1 Рандомизированный бенчмаркинг

Метод случайного тестирования (рандомизированного бенчмаркинга) состоит из следующих шагов:

- Систему из n кубитов приготавливают в определенном состоянии, описываемом матрицей плотности ρ.

- Выбирается некоторое множество операций, образующее группу, например, группу Клиффорда $\mathbf{Cl^n}$.

- Из этой группы выбирается последовательность операций длины l. Первые $l-1$ элементов последовательности V_i выбираются случайным образом. Последний элемент последовательности R - это обратная операция, которая в идеальном случае превращает последовательность в тождественную:
$$RV_{l-1}...V_2V_1 = \mathbb{I} \tag{1}$$
Т. к. V_i - это элементы группы, то R также принадлежит этой группе. Таким образом, задача состоит в том, чтобы по известной случайной последовательности элементов группы $\mathbf{Cl^n}$ найти элемент из $\mathbf{Cl^n}$, соответствующий обратной операции.

- Генерируется новая последовательность, проделывается та же схема и так ещё некоторое число раз. В итоге, для группы $\mathbf{Cl^n}$ и длины l получается усреднённая по случайным последовательностям вероятность сохранить исходное состояние. Выполняя то же самое для других l, можно получить зависимость этой вероятности от l.

2 Случай одного кубита

Рассмотрим группу Клиффорда для одного кубита, обозначим её $\mathbf{Cl^1}$. В качестве порождающего множества группы $\mathbf{Cl^1}$ можно выбрать два гейта - однокубитные вращения вокруг осей x и z на угол $\pi/2$. Соответствующие матрицы этих гейтов:

$$X_{\pi/2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \tag{2}$$

$$Z_{\pi/2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1-i & 0 \\ 0 & 1+i \end{pmatrix} \tag{3}$$

Группу $\mathbf{Cl^1}$ можно разбить на четыре класса эквивалентности [1]:

1. класс $\mathbf{Cl^1}(G_1)$, элементы которого эквивалентны элементам группы Паули G_1;

2. класс $\mathbf{Cl^1}(\pi/2)$, элементы которого эквивалентны вращениям на угол $\pi/2$;

3. класс $\mathbf{Cl^1}(2\pi/3)$, элементы которого эквивалентны вращениям на угол $2\pi/3$;

4. класс $\mathbf{Cl^1}(H)$, элементы которого эквивалентны вентилю Адамара H.

Элементы каждого класса приведены в Таблице 1. Всего группа $\mathbf{Cl^1}$ содержит 24 элемента.

Таблица 1: Классификация элементов однокубитной группы Клиффорда.

Класс	Элементы
$\mathbf{Cl^1}(G_1)$	I $X_{\pi/2}^2$ $X_{\pi/2}^2 Z_{\pi/2}^2$ $Z_{\pi/2}^2$
$\mathbf{Cl^1}(\pi/2)$	$X_{\pi/2}$ $Z_{\pi/2}^2 X_{\pi/2} Z_{\pi/2}^2$ $Z_{\pi/2} X_{\pi/2} Z_{\pi/2}^3$ $Z_{\pi/2}^3 X_{\pi/2} Z_{\pi/2}$ $Z_{\pi/2}$ $Z_{\pi/2}^3$
$\mathbf{Cl^1}(2\pi/3)$	$X_{\pi/2} Z_{\pi/2}$ $Z_{\pi/2} X_{\pi/2}$ $X_{\pi/2} Z_{\pi/2}^3$ $Z_{\pi/2} X_{\pi/2} Z_{\pi/2}^2$ $Z_{\pi/2}^2 X_{\pi/2} Z_{\pi/2}$ $Z_{\pi/2}^3 X_{\pi/2}$ $Z_{\pi/2}^2 X_{\pi/2} Z_{\pi/2}^3$ $Z_{\pi/2}^3 X_{\pi/2} Z_{\pi/2}^2$
$\mathbf{Cl^1}(H)$	$X_{\pi/2}^2 Z_{\pi/2}$ $X_{\pi/2} Z_{\pi/2}^2$ $Z_{\pi/2} X_{\pi/2}^2$ $Z_{\pi/2} X_{\pi/2} Z_{\pi/2}$ $Z_{\pi/2}^2 X_{\pi/2}$ $Z_{\pi/2}^3 X_{\pi/2} Z_{\pi/2}^3$

Элементы группы $\mathbf{Cl^1}$ определяем в модуле gates.py.

```
"""Группа Клиффорда для одного кубита"""
#Группа Паули
c[0] = I
c[1] = X @ X
c[2] = X @ X @ Z @ Z
c[3] = Z @ Z

#Вращения на pi/2
c[4] = X
c[5] = Z @ Z @ X @ Z @ Z
c[6] = Z @ X @ Z @ Z @ Z
c[7] = Z @ Z @ Z @ X @ Z
c[8] = Z
c[9] = Z @ Z @ Z

#Вращения на 2*pi/3
c[10] = X @ Z
c[11] = Z @ X
c[12] = X @ Z @ Z @ Z
c[13] = Z @ X @ Z @ Z
c[14] = Z @ Z @ X @ Z
c[15] = Z @ Z @ Z @ X
c[16] = Z @ Z @ X @ Z @ Z @ Z
c[17] = Z @ Z @ Z @ X @ Z @ Z

#Адамары
c[18] = X @ X @ Z
c[19] = X @ Z @ Z
c[20] = Z @ X @ X
c[21] = Z @ X @ Z
c[22] = Z @ Z @ X
c[23] = Z @ Z @ Z @ X @ Z @ Z @ Z
```

Из элементов группы $\mathbf{Cl^1}$ составляем последовательность операций длины l. Первые $l-1$ элементов последовательности V_i выбираются случайным образом. За генерацию случайной последовательности операций отвечает функция random_sequence_1 в модуле main.py.

```
def random_sequence_1(l):
    """Генерация случайной последовательности длины l-1 для одного кубита"""
    r = random.randint(0, 23)
    V = c[r]
    for k in range(l-1):
        i = random.randint(0, 23)
        V = c[i] @ V
    return V
```

Пусть R_1 - это обратная операция, которая превращает последовательность в тождественную:

$$R_1 V_{l-1} ... V_2 V_1 = \mathbb{I} \tag{4}$$

Необходимо выразить операцию R_1 через гейты $X_{\pi/2}$ и $Z_{\pi/2}$. Для этого определим матрицу F_i размера 4×2. В первую строку матрицы F_i запишем по порядку матричные элементы i-го оператора однокубитной группы Клиффорда c_i. Во вторую строку матрицы F_i запишем по порядку элементы матрицы R_1.

$$F_i = \begin{pmatrix} c_i^{11} & c_i^{12} & c_i^{21} & c_i^{22} \\ R_1^{11} & R_1^{12} & R_1^{21} & R_1^{22} \end{pmatrix} \tag{5}$$

Если ранг матрицы F_i равен единице, то матрица R_1 равна матрице $c_i \in \mathbf{Cl^1}$ с точностью до постоянного множителя. Матрица c_i, в свою очередь, выражается известным образом через $X_{\pi/2}$ и $Z_{\pi/2}$. Перебирая все элементы $\mathbf{Cl^1}$, находим номер i, при котором выполнено условие $\mathrm{rg}F_i = 1$. Таким образом, найдено представление R_1 через гейты $X_{\pi/2}$ и $Z_{\pi/2}$.

За нахождение обратной операции в случае одного кубита отвечает функция find_inverse_1 в модуле main.py.

```
def find_inverse_1(V):
    """Нахождение обратной операции для одного кубита"""
    q = 0
    R1 = np.linalg.inv(V)
    for i in range (len(c)):
        F = np.array([[1j]*4, [1j]*4])
        p = 0
        for k in range(2):
            for l in range(2):
                F[0][p] = c[i][l][k]
                F[1][p] = R1[l][k]
                p = p + 1
        r = np.linalg.matrix_rank(F)
        if (r == 1):
            show_1(i, V)
            q = i
    return q
```

Примеры обратных операций к случайным последовательностям из 1000 гейтов, получаемых с помощью программы:

```
-1(Z · Z · Z · X)

(Z · Z · X · Z)

-1(X · Z)
```

3 Случай двух кубитов

Рассмотрим группу Клифорда для двух кубитов, обозначим её $\mathbf{Cl^2}$. В качестве порождающего множества группы $\mathbf{Cl^2}$ можно выбрать три гейта - однокубитные вращения вокруг осей x и z на угол $\pi/2$ и гейт CZ, матрица которого записывается следующим образом:

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{6}$$

Элементы двухкубитной группы Клиффорда могут быть выражены с помощью группы $\mathbf{S} = \{\mathbb{I}, S_1, S_2\}$, где S_1 - операция циклической перестановки координатных осей на сфере Блоха $(x, y, z) \to (z, x, y)$; $S_2 = S_1^2$.
Операцию S_1 можно выразить через оператор поворота на сфере Блоха вокруг оси $\vec{k}$:

$$R_{\vec{k}}(\alpha) = \exp\left(-i(\vec{k}, \vec{\sigma})\frac{\alpha}{2}\right) \tag{7}$$

$$S_1 = R_y\left(-\frac{\pi}{2}\right)R_x\left(-\frac{\pi}{2}\right) = \frac{1}{2}\begin{pmatrix} 1+i & 1+i \\ -1+i & 1-i \end{pmatrix} \tag{8}$$

$$S_2 = S_1^2 \tag{9}$$

Заметим, что

$$S_1 = -Z_{\pi/2}X_{\pi/2}Z_{\pi/2}Z_{\pi/2}, \tag{10}$$

поэтому S_1 и S_2 принадлежат группе $\mathbf{Cl^1}$.

Группу $\mathbf{Cl^2}$ можно разбить на четыре класса эквивалентности [2]:

1. Класс $\mathbf{Cl^2}(||)$, элементы которого представляют собой параллельно выполняемые гейты однокубитной группы Клиффорда $c_i \in \mathbf{Cl^1}$:

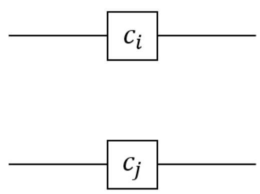

Рис. 1: Элемент класса $\mathbf{Cl^2}(||)$

2. Класс $\mathbf{Cl^2}(CNOT)$, элементы которого представляют собой следующую комбинацию элементов $c_i \in \mathbf{Cl^1}$, $s_i \in \mathbf{S}$ с гейтом CNOT:

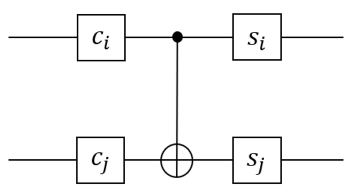

Рис. 2: Элемент класса $\mathbf{Cl^2}(CNOT)$

Элемент CNOT можно построить из элемента CZ и двух элементов Адамара:

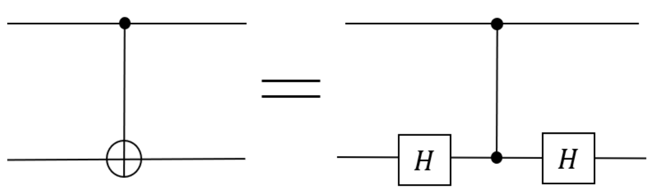

Рис. 3: Элемент CNOT

$$CNOT = (I \otimes H) \cdot CZ \cdot (I \otimes H) \tag{11}$$

Элемент Адамара можно выразить через $X_{\pi/2}$ и $Z_{\pi/2}$ следующим образом:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = i \cdot Z_{\pi/2} \cdot X_{\pi/2} \cdot Z_{\pi/2} \;\in\; \mathbf{Cl^1} \tag{12}$$

3. Класс $\mathbf{Cl^2}(SWAP)$, элементы которого представляют собой следующую комбинацию элементов $c_i \in \mathbf{Cl^1}$ с гейтом SWAP:

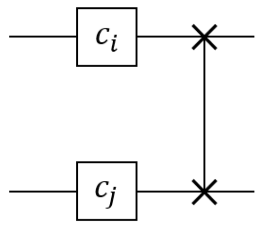

Рис. 4: Элемент класса $\mathbf{Cl^2}(SWAP)$

Элемент SWAP можно построить из трёх элементов CNOT:

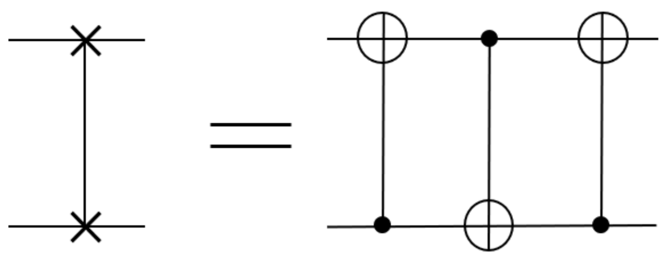

Рис. 5: Элемент SWAP

Затем SWAP можно выразить через CZ и элементы Адамара:

$$SWAP = (H \otimes I) \cdot CZ \cdot (H \otimes H) \cdot CZ \cdot (H \otimes H) \cdot CZ \cdot (H \otimes I) \tag{13}$$

4. Класс $\mathbf{Cl^2}(iSWAP)$, элементы которого представляют собой следующую комбинацию элементов $c_i \in \mathbf{Cl^1}$, $s_i \in \mathbf{S}$ с гейтом iSWAP:

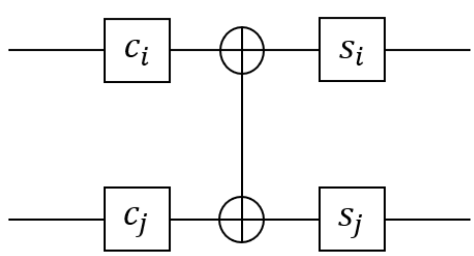

Рис. 6: Элемент класса $\mathbf{Cl^2}(iSWAP)$

Элемент iSWAP можно построить из элемента CNOT, двух элементов Адамара и двух фазовых гейтов Sph. Матрица гейта Sph:

$$Sph = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = e^{i\pi/4} \cdot Z_{\pi/2} \tag{14}$$

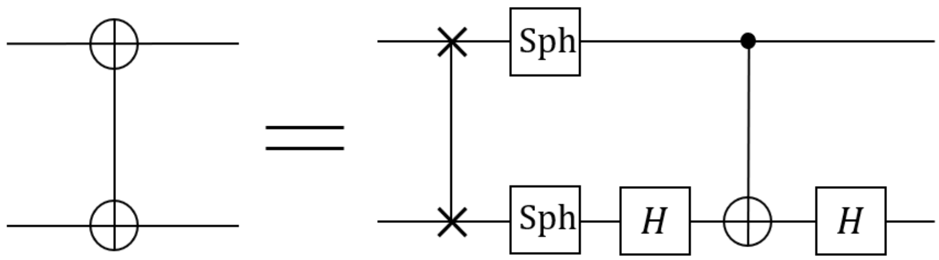

Рис. 7: Элемент iSWAP

$$iSWAP = (I \otimes H) \cdot CNOT \cdot (I \otimes H) \cdot (Sph \otimes Sph) \cdot SWAP \tag{15}$$

Количество элементов в кажом классе двухкубитной группы Клиффорда $\mathbf{Cl^2}$ приведено в Таблице 2.

Таблица 2: Классификация элементов двухкубитной группы Клиффорда.

Класс	Количество элементов
$\mathbf{Cl^2}(\|\|)$	$24^2 = 576$
$\mathbf{Cl^1}(CNOT)$	$24^2 \cdot 3^2 = 5184$
$\mathbf{Cl^2}(SWAP)$	$24^2 = 576$
$\mathbf{Cl^2}(iSWAP)$	$24^2 \cdot 3^2 = 5184$

Всего группа Клиффорда для двух кубитов содержит 11520 элементов. Элементы группы $\mathbf{Cl^2}$ определяем в модуле gates.py.

```
#Двухкубитные гейты
CNOT = tens(I, H) @ CZ @ tens(I, H)
SWAP = tens(H, I) @ CZ @ tens(H, H) @ CZ @ tens(H, H) @ CZ @ tens(H, I)
iSWAP = tens(I, H) @ CNOT @ tens(I, H) @ tens(SPh, SPh) @ SWAP

"""Группа Клиффорда для двух кубитов"""
g = []

for i in range(24):
    for j in range(24):

        W1 = tens(c[i], c[j]) #класс Cl(||)
        g.append(W1)

        W2 = SWAP @ tens(c[i], c[j]) #класс Cl(SWAP)
        g.append(W2)

for i in range(3):
    for j in range(3):
        for k in range(24):
            for l in range(24):

                U1 = tens(S[i], S[j]) @ CNOT @ tens(c[k], c[l]) #класс Cl(CNOT)
                g.append(U1)

                U2 = tens(S[i], S[j]) @ iSWAP @ tens(c[k], c[l]) #класс Cl(iSWAP)
                g.append(U2)
```

Из элементов группы $\mathbf{Cl^2}$ составляем последовательность операций длины l. Первые $l-1$ элементов последовательности W_i выбираются случайным образом. За генерацию случайной последовательности операций отвечает функция random_sequence_2 в модуле main.py.

```
def random_sequence_2(l):
    """Генерация случайной последовательности длины l-1 для двух кубитов"""
    r = random.randint(0, 11519)
    W = g[r]
    for k in range(l-1):
        i = random.randint(0, 11519)
        W = g[i] @ W
    return W
```

Пусть R_2 - это обратная операция, которая превращает последовательность в тождественную:

$$R_2 W_{l-1} ... W_2 W_1 = \mathbb{I} \tag{16}$$

Необходимо выразить операцию R_2 через гейты $X_{\pi/2}$ и $Z_{\pi/2}$. Для этого определим матрицу D_i размера 16×2. В первую строку матрицы D_i запишем по порядку матричные элементы i-го оператора двухкубитной группы Клиффорда g_i. Во вторую строку матрицы D_i запишем по порядку элементы матрицы R_2.

$$D_i = \begin{pmatrix} g_i^{11} & g_i^{12} & g_i^{21} & g_i^{22} & \cdots & g_i^{44} \\ R_2^{11} & R_2^{12} & R_2^{21} & R_2^{22} & \ldots & R_2^{44} \end{pmatrix} \tag{17}$$

Если ранг матрицы D_i равен единице, то матрица R_2 равна матрице $g_i \in \mathbf{Cl^2}$ с точностью до постоянного множителя. Матрица g_i, в свою очередь, выражается известным образом через $X_{\pi/2}$ и $Z_{\pi/2}$. Перебирая все элементы $\mathbf{Cl^2}$, находим номер i, при котором выполнено условие $\mathrm{rg} D_i = 1$. Таким образом, найдено представление R_2 через гейты $X_{\pi/2}$ и $Z_{\pi/2}$.

За нахождение обратной операции в случае одного кубита отвечает функция find_inverse_1 в модуле main.py.

```
def find_inverse_2(W):
    """Нахождение обратной операции для двух кубитов"""
    q = 0
    R2 = np.linalg.inv(W)
    for i in range (len(g)):
        D = np.array([[1j]*16, [1j]*16])
        p = 0
        for k in range(4):
            for l in range(4):
                D[0][p] = g[i][l][k]
                D[1][p] = R2[l][k]
                p = p + 1
        r = np.linalg.matrix_rank(D)
        if (r == 1):
            show_2(i, W)
            q = i
    return q
```

Примеры обратных операций к случайным последовательностям из 1000 гейтов, получаемых с помощью программы:

```
(2^(-1/2))(1 + i)[((-1)(Z · X · Z · Z) ⊗I) · (I ⊗i(Z · X · Z)) · CZ · (I ⊗i(Z · X · Z)) · ((Z · X · Z · Z) ⊗(Z · X · Z · Z))]


-i[((Z · X · Z · Z) · (Z · X · Z · Z) ⊗(Z · X · Z · Z) · (Z · X · Z · Z)) · (I ⊗i(Z · X · Z)) · CZ · (I ⊗i(Z · X · Z)) · ((Z · X · Z · Z · Z) ⊗(Z · X · Z · Z))]


-1[((-1)(Z · X · Z · Z) ⊗I) · (I ⊗i(Z · X · Z)) · (I ⊗i(Z · X · Z)) · CZ · (I ⊗i(Z · X · Z)) · (I ⊗i(Z · X · Z)) · ((2^(-1/2))(1 + i)Z ⊗(2^(-1/2))(1 + i)Z) ·
(i(Z · X · Z) ⊗I) · CZ · (i(Z · X · Z) ⊗i(Z · X · Z)) · CZ · (i(Z · X · Z) ⊗i(Z · X · Z)) · CZ · (i(Z · X · Z) ⊗I) · ((Z · Z · X) ⊗(Z · X · Z · Z))]
```

Подпрограмма output.py отвечает за вывод обратных операций в символьном виде.

Список литературы

[1] Randomized Benchmarking of Two-Qubit Gates, Samuel Haberthur, Swiss Federal Institute of Technology Zurich, Department of Physics, Laboratory for Solid State Physics, Quantum Device Lab, August 28, 2015

[2] Process verification of two qubit quantum gates by randomized benchmarking [Text] / A. D. Corcoles [et al.] // Physical Review A, Atomic, Molecular, and Optical Physics. — 2013. — Vol. 87, no. 3.