University of Science and Technology Houari Boumediene

Faculté d'Informatique

Master MIV 1

Projet : Quantum Image Processing

Academic year 2025–2026

FRQI Model Implementation

(Part 2 – Group 4)

Prepared by :
Djaoud Khadidja
Guarbi Feriel

Introduction

Quantum computing is a relatively new concept in science and technology, and research is underway to enable its widespread use in solving problems that seem too complex for current conventional computers. Despite the differences and complexity of quantum computers, the potential for significantly increasing the speed and efficiency of problem-solving has sparked interest among researchers worldwide in exploring its applications. Even problems that seem extremely complex for supercomputers, such as modeling atoms in a compound, can be performed with relative ease by quantum computers. Quantum computers and quantum technology in general are currently used in diverse applications, such as electric vehicles, solving complex energy challenges, exploring the mysteries of space and the universe, image processing, and many others . Quantum image processing is one of the most exciting areas where quantum science and technology are used. With tremendous technological advancements, image processing has become a broad research field. It is used in various disciplines and research areas. Facial recognition technologies, robotic vehicles, Photoshop, and many other technologies rely on image processing as their foundation. The core of quantum image processing lies in the use of quantum technology.

Image compression, edge detection, image storage and retrieval, and binary image line detection are just some of the tasks that can be accomplished through quantum image processing. To perform quantum image processing, the image must first be converted into its quantum counterpart, known as the quantum image. This state can be achieved through a number of different operations such as FRQI (Flexible Quantum Image Interface), NEQR (New Enhanced Quantum Image Interface), QBIP (QoD Logical Quantum Image Processing), and others. Here, we discuss FRQI technology in detail. The FRQI state represents classical images after conversion and is one of the image representations in a quantum computer.

The mathematical aspect of The FRQI :

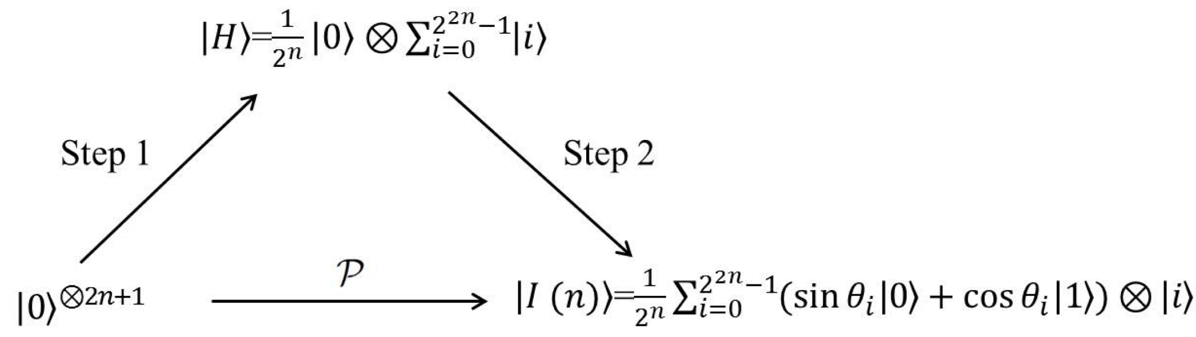
The Flexible Representation Of Quantum Images also known as Fourier Deconstruction Quantum Imaging is a mathematicale Formula that allows as to presents an image in quantum computers .

This representation is based on two types of quantum bits the **position** qubits and **color** qubit , an image $2^n \times 2^n$ is represented by 2n + 1 quantum bits , 2n qubit for positions , and one for intensities

Mathematical Formula :

The FRQI state contains coded information in the form of color and it's related position as show below .

FRQI state is prepared through a unitary transformation which has two steps , after all qubits are initialized to state 0 we: **Apply the Hadamard transform** on position qubits, and then **Applying controlled rotations on the $|H\rangle$ state**.[2]



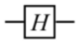
$$|I(\theta)\rangle = \frac{1}{2^n}\sum_{i=0}^{2^{2n}-1}\Big(\cos\theta_i\,|0\rangle + \sin\theta_i\,|1\rangle\Big) \otimes |i\rangle$$

Where

$$\theta_i \in \left[0, \frac{\pi}{2}\right], \qquad i = 1, 2, \ldots, 2^{2n} - 1.$$

1. **Applying the Hadamard transform :**

Hadamard gate :

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

This gate is used in almost all circuits and quantum algorithmes.

It is a single qubit gate but it can be expend to be applayed on multiple qubit .

We can express the quantum Hadamard gate as unitary matrix , and the qubit as vector . So talking about applaying quantum gate on a qubit is exactlly multipplaying a matrix by a vector .
A key use of Hadamard gate is that we give input a qubit in pure state , and we get as output a qubit in equal superposition .
Matrix of Hadamrd gate :

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

The qubit representation :

$$\psi = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$$

Base State : We represent the states on colomn vector :

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

Applaying Hadamard on $|0\rangle$

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ 1\end{pmatrix}.$$

$$H|0\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big).$$

Applaying Hadamard on $|1\rangle$

$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ -1\end{pmatrix}.$$

$$H|1\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big).$$

Measurement probabilities :

For a general state :

$\psi = a|0\rangle + b|1\rangle$,

the probability of measuring $|0\rangle$ is $|a|^2$ and that of measuring $|1\rangle$ is $|b|^2$.

So :

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \Rightarrow P(0) = \frac{1}{2}, \quad P(1) = \frac{1}{2}.$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \Rightarrow P(0) = \frac{1}{2}, \quad P(1) = \frac{1}{2}.$$

a and b are the root of the probability of getting the state 1 or 0 and it called Amplitude.

Probability = $(|\text{Amplitude}|)^2$

Important properties :

- H is unitary :

$$H^\dagger H = I.$$

- H is its own inverse :

$$H^2 = I.$$

  That is, applying the Hadamard gate twice to any quantum state returns the state itself.

  Proof :

$$H\big(H\,|1\rangle\big) = H\Big(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\Big) = \frac{1}{\sqrt{2}}\big(H\,|0\rangle - H\,|1\rangle\big)$$

$$= \frac{1}{\sqrt{2}}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} - \frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$= \frac{1}{2}\Big((|0\rangle + |1\rangle) - (|0\rangle - |1\rangle)\Big)$$

$$= \frac{1}{2}\Big(|0\rangle + |1\rangle - |0\rangle + |1\rangle\Big)$$

$$= \frac{1}{2}(2\,|1\rangle) = |1\rangle\,.$$

$$\boxed{H(H\,|1\rangle) = |1\rangle}$$

$$H\Big(H\,|0\rangle\Big) = H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) = \frac{1}{\sqrt{2}}\Big(H\,|0\rangle + H\,|1\rangle\Big)$$

$$= \frac{1}{\sqrt{2}}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$= \frac{1}{2}\Big((|0\rangle + |1\rangle) + (|0\rangle - |1\rangle)\Big)$$

$$= \frac{1}{2}\Big(|0\rangle + |1\rangle + |0\rangle - |1\rangle\Big)$$

$$= \frac{1}{2}(2\,|0\rangle) = |0\rangle\,.$$

$$\boxed{H(H\,|0\rangle) = |0\rangle}$$

Applaying Hadamard gate on multiple qubits :

$$\boxed{H^{\otimes n}\,|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{x\cdot y}\,|y\rangle}$$

Where

$$x \cdot y = \sum_{i=1}^{n} x_i y_i \pmod 2.$$

Proof :

Hadamard on 1 qubit. The Hadamard matrix applied on one qubit is :

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

Its tronsformation on the basic states is written, like this $x \in \{0,1\}$,

$$H|x\rangle = \frac{1}{\sqrt{2}} \sum_{y \in \{0,1\}} (-1)^{xy} |y\rangle ,$$

where, here, $x.y$ is the product in $\{0,1\}$ (equivalent to integer multiplication, then taken modulo 2 if we want to specify).

Quick check: for $x = 0$, $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and for $x = 1$, $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. These two cases correspond well to the formula above.

Transition to n qubits per tensor product. The Hadamard on n qubits is defined by $H^{\otimes n} = H \otimes H \otimes \cdots \otimes H$ (tensor product n times). Let a basic state $|x\rangle$ with $x = (x_1, \ldots, x_n) \in \{0,1\}^n$,

$$|x\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle .$$

Let's apply $H^{\otimes n}$ à $|x\rangle$using the bilinearity of the tensor product:

$$\begin{aligned} H^{\otimes n}|x\rangle &= \big(H|x_1\rangle\big) \otimes \big(H|x_2\rangle\big) \otimes \cdots \otimes \big(H|x_n\rangle\big) \\ &= \left(\frac{1}{\sqrt{2}} \sum_{y_1 \in \{0,1\}} (-1)^{x_1 y_1} |y_1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}} \sum_{y_2 \in \{0,1\}} (-1)^{x_2 y_2} |y_2\rangle\right) \otimes \cdots \\ &\quad \cdots \otimes \left(\frac{1}{\sqrt{2}} \sum_{y_n \in \{0,1\}} (-1)^{x_n y_n} |y_n\rangle\right). \end{aligned}$$

By factoring the constants $1/\sqrt{2}$ (one for each qubit) and by expanding the product of sums, we obtain a sum over all the n-bits $y = (y_1, \ldots, y_n) \in \{0,1\}^n$:

$$\begin{aligned} H^{\otimes n}|x\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y_1,\ldots,y_n \in \{0,1\}} \Big((-1)^{x_1 y_1}(-1)^{x_2 y_2} \cdots (-1)^{x_n y_n}\Big) |y_1\rangle \otimes |y_2\rangle \otimes \cdots \otimes |y_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{\sum_{i=1}^n x_i y_i} |y\rangle . \end{aligned}$$

Interpretation using the binary scaler product:

The binary scaler product is defined:

$$x \cdot y := \sum_{i=1}^{n} x_i y_i \pmod 2.$$

So :

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle .$$

This is exactly the general formula.

In FRQI, we apply the Hadamard gate only to the position register, and the register is initially $|00\ldots0\rangle$.

So instead of using the general Hadamard formula:

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y}|y\rangle,$$

we take the special case where

$$x = 00\ldots0.$$

Special case: $x = 00\ldots0$

Compute the scalar product:

$$x \cdot y = 0,$$

because

$$0 \cdot y_1 + 0 \cdot y_2 + \cdots + 0 \cdot y_n = 0.$$

So

$$(-1)^{x \cdot y} = (-1)^0 = 1.$$

Therefore:

$$H^{\otimes n}|00\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle.$$

In this way we have a uniform superposition over all pixel positions.

2. **Applying controlled rotations Ry on the $|H\rangle$ state :**
After Hadamard , the position qbits are in superposition , in other term all pixels are existing at once Now we need to encode the pixel colors (intensities) into the color qubit

Assign θ values to each pixel intensity:
Each pixel intensity (0–255) is converted into an angle using this formula:

$$\theta_i = \frac{\pi}{2} \cdot \frac{\text{pixel}}{255}$$

Example with 4 pixels: Let the pixel intensities be 0, 64, 128, and 255. Their corresponding angles are calculated as:

Pixel	Intensity	θ_i (rad)
1	0	0
2	64	$\frac{\pi}{2} \cdot \frac{64}{255} \approx 0.394$
3	128	$\frac{\pi}{2} \cdot \frac{128}{255} \approx 0.789$
4	255	$\frac{\pi}{2} \cdot \frac{255}{255} = \frac{\pi}{2} \approx 1.571$

So the color cubit is represented by : (0 , 0.39 , 0.789 , 1.57)

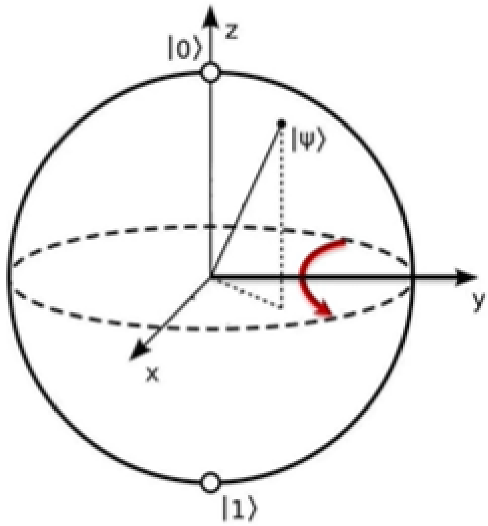
Rotation Gate Ry :

Consider the rotation matrices $R_y(2\theta_i)$ (rotation along the y-axis by the angle $2\theta_i$) and controlled rotation matrices R_i $(i = 0, 1, \ldots, 2^{2n} - 1)$:

$$R_y(2\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}$$

Applying $R_y(2\theta)$ to $|0\rangle$ gives:

$$R_y(2\theta)\,|0\rangle = \cos(\theta)\,|0\rangle + \sin(\theta)\,|1\rangle$$



Controlled-R_y gate (CR_y): The target qubit controlled-R_y gate (CR_y) is a two qubit gate where the first qubit is the control and the second is the target qubit. .

The gate applies a rotation around the y-axis on the taget qubit only when the control qubit is in the state $|1\rangle$.

But to clarify how to use this gate in FRQI , we are gonna explain first , how it's used on 2 qubits (control , target):

We start with the single-qubit rotation around the y-axis:

$$R_y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}.$$

The computational basis for 2 qubits is:

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle$$

Control = 0: states $|00\rangle, |01\rangle \rightarrow$ apply identity:

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

where I_2 is the 2×2 identity matrix acting on the target qubit when the control qubit is $|0\rangle$.
Control = 1: states $|10\rangle, |11\rangle \rightarrow$ apply $R_y(\theta)$ to the target:

$$R_y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

To make it a controlled rotation (CR_y) with a control qubit and a target qubit, we define[1]:

$$CR_y(\theta) = \begin{bmatrix} I_2 & 0 \\ 0 & R_y(\theta) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\theta/2) & -\sin(\theta/2) \\ 0 & 0 & \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

We represent this matrix by this formula :

$$CR_y(\theta) = I \otimes |0\rangle\langle 0| \; + \; R_y(\theta) \otimes |1\rangle\langle 1|.$$

This is the controlled-R_y gate on one controle qubit , but in our case , we have n controle qubits .A multi-controlled rotation applied only when the control register is in state $|i\rangle$ is:

$$R_i = I \otimes \sum_{j \neq i} |j\rangle\langle j| \; + \; R_y(2\theta_i) \otimes |i\rangle\langle i|.$$

For every position $|j\rangle$ with $j \neq i$, it applies the identity on the color qubit (so nothing happens). When the position register is in the state $|i\rangle$, it applies the rotation $R_y(2\theta_i)$ to the color qubit. Thus, R_i performs the rotation only on the pixel indexed by i.

Construction of the FRQI state using Ri- controlled operators [3] :

Initial stat :

The initial stat is a uniform superposition of positions with a color qubit initilised by $|0\rangle$:

$$|H\rangle = \frac{1}{2^n}|0\rangle \otimes \sum_{i=0}^{2^{2n}-1} |i\rangle.$$

We're gonna show how the successive application of the $\{R_i\}$ is going to lead us exactly to the FRQI state :

Application of R_k on the $|H\rangle$ state :

We separate the component corresponding to position k :

$$|H\rangle = \frac{1}{2^n}\left(|0\rangle|k\rangle + \sum_{i=0,\, i\neq k}^{2^{2n}-1} |0\rangle|i\rangle\right).$$

For each $i \neq k$, R_k acts as the identity :

$$R_k(|0\rangle|i\rangle) = |0\rangle|i\rangle.$$

For the position k, we applay the rotation :

$$R_y(2\theta_k)|0\rangle = \cos\theta_k|0\rangle + \sin\theta_k|1\rangle.$$

Thus :

$$\boxed{R_k|H\rangle = \frac{1}{2^n}\left[\sum_{i=0,\, i\neq k}^{2^{2n}-1} |0\rangle|i\rangle + (\cos\theta_k|0\rangle + \sin\theta_k|1\rangle)|k\rangle\right].}$$

Application of $R_l R_k$ on $|H\rangle$

Then , we applay R_l with the same way :

$$\boxed{R_l R_k|H\rangle = \frac{1}{2^n}\left[\sum_{i=0,\, i\neq k,l}^{2^{2n}-1} |0\rangle|i\rangle + (\cos\theta_k|0\rangle + \sin\theta_k|1\rangle)|k\rangle + (\cos\theta_l|0\rangle + \sin\theta_l|1\rangle)|l\rangle\right].}$$

Each operator R_i only modifies the contribution corresponding to address i.

Product of all rotations :

By successively applying all the operators $R_0, R_1, \ldots, R_{2^{2n}-1}$, we obtain :

$$\mathcal{R}|H\rangle = \left(\prod_{i=0}^{2^{2n}-1} R_i \right) |H\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} (\cos\theta_i|0\rangle + \sin\theta_i|1\rangle) \otimes |i\rangle.$$

Final State: FRQI

We find exactly the definition of the FRQI quantum state :

$$\boxed{|\psi_{FRQI}\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} (\cos\theta_i|0\rangle + \sin\theta_i|1\rangle) \otimes |i\rangle.}$$

Exemple :

Let's apply the FRQI on 2x2 matrix (4 pixels) :
First, applying the Hadamard transform on the 2 position qubits and then we have this initial state[2]:

$$|H\rangle = \frac{1}{2} |0\rangle \otimes \sum_{i=0}^{3} |i\rangle.$$

State $|0\rangle$ is initialised on all three qubits and the Hadamard gate is applied on the first two, creating superposition. The third qubit is the color qubit. In the second step, controlled rotations are applied on the state $|H\rangle$ as defined by:

$$R_i = \left(I \otimes \sum_{\substack{j=0 \\ j\neq i}}^{3} |j\rangle\langle j| \right) + \left(R_y(2\theta_i) \otimes |i\rangle\langle i|\right),$$

The controlled rotations are applied in succession corresponding to the number of pixels, which in our case is 4. This corresponds to the unitary operation R defined as:

$$R|H\rangle = \prod_{i=0}^{3} R_i\,|H\rangle. \tag{5}$$

The initial state is the state obtained after applying the Hadamard transform. Now, the controlled rotation operators are applied in succession as follows.

—

First rotation

$$\begin{aligned} R_0|H\rangle &= \left(I \otimes \sum_{i=0,\, i\neq 0}^{3} |i\rangle\langle i| + R_y(2\theta_0) \otimes |0\rangle\langle 0|\right)\left(\frac{1}{2}|0\rangle \otimes \sum_{i=0}^{3}|i\rangle\right) \\ &= \frac{1}{2}\left[|0\rangle \otimes \sum_{i=0,\, i\neq 0}^{3} |i\rangle + (\cos\theta_0|0\rangle + \sin\theta_0|1\rangle) \otimes |0\rangle\right]. \end{aligned}$$

—

Second rotation

$$\begin{aligned} R_1(R_0|H\rangle) &= \left(I \otimes \sum_{i=0,\, i\neq 1}^{3} |i\rangle\langle i| + R_y(2\theta_1) \otimes |1\rangle\langle 1|\right)(R_0|H\rangle) \\ &= \frac{1}{2}\Big[|0\rangle \otimes \sum_{i=0,\, i\neq 0,1}^{3} |i\rangle + (\cos\theta_0|0\rangle + \sin\theta_0|1\rangle) \otimes |0\rangle \\ &\quad + (\cos\theta_1|0\rangle + \sin\theta_1|1\rangle) \otimes |1\rangle\Big]. \end{aligned}$$

—

Third rotation

$$\begin{aligned} R_2(R_1R_0|H\rangle) = \frac{1}{2}\Big[&|0\rangle \otimes \sum_{i=0,\, i\neq 0,1,2}^{3} |i\rangle + (\cos\theta_0|0\rangle + \sin\theta_0|1\rangle) \otimes |0\rangle \\ &+ (\cos\theta_1|0\rangle + \sin\theta_1|1\rangle) \otimes |1\rangle + (\cos\theta_2|0\rangle + \sin\theta_2|1\rangle) \otimes |2\rangle\Big]. \end{aligned}$$

—

Fourth rotation

$$
\begin{aligned}
R_3(R_2R_1R_0|H\rangle) = \frac{1}{2}\Big[&(\cos\theta_0|0\rangle + \sin\theta_0|1\rangle) \otimes |0\rangle + (\cos\theta_1|0\rangle + \sin\theta_1|1\rangle) \otimes |1\rangle \\
&+ (\cos\theta_2|0\rangle + \sin\theta_2|1\rangle) \otimes |2\rangle + (\cos\theta_3|0\rangle + \sin\theta_3|1\rangle) \otimes |3\rangle\Big].
\end{aligned}
$$

—

The states $\{|i\rangle\}_{i=0}^{3}$ correspond to the 2-qubit basis states in the Zeeman computational basis:

1. $|0\rangle = |00\rangle$ 2. $|1\rangle = |01\rangle$ 3. $|2\rangle = |10\rangle$ 4. $|3\rangle = |11\rangle$

Hence, the final FRQI state obtained is:

$$
\frac{1}{2}\begin{bmatrix} (\cos\theta_0|0\rangle + \sin\theta_0|1\rangle) \otimes |00\rangle \\ + (\cos\theta_1|0\rangle + \sin\theta_1|1\rangle) \otimes |01\rangle \\ + (\cos\theta_2|0\rangle + \sin\theta_2|1\rangle) \otimes |10\rangle \\ + (\cos\theta_3|0\rangle + \sin\theta_3|1\rangle) \otimes |11\rangle \end{bmatrix}.
$$

The Practical aspect of The FRQI on Qiskit :

Befor we get start with the code , we will explain the functions thet we are going to use :

- h(qubit) : The hadamard gate

QuantumCircuit (int) : creates a new quantum circuit with the given number of qubits.

- x(qubit) : applies the NOT gate on the specified qubit.
- mcry(angle, [controle_qubits], target_qubit, mode) : Multi-Controlled RY gate, Apply a rotation around Y axes only if all the control qubits are in the correct state.

To demonstrate the FRQI on Qiskit we are going to use this matrix :

```
img = np.array([
    [25, 100],
    [180, 230]
], dtype=float)
```

Then , we need to transform each pixel intensity to an angle :

```
# Transformation pixel     angle     = (  /2)*(pixel/255)
theta = (np.pi/2) * (img.flatten() / 255.0)
```

Calulate the number of qubits :

```
Calulate the number of qubits :
nb_pixels = img.size            # ex : 4
nb_addr = int(math.log2(nb_pixels))    # ex : 2 qubits
color = nb_addr
```

Create a cirquit with nb_addr + 1

```
qc = QuantumCircuit(nb_addr + 1)
```

Applay Hadamard gate on all position qubits

```
for q in range(nb_addr):
    qc.h(q)
```

Applay the MCRy sur chaque position

```
# ----- PIXEL 0      adresse |00> -----
qc.x(1); qc.x(0)                                  # qubits d'adresse
qc.mcry(2*theta[0], [1,0], color, mode='noancilla')
qc.x(1); qc.x(0)

# ----- PIXEL 1      adresse |01> -----
qc.x(1)
qc.mcry(2*theta[1], [1,0], color, mode='noancilla')
qc.x(1)

# ----- PIXEL 2      adresse |10> -----
qc.x(0)
qc.mcry(2*theta[2], [1,0], color, mode='noancilla')
qc.x(0)

# ----- PIXEL 3      adresse |11> -----
qc.mcry(2*theta[3], [1,0], color, mode='noancilla')
```

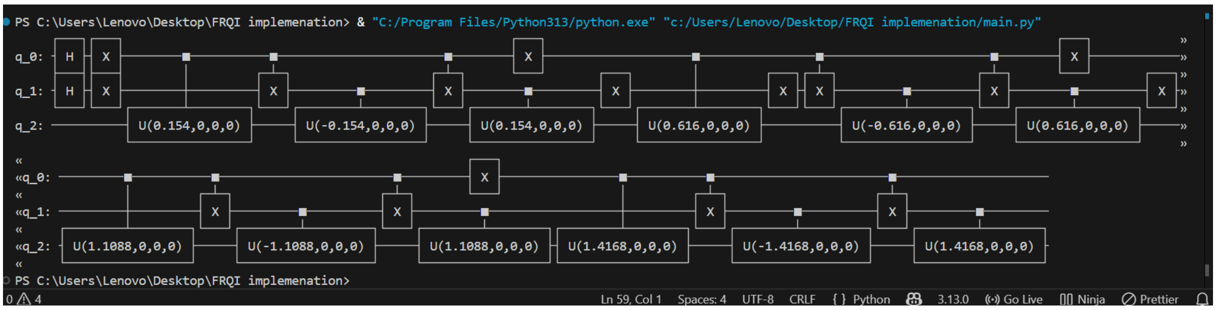
Befor using the mcry we need to make sur that the postion (control) qubits are all in state 1 , cause the rotation will not work otherwise , so we test all the probabilities for all positions and invers the 0 to 1 using the X gate, for exemple if the register is |00> , then when he applays qc . x (1) ; qc . x (0) both the 2 qubits are going to be 1 , so the mrcy is applayed , and the color qubit is going to be rotated by the angle teta0 , and this condition will fail for all other tests .

After inversing we use the mrcy with is CRy with multiple control qubits with the parameters the angle : 2* theta [i] , the control qubits : [1,0] , the target qubit : color and the mode = 'noancilla' .

At the end we have to return the qubit back to his initial stat by inversing it again .

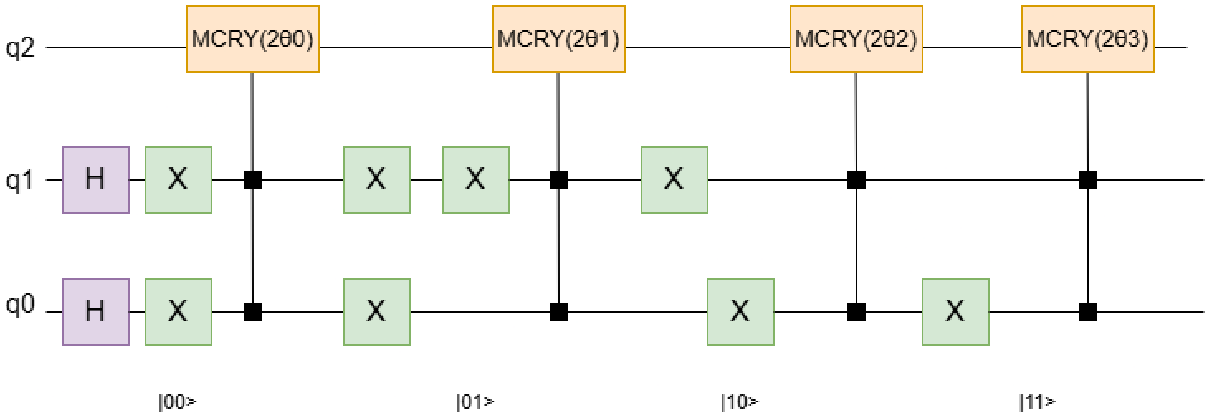
Circuit display

```
print(qc.draw())
```



We are going to see a lot of gates X and U (cry) applayed even thought i did not applay them on my code , that's because Qiskit devide the mrcy gate to multiple base gate to acheave the mrcy .

Here is a simplified circuit of our image :



FRQI Limits :

(a) **High circuit complexity.** Preparing an FRQI image requires a controlled rotation for each pixel. For an image with 2^n pixels, the state preparation needs 2^n controlled-R_y gates, which leads to deep circuits and hug complexity.

(b) **Only single–channel (grayscale) representation.** FRQI encodes each pixel intensity into a single angle θ_k. It cannot directly represent RGB images without extensions.

(c) **Probabilistic measurement.** The colour of each pixel is stored in amplitudes. Reconstructing the full image requires many measurements (shots), which increases error sensitivity.

(d) **Incompatibility with arbitrary image sizes.** FRQI requires that the number of pixels be a power of two. It only supports images of size $2^n \times 2^n$. Non-square or non-power-of-two images (such as 3×3 or rectangular images) must be padded into the nearest $2^n \times 2^n$ grid, wasting qubits and increasing circuit size.

Conclusion:

FRQI provides an elegant and compact way to encode images within a quantum system using position qubits and one color qubit. Due to quantum parallelism, this representation enables global operations to be applied to all the pixels simultaneously. However, its feasibility relies heavily on the accuracy of controlled rotations and the depth of the circuit required to prepare the state. Moreover, it is still hard to implement for high-resolution images. Despite such limitations, FRQI is an important foundation for future work in quantum image processing.

Bibliography

[1] MathWorks, *cryGate — Controlled Y-Rotation Gate*, Available at: https://github.com/Qiskit/textbook/blob/main/notebooks/ch-applications/image-processing-frqi-neqr.ipynb, accessed 2025.

[2] F. Yan, A. M. Iliyasu, and Z. Jiang, *Quantum Computation-Based Image Representation, Processing Operations and Their Applications*, Entropy, vol. 16, pp. 5290–5338, 2014. DOI: 10.3390/e16105290. Available at: https://physlab.org/wp-content/uploads/2023/04/QuantumImageRepresentation_22120005.pdf.

[3] G. R. Haider and W. Rizwan, *Quantum Image Processing — FRQI Image Representation*, Syed Babar Ali School of Science and Engineering, Lahore University of Management Sciences, 2022. Available at: https://www.mdpi.com/1099-4300/16/10/5290.

[4] I. Glover, *FRQI Tutorial (Jupyter Notebook)*, GitHub Repository, 2022. Available at: https://github.com/isaacglo/FRQI-tutorial/blob/main/FRQI_Tutorial.ipynb.

[5] Qiskit Community, *FRQI and NEQR Quantum Image Processing Tutorial*, Qiskit Textbook Notebook (GitHub), 2023. Available at: https://github.com/Qiskit/textbook/blob/main/notebooks/ch-applications/image-processing-frqi-neqr.ipynb.

[6] IBM, *What Is Quantum Computing?* Available at: https://www.ibm.com/topics/quantum-computing, accessed 2025.