

IBM2021 量子挑戰賽

作者：黃琮暉、張仁瑀、徐育兆、林祐睿、林橋毅

作者簡介

黃琮暉是臺灣大學物理學博士，現任職於中原大學資訊工程系與中原大學量子資訊中心，研究的領域是量子計算、量子資訊。

張仁瑀、徐育兆、林祐睿、林橋毅是學生量子電腦交流會的成員。

1981 年麻省理工學院與 IBM 舉辦的一場物理計算會議開啟了量子計算元年 [1]。在 2021 年 5 月 IBM 爲了慶祝量子計算 40 年舉辦線上量子挑戰賽，其中提及了量子計算中 5 個非常重要的議題，如下圖 1。而這 5 題也代表了量子電腦從理論邁入實作進而進入即將應用於實際課題的重要轉捩點，因此格外有意義。

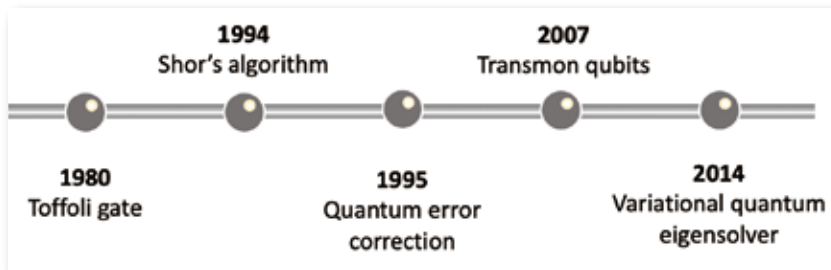


圖 1：2021 年 IBM 挑戰賽的 5 大題。

第一題：利用單位閘（CX, RZ, SX, X 閘）組成托弗里閘

1980 年代，美籍教授托弗里（Tommaso Toffoli）建構托弗里閘（Toffoli gate）[2-3]，其重要性是因為量子計算中可以藉由托弗里閘組成古典電腦中的 NAND 和 AND 閘，這些邏輯閘是建構古典邏輯運算的根基。除此之外在設計量子演算法最核心又重要的「神諭」（oracle）時，常常會有托弗里閘的存在，因此托弗里閘是非常重要的量子邏輯閘。除此之外根據數學理論，量子邏輯閘的通用閘（universal gate）只需要所有的單閘與 CNOT 閘（所謂的所有單閘爲可以實現布洛赫球面上任一點，

CNOT 閘爲控制否閘，而通用量子邏輯閘也可以是所有的單閘加上 SWAP 閘：交換閘）。因此 IBM 挑戰賽的第一題便是利用單位閘（CX, RZ, SX, X 閘）組成托弗里閘。

這一題不僅是一個數學題，我們當然可以找出每一個矩陣並算出如何符合這個 8×8 矩陣的最佳解（其實在維基百科就可以找到解答），只需要做些許的修改便能通過挑戰，這並不代表沒有深度，它的價值在於將我們帶回 50 年前的量子電腦初期，用當時科學家手邊有的資訊來創建這一個邏輯閘，學到的知識有限，卻帶給我們深刻的歷史之旅。而此題我們的解法如下圖 2：

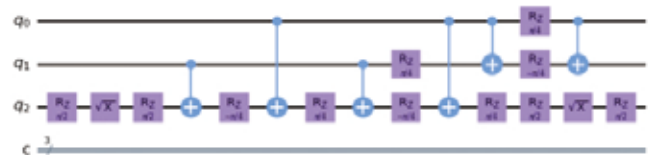


圖 2：第一題解答。可知所需的量子邏輯閘只有 R_Z 、 $\sqrt{X}$ 與 CNOT 閘。

而這題帶給我們更多的知識，當我們回顧托弗里的背景時，才知此閘是由研究可逆計算發展而來的，可逆運算是量子力學的基本定理，因為最後的機率合必爲 1，設計邏輯閘要符合量子力學的可逆原理來設計，才能發揮量子電腦的量子特性；從 1960 年代人們研究可逆邏輯閘時便想減少計算過程的熱量產生，因為根據熵理論，信息的散失會以熱的形式耗散在環境中，而可逆邏輯閘能實現只將

—— 本文特別致謝科技部的支持，其計畫編號為 MOST111-2119-M-033-001-。 ——

信息狀態從輸入搬移到輸出，不會損失信息，來維持量子態資訊的保持。

第二題：秀爾演算法

在計算中，演算法如何隨著輸入問題的大小而增長來衡量演算法的性能。例如，有一種演算法，該演算法會隨著數位大小而線性增長，這類的問題被視為可解的。然而有一些計算問題，演算法隨著輸入的大小而呈指數級增長，這意味著當問題變大或大到某個地步，地球上的任何計算機也無法解決。1994 年，秀爾（Peter Shor）表明，在量子計算機上能有效的做質因數分解 [4]。這是一個大新聞，因為我們知道最好的經典演算法，面對該類問題為指數級增長。事實上，RSA 加密，就是依賴於考慮對足夠大的半質數（semiprime，只有 1、自己與兩個因數的數）作質因數分解是不可行的。因此秀爾演算法被譽為可能破解 RSA 加密問題。

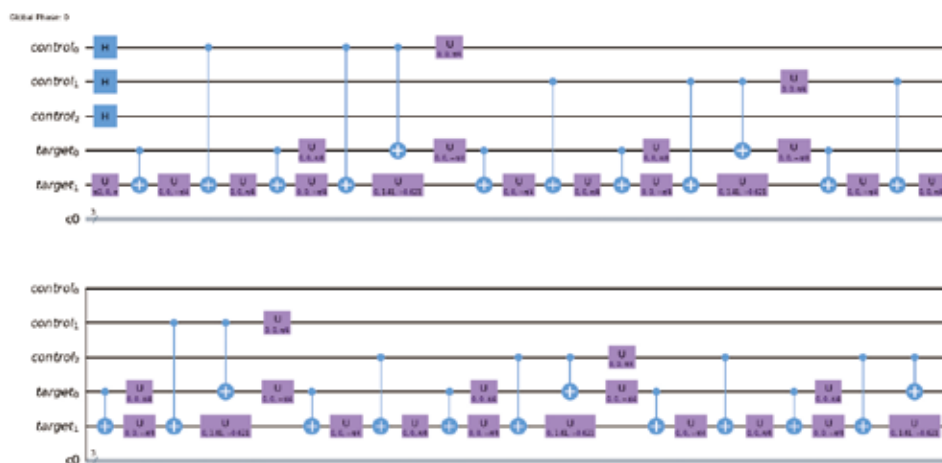
秀爾演算法：

結合古典電腦以及量子電腦，使用量子電腦尋找

週期函數的週期。

量子電腦所處理的流程：

- 1 選擇任意的數字 a ， $a < N$ ，定義 $f(x) = a^x \bmod N$ ；
 - 2 利用秀爾演算法尋找 $f(x)$ 的週期 r ；
 - 3 如果 r 是奇數，就回到第一步
 - 4 如果 r 是偶數就計算 $f(r/2) + 1$ 和 $f(r/2) - 1$ ，該兩數字與 N 有公因數。可利用古典計算中的公因數求解得到 N 的因數。若該因數有意義則完成質因數分解，若該因數無意義則回到第一步。
- 由上可知秀爾演算法創建電路的困難在於創建計算受控 $a^y \bmod N$ 的電路。雖然我們知道如何使用多數的門創建這些電路，但這些電路對於當今的計算機來說仍然太大。幸運的是，如果我們知道一些有關這個問題的資訊，那麼我們有時可以「作弊」[5]，並創建更高效的電路。因此第二題主要是對 35 作質因數分解，並透過已知的資訊創建一個 $35^y \bmod N$ 的電路，該電路可用於秀爾演算法，並且可以在 "ibmq_santiago" 上運行。而此題我們的解法如下圖 3：



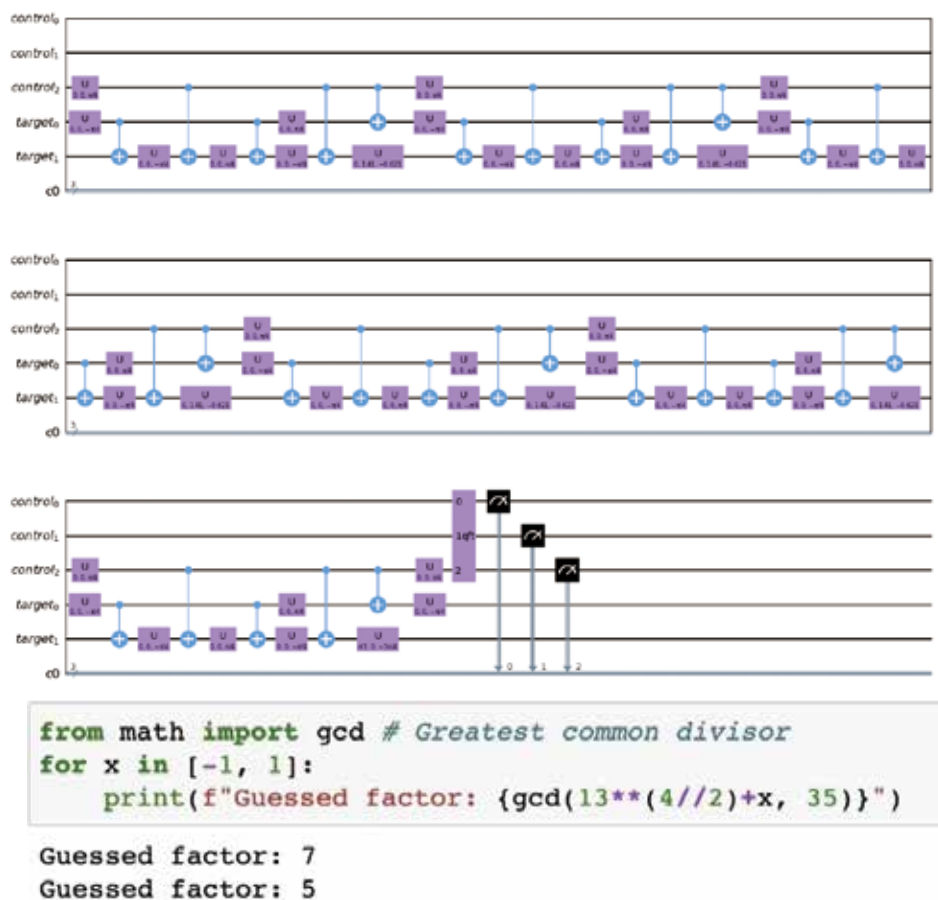


圖 3：上半部為使用秀爾演算法作 35 質因數分解的方法，下圖利用其結果結合古典電腦計算出來的質因數。

第三題：量子糾錯

秀爾演算法為量子計算機提供了一個有價值的應用案例（如第二題），但根據量子力學所有的量子電腦中帶有雜訊，這意味著要構建能夠運行這種演算法的硬體將非常困難。1995 年，秀爾發表了另一篇具有里程碑意義的論文：一個通過多個量子位共用量子資訊以減少錯誤的演算法。[6]

自那以後的幾十年裡已經取得了很大的進展。新的糾錯代碼形式已經被發現，並圍繞它們建立了一個龐大的理論框架。基塔耶夫（Alexei Kitaev）在

1997 年所提出的理論已經成為主要的候選者，從那時起，在原始設計的基礎上出現了许多變體。但在為量子硬體的具體細節定製編碼方面仍有很大的進展。[7]

第三題你需要考慮如何在一個真實的設備上實現量子糾錯的電路。這意味著需要根據量子位元的位置來調整量子位元的編碼。第三題中將可以學到如何利用量子糾錯理論辨識出哪個量子位元出錯以及發生哪種錯誤，並且如何將其實現真實的量子計算上。第三題的解答如下圖 4：另外這一題我們將詳細寫一篇專文描述量子糾錯在未來的重要性。

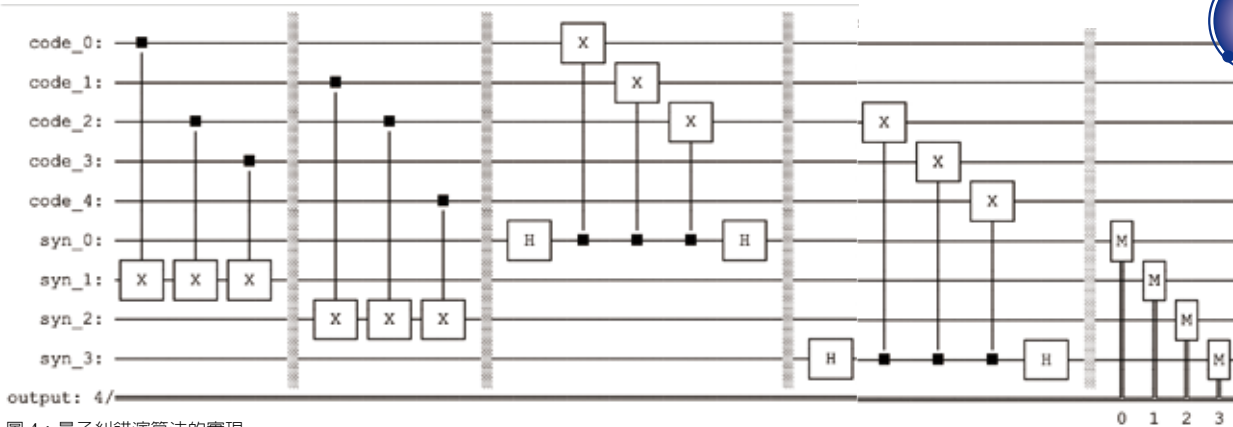


圖 4：量子糾錯演算法的實現。

第四題：Transmon 量子位元

在量子計算中，量子位元是量子訊息的基本單位，它在本質上是一個兩能階的量子力學系統，可以在許多物理系統中實現，包括電子自旋（spin qubit）等自然系統、離子中的原子能級（捕獲離子井）或超導電路中的狀態等人工系統（超導量子位元）。

最早的量子位元發明在 1999 年的庫柏對盒（Cooper pair box）[8]，數量明確的庫柏對（Cooper pair，超導中的束縛電子）佔據一個由約瑟夫森結（Josephson junction）與導線弱耦合的超導體島（an island of superconductor）。在庫柏對盒中的狀態 $|0\rangle$ （沒有庫柏對）與狀態 $|1\rangle$ （有一個庫柏對）的能量差與在系統內或系統周圍的電子分佈有強烈的關係。這種電子造成的雜躁很容易造成量子位元退相位（dephasing）。庫柏對盒量子位元的量子退相位時間 T_2 通常限制在 $\sim 1\mu s$ 。

Transmon 量子位元首次提出於 2007 年 [9]，而發現 transmon 量子位元的關鍵是兩狀態的能量差與電壓（稱為色散）是週期性的關係。通過引入一個分流電容，約瑟夫森能量與電容能量的比值變得非常大 $E_J/E_C \sim 50$ ，導致電荷色散平坦化。雖然庫柏對盒對電荷雜訊非常敏感（即圖 5a），但它在 transmon 量子位元中被極大地抑制（圖 5d）。transmon 量子位元的退相位時間顯著提高。

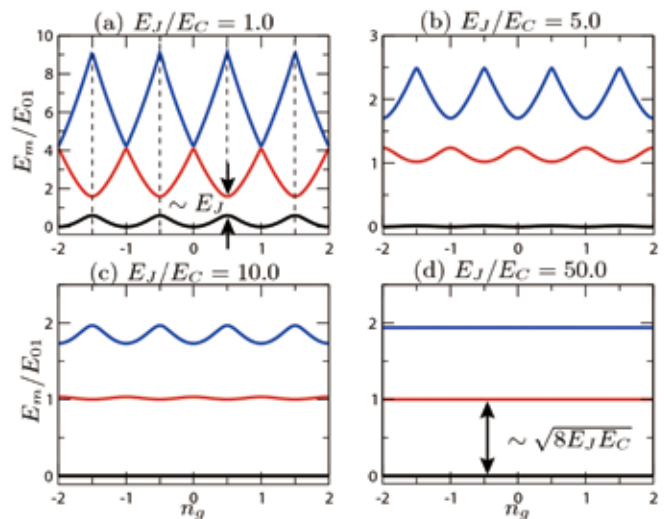


圖 5：在不同 E_J/E_C 下的量子位元的耗散。[9]

在前面三題都是藉由抽象的量子邏輯閘去操作量子位元，而第四題是藉由操作微波脈衝來操作量子位元。在 IBM 系統中是利用 Qiskit Pulses 來操作微波脈衝來操作量子位元。並且利用 Qiskit Pulses 去計算 transmon 量子位元中的 E_J/E_C 。而第四題的解答如下：

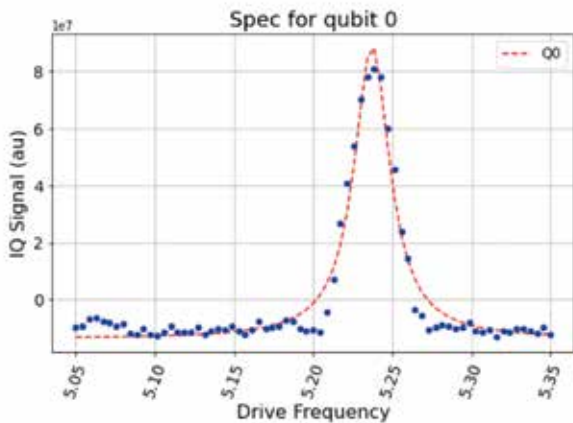
這一題的挑戰簡單來說，使用 Qiskit 提供的 Qiskit pulse 套件，調控脈衝的發射時間，而挑戰的演示如何校準脈衝以區分一個量子位元從狀態 $|0\rangle$ 到狀態 $|1\rangle$ ，問要怎麼做同樣的事情來區分狀態 $|1\rangle$ 到狀態 $|2\rangle$ 。

步驟 1：找到 $|0\rangle \rightarrow |1\rangle$ 過渡的驅動頻率

主要代碼：

```
with pulse.align_sequential():
# Pay attention to this part to solve the problem at the end
pulse.set_frequency(freq*GHz, DriveChannel(qubit))
pulse.play(spec_pulse, DriveChannel(qubit))
pulse.call(meas)
```

通常我們在 5.05 到 5.35GHz 的區間內嘗試一組頻率，並測量對應 sign。當我們達到正確的轉換頻率時，響應將達到峰值，在這種情況下，我們得到的值為 5.237220GHz。



一個量子位元態 (qubit state)，初始狀態為 $|0\rangle$ 經過擬合後，得到吸收頻率為 5.237207GHz。

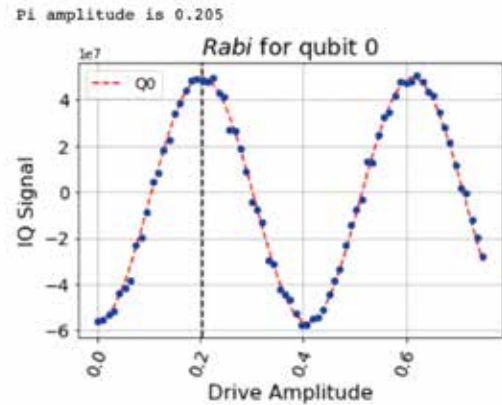
步驟 2：找到 $|0\rangle \rightarrow |1\rangle$ 過渡的驅動幅度

接下來，我們將了解如何校准我們的 pi 脈衝。此步驟的目的是找到將量子位元從 $|0\rangle$ 帶到 $|1\rangle$ 的適當驅動幅度。

主要代碼：

```
with pulse.align_sequential():
pulse.set_frequency(f01*GHz, DriveChannel(qubit))
rabi_pulse = Gaussian(duration=drive_duration, amp=amp,
sigma=drive_sigma, name=f"Rabi drive amplitude = {amp}")
pulse.play(rabi_pulse, DriveChannel(qubit))
pulse.call(meas)
```

我們將執行與步驟 1 類似的過程，只是現在我們一點一點的增加幅度，直到通過查看我們的測量信號再次達到最佳值。



在這種情況下，我們的振幅（方便的用黑色虛線表示）為 0.205。

步驟 3：重複步驟 1，找到驅動頻率，但是這一次為 $|1\rangle \rightarrow |2\rangle$ 過渡

為了開始步驟 3，我們需要獲得 $|1\rangle$ 上的量子位元。我們只需應用步驟 1 和 2 的操作即可做到這一點。此後，我們再次以增量方式掃描頻率間隔，以找到正確的 $|1\rangle$ 到 $|2\rangle$ 驅動頻率。

```
# Promote qubit from  $|0\rangle$  to  $|1\rangle$ 
pulse.play(x_pulse, DriveChannel(qubit))# Incrementally
sweeping the frequency interval to find  $|1\rangle \rightarrow |2\rangle$  transition
frequency
pulse.set_frequency(((freq+anharm_guess_GHz)*GHz,
DriveChannel(qubit))
pulse.play(spec_pulse, DriveChannel(qubit))
```

第五題：變分量子演算法求解 LiH 基態能量

量子計算發展至今，已經漸漸的有雛型並且接近費曼當時的想法——用量子計算模擬自然界的系統，因為只有量子能夠模擬量子世界。而在 2014

年，佩魯佐（Alberto Peruzzo）首次發表了一篇有關於尋找最低能量解的論文——變分量子演算法（VQE）[10]，並且這個方法也比較快速、誤差更低，過了 3 年，2017 年 IBM Qiskit 社群在他們的網站上呈現出了他們用 VQE 演算法模擬氫化鋰（化學式：LiH）的基態能量 [11]。

VQE 演算法最大的特色是它採用混合式的計算模式，用量子電腦作為計算的主體，設計「擬設態」（也就是參數優化目標）、存取系統波函數，並且用古典電腦做參數優化（例如：梯度下降法）。

此題目的目標在於，讓使用者能夠用 Qiskit Nature 套件成功模擬費曼的想法，設計 VQE 演算法來計算 LiH 的基態能量。[12]

而本題也是這次比賽最主要的題目，我們將詳盡的一步一步解說，所有的步驟如下：

1 驅動程式（driver）

首先我們需要輸入我們的目標分子，利用 PSI4Driver 匯入即可。

```
#匯入分子資訊
from qiskit_nature.drivers import PySCFDriver

molecule = 'Li 0.0 0.0 0.0; H 0.0 0.0 1.5474'
driver = PySCFDriver(atom=molecule)
qmolecule = driver.run()
```

2 電子結構問題（electronic structure problem）

我們在此寫下系統的哈密頓量。

```
#寫下系統總能量
from qiskit_nature.problems.second_quantization.electronic import ElectronicStructureProblem
from qiskit_nature.transformers import FreezeCoreTransformer
```

```

# 冷凍可以忽略計算的軌域
freezeCoreTransformer = FreezeCoreTransformer(freeze_core=True,remove_orbitals=[3,4])

#這裡是建立要解決問題的地方，我們改動了函式的相關參數
#可以在這裡找到更多答案：https://qiskit.org/documentation/nature/tutorials/01_electronic_structure.html
qmolecule = freezeCoreTransformer.transform(qmolecule)
problem = ElectronicStructureProblem(driver,q_molecule_transformers=[freezeCoreTransformer])

# 創建二次量子化運算符 - 二次量子化的目的在於得到系統哈密頓量也就是總能量
second_q_ops = problem.second_q_ops()

# 哈密頓量
main_op = second_q_ops[0]

#在這裡我建議print哈密頓量查看
print(main_op)

```

3 量子位元轉換器 (QubitConverter)

接著用二次量子化其映射到量子位元上，由於 QISKIT 皆有完整模組，我們只需套用即可。

```

from qiskit_nature.mappers.second_quantization import ParityMapper, BravyiKitaevMapper, JordanWignerMapper
from qiskit_nature.converters.second_quantization.qubit_converter import QubitConverter
from qiskit.opflow.primitive_ops import PauliSumOp
# 設置映射器和量子比特轉換器 - 在這裡我們轉化哈密頓量為 Qubits的相關參數
mapper_type = 'ParityMapper'

if mapper_type == 'ParityMapper':
    mapper = ParityMapper()
elif mapper_type == 'JordanWignerMapper':
    mapper = JordanWignerMapper()
elif mapper_type == 'BravyiKitaevMapper':
    mapper = BravyiKitaevMapper()

#two_qubit_reduction - 使我們能忽略兩個位元，進而降低CX閘的使用
#z2symmetry_reduction - Z2對稱約化 - 可以搜尋群論找到更多答案
converter = QubitConverter(mapper=mapper, two_qubit_reduction=True,z2symmetry_reduction = [-1])

# 費米子算符映射到量子比特算符
num_particles = (problem.molecule_data_transformed.num_alpha,
                 problem.molecule_data_transformed.num_beta)
qubit_op = converter.convert(main_op, num_particles=num_particles)

```

4 初始態 (initial state)

首先我們使用量子化學中的哈吹／福克演算法 (Hartree-Fock algorithm) 來映射出系統的哈密頓量。

```
from qiskit_nature.circuit.library import HartreeFock

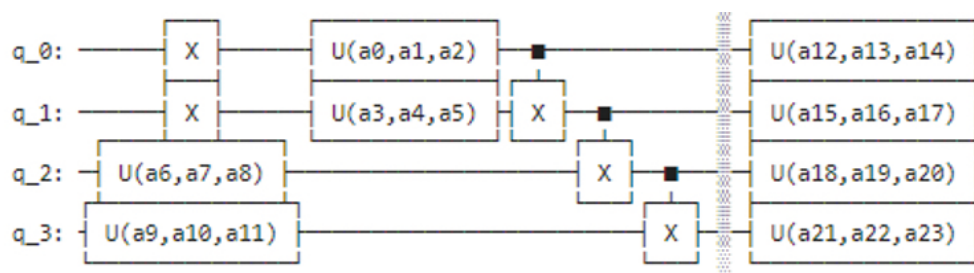
# 在這裡引入HartreeFock作為初始值 - 用來解多體量子系統的波函數
# 相關連結 : https://en.wikipedia.org/wiki/Hartree%E2%80%93Fock\_method

num_particles = (problem.molecule_data_transformed.num_alpha,
                 problem.molecule_data_transformed.num_beta)
num_spin_orbitals = 2 * problem.molecule_data_transformed.num_molecular_orbitals
init_state = HartreeFock(num_spin_orbitals, num_particles, converter)

#我猜可能的波函數
print(init_state)
```

5 擬設態

緊接著我們開始猜測系統的波函數，這裡也是整道題目的重點，我們需要在這裡求解出正確的擬設態使得演算法答案趨近於正確能量解。



這裡會使用 U_3 的原理是優化器 (optimizer) 會自動幫我們旋轉每個 U_3 的 X、Y、Z 三個角度，雖然計算過程會比較久，但能以最小的成本得到基態能量。

6 優化器

在這裡便不是量子的範圍了，這裡採用古典的優化方式求解正確答案，如果結果不正確，那優化器便會給出下一組角度再次放進演算法中求解，不斷的循環直到得到正確答案。

7 VQE 與擬設態的初始參數

因為我們已經將所有的部分都已經建好了，接下來只要使用 IBM 所提供的套件便可直接計算出正確答案，我們可以得知計算結果為

`'optimal_value': -1.0863572878179788,`

```
#匯入 VQE 函式 - 這個是IBM已經寫好的我們可以直接使用
from qiskit.algorithms import VQE
from IPython.display import display, clear_output

# Print and save the data in Lists
def callback(eval_count, parameters, mean, std):
    # Overwrites the same line when printing
    display("Evaluation: {}, Energy: {}, Std: {}".format(eval_count, mean, std))
    clear_output(wait=True)
    counts.append(eval_count)
    values.append(mean)
    params.append(parameters)
    deviation.append(std)

counts = []
values = []
params = []
deviation = []

# Set initial parameters of the ansatz
# We choose a fixed small displacement
# So all participants start from similar starting point
try:
    initial_point = [0.01] * len(ansatz.ordered_parameters)
except:
    initial_point = [0.01] * ansatz.num_parameters
#執行VQE
#在這裡我們可以看到我們上面所定義的參數都在這裡被輸入
algorithm = VQE(ansatz,
                 optimizer=optimizer,
                 quantum_instance=backend,
                 callback=callback,
                 initial_point=initial_point)

result = algorithm.compute_minimum_eigenvalue(qubit_op)

print(result)
```

結語

IBM2021 年的量子挑戰賽相對於以前的挑戰賽而言每一題之間的關係較少，比較沒有連接性，也相較比以往的應用性較低。比較像複習這 40 年來最重要的關鍵技術複習，或許在應用性的價值較低，然而對於使用者來說是非常珍貴的學習機會，而且最重要的是這次的挑戰賽可以讓參加者從底層的硬體到演算法使用、從歷史的發展到現今的展望，面面兼顧實在是非常好的挑戰賽。也希望藉由

這篇文章開啟更多人對量子電腦、量子計算與量子科技的重視。∞

本文參考資料請見〈數理人文資料網頁〉

<http://yaucenter.nctu.edu.tw/periodical.php>

延伸閱讀

- ▶ <https://www.youtube.com/channel/UCrK2BbGnhPrx6tUf1hgfkvg>
這是學生量子電腦交流會的 YouTube 首頁，有許多教學影片。
- ▶ <https://sqcs.tw/>
這是學生量子電腦交流會的網址。

特別介紹：學生量子電腦交流會

「學生量子電腦交流會」社群是爲了集結對於量子科技有興趣的台灣學生，一起學習和討論，至少我們在學習的道路上不是孤單的，並且我們將社群中的學習成果以及各個學習資源統合後以各個是社群軟體分享給社會大眾，讓更多的莘莘學子甚至是一般人士加入我們打造出台灣自己的 Q 世代。

目前我們社群以教育、推廣、交流爲主要方向，我們社群主要經營 Facebook、Instagram 和 YouTube 頻道，在每個星期都會不定時發放貼文，內容涵蓋新聞資訊、量子競賽推廣、量子科技知識推廣等。並且我們會邀請中原大學資工系黃琮暉教授以及 IBM 呂文森博士不定期舉行講座。

而我們最主要的溝通以及交流管道是 Discord，利用 SQCS 的伺服器進行講座和比賽交流，每當有國外的量子競賽甚至是量子黑客松都可以在我們的 Discord 的交流區上進行討論。

我們也會不定期的舉辦活動，例如線上研討會、線上黑客松等並且會在寒暑期時舉行大型量子營隊活動，讓全國對於量子計算有興趣的學生一起在幾天的活動與教學裡認識彼此、精進學習、獲取知識。