@{1, 2}]
        elm = GroupElements[grp, {1, 2, 11 520 + 1, 11 520 + 2}]
```

```
Out[•]= FullCliffordGroup[{S1, S2}]
```

$$\text{Out}[\bullet]= \left\{\frac{1}{2\sqrt{2}}+\frac{S_1^x S_2^x}{2\sqrt{2}}+\frac{S_1^x S_2^z}{2\sqrt{2}}-\frac{i S_1^y S_2^x}{2\sqrt{2}}+\frac{i S_1^y S_2^z}{2\sqrt{2}}-\frac{i S_1^z S_2^y}{2\sqrt{2}}+\frac{S_1^z}{2\sqrt{2}}+\frac{i S_2^y}{2\sqrt{2}},\right.$$
$$\frac{1}{2\sqrt{2}}+\frac{S_1^x S_2^x}{2\sqrt{2}}+\frac{S_1^x S_2^z}{2\sqrt{2}}+\frac{i S_1^y S_2^x}{2\sqrt{2}}-\frac{i S_1^y S_2^z}{2\sqrt{2}}+\frac{i S_1^z S_2^y}{2\sqrt{2}}+\frac{S_1^z}{2\sqrt{2}}-\frac{i S_2^y}{2\sqrt{2}},$$
$$e^{\frac{i\pi}{4}}\left(\frac{1}{2\sqrt{2}}+\frac{S_1^x S_2^x}{2\sqrt{2}}+\frac{S_1^x S_2^z}{2\sqrt{2}}-\frac{i S_1^y S_2^x}{2\sqrt{2}}+\frac{i S_1^y S_2^z}{2\sqrt{2}}-\frac{i S_1^z S_2^y}{2\sqrt{2}}+\frac{S_1^z}{2\sqrt{2}}+\frac{i S_2^y}{2\sqrt{2}}\right),$$
$$\left.e^{\frac{i\pi}{4}}\left(\frac{1}{2\sqrt{2}}+\frac{S_1^x S_2^x}{2\sqrt{2}}+\frac{S_1^x S_2^z}{2\sqrt{2}}+\frac{i S_1^y S_2^x}{2\sqrt{2}}-\frac{i S_1^y S_2^z}{2\sqrt{2}}+\frac{i S_1^z S_2^y}{2\sqrt{2}}+\frac{S_1^z}{2\sqrt{2}}-\frac{i S_2^y}{2\sqrt{2}}\right)\right\}$$

In the full Clifford group $\tilde{C}(n)$, eight different global phase factors appear, $e^{i\,m\,\pi/4}$ ($m = 0, 1, 2, \ldots, 7$). In many cases, these factors play no role and it is pointless to distinguish Clifford operators differing only by these factors. Mathematically, as in the full Pauli group, one can ignore these factors by introducing a quotient group

$$C(n) := \tilde{C}(n) \big/ \mathcal{Z}(n),$$

where

$$\mathcal{Z}(n) := \left\{e^{i\,m\,\pi/4}\, I^{\otimes n} : m = 0, 1, 2, \ldots, 7\right\}.$$

is the center of the full Clifford group. This quotient group $C(n)$ is called the *Clifford group*.

```
In[•]:= grp = CliffordGroup[S@{1, 2}]
        GroupOrder[grp]
```

```
Out[•]= CliffordGroup[{S1, S2}]
```

```
Out[•]= 11 520
```

The Clifford group is still huge although its order has been reduced by factor 8.

In[•]:=
```
GroupElements[grp]
```

CliffordGroup: There are about $1.2\times10^4$ elements in the group. Only the first 10 elements are returned.

Out[•]=

$$\left\{\frac{1}{2\sqrt{2}}+\frac{S_1^x S_2^x}{2\sqrt{2}}+\frac{S_1^x S_2^z}{2\sqrt{2}}-\frac{i S_1^y S_2^x}{2\sqrt{2}}+\frac{i S_1^y S_2^z}{2\sqrt{2}}-\frac{i S_1^z S_2^y}{2\sqrt{2}}+\frac{S_1^z}{2\sqrt{2}}+\frac{i S_2^y}{2\sqrt{2}},\right.$$

$$\frac{1}{2\sqrt{2}}+\frac{S_1^x S_2^x}{2\sqrt{2}}+\frac{S_1^x S_2^z}{2\sqrt{2}}+\frac{i S_1^y S_2^x}{2\sqrt{2}}-\frac{i S_1^y S_2^z}{2\sqrt{2}}+\frac{i S_1^z S_2^y}{2\sqrt{2}}+\frac{S_1^z}{2\sqrt{2}}-\frac{i S_2^y}{2\sqrt{2}},$$

$$\frac{i}{2\sqrt{2}}+\frac{i S_1^x S_2^y}{2\sqrt{2}}-\frac{S_1^y S_2^x}{2\sqrt{2}}+\frac{S_1^y S_2^z}{2\sqrt{2}}+\frac{S_1^z S_2^x}{2\sqrt{2}}+\frac{S_1^z S_2^z}{2\sqrt{2}}+\frac{S_1^x}{2\sqrt{2}}+\frac{S_2^y}{2\sqrt{2}},$$

$$\frac{1}{2}S_1^x S_2^x+\frac{1}{2}S_1^x S_2^z+\frac{1}{2}S_1^z S_2^x+\frac{1}{2}S_1^z S_2^z,\ \frac{1}{2}i S_1^x S_2^y+\frac{1}{2}S_1^z S_2^x+\frac{1}{2}S_1^z S_2^z+\frac{S_1^x}{2},$$

$$-\frac{1}{2}i S_1^x S_2^y+\frac{1}{2}S_1^z S_2^x+\frac{1}{2}S_1^z S_2^z+\frac{S_1^x}{2},\ \frac{1}{2}i S_1^x S_2^x-\frac{1}{2}i S_1^x S_2^z+\frac{1}{2}S_1^z S_2^x+\frac{1}{2}S_1^z S_2^z,$$

$$\frac{S_2^x}{\sqrt{2}}+\frac{S_2^z}{\sqrt{2}},\ \frac{1}{2\sqrt{2}}-\frac{i S_1^y S_2^x}{2\sqrt{2}}-\frac{S_1^y S_2^y}{2\sqrt{2}}-\frac{i S_1^y S_2^z}{2\sqrt{2}}+\frac{i S_1^y}{2\sqrt{2}}+\frac{S_2^x}{2\sqrt{2}}+\frac{i S_2^y}{2\sqrt{2}}+\frac{S_2^z}{2\sqrt{2}},$$

$$\left.\frac{1}{2\sqrt{2}}-\frac{i S_1^y S_2^x}{2\sqrt{2}}+\frac{S_1^y S_2^y}{2\sqrt{2}}-\frac{i S_1^y S_2^z}{2\sqrt{2}}+\frac{i S_1^y}{2\sqrt{2}}+\frac{S_2^x}{2\sqrt{2}}-\frac{i S_2^y}{2\sqrt{2}}+\frac{S_2^z}{2\sqrt{2}}\right\}$$

This gives the first, tenth, and twentieth elements of the Clifford group.

In[•]:=
```
GroupElements[grp, {1, 10, 20}]
```

Out[•]=

$$\left\{\frac{1}{2\sqrt{2}}+\frac{S_1^x S_2^x}{2\sqrt{2}}+\frac{S_1^x S_2^z}{2\sqrt{2}}-\frac{i S_1^y S_2^x}{2\sqrt{2}}+\frac{i S_1^y S_2^z}{2\sqrt{2}}-\frac{i S_1^z S_2^y}{2\sqrt{2}}+\frac{S_1^z}{2\sqrt{2}}+\frac{i S_2^y}{2\sqrt{2}},\right.$$

$$\frac{1}{2\sqrt{2}}-\frac{i S_1^y S_2^x}{2\sqrt{2}}+\frac{S_1^y S_2^y}{2\sqrt{2}}-\frac{i S_1^y S_2^z}{2\sqrt{2}}+\frac{i S_1^y}{2\sqrt{2}}+\frac{S_2^x}{2\sqrt{2}}-\frac{i S_2^y}{2\sqrt{2}}+\frac{S_2^z}{2\sqrt{2}},$$

$$-\frac{i}{4}+\frac{1}{4}S_1^x S_2^x+\frac{1}{4}i S_1^x S_2^y+\frac{1}{4}S_1^x S_2^z+\frac{1}{4}S_1^y S_2^x-\frac{1}{4}i S_1^y S_2^y+\frac{1}{4}S_1^y S_2^z+$$

$$\left.\frac{1}{4}S_1^z S_2^x+\frac{1}{4}i S_1^z S_2^y+\frac{1}{4}S_1^z S_2^z+\frac{S_1^x}{4}-\frac{S_1^y}{4}+\frac{S_1^z}{4}+\frac{i S_2^x}{4}+\frac{S_2^y}{4}+\frac{i S_2^z}{4}\right\}$$

## Gottesman Matrices

Let us now examine the structure of the Clifford group more closely. By definition, Clifford operators leave the Pauli group invariant under conjugation. Therefore, a **Clifford operator** $U \in C(n)$ transform any Pauli operator, especially Pauli operators $X_j$ and $Z_j$ on $j^{\text{th}}$ qubit, to a tensor product of single-qubit Pauli operators,

$$U X_j U^\dagger = (-1)^{p_j} \prod_{i=1}^n X_i^{a_{ij}} Z_i^{b_{ij}}$$

$$U Z_j U^\dagger = (-1)^{q_j} \prod_{i=1}^n X_i^{c_{ij}} Z_i^{d_{ij}}$$

with $p_j,\ q_j,\ a_{ij},\ b_{ij},\ c_{ij},\ d_{ij} \in \{0, 1\}$. Since the single-qubit Pauli operators, $X_j$ and $Z_j$, generate the full Pauli group, a Clifford operator is completely specified (up to global phase factor) by this set of binary coefficients $p_j,\ q_j,\ a_{ij},\ b_{ij},\ c_{ij},\ d_{ij}$.

Among the binary coefficients, coefficients $p_j$ and $q_j$ that determine the sign of the resulting Pauli operators should come from Pauli operators because no elementary Clifford gates (the Hadamard, quadrant phase, or CNOT gates) can induce –1. Recalling that the Pauli group is an invariant subgroup of the Clifford group, one can then attribute the rest coefficients $a_{ij}$, $b_{ij}$, $c_{ij}$, $d_{ij}$ to the quotient group $C(n)/\mathcal{P}(n)$.

Furthermore, since any unitary operator ***must preserve the commutation relations***, the commutation relations between $X_j$ and $Z_j$ leads to certain constraints on binary coefficients $a_{ij}$, $b_{ij}$, $c_{ij}$, $d_{ij}$. Recall that the commutation relations between Pauli operators may be described by the Gottesman inner product between the corresponding Gottesman vectors. Therefore, the constraints on binary coefficients $a_{ij}$, $b_{ij}$, $c_{ij}$, $d_{ij}$ can be conveniently described in terms of the Gottesman inner product. This can be done most efficiently by arranging the binary coefficients in a $2n \times 2n$ binary matrix of the form

$$S(U) := \begin{pmatrix} a_{11} & c_{11} & a_{12} & c_{12} & \cdots & \cdots \\ b_{11} & d_{11} & b_{12} & d_{12} & \cdots & \cdots \\ a_{21} & c_{21} & a_{22} & c_{22} & \cdots & \cdots \\ b_{21} & d_{21} & b_{22} & d_{22} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{pmatrix}.$$

Then, the commutation relations between single-qubit Pauli operators leads to the condition

$$S(U) \cdot J \cdot S^{\dagger}(U) = J,$$

where $J$ is the same matrix defining the Gottesman inner product in the Gottesman vector space $\mathbb{Z}_2^{2n}$, that is,

$$J := I_n \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

This means that matrix $S(U)$ is a symplectic matrix. In other words, the quotient group $C(n)/\mathcal{P}(n)$ is isomorphic to the binary symplectic group $\mathrm{Sp}(2n, \mathbb{Z}_2)$, the group of $2n \times 2n$ binary symplectic matrices. We call the **binary symplectic matrix** $S(U)$ the ***Gottesman matrix*** of Clifford operator $U$.

For example, consider a Clifford operator.

```
In[•]:= op = First@GroupElements[grp, {3}];
        op // PauliForm
```

$$\text{Out[•]=} \quad \frac{\mathbb{i}\, \mathrm{I}\otimes\mathrm{I}}{2\sqrt{2}} + \frac{\mathrm{I}\otimes\mathrm{Y}}{2\sqrt{2}} + \frac{\mathrm{X}\otimes\mathrm{I}}{2\sqrt{2}} + \frac{\mathbb{i}\, \mathrm{X}\otimes\mathrm{Y}}{2\sqrt{2}} - \frac{\mathrm{Y}\otimes\mathrm{X}}{2\sqrt{2}} + \frac{\mathrm{Y}\otimes\mathrm{Z}}{2\sqrt{2}} + \frac{\mathrm{Z}\otimes\mathrm{X}}{2\sqrt{2}} + \frac{\mathrm{Z}\otimes\mathrm{Z}}{2\sqrt{2}}$$

Get the Gottesman matrix corresponding to the Clifford operator.

```
In[•]:= mat = GottesmanMatrix[op, S@{1, 2}];
        mat // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

You can also recover the Clifford operator (up to a factor of Pauli operator) from the Gottesman matrix.

```
In[•]:= new = FromGottesmanMatrix[mat, S@{1, 2}];
        new // PauliForm
```

Out[•]= $\frac{\mathrm{i}\, \mathrm{I}\otimes\mathrm{I}}{2\sqrt{2}} + \frac{\mathrm{I}\otimes\mathrm{Y}}{2\sqrt{2}} + \frac{\mathrm{X}\otimes\mathrm{I}}{2\sqrt{2}} + \frac{\mathrm{i}\, \mathrm{X}\otimes\mathrm{Y}}{2\sqrt{2}} - \frac{\mathrm{Y}\otimes\mathrm{X}}{2\sqrt{2}} + \frac{\mathrm{Y}\otimes\mathrm{Z}}{2\sqrt{2}} + \frac{\mathrm{Z}\otimes\mathrm{X}}{2\sqrt{2}} + \frac{\mathrm{Z}\otimes\mathrm{Z}}{2\sqrt{2}}$

```
In[•]:= new - op
```

Out[•]= 0

## The Gottesman-Knill Theorem

The peculiarity of the Clifford group is most pronounced in the Gottesman-Knill theorem (Gottesman, 1999). It states that the so-called *Clifford circuits*, quantum circuits consisting of unitary operators from the Clifford group, can be efficiently simulated on a classical computer.

This sounds surprising because the Clifford operators are sufficient to generate a rich range of quantum effects including the Greenberger-Horne-Zeilinger (GHZ) experiment (Greenberger et al., 1989), quantum teleportation, and super dense coding. They are also sufficient to encode and decode quantum error-correction codes (Section 6.4 in the Quantum Workbook). Nevertheless, the theorem indicates that the Clifford group "falls short of the full power of quantum computation." Indeed, we recall that in addition to Clifford gates, we need octant phase gates or Toffoli gates to implement an arbitrary unitary gate to an arbitrary accuracy (Section 2.4 of the Quantum Workbook).

How is it possible to simulate Clifford circuits efficiently on a classical computer? The key point is that stabilizer formalism tracks the generators of the stabilizer rather than the quantum states as in the usual methods. To follow the evolution of the quantum states, one needs to keep track of an exponential number of complex amplitudes. However, for a system of $n$ qubits, there are an order of $n$ generators of the stabilizer because any finite group $\mathcal{G}$ has a generating set of at most $\log_2 |\mathcal{G}|$ elements. Each generator takes $2n$ bits for the Pauli $X$ or $Y$ operators on $n$ qubits and an additional bit for the phase factor $\pm 1$. Overall, one needs an order of $n^2$ bits to track the evolution of the stabilizer. We have seen that the Clifford group is generated by the Hadamard, quadrant phase, and CNOT gates and that the generators under conjugation by these elementary gates can be updated in polynomial time, $O(n)$ time, for the gate.

A specific algorithm to simulate Clifford circuits was proposed in Aaronson and Gottesman (2004). Recently, a simulation algorithm was also proposed to conduct a fast simulation of the stabilizer circuits using graph states.

# References


- D. Gottesman, Physical Review A 54, 1862 (1996) , “Class of quantum error- correcting codes saturating the quantum Hamming bound”.
- D. Gottesman, Stabilizer Codes and Quantum Error Correction (Ph.D. Thesis, California Institute . of Technology, 1997)
- D. Gottesman, Physical Review A 57, 127 (1998) , “Theory of fault-tolerant quantum computation”.
- D. Browne and H. Briegel, “One–Way Quantum Computation,” in Quantum Information: From . Foundations to Quantum Technology Applications, edited by Bruß, D. & G. Leuchs (Wiley, 2016)
- D. M. Greenberger, M. A. Horne, and A. Zeilinger, “Going beyond Bell’s theorem,” in Bell’s . Theorem, Quantum Theory, and Conceptions of the Universe, edited by Kafatos, M. (Kluwer Academic, Dordrecht, The Netherlands, 1989)
- S. Aaronson and D. Gottesman, Physical Review A 70 (5) (2004) , “Im- proved simulation of stabilizer circuits”.
- *M. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information (Cambridge . University Press, New York, 2011)*
- Mahn–Soo Choi, A Quantum Computation Workbook (Springer, 2022) .