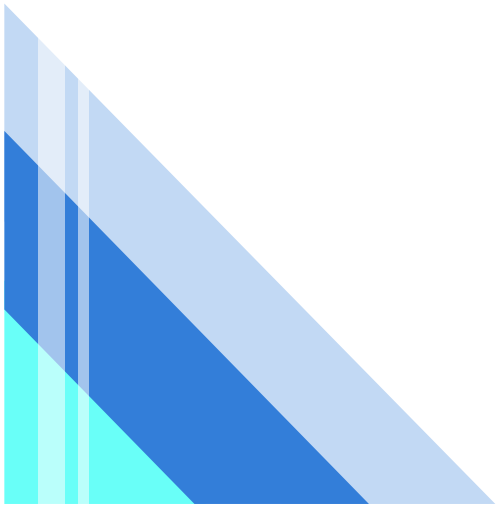
2025 / Spring Semester

Topics on Quantum Computing

Lecture Note – Variational Quantum Algorithm

Junghee Ryu

KISTI

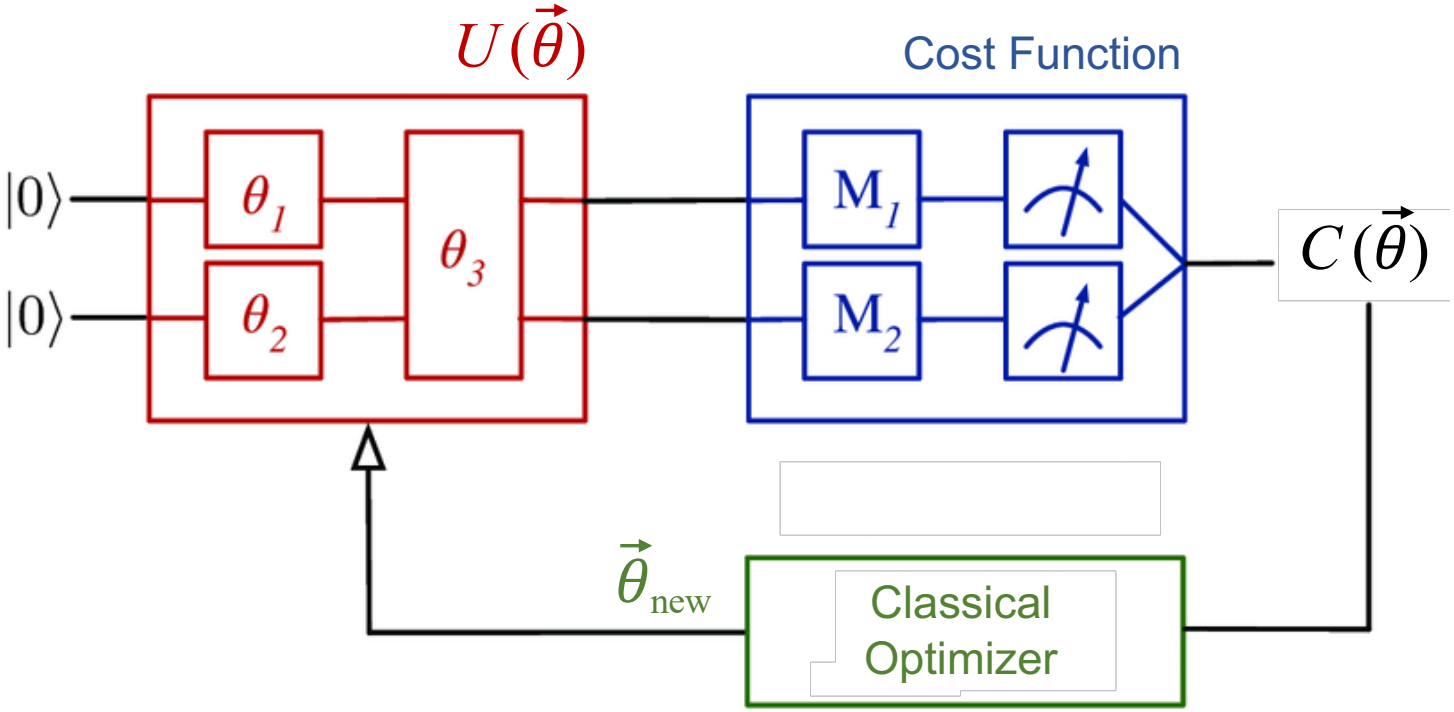


Variational Quantum Algorithm (VQA)

Noise-Intermediate-Scale Quantum (NISQ) algorithm

Variational Quantum Algorithm (VQA)

- Won't correct noise, but overcome with iterative processes



Flow of Operation

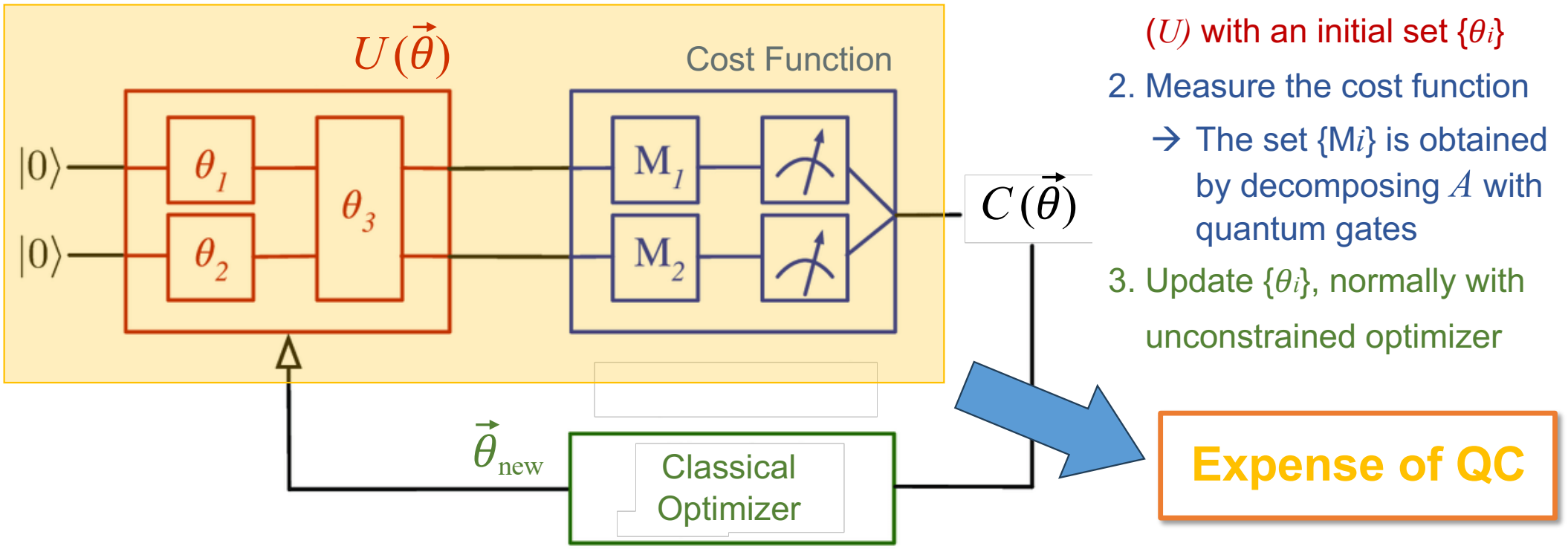
1. Construct the ansatz circuit (U) with an initial set $\{\theta_i\}$
2. Measure the cost function
   → The set $\{M_i\}$ is obtained by decomposing A with quantum gates
3. Update $\{\theta_i\}$, normally with unconstrained optimizer

Variational Quantum Algorithm (VQA)

Noise-Intermediate-Scale Quantum (NISQ) algorithm

Variational Quantum Algorithm (VQA)

- Won't correct noise, but overcome with iterative processes



Flow of Operation

1. Construct the ansatz circuit (U) with an initial set $\{\theta_i\}$
2. Measure the cost function
   → The set $\{M_i\}$ is obtained by decomposing A with quantum gates
3. Update $\{\theta_i\}$, normally with unconstrained optimizer

Variational Quantum Algorithm (VQA)

Noise-Intermediate-Scale Quantum (NISQ) algorithm

Classical Optimizer

- Take multi variables as input & executed in a classical computer
- Find a set of parameters that minimizes the output of a function f

$$c = f(\theta_1, \theta_2, \ldots, \theta_{n-2}, \theta_{n-1})$$

- Broyden-Fletcher-Goldfarb-Shanno (BFGS) & COBYLA method (supported by SciPy)
- PENNYLANE also support a python-native Adam Optimizer (stochastic optimization)

```
class AdamOptimizer(stepsize=0.01, beta1=0.9, beta2=0.99, eps=1e-08)
```

Two most well-known VQAs

- Variational Quantum Eigensolver (VQE): Approximate the lowest eigenpair of a symmetric matrix
- Variational Quantum Linear Solver (VQLS): Approximate the solution of a linear system described with a symmetric matrix

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

General sequence of a VQE process

- Let's say we have a Hermitian matrix = H, and its eigenpair = (λ(eigenvalue), ψ(eigenvector)). Then, we have the following equation,

$$H\psi = \lambda\psi$$

- Since ψ is usually represented with a normalized vector, it can be (in principle) a quantum state and the above equation becomes as follows,

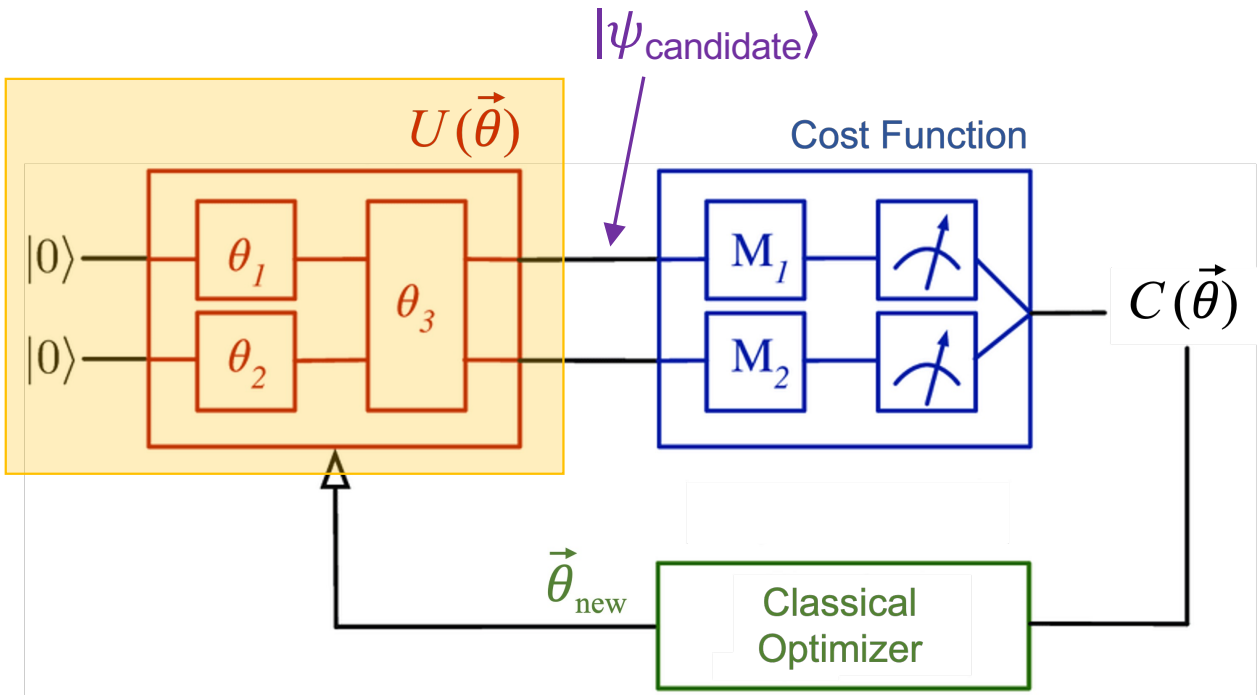
$$\langle\psi|H|\psi\rangle = \lambda\langle\psi|\psi\rangle = \lambda$$

- So, the general sequence of a VQE process becomes:
  1) Get a candidate of ψ from a given set of parameters
  2) Compute the expectation value $C = \langle\psi|H|\psi\rangle$
  3) Check if c converges: (if yes) the process is completed (if no) go to the step 4)
  4) Secure a new set of parameters through optimization and go to the step 1)

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

STEP 1): Get a candidate of ψ from a given set of parameters



- Design an Ansatz circuit U such that

$$|\psi_{candidate}\rangle = U(\theta_1, \ldots, \theta_{n-1})|0\rangle^{\otimes m}$$

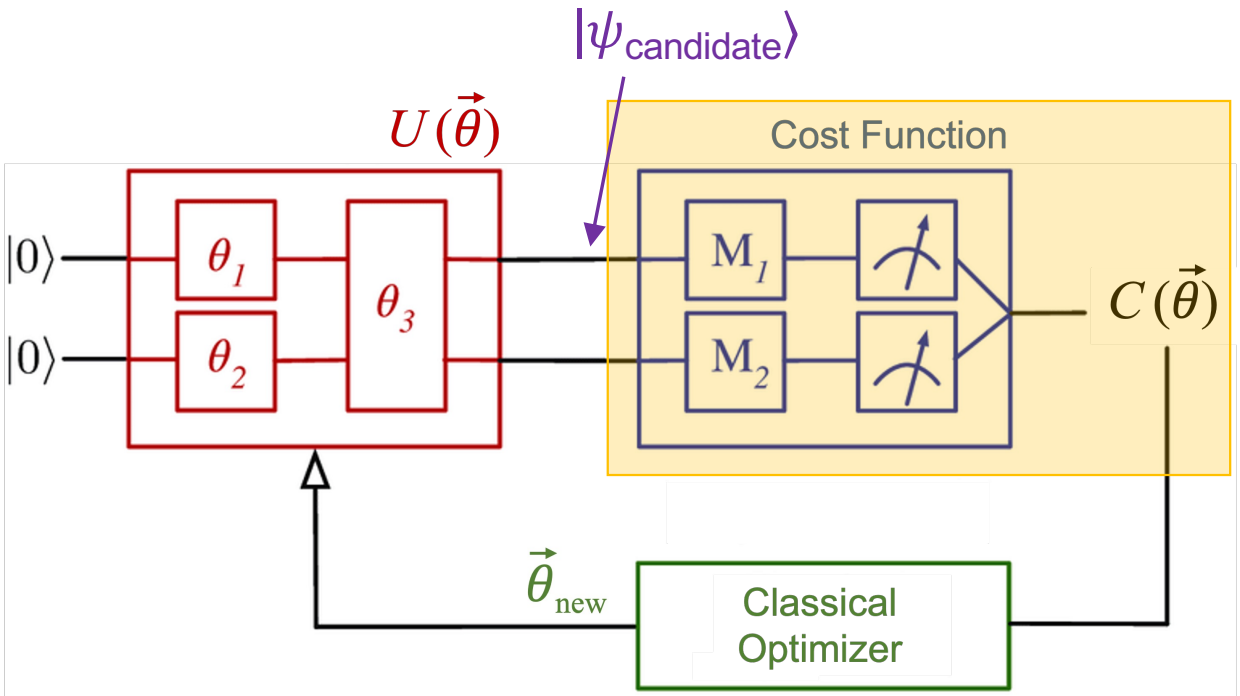
- The set of parameters $(\theta_1, \ldots, \theta_{n-1})$ consist of rotating angles of logic gates
- Design of cost-efficient Ansatz circuits is one of important research topics in quantum computation

  → Cost-efficiency is defined as # of quantum logic gates & circuit layers

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

STEP 2): Compute the expectation value $\mathbf{C} = \langle \psi_{candidate} | H | \psi_{candidate} \rangle$



- Design a cost-function C
  - → C is also a quantum circuit that takes $|\psi_{candidate}\rangle$ as the input
  - → Design of the C circuit is all about measurements (Recall the case of a single-qubit case - what do the following equations mean?)

$$\langle \psi | Z | \psi \rangle, \langle \psi | Y | \psi \rangle, \langle \psi | X | \psi \rangle$$

- Design of a cost-efficient cost-function is also an important research topics
  - → Cost-efficiency is defined as # of quantum logic gates & circuit layers

Variational Quantum Eigensolver (VQE)

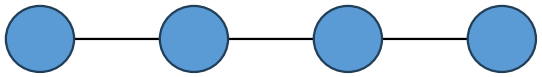
Find the lowest energy system in the quantum system

STEP 2): Compute the expectation value $\mathbf{C} = \langle \psi_{candidate} | \boldsymbol{H} | \psi_{candidate} \rangle$

- Example: A simple Hamiltonian matrix representing a 1D atomic chain
  - → Each atom is represented with a single basis
  - → On-site energy & coupling energy between nearest atoms
  - → The system has closed boundaries

$$H = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

Onsite energy of each atom = 2
Coupling strength between nearest atoms = 1



[1] $\{\hat{\sigma}\}_n = \{I, X, Y, Z\}^{\otimes n}$

[2] $\hat{H}_k = \sum_{i=1}^{4^n} c_{ik} \hat{\sigma}_i$

[3] $\mathrm{Tr}\left(\hat{\sigma}_i^{\dagger} \hat{\sigma}_j\right) = 2^n \delta_{ij}$

[4] $c_i = \dfrac{\mathrm{Tr}\left(\hat{H}_k^{\dagger} \hat{\sigma}_i\right)}{2^n}$

F. Carasoli *et al.*, *Physical Chemistry Chemical Physics* **22**, 21816

- How to compute the expectation value of a state $|\psi\rangle$ given a matrix H ($\langle \psi | H | \psi \rangle$)?
  - → The first step should be decomposition of H into a set of elementary gates
  - → The most primitive way of circuit decomposition is based on {I, X, Y, Z}

Variational Quantum Eigensolver (VQE)

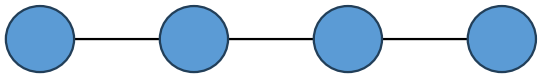
Find the lowest energy system in the quantum system

STEP 2): Compute the expectation value $\mathbf{C} = \langle \psi_{candidate} | \boldsymbol{H} | \psi_{candidate} \rangle$

- Example: A simple Hamiltonian matrix representing a 1D atomic chain

$$H = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

Onsite energy of each atom = 2
Coupling strength between nearest atoms = 1



$$H = 2I \otimes I - I \otimes X - 0.5X \otimes X - 0.5Y \otimes Y$$

- Strictly speaking, we need four 2-qubit circuits to evaluate $(\langle \psi | H | \psi \rangle)$

  → Effectively three since the first term does not have to necessarily computed

$$\langle \psi | H | \psi \rangle = 2\langle \psi | I \otimes I | \psi \rangle - \langle \psi | I \otimes X | \psi \rangle - 0.5\langle \psi | X \otimes X | \psi \rangle - 0.5\langle \psi | Y \otimes Y | \psi \rangle$$

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

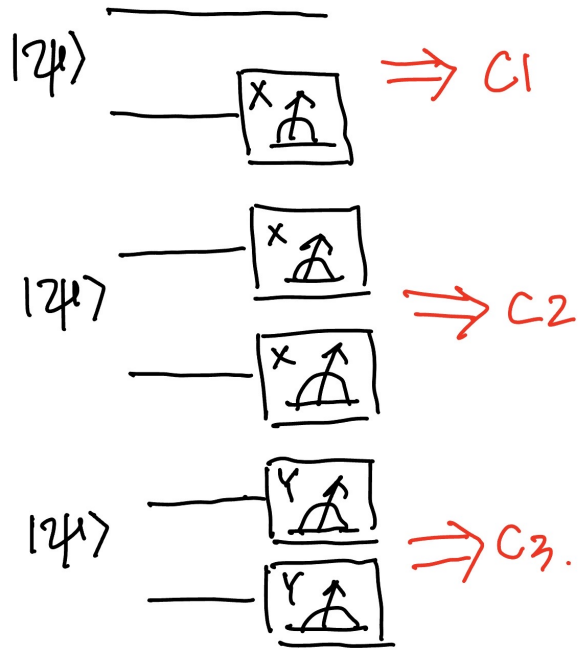
STEP 2): Compute the expectation value $\mathbf{C} = \langle \psi_{candidate} | H | \psi_{candidate} \rangle$

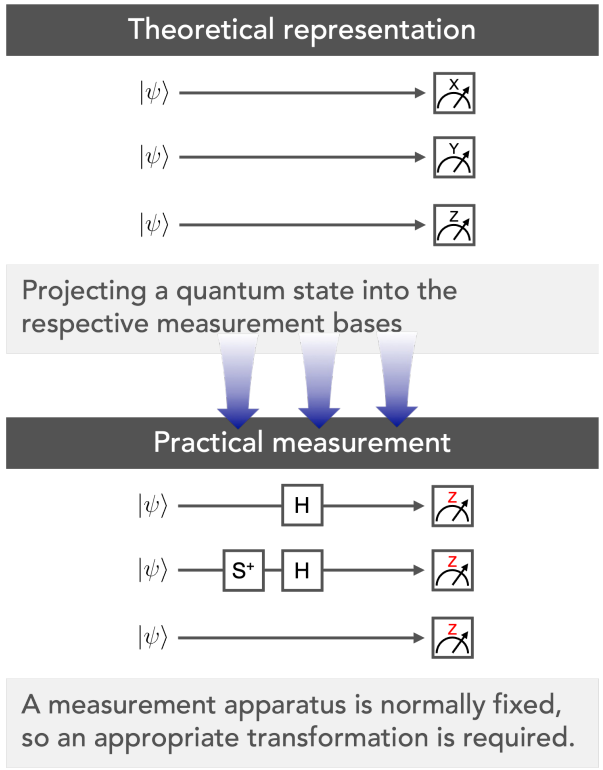
- Example: A simple Hamiltonian matrix representing a 1D atomic chain

$$C1 = \langle \psi | I \otimes X | \psi \rangle$$

$$C2 = \langle \psi | X \otimes X | \psi \rangle$$

$$C3 = \langle \psi | Y \otimes Y | \psi \rangle$$





Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

Flow of Programming (in case we need 2 circuits to evaluate the cost function)

```
def ansatz_circuit(param)
    ... circuit ...
    return qml.state()

def cost1(param)
    ansatz_circuit(param)
    return qml.expval(measurement operators 1)

def cost2(param)
    ansatz_circuit(param)
    return qml.expval(measurement operators 2)
```

```
conv = 1e-4 # for example
p = [initial values ...]
mcost = cost1(p) + cost2(p)

iterate:
    pcost = mcost
    p = optimize_and_update(prevcost)
    mcost = cost1(p) + cost2(p)
    if(abs(mcost – pcost) < conv)
        break;
```

Do you recognze that the working flow include STEP 1-4 appropriately?

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

Sample Code: A simple Hamiltonian representing a 1D atomic chain (4-by-4, 2 qubits)

```
import pennylane as qml
from pennylane import numpy as np
import matplotlib.pyplot as plt

dev = qml.device("default.qubit", wires=2)
@qml.qnode(dev, interface="autograd")
def C1(param):
    qml.RY(param[0], wires=0)
    qml.RY(param[1], wires=1)
    qml.CNOT(wires=[0,1])
    qml.RY(param[2], wires=0)
    qml.RY(param[3], wires=1)
    return qml.expval(qml.Identity(0)@qml.X(1))
```

```
@qml.qnode(dev, interface="autograd")
def C2(param):
    qml.RY(param[0], wires=0)
    qml.RY(param[1], wires=1)
    qml.CNOT(wires=[0,1])
    qml.RY(param[2], wires=0)
    qml.RY(param[3], wires=1)
    return qml.expval(qml.X(0)@qml.X(1))

# continue to the next slide
```

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

Sample Code: A simple Hamiltonian representing a 1D atomic chain (4-by-4, 2 qubits)

```
# continued from the previous slide

@qml.qnode(dev, interface="autograd")
def C3(param):
    qml.RY(param[0], wires=0)
    qml.RY(param[1], wires=1)
    qml.CNOT(wires=[0,1])
    qml.RY(param[2], wires=0)
    qml.RY(param[3], wires=1)
    return qml.expval(qml.Y(0)@qml.Y(1))

def cost_fn(param):
    cost = 2-C1(param)-C2(param)/2-C3(param)/2
    return cost
```

```
opt = qml.GradientDescentOptimizer \
      (stepsize =0.4)

theta = np.array([np.pi/2,np.pi/2,np.pi/2, \
        np.pi/2], requires_grad=True)

energy = [cost_fn(theta)]
angle = [theta]

max_iterations = 100
conv_tol = 1e-04

# continue to the next slide
```

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

Sample Code: A simple Hamiltonian representing a 1D atomic chain (4-by-4, 2 qubits)

```
# continued from the previous slide
for n in range(max_iterations):
theta, prev_energy = opt.step_and_cost(cost_fn, theta)

energy.append(cost_fn(theta))
angle.append(theta)

conv = np.abs(energy[-1] - prev_energy)

if n % 2 == 0:
    print(f"Step = {n}, Energy = {energy[-1]:.8f}")
if conv <= conv_tol:
    print(f"Converged at Step = {n}, Energy = {energy[-1]:.8f}")
    break
```

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

Sample Code: A simple Hamiltonian representing a 1D atomic chain (4-by-4, 2 qubits)

```
# continued from the previous slide
for n in range(max_iterations):
theta, prev_energy = opt.step_and_cost(cost_fn,


energy.append(cost_fn(theta))
angle.append(theta)

conv = np.abs(energy[-1] - prev_energy)

if n % 2 == 0:
    print(f"Step = {n}, Energy = {energy[-1]:.8
if conv <= conv_tol:
    print(f"Converged at Step = {n}, Energy = {energy[-1]:.8f}")
    break
```

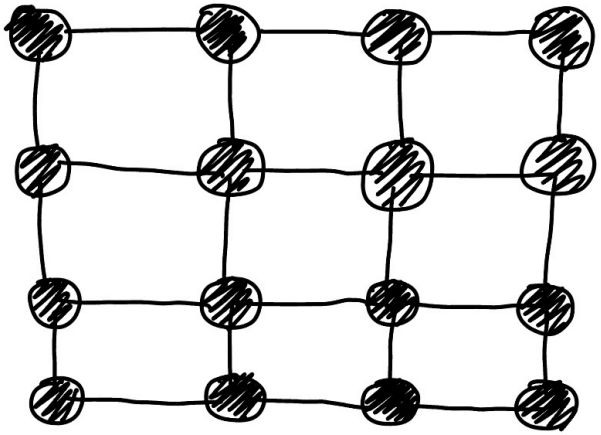
Ref. solution: 0.38196601

```
Step = 0, Energy = 1.61058166
Step = 2, Energy = 0.92919395
Step = 4, Energy = 0.54090589
Step = 6, Energy = 0.43152534
Step = 8, Energy = 0.40144038
Step = 10, Energy = 0.38991480
Step = 12, Energy = 0.38509493
Step = 14, Energy = 0.38315260
Step = 16, Energy = 0.38240480
Step = 18, Energy = 0.38212581
Converged at Step = 19, Energy = 0.38206210
```

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

❑ Mission: Finding the ground state energy of a 4x4-atom 2D chain



$$H_{2D} = I_4 \otimes H_{1D} + H_{1D} \otimes I_4$$

(I_4 = a 4x4 identity matrix)

$$H_{1D} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

- Using the code we have discussed so far, write a code that compute the lowest eigenstate of a 16-atom 2D chain.
- To find the expectation value of a circuit involving identity measurements, use the identity operator (`qml.Identity`) appropriately ($I_4 = I_2 \otimes I_2$),

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

❑ **Mission: Finding the ground state energy of a 4x4-atom 2D chain** (cont.)

$$H_{2D} = I_4 \otimes H_{1D} + H_{1D} \otimes I_4$$

$$= I_4 \otimes (2I_4 - I_2 \otimes X - 0.5X \otimes X - 0.5Y \otimes Y)$$

$$+ (2I_4 - I_2 \otimes X - 0.5X \otimes X - 0.5Y \otimes Y) \otimes I_4$$

$\otimes$ operation is not generally commutative.

$$I_4 = I_2 \otimes I_2$$

Variational Quantum Eigensolver (VQE)

Find the lowest energy system in the quantum system

❑ **Mission: Finding the ground state energy of a 4x4-atom 2D chain** (cont.)

- Consider the following Ansatz circuit
- Use the block surrounded with a dashed line once, twice and three times. What can you learn from the convergence patterns?