

# Applied QML

*Lecture 7: Variational Quantum Algorithms*

Christophe Pere

2024-02-22

# Table des matières

Rappel du cours Optimisation

QAOA

VQA

VQE

VQC

# Rappel cours d'optimisation

# Modèle d'Ising

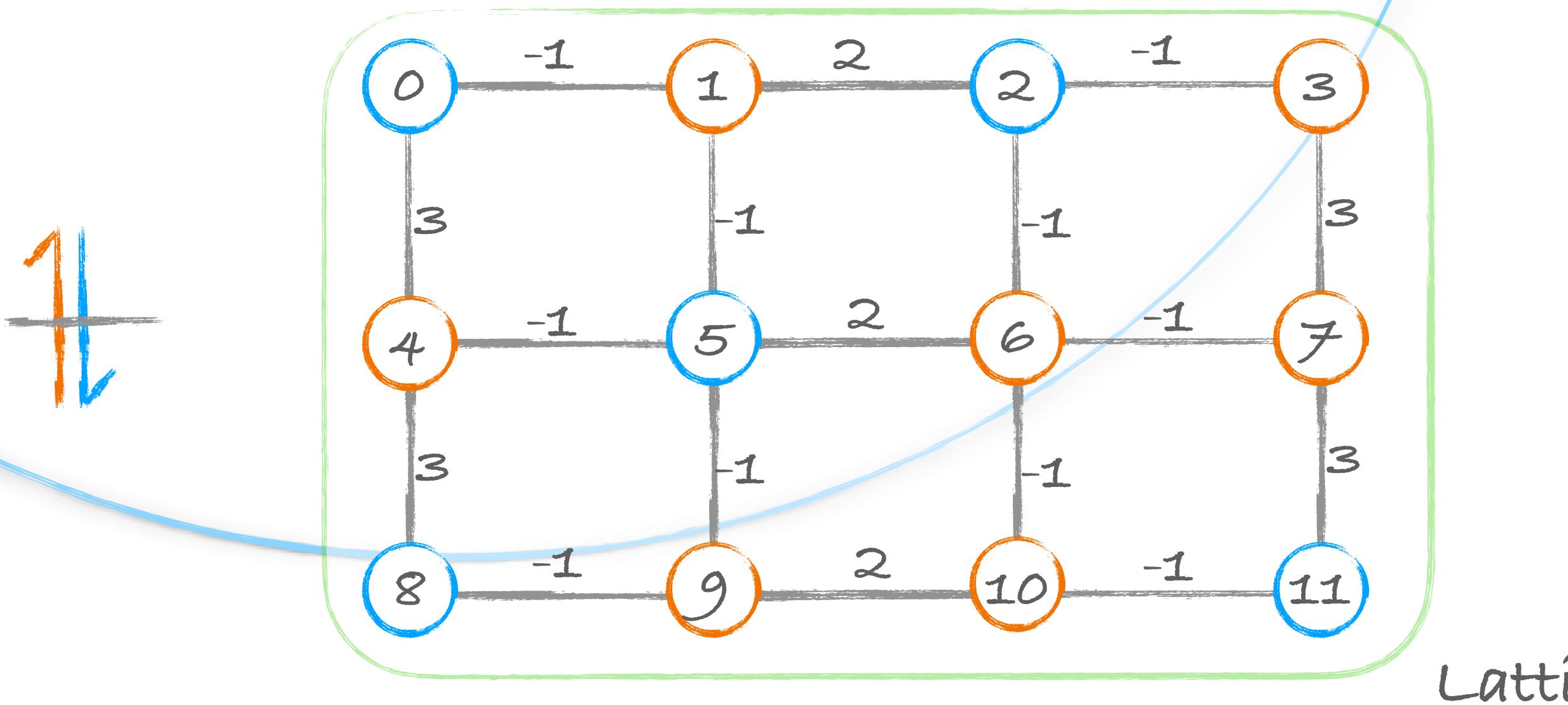
Ising

Physique

Modèle ferromagnétique qui représente les "spins" des atomes et les interactions avec les atomes voisins. La structure est appelée un lattice.

Informatique

Modèle de graphe dont les noeuds sont binaires (+1, -1).



Hamiltonien

$$H = - \sum_{j,k} J_{jk} z_j z_k - \sum_j h_j z_j$$

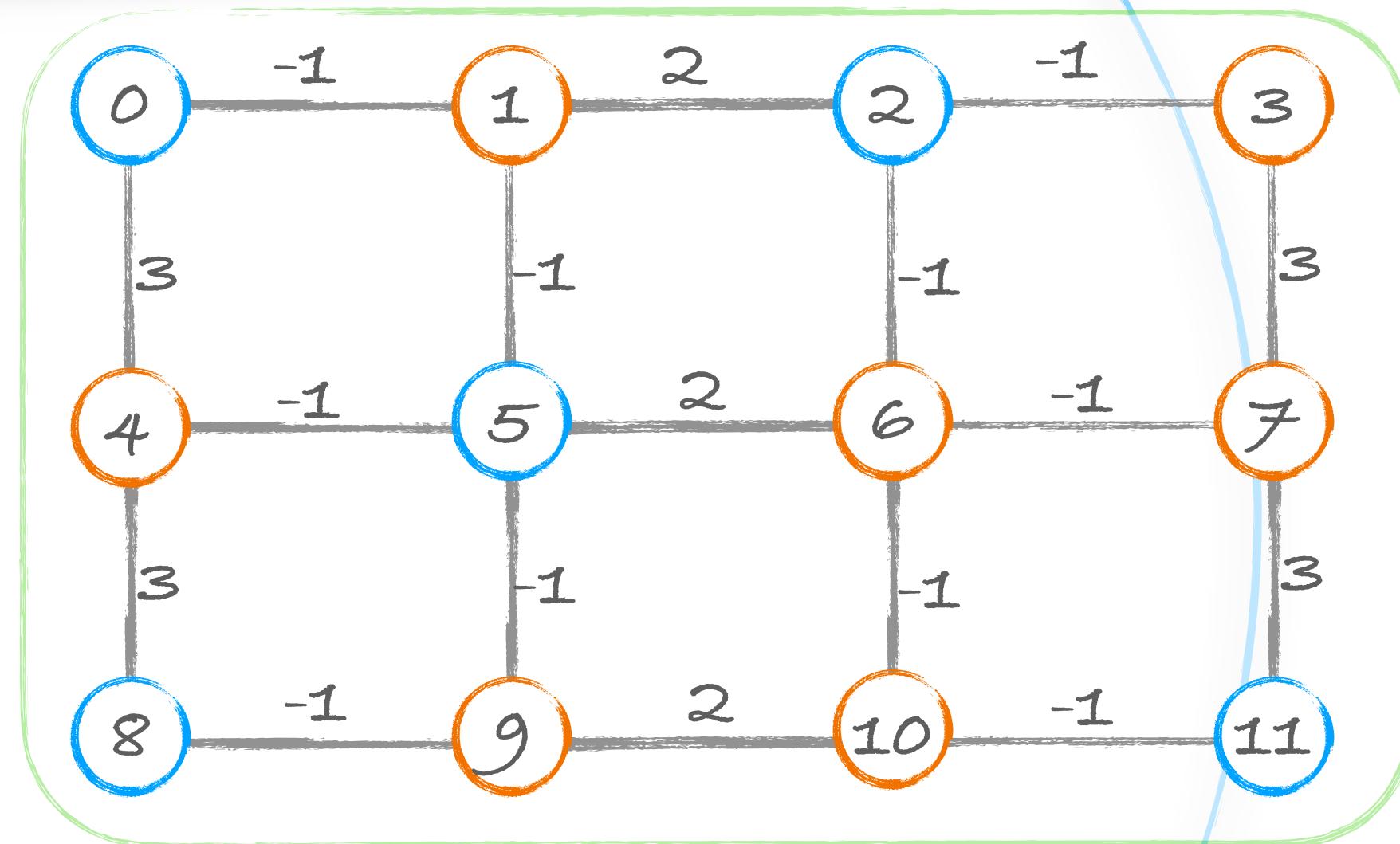
$J_{jk}$  interaction particule j et k

$h_j$  champ magnétique externe

$z$  particule

# Modèle d'Ising

Ising



Comment formuler l'hamiltonien avec:

$$J_{jk} = 1$$

$$h_j = 1$$

Hamiltonien

$$H = - \sum_{j,k} J_{jk} z_j z_k - \sum_j h_j z_j$$

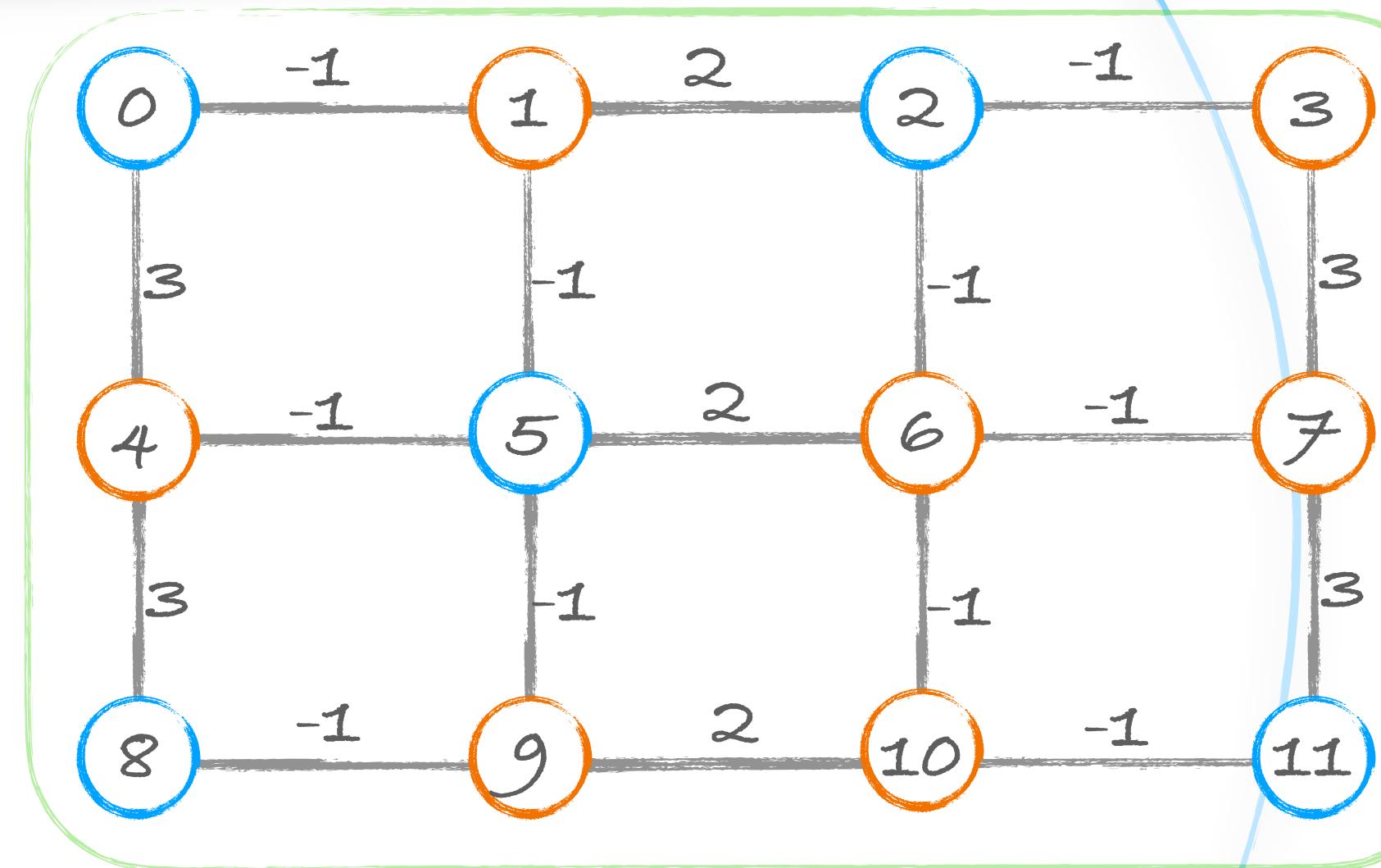
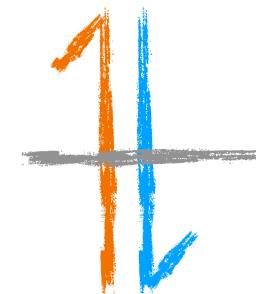
$J_{jk}$  interaction particule j et k

$h_j$  champ magnétique externe

$z$  particule

# Modèle d'Ising

Ising



Lattice

Comment formuler l'hamiltonien avec:

$$J_{jk} = 1$$

$$h_j = 1$$

$$-\sum_{j,k} J_{jk} z_j z_k$$

Hamiltonien

$$H = - \sum_{j,k} J_{jk} z_j z_k - \sum_j h_j z_j$$

$J_{jk}$  interaction particule j et k

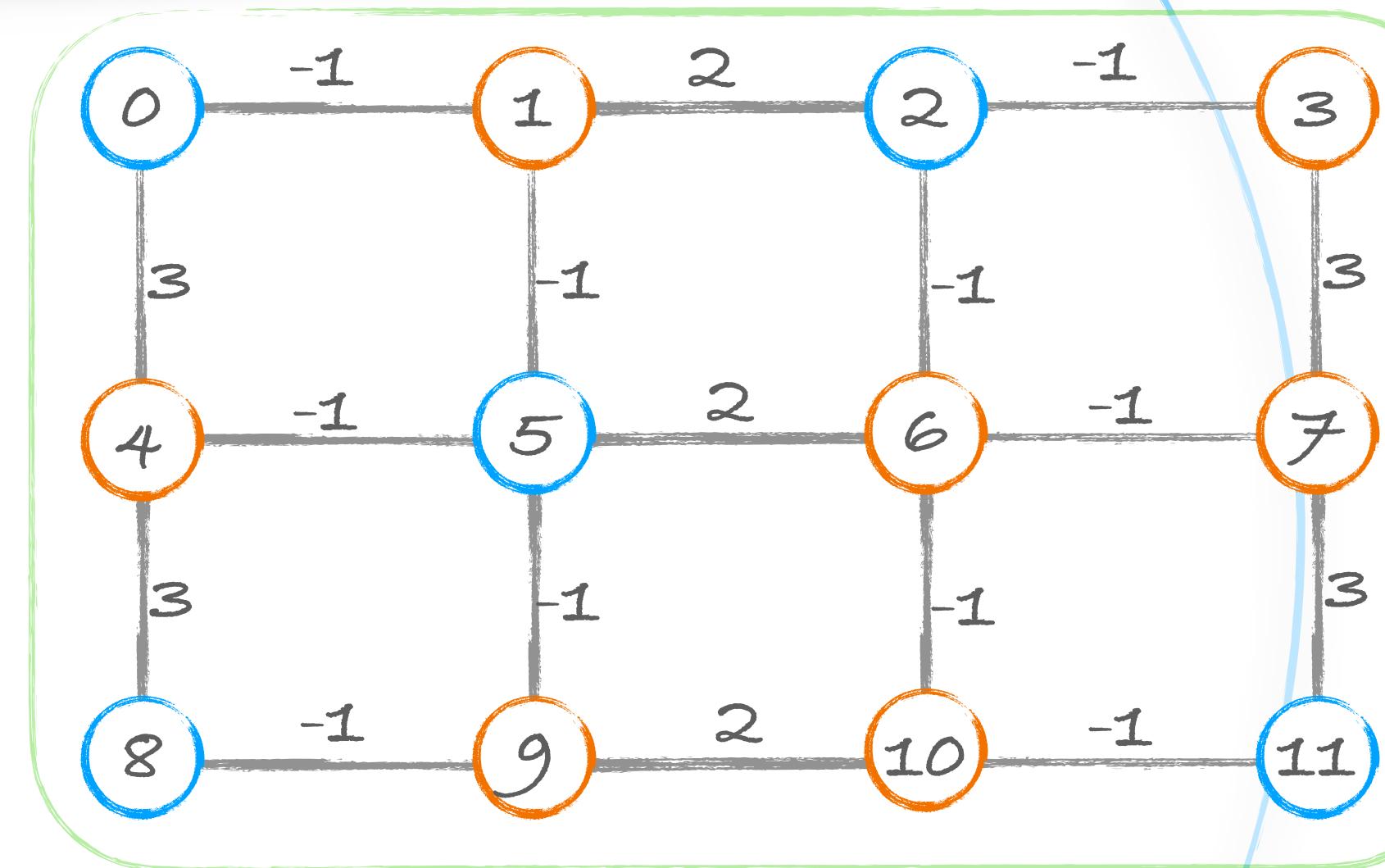
$h_j$  champ magnétique externe

$z$  particule

$$\begin{aligned} & z_0 z_1 - 2 z_1 z_2 + z_2 z_3 - 3 z_0 z_4 + z_1 z_5 + z_2 z_6 - 3 z_3 z_7 \\ & + z_4 z_5 - 2 z_5 z_6 + z_6 z_7 - 3 z_4 z_8 + z_5 z_9 + z_6 z_1 0 - 3 z_7 z_{11} \\ & + z_8 z_9 - 2 z_9 z_{10} + z_{10} z_{11} \end{aligned}$$

# Modèle d'Ising

Ising



Lattice

Comment formuler l'hamiltonien avec:

$$J_{jk} = 1$$

$$h_j = 1$$

$$-\sum_{j,k} J_{jk} z_j z_k$$

$$-\sum_j h_j z_j$$

Hamiltonien

$$H = - \sum_{j,k} J_{jk} z_j z_k - \sum_j h_j z_j$$

$J_{jk}$  interaction particule j et k

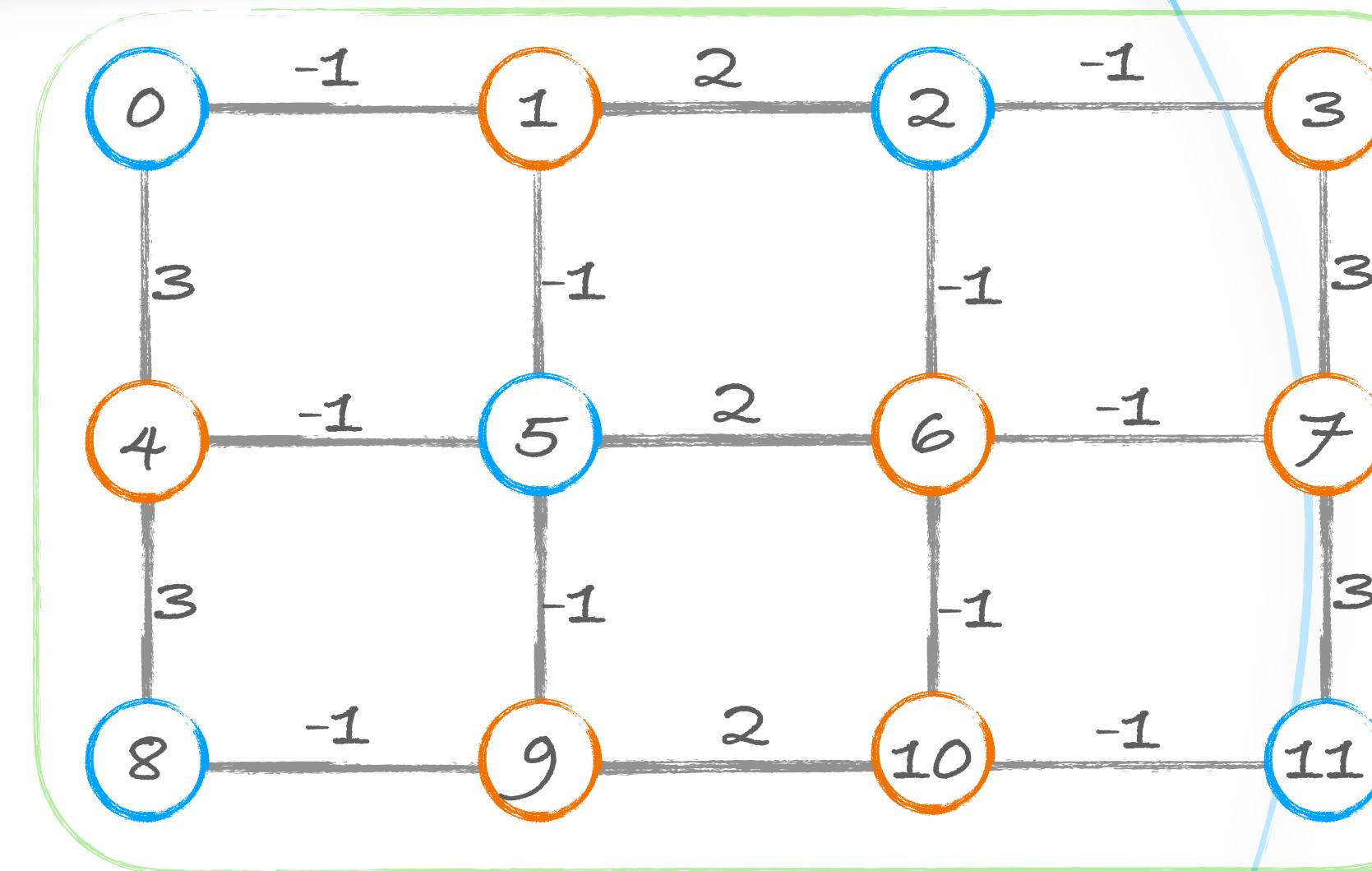
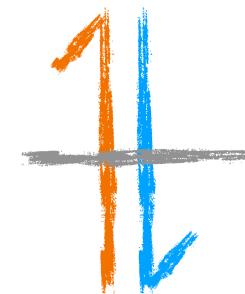
$h_j$  champ magnétique externe

$z$  particule

$$\begin{aligned} & z_0 z_1 - 2 z_1 z_2 + z_2 z_3 - 3 z_0 z_4 + z_1 z_5 + z_2 z_6 - 3 z_3 z_7 \\ & + z_4 z_5 - 2 z_5 z_6 + z_6 z_7 - 3 z_4 z_8 + z_5 z_9 + z_6 z_{10} - 3 z_7 z_{11} \\ & + z_8 z_9 - 2 z_9 z_{10} + z_{10} z_{11} \\ & - z_0 - z_1 - z_2 - z_3 - z_4 - z_5 - z_6 - z_7 - z_8 - z_9 \\ & - z_{10} - z_{11} \end{aligned}$$

# Modèle d'Ising

Ising



Lattice

Comment formuler l'hamiltonien avec:

$$J_{jk} = 1$$

$$h_j = 1$$

Minimiser

$$-\sum_{j,k} J_{jk} z_j z_k - \sum_j h_j z_j$$

$$z_j = \{1, -1\} \quad j = 0, \dots, 11$$

Hamiltonien

$$H = - \sum_{j,k} J_{jk} z_j z_k - \sum_j h_j z_j$$

$J_{jk}$  interaction particule  $j$  et  $k$

$h_j$  champ magnétique externe

$z$  particule

$$\begin{aligned} & z_0 z_1 - 2 z_1 z_2 + z_2 z_3 - 3 z_0 z_4 + z_1 z_5 + z_2 z_6 - 3 z_3 z_7 \\ & + z_4 z_5 - 2 z_5 z_6 + z_6 z_7 - 3 z_4 z_8 + z_5 z_9 + z_6 z_1 - 3 z_7 z_{11} \\ & + z_8 z_9 - 2 z_9 z_{10} + z_{10} z_{11} \end{aligned}$$

$$\begin{aligned} & -z_0 - z_1 - z_2 - z_3 - z_4 - z_5 - z_6 - z_7 - z_8 - z_9 \\ & - z_{10} - z_{11} \end{aligned}$$

# QUBO

subset sum pb

Réécriture selon un Ising ground state

1 set de données  
1 cible (target)

Est-ce que la somme de plusieurs éléments peut être égale à la cible?

Problème de minimisation avec des variables binaires

$$S = \{a_0, \dots, a_m\}$$

variables binaires  $x_j \quad j = 0, \dots, m$

$$c(x_0, x_1, \dots, x_m) = (a_0x_0 + a_1x_1 + \dots + a_mx_m - T)^2$$

Équation quadratique

Le problème revient à minimiser  $c(x_0, x_1, \dots, x_m)$

S'il la minimisation est égale à 0 il y une solution sinon il n'y a pas de solution

optimisation combinatoire

# QUBO

$$c(x_0, x_1, \dots, x_m) = (a_0x_0 + a_1x_1 + \dots + a_mx_m - T)^2$$

Toujours de degré 2  Quadratic Unconstrained Binary Optimization (QUBO) problem

Minimiser  $q(x_0, \dots, x_m)$

Avec  $x_j \in \{0,1\}, j = 0, \dots, m$

QUBO car on cherche à minimiser des expressions quadratiques avec des variables binaires sans restrictions car toutes les combinaisons de 0 et 1 sont permises.

# QUBO

$$c(x_0, x_1, \dots, x_m) = (a_0x_0 + a_1x_1 + \dots + a_mx_m - T)^2$$

Toujours de degré 2  Quadratic Unconstrained Binary Optimization (QUBO) problem

Minimiser  $q(x_0, \dots, x_m)$

Avec  $x_j \in \{0,1\}, j = 0, \dots, m$

Formulation pour un Ising?

$$H = - \sum_{j,k} J_{jk} z_j z_k - \sum_j h_j z_j$$

QUBO à Ising

$$x_j = \frac{(1 - z_j)}{2}$$

$$\begin{aligned} x_j = 0 &\rightarrow z_j = 1 \\ x_j = 1 &\rightarrow z_j = -1 \end{aligned}$$

Ising à QUBO

$$z_j = 1 - 2x_j$$

$$\begin{aligned} x_j = 0 &\rightarrow z_j = 1 \\ x_j = 1 &\rightarrow z_j = -1 \end{aligned}$$

Ising à QUBO

$$z_j = 2x_j - 1$$

Polynôme quadratique qui représente exactement la fonction d'énergie

QAOA

# QAOA

Permet de discréteriser une évolution continue (quantum annealing) en petites étapes de changements discrets.

Trotterization, comme vu avec l'encodage Hamiltonien, est le processus de transformation d'une évolution continue à une évolution discrète.

QAOA permet d'approximer une solution optimale à des problèmes combinatoires.

# QAOA

Permet de discréteriser une évolution continue (quantum annealing) en petites étapes de changements discrets.

Trotterization, comme vu avec l'encodage Hamiltonien, est le processus de transformation d'une évolution continue à une évolution discrète.

QAOA permet d'approximer une solution optimale à des problèmes combinatoires.

Quantum  
Approximate  
Optimization  
Algorithm

# QAOA

Permet de discréteriser une évolution continue (quantum annealing) en petites étapes de changements discrets.

Trotterization, comme vu avec l'encodage Hamiltonien, est le processus de transformation d'une évolution continue à une évolution discrète.

QAOA permet d'approximer une solution optimale à des problèmes combinatoires.

Quantum  
Approximate  
Optimization  
Algorithm

Hamiltonien adiabatique  
 $H(t) = A(t)H_0 + B(t)H_1$

Avec  $H_0$  et  $H_1$  deux Hamiltoniens et  
 $A(t)$  et  $B(t)$  des fonctions qui puissent satisfaire  
 $A(0)=B(T)=1$  et  $A(T)=B(0)=0$  avec  
T le temps total du processus.

# QAOA

Permet de discréteriser une évolution continue (quantum annealing) en petites étapes de changements discrets.

Trotterization, comme vu avec l'encodage Hamiltonien, est le processus de transformation d'une évolution continue à une évolution discrète.

QAOA permet d'approximer une solution optimale à des problèmes combinatoires.

Quantum  
Approximate  
Optimization  
Algorithm

Hamiltonien adiabatique  
 $H(t) = A(t)H_0 + B(t)H_1$

Avec  $H_0$  et  $H_1$  deux Hamiltoniens et  $A(t)$  et  $B(t)$  des fonctions qui puissent satisfaire  $A(0)=B(T)=1$  et  $A(T)=B(0)=0$  avec  $T$  le temps total du processus.

L'évolution d'un système quantique est gouvernée par l'équation de Schrödinger dépendante du temps. La résoudre permet d'obtenir l'état du système à tout moments  $t$  entre 0 et  $T$ .

$$i\hbar \frac{d}{dt} |\Psi(t)\rangle = \hat{H} |\Psi(t)\rangle$$

# QAOA

Hamiltonien adiabatique

$$H(t) = A(t)H_0 + B(t)H_1$$

$$H(t) = A(t)H_0 + B(t)H_1$$

Discretisation

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)}$$

$t_c$  est un temps fixe entre  $[0, T]$

# QAOA

Hamiltonien adiabatique

$$H(t) = A(t)H_0 + B(t)H_1$$

$$H(t) = A(t)H_0 + B(t)H_1$$

Discretisation

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)}$$

$t_c$  est un temps fixe entre  $[0, T]$

État quantique final:

$$\left( \prod_{m=0}^p e^{i\Delta t(A(t_m)H_0+B(t_m)H_1)} \right) |\psi_0\rangle$$

Avec

$$t_m = m \frac{\Delta t}{T}$$

: le temps de chaque pas  $m$

$$p = \frac{T}{\Delta t}$$

: le nombre maximal de pas  $m$

# QAOA

Hamiltonien adiabatique

$$H(t) = A(t)H_0 + B(t)H_1$$

$$H(t) = A(t)H_0 + B(t)H_1$$

Discrétisation

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)}$$

$t_c$  est un temps fixe entre  $[0, T]$

État quantique final:

$$\left( \prod_{m=0}^p e^{i\Delta t(A(t_m)H_0+B(t_m)H_1)} \right) |\psi_0\rangle$$

Avec

$m$ : le nombre de pas d'évolution

$$t_m = m \frac{\Delta t}{T}$$

: le temps de chaque pas  $m$

$$p = \frac{T}{\Delta t}$$

: le nombre maximal de pas  $m$

Application de portes  
de rotations unitaires

Circuit quantique répété

# QAOA

Hamiltonien adiabatique

$$H(t) = A(t)H_0 + B(t)H_1$$

$$H(t) = A(t)H_0 + B(t)H_1$$

Discretisation

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)}$$

$t_c$  est un temps fixe entre  $[0, T]$

État quantique final:

$$\left( \prod_{m=0}^p e^{i\Delta t(A(t_m)H_0+B(t_m)H_1)} \right) |\psi_0\rangle$$

Avec

$m$ : le nombre de pas d'évolution

$$t_m = m \frac{\Delta t}{T}$$

: le temps de chaque pas  $m$

$$p = \frac{T}{\Delta t}$$

: le nombre maximal de pas  $m$

# QAOA

Hamiltonien adiabatique

$$H(t) = A(t)H_0 + B(t)H_1$$

$$H(t) = A(t)H_0 + B(t)H_1$$

Discretisation

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)}$$

$t_c$  est un temps fixe entre  $[0, T]$

État quantique final:

$$\left( \prod_{m=0}^p e^{i\Delta t(A(t_m)H_0+B(t_m)H_1)} \right) |\psi_0\rangle$$

Nouvelle approximation

$\approx$

Avec

$m$ : le nombre de pas d'évolution

$$t_m = m \frac{\Delta t}{T}$$

: le temps de chaque pas  $m$

$$p = \frac{T}{\Delta t}$$

: le nombre maximal de pas  $m$

# QAOA

Hamiltonien adiabatique

$$H(t) = A(t)H_0 + B(t)H_1$$

$$H(t) = A(t)H_0 + B(t)H_1$$

Discrétisation

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)}$$

$t_c$  est un temps fixe entre  $[0, T]$

État quantique final:

$$\left( \prod_{m=0}^p e^{i\Delta t(A(t_m)H_0+B(t_m)H_1)} \right) |\psi_0\rangle$$

Nouvelle approximation

$\approx$

$$\left( \prod_{m=0}^p e^{i\Delta tA(t_m)H_0} e^{i\Delta tB(t_m)H_1} \right) |\psi_0\rangle$$

Avec

$m$ : le nombre de pas d'évolution

$$t_m = m \frac{\Delta t}{T}$$

: le temps de chaque pas  $m$

$$p = \frac{T}{\Delta t}$$

: le nombre maximal de pas  $m$

Connue sous le nom de formule de Lie-Trotter

# QAOA

Hamiltonien adiabatique

$$H(t) = A(t)H_0 + B(t)H_1$$

$$H(t) = A(t)H_0 + B(t)H_1$$

Discrétisation

$$e^{i\Delta t(A(t_c)H_0+B(t_c)H_1)}$$

$t_c$  est un temps fixe entre  $[0, T]$

État quantique final:

$$\left( \prod_{m=0}^p e^{i\Delta t(A(t_m)H_0+B(t_m)H_1)} \right) |\psi_0\rangle$$

Nouvelle approximation

$\approx$

$$\left( \prod_{m=0}^p e^{i\Delta tA(t_m)H_0} e^{i\Delta tB(t_m)H_1} \right) |\psi_0\rangle$$

Avec

$m$ : le nombre de pas d'évolution

$$t_m = m \frac{\Delta t}{T} : \text{le temps de chaque pas } m$$

$$p = \frac{T}{\Delta t} : \text{le nombre maximal de pas } m$$

Deux formes Hamiltoniennes d'évolution

Forme de modèle d'Ising pour encoder le problème

L'objectif est de trouver le "ground state energy" à partir d'un circuit quantique

# QAOA

Formulation digitale

$$\left( \prod_{m=0}^p e^{i\Delta t A(t_m) H_0} e^{i\Delta t B(t_m) H_1} \right) |\psi_0\rangle$$



Posons

$$e^{i\Delta t A(t_m) H_0} = e^{i\gamma H_0}$$

$$e^{i\Delta t B(t_m) H_1} = e^{i\beta H_1}$$

# QAOA

## Formulation digitale

$$\left( \prod_{m=0}^p e^{i\Delta t A(t_m) H_0} e^{i\Delta t B(t_m) H_1} \right) |\psi_0\rangle$$

Posons

$$e^{i\Delta t A(t_m) H_0} = e^{i\gamma H_0}$$

$$e^{i\Delta t B(t_m) H_1} = e^{i\beta H_1}$$

On va venir appliquer  $p$  fois  
l'alternance de ces deux termes  
pour  $\gamma$  et  $\beta$  donnés

$$e^{i\gamma_p H_0} e^{i\beta_p H_1} \dots e^{i\gamma_2 H_0} e^{i\beta_2 H_1} e^{i\gamma_1 H_0} e^{i\beta_1 H_1} |\psi_0\rangle$$

# QAOA

## Formulation digitale

$$\left( \prod_{m=0}^p e^{i\Delta t A(t_m) H_0} e^{i\Delta t B(t_m) H_1} \right) |\psi_0\rangle$$

Peuvent être implanté avec des portes à 1 et 2 qubits

Nombres réels

Posons

$$e^{i\Delta t A(t_m) H_0} = e^{i\gamma H_0}$$

$$e^{i\Delta t B(t_m) H_1} = e^{i\beta H_1}$$

On va venir appliquer  $p$  fois l'alternance de ces deux termes pour  $\gamma$  et  $\beta$  donnés

$$e^{i\gamma_p H_0} e^{i\beta_p H_1} \dots e^{i\gamma_2 H_0} e^{i\beta_2 H_1} e^{i\gamma_1 H_0} e^{i\beta_1 H_1} |\psi_0\rangle$$

$$\begin{aligned}\gamma &= (\gamma_1, \dots, \gamma_p) \\ \beta &= (\beta_1, \dots, \beta_p)\end{aligned}$$

$$|\beta, \gamma\rangle$$

# QAOA

## Formulation digitale

$$\left( \prod_{m=0}^p e^{i\Delta t A(t_m) H_0} e^{i\Delta t B(t_m) H_1} \right) |\psi_0\rangle$$



$$e^{i\gamma_p H_0} e^{i\beta_p H_1} \dots e^{i\gamma_2 H_0} e^{i\beta_2 H_1} e^{i\gamma_1 H_0} e^{i\beta_1 H_1} |\psi_0\rangle$$

$$\gamma = (\gamma_1, \dots, \gamma_p)$$

$$\beta = (\beta_1, \dots, \beta_p)$$

$|\beta, \gamma\rangle$

On va chercher à trouver la valeur de plus faible énergie pour  $H_1$  en faisant:

$$\langle \beta, \gamma | H_1 | \beta, \gamma \rangle$$

On va donc passer d'un problème d'optimisation à un problème combinatoire  
Pour déterminer les meilleures valeurs de  $\beta$  et  $\gamma$

Simple estimation de valeurs réelles pour un output réel.

Le problème est que  $|\beta, \gamma\rangle$  grossit exponentiellement et ne peut pas être simulé classiquement

# QAOA

## Formulation digitale

$$\left( \prod_{m=0}^p e^{i\Delta t A(t_m) H_0} e^{i\Delta t B(t_m) H_1} \right) |\psi_0\rangle$$

$$e^{i\gamma_p H_0} e^{i\beta_p H_1} \dots e^{i\gamma_2 H_0} e^{i\beta_2 H_1} \dots e^{i\gamma_1 H_0} e^{i\beta_1 H_1} |\psi_0\rangle$$

$\gamma = (\gamma_1, \dots)$   
 $\beta = (\beta_1, \dots)$

On va chercher à trouver la valeur de plus faible en  $\langle \beta, \gamma | H_1 | \beta, \gamma \rangle$

On va donc passer d'un problème d'optimisation pour déterminer les nvaleurs de  $\beta$  et  $\gamma$

Simple estimation

Le problème est que  $|\beta, \gamma\rangle$  grossit exponentiellement et ne peut pas être simulé classiquement

Efficace avec un ordinateur quantique si + a un nombre de terme qui grossit polynomiallement

réelles pour un output réel.

# QAOA

## Formulation digitale

$$\left( \prod_{m=0}^p e^{i\Delta t A(t_m) H_0} e^{i\Delta t B(t_m) H_1} \right) |\psi_0\rangle$$

$$e^{i\gamma_p H_0} e^{i\beta_p H_1} \dots e^{i\gamma_2 H_0} e^{i\beta_2 H_1} e^{i\gamma_1 H_0} e^{i\beta_1 H_1} |\psi_0\rangle$$

Algorithme hybride qui a besoin de l'info classique pour calculer la fonction de coût et les nouveaux paramètres  $\beta$  et  $\gamma$  par descente de gradient

L'ordinateur quantique sera ensuite utilisé pour créer l'état quantique en utilisant les valeurs  $\beta^*$  et  $\gamma^*$  qui sont les valeurs approximées pour le niveau d'énergie le plus bas. Cet état a le maximum d'overlap avec  $H_1$  et donc le bitstring final est la valeur des coefficients du problème encodé dans  $H_1$

0100110010011

# QAOA

## Algorithme

classique	Choix de la valeur de $p$	
classique	Choix de valeurs d'initialisation de $\beta = (\beta_1, \dots, \beta_p)$ et $\gamma = (\gamma_1, \dots, \gamma_p)$	
classique	<b>while</b> critère d'arrêt non atteint:	
Quantique	Prépare l'état $ \beta, \gamma\rangle$	<i>Cette étape se fait sur un QC</i>
classique	Mesure de $ \beta, \gamma\rangle$ et estimation de $E(\beta, \gamma)$	
classique	Mise à niveau des paramètres $ \beta, \gamma\rangle$ grâce à un algorithme de minimisation	
classique	Optention des valeurs optimales de $\beta^*$ et $\gamma^*$	
Quantique	Prépare l'état $ \beta^*, \gamma^*\rangle$	<i>Cette étape se fait sur un QC</i>
classique	Mesure de $ \beta^*, \gamma^*\rangle$ et obtention de $E(\beta^*, \gamma^*)$	

# QAOA

Algorithme

En pratique

$$|\beta, \gamma\rangle = e^{i\gamma_p H_0} e^{i\beta_p H_1} \dots e^{i\gamma_2 H_0} e^{i\beta_2 H_1} e^{i\gamma_1 H_0} e^{i\beta_1 H_1} |\psi_0\rangle$$

État d'énergie fondamentale de  $H_0$

Préparé dans l'état  $|+\rangle$  avec des portes  
de Hadamard sur tous les qubits

# QAOA

Algorithme

En pratique

$$|\beta, \gamma\rangle = e^{i\gamma_p H_0} e^{i\beta_p H_1} \dots e^{i\gamma_2 H_0} e^{i\beta_2 H_1} e^{i\gamma_1 H_0} e^{i\beta_1 H_1} |\psi_0\rangle$$

Pauli X

$$H_0 = - \sum_{j=0}^{n-1} X_j$$

État d'énergie fondamentale de  $H_0$

Préparé dans l'état  $|+\rangle$  avec des portes de Hadamard sur tous les qubits

Application de la porte sur tous les qubits

$$R_X(2\beta)$$

# QAOA

Algorithme

En pratique

$$|\beta, \gamma\rangle = e^{i\gamma_p H_0} e^{i\beta_p H_1} \dots e^{i\gamma_2 H_0} e^{i\beta_2 H_1} e^{i\gamma_1 H_0} e^{i\beta_1 H_1} |\psi_0\rangle$$

Pauli X  
"Mixer"

$$H_0 = - \sum_{j=0}^{n-1} X_j \rightarrow R_X(2\beta)$$

Ising  
"Cost"

$$H_1 = - \sum_{j,k} J_{jk} Z_j Z_k - \sum_j h_j Z_j \rightarrow R_Z(2\gamma)$$

$J_{jk}, h_j$  sont des nombres réels

État d'énergie fondamentale de  $H_0$   
Préparé dans l'état  $|+\rangle$  avec des portes  
de Hadamard sur tous les qubits

Application de la porte sur  
tous les qubits

Et CNOTS entre les qubits  $j, k$

Role du "Mixer": Cet Hamiltonien est là pour s'assurer que si on tombe sur un "eigenstate" le mixer s'assure que l'algorithme ne reste pas bloqué sur cet état

Permet d'explorer dans des nouvelles directions

Le mixer anti-commute avec le cost

# QAOA

Algorithme

En pratique

$$|\beta, \gamma\rangle = e^{i\gamma_p H_0} e^{i\beta_p H_1} \dots e^{i\gamma_2 H_0} e^{i\beta_2 H_1} e^{i\gamma_1 H_0} e^{i\beta_1 H_1} |\psi_0\rangle$$

État d'énergie fondamentale de  $H_0$

Préparé dans l'état  $|+\rangle$  avec des portes de Hadamard sur tous les qubits

Pauli X

$$H_0 = - \sum_{j=0}^{n-1} X_j \rightarrow R_X(2\beta)$$

Application de la porte sur tous les qubits

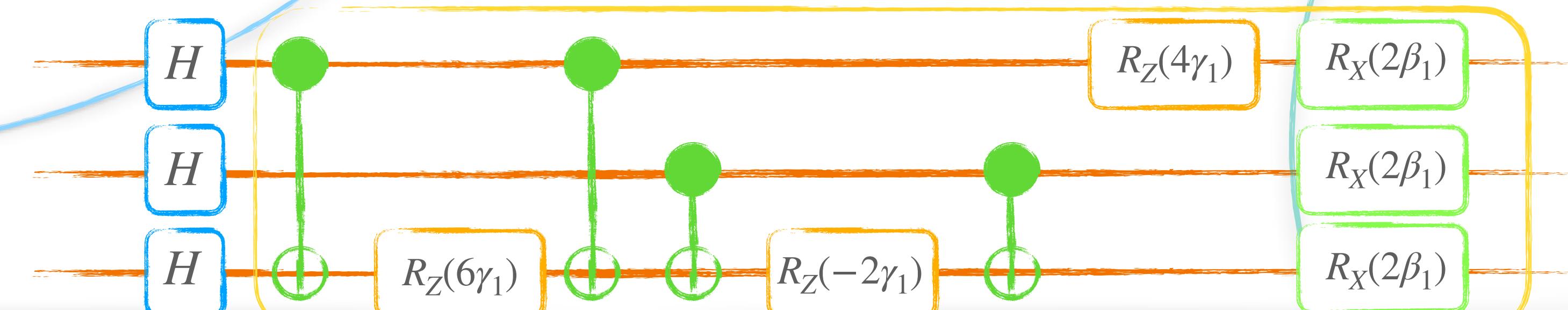
Ising

$$H_1 = - \sum_{j,k} J_{jk} Z_j Z_k - \sum_j h_j Z_j \rightarrow R_Z(2\gamma)$$

Et CNOTS entre les qubits j,k

$J_{jk}, h_j$  sont des nombres réels

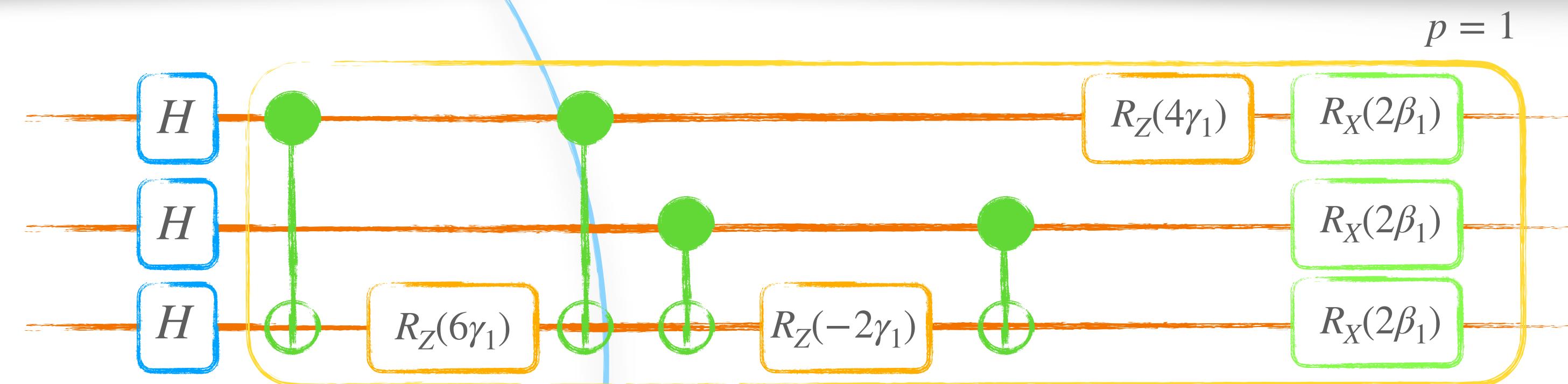
Pour 3 qubits, avec un Ising de la forme  $3Z_0Z_1 - Z_1Z_2 + 2Z_0$



# QAOA

Algorithme

valeur estimées



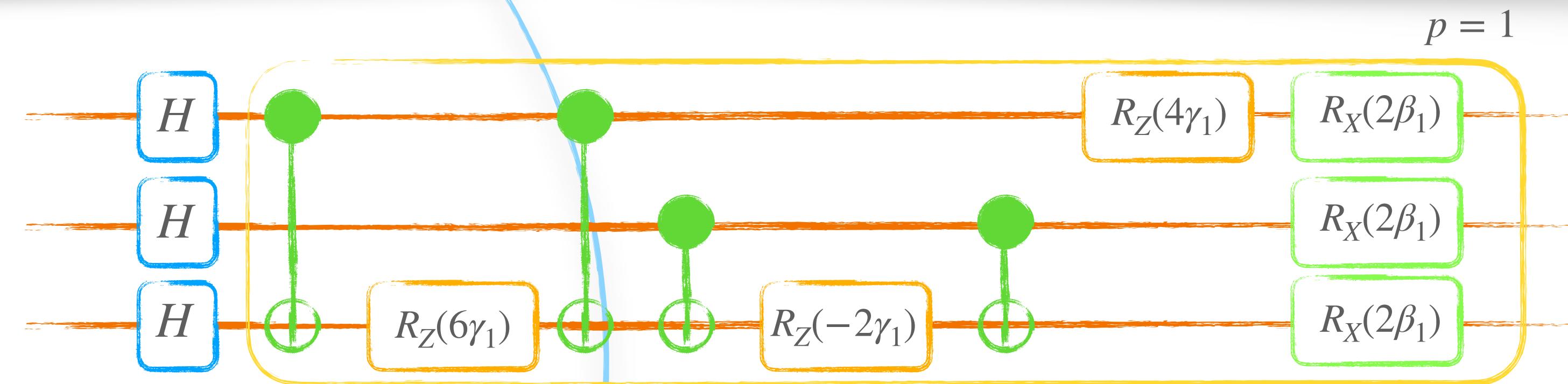
Pour 3 qubits, avec un Ising de la forme  $3Z_0Z_1 - Z_1Z_2 + 2Z_0$

On veut directement estimer  $\langle x | H_1 | x \rangle$

# QAOA

Algorithme

valeur estimées



Pour 3 qubits, avec un Ising de la forme  $3Z_0Z_1 - Z_1Z_2 + 2Z_0$

On veut directement estimer  $\langle x | H_1 | x \rangle$

Rappels  $\langle x | Z_j | x \rangle = 1$  si  $x = 0$

$\langle x | Z_j | x \rangle = -1$  le reste du temps

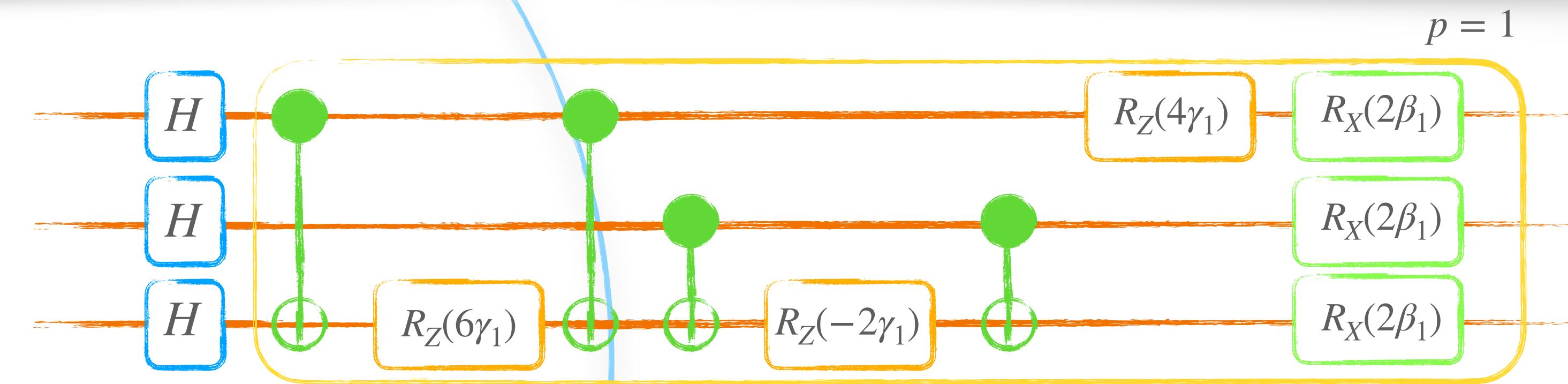
$\langle x | Z_j Z_k | x \rangle = 1$  si  $x_j = x_k$

$\langle x | Z_j Z_k | x \rangle = -1$  si  $x_j \neq x_k$

# QAOA

Algorithme

valeur estimées



Pour 3 qubits, avec un Ising de la forme  $3Z_0Z_1 - Z_1Z_2 + 2Z_0$

On veut directement estimer  $\langle x | H_1 | x \rangle$

Rappels  $\langle x | Z_j | x \rangle = 1$  si  $x = 0$

$\langle x | Z_j | x \rangle = -1$  le reste du temps

$\langle x | Z_j Z_k | x \rangle = 1$  si  $x_j = x_k$

$\langle x | Z_j Z_k | x \rangle = -1$  si  $x_j \neq x_k$

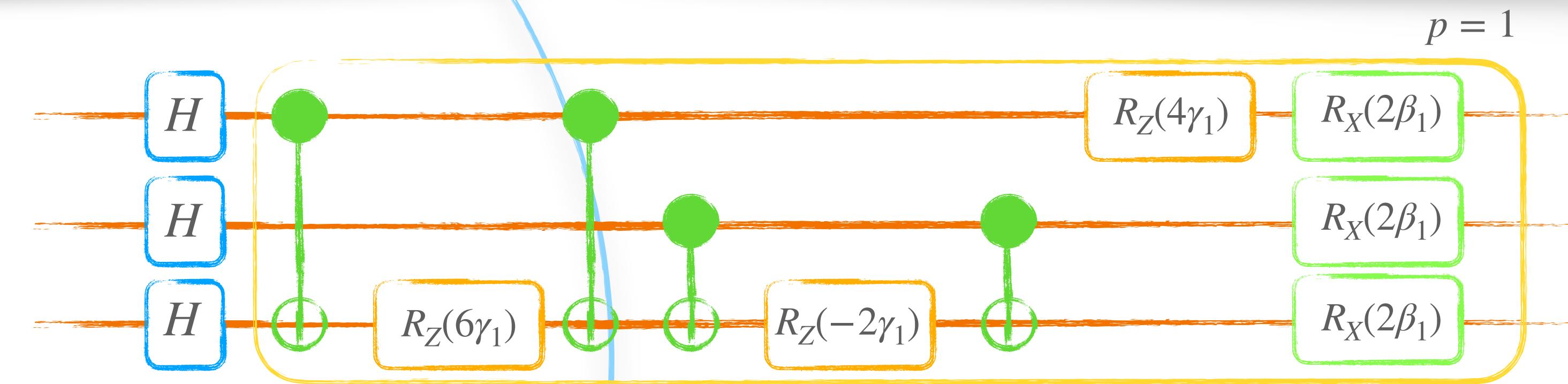
Pour l'état  $|101\rangle$   $\langle 101 | H_1 | 101 \rangle = 3\langle 101 | Z_0Z_2 | 101 \rangle - \langle 101 | Z_1Z_2 | 101 \rangle + 2\langle 101 | Z_0 | 101 \rangle = 3 + 1 - 2 = 2$

Pour l'état  $|100\rangle$   $\langle 100 | H_1 | 100 \rangle = 3\langle 100 | Z_0Z_2 | 100 \rangle - \langle 100 | Z_1Z_2 | 100 \rangle + 2\langle 100 | Z_0 | 100 \rangle = ?$

# QAOA

Algorithme

valeur estimées



Pour 3 qubits, avec un Ising de la forme  $3Z_0Z_1 - Z_1Z_2 + 2Z_0$

On veut directement estimer  $\langle x | H_1 | x \rangle$

Rappels  $\langle x | Z_j | x \rangle = 1$  si  $x = 0$

$\langle x | Z_j | x \rangle = -1$  le reste du temps

$\langle x | Z_j Z_k | x \rangle = 1$  si  $x_j = x_k$

$\langle x | Z_j Z_k | x \rangle = -1$  si  $x_j \neq x_k$

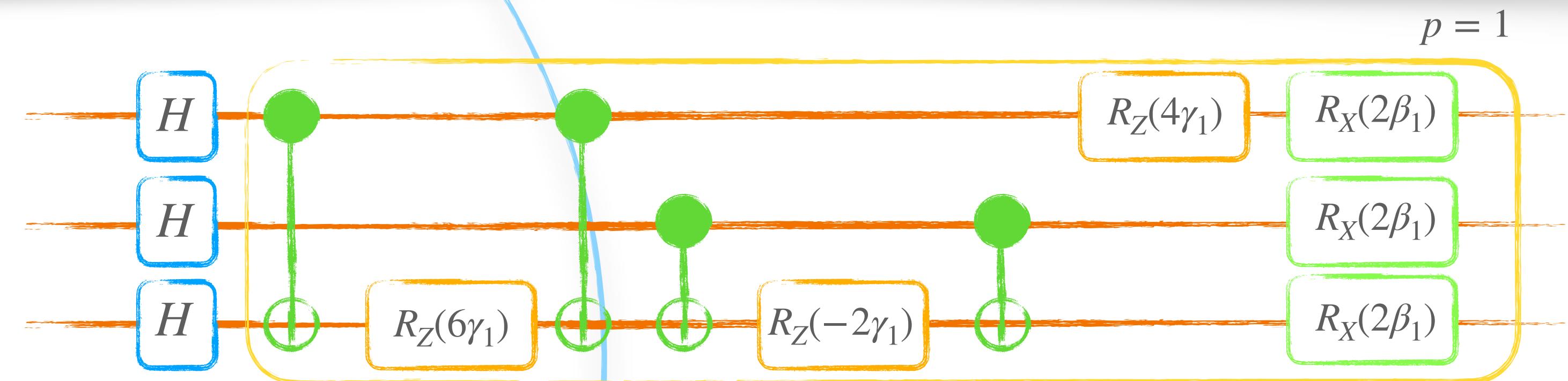
Pour l'état  $|101\rangle$   $\langle 101 | H_1 | 101 \rangle = 3\langle 101 | Z_0 Z_2 | 101 \rangle - \langle 101 | Z_1 Z_2 | 101 \rangle + 2\langle 101 | Z_0 | 101 \rangle = 3 + 1 - 2 = 2$

Pour l'état  $|100\rangle$   $\langle 100 | H_1 | 100 \rangle = 3\langle 100 | Z_0 Z_2 | 100 \rangle - \langle 100 | Z_1 Z_2 | 100 \rangle + 2\langle 100 | Z_0 | 100 \rangle = -3 - 1 - 2 = -6$

# QAOA

Algorithme

valeur estimées



Pour 3 qubits, avec un Ising de la forme  $3Z_0Z_1 - Z_1Z_2 + 2Z_0$

On veut directement estimer  $\langle x | H_1 | x \rangle$

Rappels  $\langle x | Z_j | x \rangle = 1$  si  $x = 0$

$\langle x | Z_j | x \rangle = -1$  le reste du temps

$\langle x | Z_j Z_k | x \rangle = 1$  si  $x_j = x_k$

$\langle x | Z_j Z_k | x \rangle = -1$  si  $x_j \neq x_k$

Forme générale pour sur un QC  $\langle \beta, \gamma | H_1 | \beta, \gamma \rangle \approx \sum_x \frac{m_x}{M} \langle x | H_1 | x \rangle$  pour M mesure du système

# QAOA

HOBO

Sat pb

Higher Order Binary Optimization

Polynomial Unconstrained Binary Optimization

SAT est le problème qui consiste à déterminer s'il existe une interprétation qui satisfait une formule booléenne donnée. En d'autres termes, il s'agit de savoir si les variables d'une formule booléenne donnée peuvent être systématiquement remplacées par les valeurs VRAI ou FAUX de manière à ce que la formule soit évaluée à TRUE. Si c'est le cas, la formule est dite **satisfaisable**. En revanche, si une telle affectation n'existe pas, la fonction exprimée par la formule est FALSE pour toutes les affectations de variables possibles et la formule est **insatisfaisante**.

SAT  $(x_0 \vee \neg x_1 \vee x_2) \wedge (\neg x_0 \vee x_1 \neg \vee x_2) \wedge (x_0 \vee x_1 \vee x_2)$

Minimiser  $(1 - x_0)x_1(1 - x_2) + x_0(1 - x_1)x_2 + (1 - x_0)(1 - x_1)(1 - x_2)$

Avec  $x_j \in \{0,1\}$   $j = 0,1,2$

Polynôme binaire de degrés 3

# QAOA

HOBO

Sat pb

Higher Order Binary Optimization

Polynomial Unconstrained Binary Optimization

SAT est le problème qui consiste à déterminer s'il existe une interprétation qui satisfait une formule booléenne donnée. En d'autres termes, il s'agit de savoir si les variables d'une formule booléenne donnée peuvent être systématiquement remplacées par les valeurs VRAI ou FAUX de manière à ce que la formule soit évaluée à TRUE. Si c'est le cas, la formule est dite **satisfaisable**. En revanche, si une telle affectation n'existe pas, la fonction exprimée par la formule est FALSE pour toutes les affectations de variables possibles et la formule est **insatisfaisante**.

SAT

$$(x_0 \vee x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee \neg x_2 \vee \neg x_3)$$

Minimiser

Avec

$$x_j \in \{0,1\}$$

$$j = 0,1,2,3$$

"Conjunctive normal form" ou CNF

Polynôme binaire de degrés ?

# QAOA

HOBO

Sat pb

Higher Order Binary Optimization

Polynomial Unconstrained Binary Optimization

SAT est le problème qui consiste à déterminer s'il existe une interprétation qui satisfait une formule booléenne donnée. En d'autres termes, il s'agit de savoir si les variables d'une formule booléenne donnée peuvent être systématiquement remplacées par les valeurs VRAI ou FAUX de manière à ce que la formule soit évaluée à TRUE. Si c'est le cas, la formule est dite **satisfaisable**. En revanche, si une telle affectation n'existe pas, la fonction exprimée par la formule est FALSE pour toutes les affectations de variables possibles et la formule est **insatisfaisante**.

SAT  $(x_0 \vee x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee \neg x_2 \vee \neg x_3)$

Minimiser  $(1 - x_0)(1 - x_1)x_2(1 - x_3) + x_0(1 - x_1)(1 - x_2)(1 - x_3) + x_0(1 - x_1)x_2x_3$

Avec  $x_j \in \{0,1\}$        $j = 0,1,2,3$

Polynôme binaire de degrés 4

3-SAT ou 3-CNF implique au moins 3 variables

# QAOA

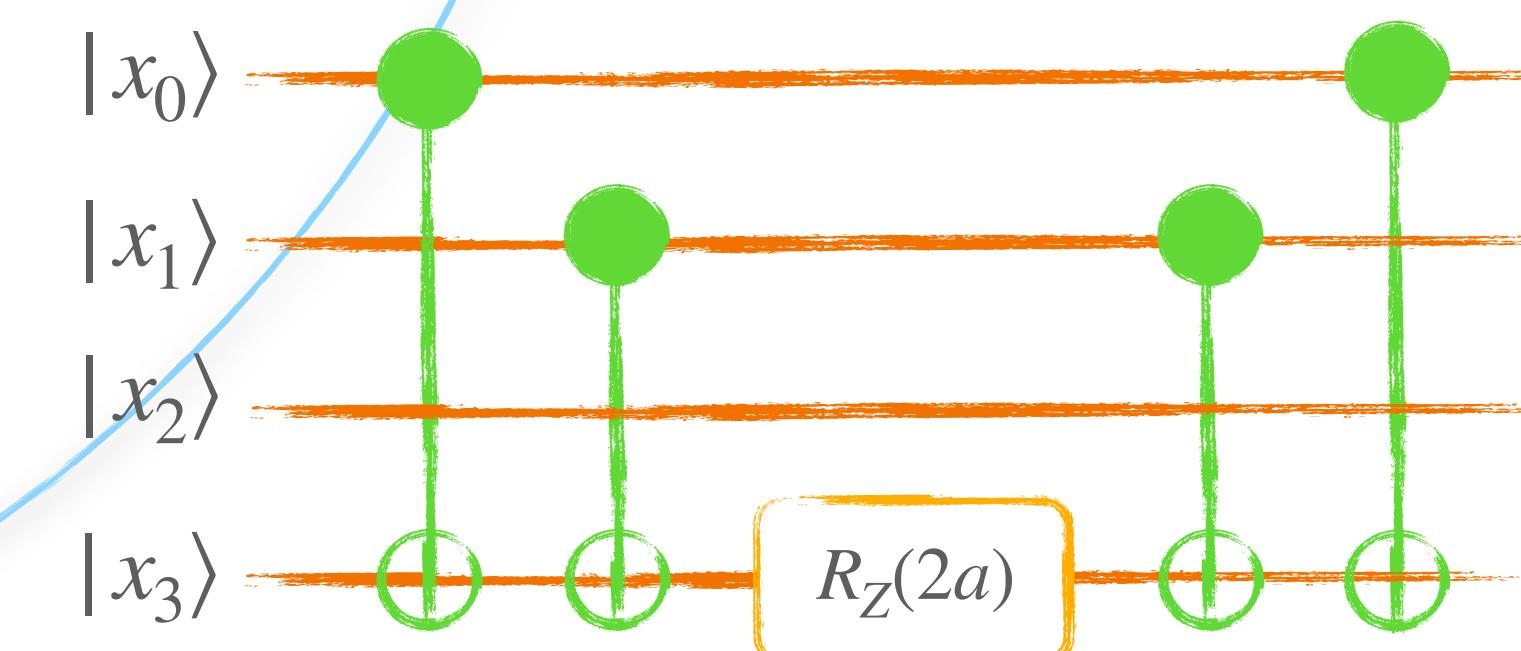
HOBO

Sat pb

SAT	$(x_0 \vee x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee \neg x_2 \vee \neg x_3)$
Minimiser	$(1 - x_0)(1 - x_1)x_2(1 - x_3) + x_0(1 - x_1)(1 - x_2)(1 - x_3) + x_0(1 - x_1)x_2x_3$
Avec	$x_j \in \{0,1\}$ $j = 0,1,2,3$

Résolution comme avec le QUBO et les portes Z

$$U(x) = e^{iaZ_0Z_1Z_3}$$



Higher Order Binary Optimization  
Polynomial Unconstrained Binary Optimization

# QAOA

HOBO

Sat pb

SAT  $(x_0 \vee x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee \neg x_2 \vee \neg x_3)$

Minimiser  $(1 - x_0)(1 - x_1)x_2(1 - x_3) + x_0(1 - x_1)(1 - x_2)(1 - x_3) + x_0(1 - x_1)x_2x_3$

Avec  $x_j \in \{0,1\}$   $j = 0,1,2,3$

Higher Order Binary Optimization

Polynomial Unconstrained Binary Optimization

Résolution comme avec le QUBO et les portes Z

$$U(x) = e^{i\frac{\pi}{4}Z_0Z_2Z_4} \quad ??$$

# QAOA

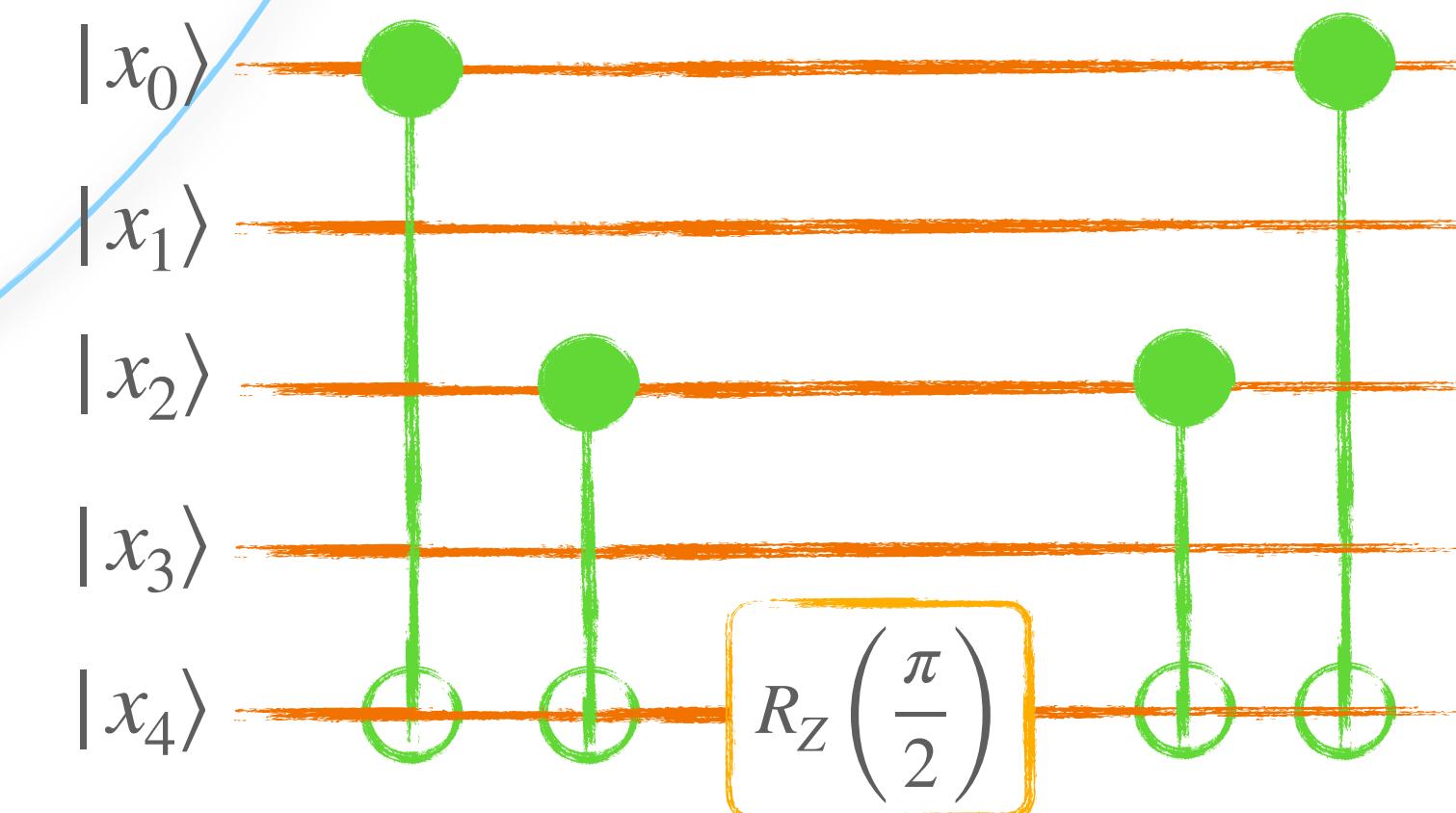
HOBO

Sat pb

SAT	$(x_0 \vee x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee \neg x_2 \vee \neg x_3)$
Minimiser	$(1 - x_0)(1 - x_1)x_2(1 - x_3) + x_0(1 - x_1)(1 - x_2)(1 - x_3) + x_0(1 - x_1)x_2x_3$
Avec	$x_j \in \{0,1\}$ $j = 0,1,2,3$

Résolution comme avec le QUBO et les portes Z

$$U(x) = e^{i\frac{\pi}{4}Z_0Z_2Z_4}$$



Higher Order Binary Optimization  
Polynomial Unconstrained Binary Optimization

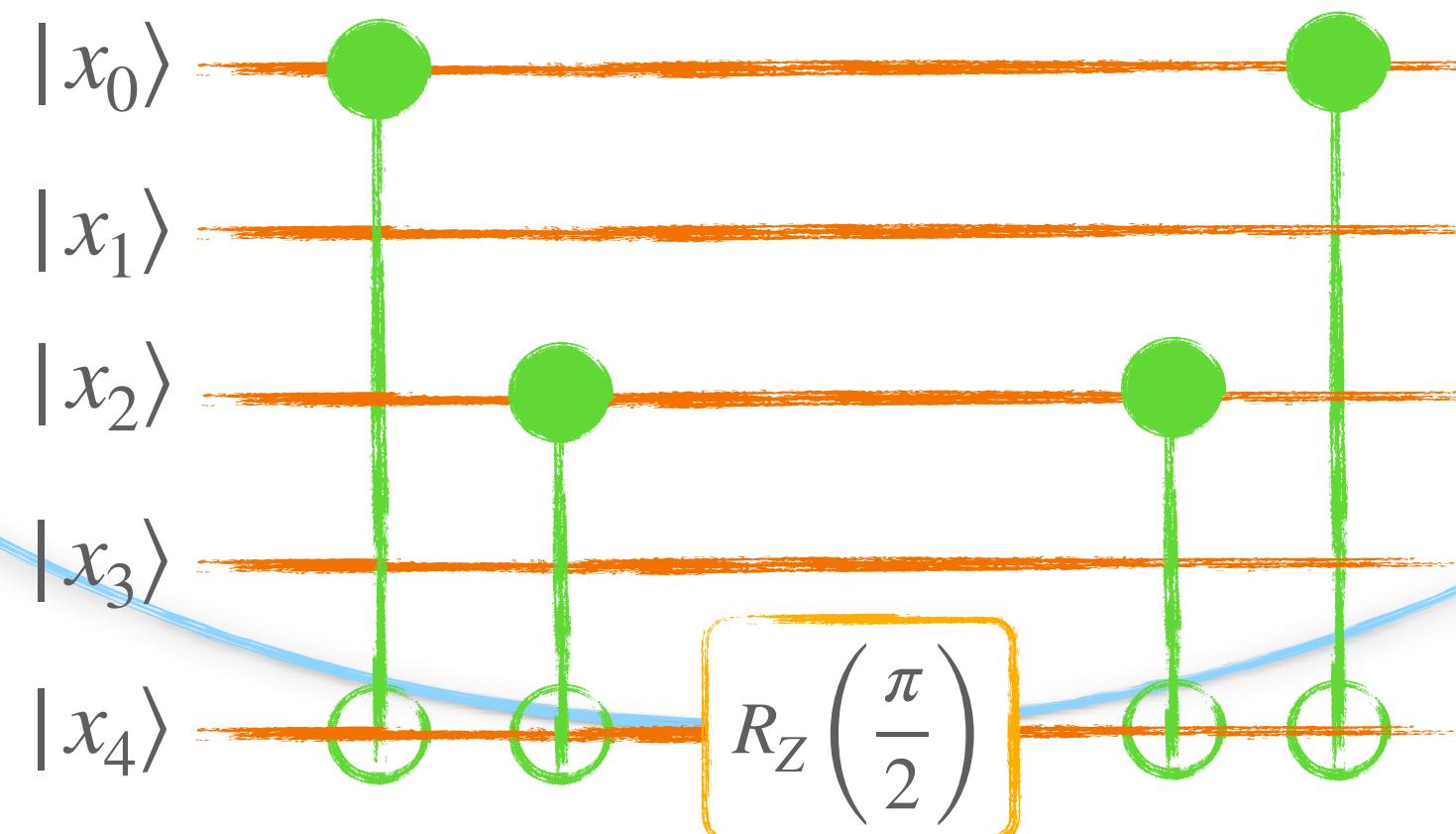
# QAOA

HOBO

Sat pb

	Higher Order Binary Optimization
	Polynomial Unconstrained Binary Optimization
SAT	$(x_0 \vee x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee x_2 \vee x_3) \wedge (\neg x_0 \vee x_1 \vee \neg x_2 \vee \neg x_3)$
Minimiser	$(1 - x_0)(1 - x_1)x_2(1 - x_3) + x_0(1 - x_1)(1 - x_2)(1 - x_3) + x_0(1 - x_1)x_2x_3$
Avec	$x_j \in \{0,1\}$
	$j = 0,1,2,3$

Résolution comme avec le QUBO et les portes Z



$$\langle x | Z_{j_1}Z_{j_2}\dots Z_{j_m} | x \rangle = 1$$

Si la somme des bits de  $x$  en position  $j_1, j_2, \dots, j_m$  est pair  
sinon  $\langle x | Z_{j_1}Z_{j_2}\dots Z_{j_m} | x \rangle = -1$

# QAOA

HOBO

$$\langle \psi | H_1 | \psi \rangle ? \quad H_1 = Z_0 Z_1 Z_2 + 3Z_0 Z_2 - Z_1 Z_2 + 2Z_0$$

$$\langle 100 | H_1 | 100 \rangle ?$$

# QAOA

HOBO

$$\langle \psi | H_1 | \psi \rangle ? \quad H_1 = Z_0 Z_1 Z_2 + 3Z_0 Z_2 - Z_1 Z_2 + 2Z_0$$

$$\langle 100 | H_1 | 100 \rangle ?$$

$$\langle 100 | H_1 | 100 \rangle = \langle 100 | Z_0 Z_1 Z_2 | 100 \rangle + 3\langle 100 | Z_0 Z_2 | 100 \rangle - \langle 100 | Z_1 Z_2 | 100 \rangle + 2\langle 100 | Z_0 | 100 \rangle = -1 - 3 - 1 - 2 = -7$$

# QAOA

Code!!

$$H = 2Z_0Z_1 - Z_0Z_2 + 3.5Z_1$$

```
import pennylane as qml  
from pennylane import PauliZ
```

```
coef = [2, -1, 3.5]
```

```
paulis = [PauliZ(0)@PauliZ(1), PauliZ(0)@PauliZ(2), PauliZ(1)]
```

```
H = qml.Hamiltonian(coef, paulis)
```

```
print(H)
```

```
>> (3.5) [Z1]  
+ (-1) [Z0 Z2]  
+ (2) [Z0 Z1]
```

Importation des modules

Construction de l'Hamiltonien

$$H = 2 * \text{PauliZ}(0)@\text{PauliZ}(1) - \text{PauliZ}(0)@\text{PauliZ}(2) + 3.5 * \text{PauliZ}(1)$$

# QAOA

Code!!

$$H = -3Z_0Z_1Z_2 + 2Z_1Z_2 - Z_2$$

```
import pennylane as qml  
from pennylane import PauliZ
```

Importation des modules

```
coef = [-3, 2, -1]
```

```
paulis = [PauliZ(0)@PauliZ(1)@PauliZ(2), PauliZ(1)@PauliZ(2), PauliZ(2)]
```

```
H = qml.Hamiltonian(coef, paulis)
```

```
print(H)  
>> (-1) [Z2]  
+ (2) [Z1 Z2]  
+ (-3) [Z0 Z1 Z2]
```

$H = -3*PauliZ(0)@PauliZ(1)@PauliZ(2)$   
 $+ 2*PauliZ(1)@PauliZ(2)$   
 $- PauliZ(2)$

# QAOA

Code!!

$$H_0 = X_0 + X_1$$
$$H_1 = Z_0 Z_1$$

```
import pennylane as qml
from pennylane import Pauliz, Paulix, qaoa

H_O = Paulix(0) + Paulix(1)
H_I = 1.0*Pauliz(0)@Pauliz(1)

wires = [0, 1]

dev = qml.device('default.qubit', wires=wires)

p=2
```

```
@qml.qnode(dev)
def energy(angles, p, wires):
    for w in wires:
        qml.Hadamard(w)
    for i in range(p):
        qaoa.cost_layer(angles[2*i+1], H_I)
        qaoa.mixer_layer(angles[2*i], H_O)
    return qml.expval(H_I)
```

# QAOA

Code!!

$$H_0 = X_0 + X_1$$
$$H_1 = Z_0 Z_1$$

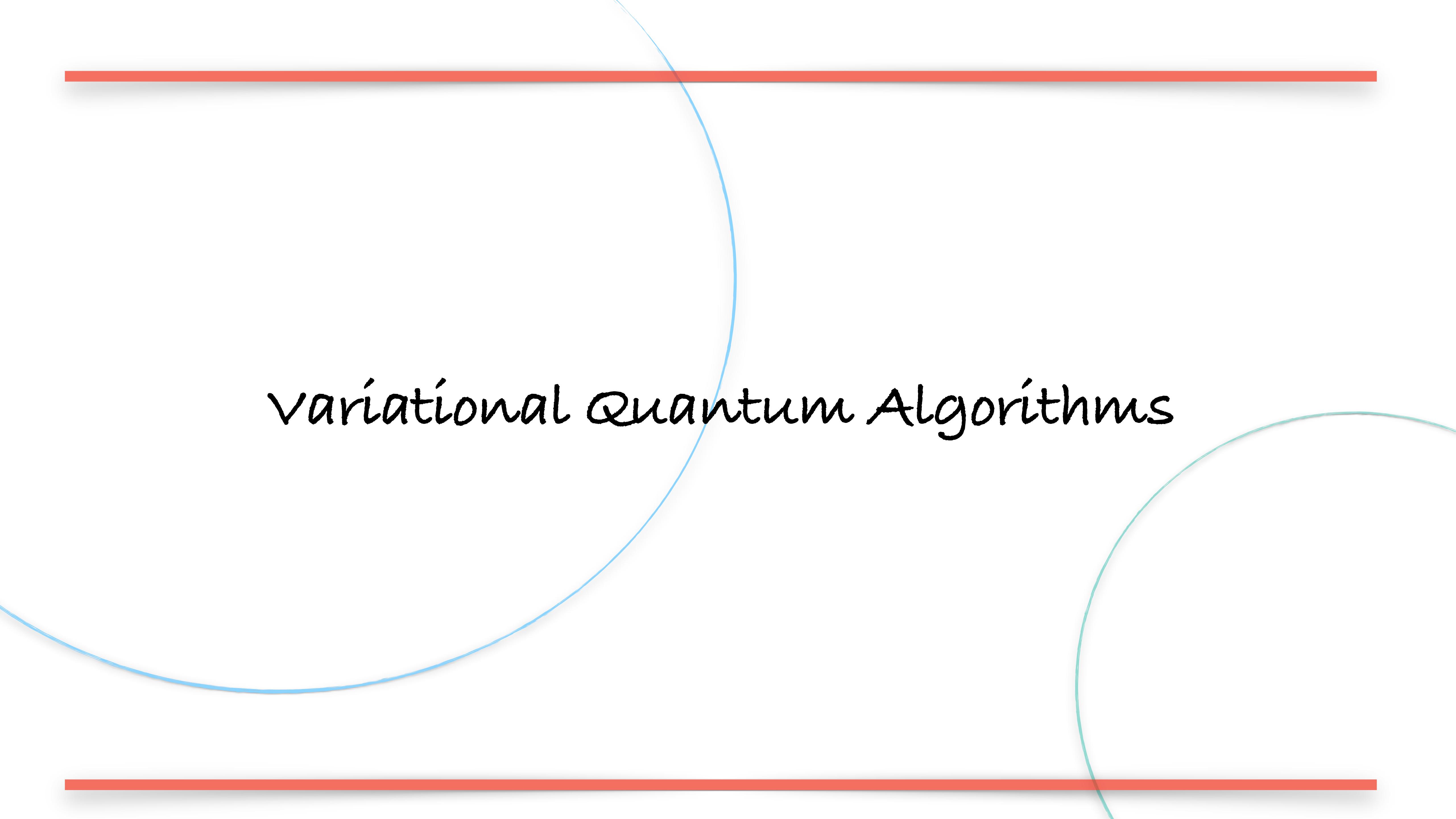
```
@qml.qnode(dev)
def energy(angles, p, wires):
    for w in wires:
        qml.Hadamard(w)
    for i in range(p):
        qaoa.cost_layer(angles[2*i+1], H_1)
        qaoa.mixer_layer(angles[2*i], H_0)
    return qml.expval(H_1)
```

```
from pennylane import numpy as np
optimizer = qml.GradientDescentOptimizer()
steps = 200
angles = np.array([1., 1., 1., 1.], requires_grad=True)

for i in range(steps):
    angles, _, _ = optimizer.step(energy, angles, p, wires)

print(f'Optimal angles : {angles}')

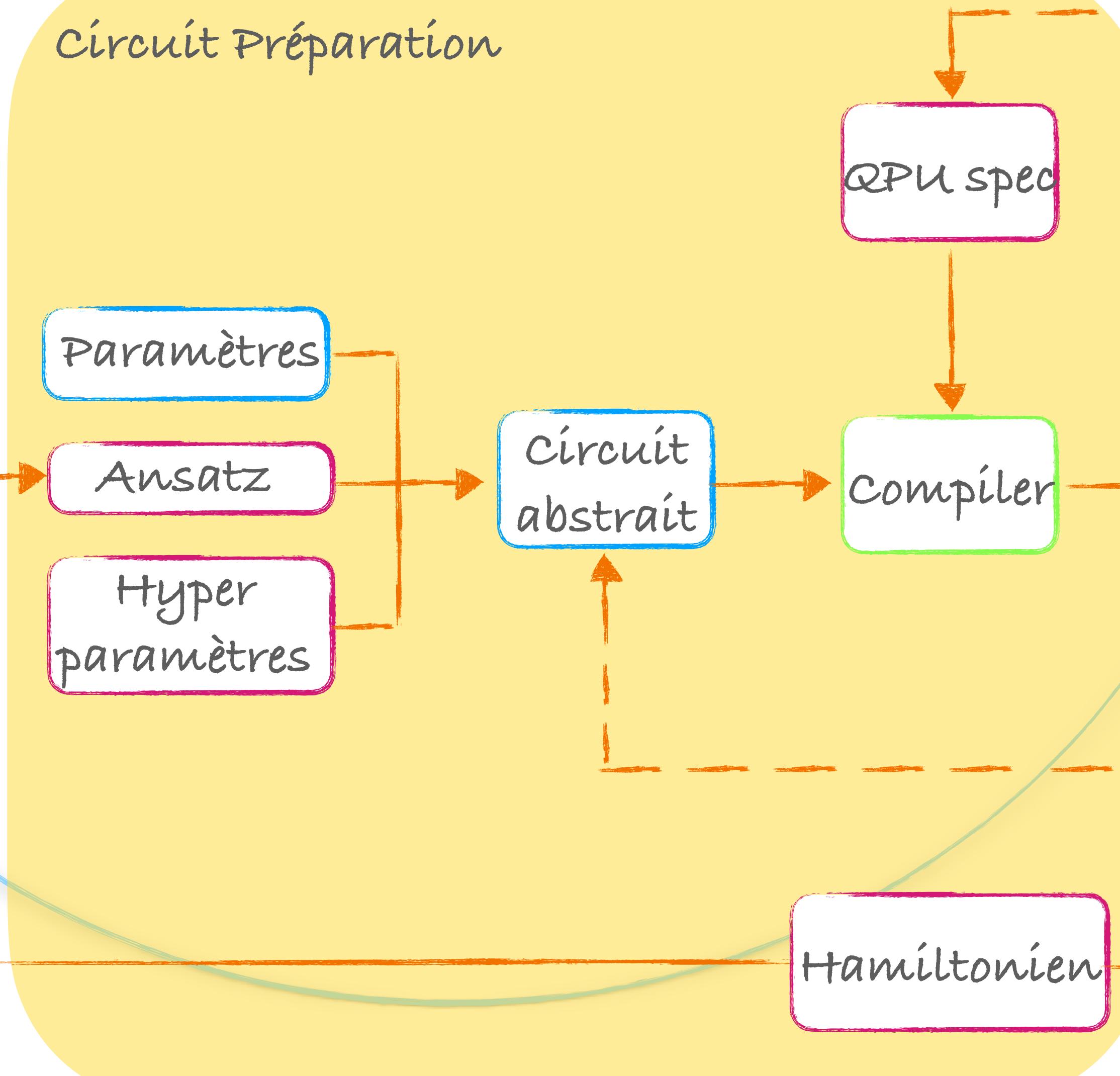
Optimal angles : [0.78856093 0.60745954
                  1.18106183 1.39284456]
```



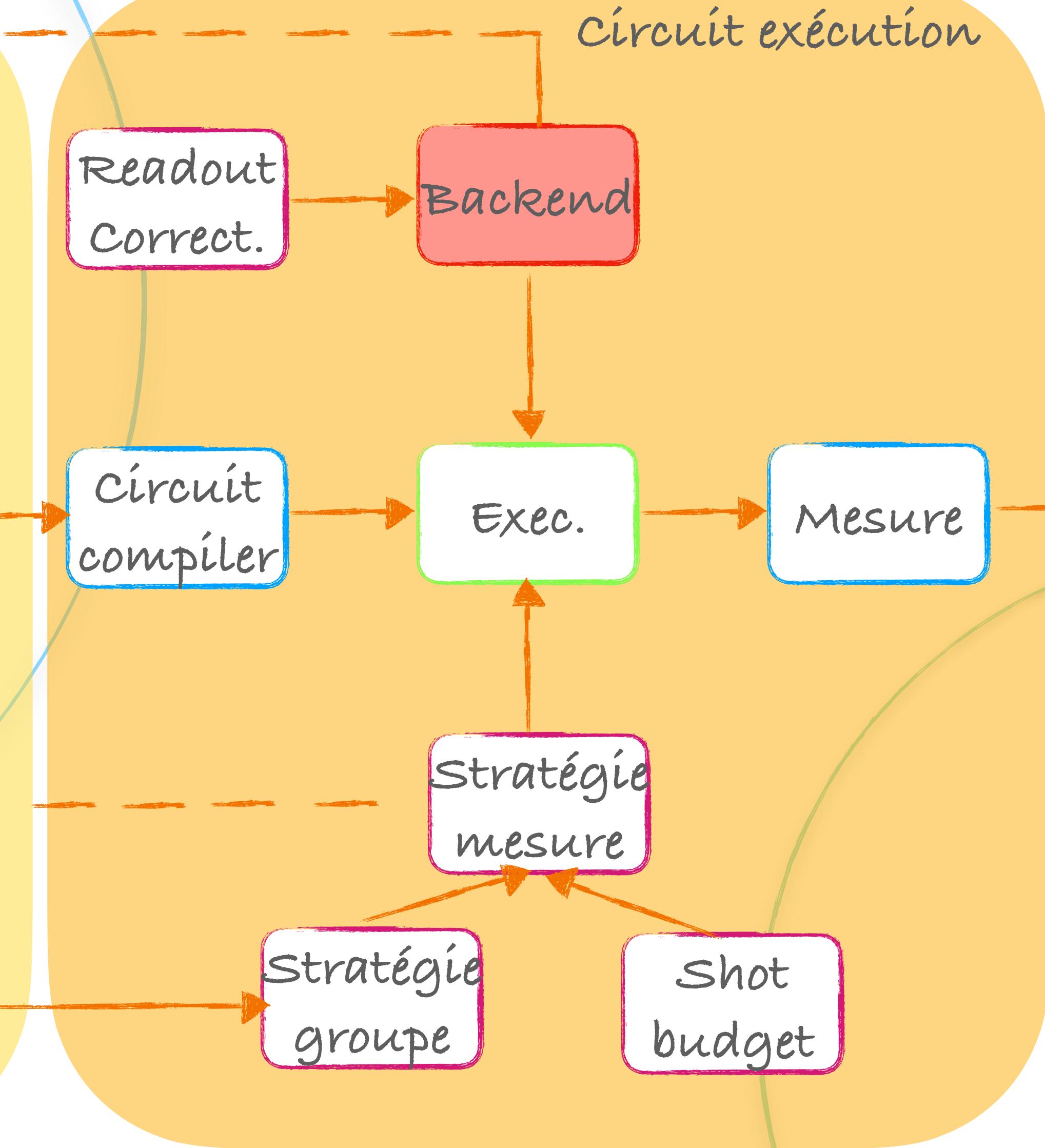
variational Quantum Algorithms

# Variational Quantum Algorithms

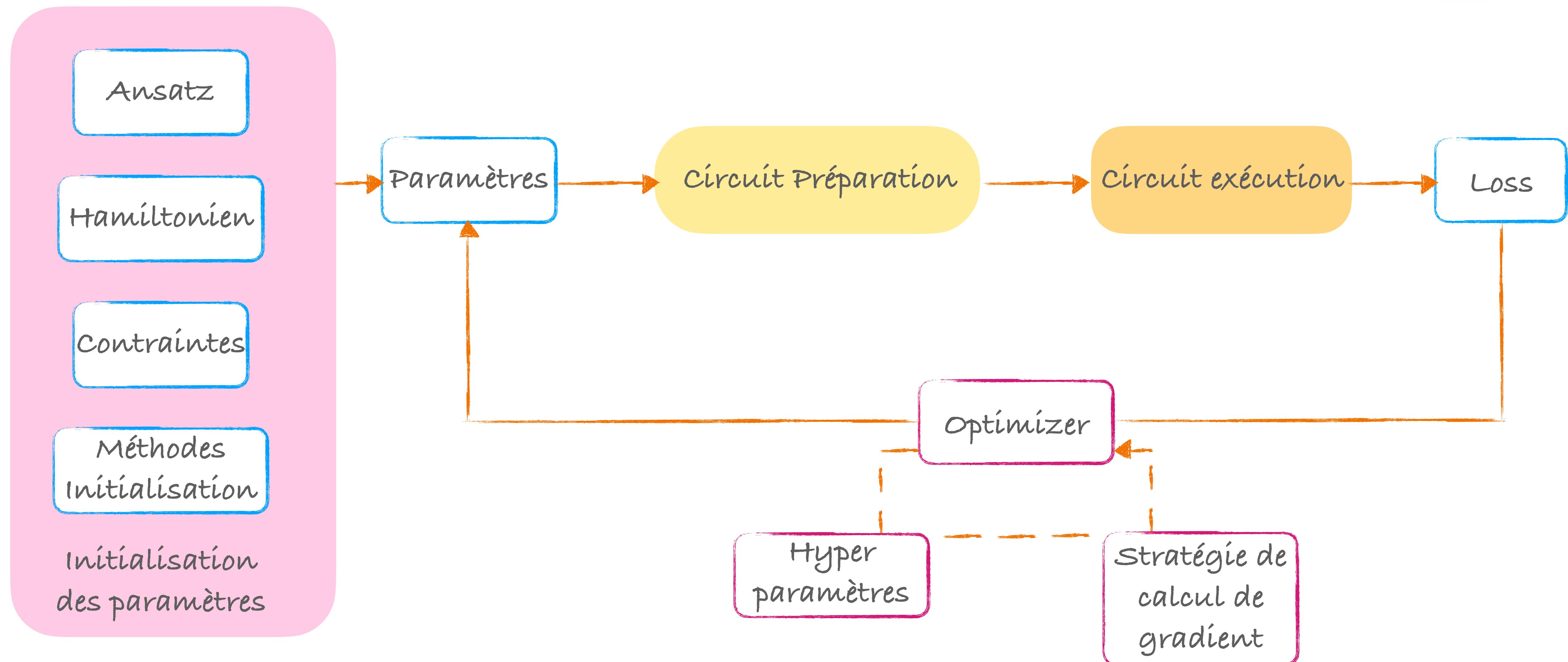
Circuit Préparation

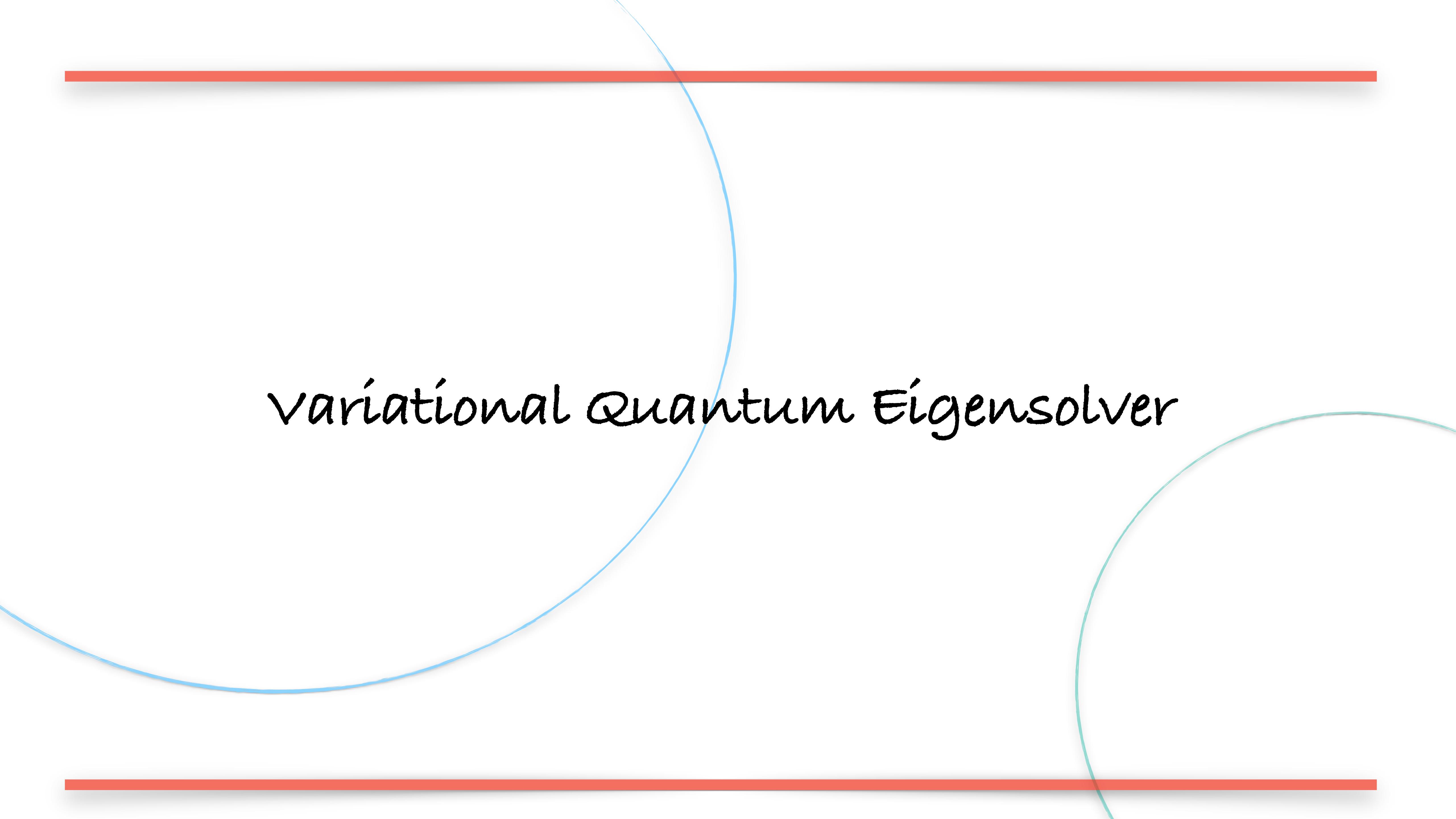


Circuit exécution



# Variational Quantum Algorithms





variational Quantum Eigensolver

# Variational Quantum Eigensolver

valeur attendue

Partons d'un Hamiltonien  $H_f$  qui est une représentation d'une fonction  $f$  tel que:

$$\langle x | H_f | x \rangle = f(x) \quad \text{pour tout bitstring } x$$

On peut trouver le "ground state"  $|\psi\rangle$  de telle sorte que  $\langle \psi | H_f | \psi \rangle$  soit minimum

Dans la base de calcul:  $H_f |x\rangle = f(x) |x\rangle$  ce qui se traduit par  $|x\rangle$  eingenvecteur

$f(x)$  eingenvalue associée

Prenons une observable  $A$  qui est "Hermitian"  $A^\dagger = A$  on peut ajouter une magnitude qui est...

$f(x)$  eingenvalue associée

La mesure est une observable

# variational Quantum Eigensolver

valeur attendue

La valeur attendue d'une observable  $A$  via l'état  $|\psi\rangle$  est définie comme:

Hamiltonien

$$\langle A \rangle_{\psi} = \sum_{j,k} |\langle \lambda_j^k | \psi \rangle|^2 \lambda_j = \langle \psi | A | \psi \rangle$$

$\alpha_j$

eigenvalue

eigenvecteur

Comment l'estimer?

$\langle A \rangle_{\psi}$  n'est pas toujours diagonalisable

# Variational Quantum Eigensolver

valeur attendue

La valeur attendue d'une observable  $A$  via l'état  $|\psi\rangle$  est définie comme:

Hamiltonien

$$\langle A \rangle_{\psi} = \sum_{j,k} |\langle \lambda_j^k | \psi \rangle|^2 \lambda_j = \langle \psi | A | \psi \rangle$$

$\alpha_j$

eigenvalue  
eigenvecteur

Rôle du VQE

inconnus

Comment l'estimer?

$\langle A \rangle_{\psi}$  n'est pas toujours diagonalisable

# Variational Quantum Eigensolver

valeur attendue

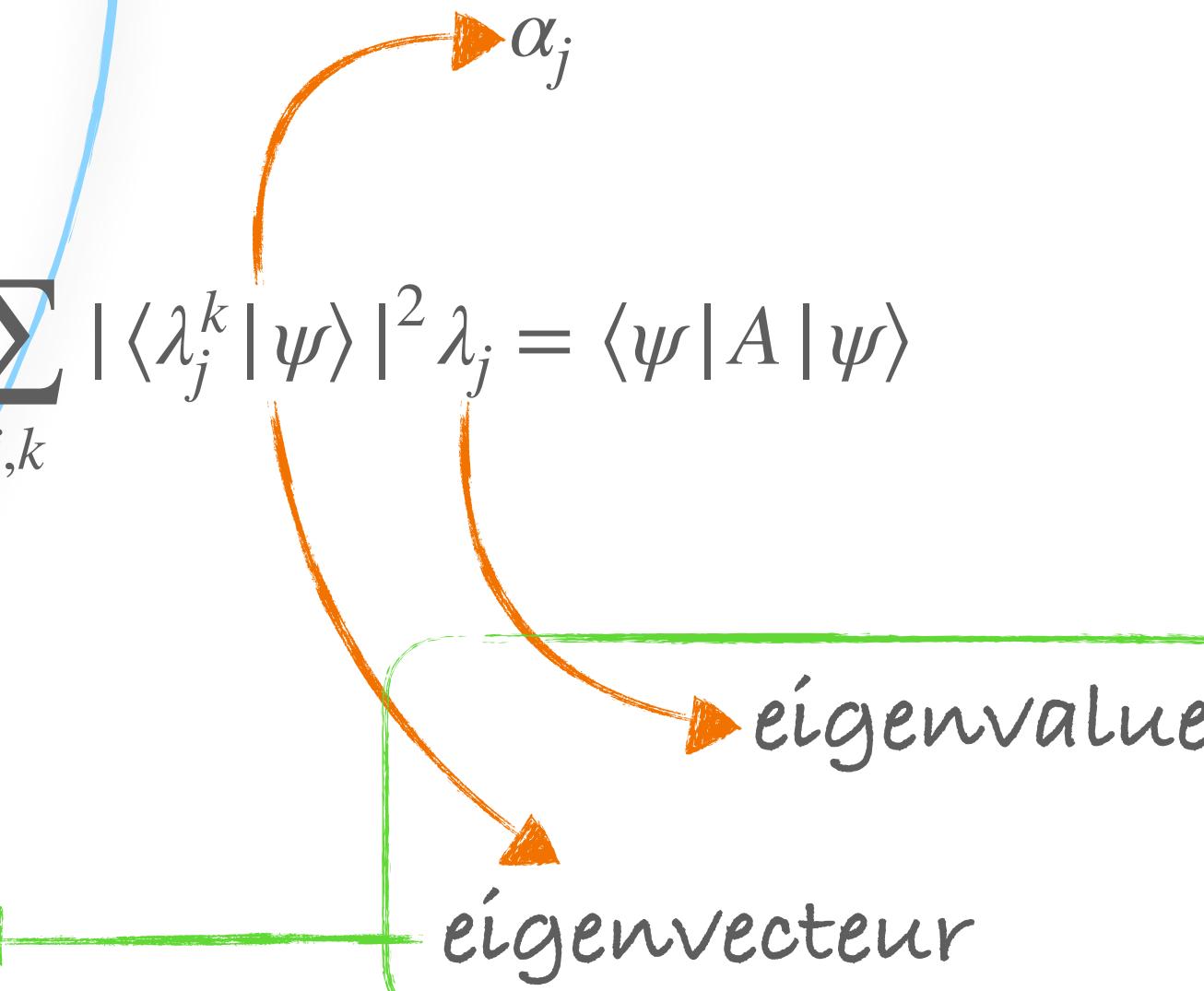
La valeur attendue d'une observable  $A$  via l'état  $|\psi\rangle$  est définie comme:

Hamiltonien  $\longrightarrow \langle A \rangle_{\psi} = \sum_{j,k} |\langle \lambda_j^k | \psi \rangle|^2 \lambda_j = \langle \psi | A | \psi \rangle$

Augmente exponentiellement avec le nombre de qubits

Comment l'estimer?

$\langle A \rangle_{\psi}$  n'est pas toujours diagonalisable



Rôle du VQE

Inconnus

# Variational Quantum Eigensolver

valeur attendue

La valeur attendue d'une observable  $A$  via l'état  $|\psi\rangle$  est définie comme:

Hamiltonien

Solution?

Somme de Pauli's Matrices  
Tensor product

Temps de calcul très grand

Augmente exponentiellement  
avec le nombre de qubits

Comment l'estimer?

$\langle A \rangle_\psi$  n'est pas toujours diagonalisable

$$\langle A \rangle_\psi = \sum_{j,k} |\langle \lambda_j^k | \psi \rangle|^2 \lambda_j = \langle \psi | A | \psi \rangle$$

$\alpha_j$

eigenvalue  
eigenvecteur

Rôle du VQE

Inconnus

# variational Quantum Eigensolver

Exemple

Hamiltonien:

$$A = \frac{1}{2}Z \otimes I \otimes X - 3I \otimes Y \otimes Y + 2Z \otimes X \otimes Z$$

valeur attendue:

$$\langle \psi | A | \psi \rangle = \langle \psi | \left( \frac{1}{2}Z \otimes I \otimes X - 3I \otimes Y \otimes Y + 2Z \otimes X \otimes Z \right) | \psi \rangle$$

Par linéarité:

$$\langle \psi | A | \psi \rangle = \frac{1}{2} \underbrace{\langle \psi | (Z \otimes I \otimes X) | \psi \rangle}_{\text{valeur attendue}} - 3 \underbrace{\langle \psi | (I \otimes Y \otimes Y) | \psi \rangle}_{\text{valeur attendue}} + 2 \underbrace{\langle \psi | (Z \otimes X \otimes Z) | \psi \rangle}_{\text{valeur attendue}}$$

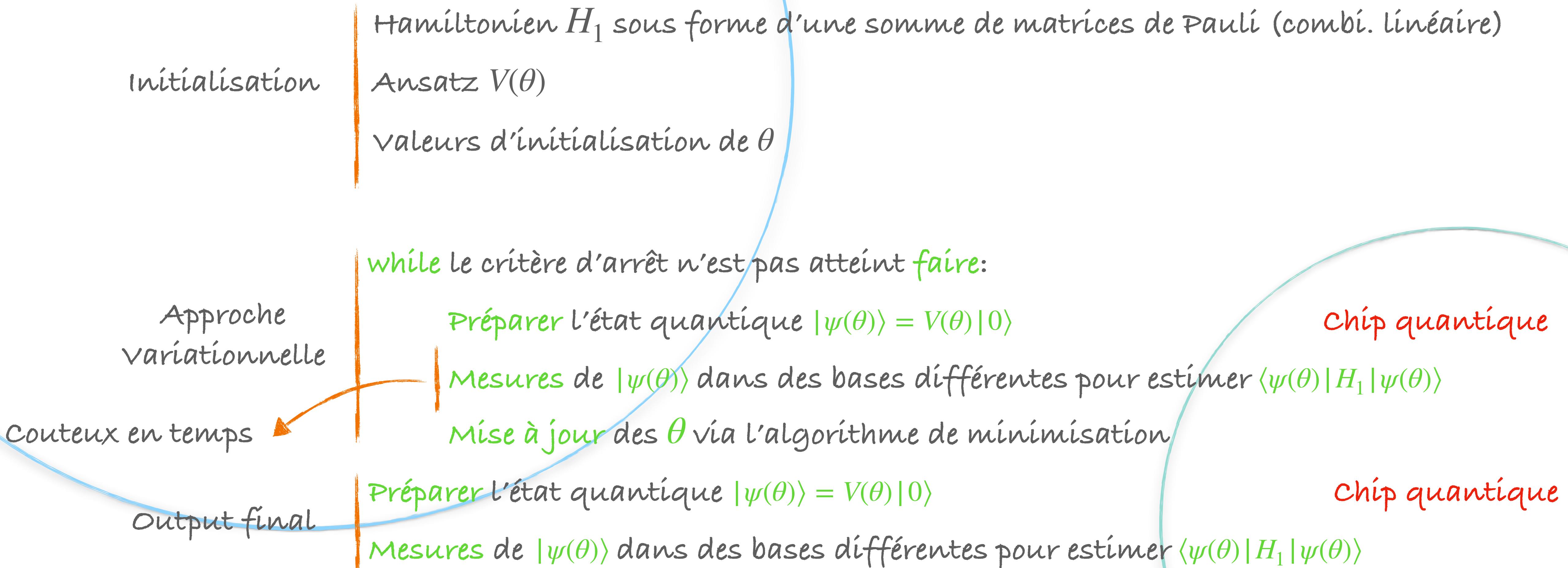


Matrices de Pauli, simple d'obtenir les Eigenvalues et Eigenvecteurs

L'objectif du VQE est de trouver la valeur minimum de l'Hamiltonien, la eigenvalue la plus faible.

# Variational Quantum Eigensolver

## Pseudo-code



# Variational Quantum Eigensolver

États excités?

Un VQE peut aussi être utilisé pour trouver des états excités et non juste des états stables. Il faut alors trouver d'autres eigenvalues.

État fondamental  $|\psi_0\rangle = V(\theta_0)|0\rangle$  avec l'énergie  $\lambda_0$  trouvé par le VQE précédent

On peut formuler un Hamiltonien modifié:

$$H' = H + C|\psi_0\rangle\langle\psi_0|$$

Hamiltonien excité

Niveau fondamental

Hermitien

Matrice carrée

Constante valeur réelle positive

$$(|\psi_0\rangle\langle\psi_0|)^\dagger = \langle\psi_0|^\dagger|\psi_0\rangle^\dagger = |\psi_0\rangle\langle\psi_0|$$

# variational quantum Eigensolver

États excités?

$$H' = H + C|\psi_0\rangle\langle\psi_0|$$

valeur attendue

$$\langle\psi|H'|\psi\rangle = \langle\psi|H|\psi\rangle + C\langle\psi|\psi_0\rangle\langle\psi_0|\psi\rangle$$

# variational quantum Eigensolver

États excités?

$$H' = H + C|\psi_0\rangle\langle\psi_0|$$

valeur attendue

$$\langle\psi|H'|\psi\rangle = \langle\psi|H|\psi\rangle + C\langle\psi|\psi_0\rangle\langle\psi_0|\psi\rangle = \langle\psi|H|\psi\rangle + C|\langle\psi_0|\psi\rangle|^2$$

# variational quantum Eigensolver

États excités?

$$H' = H + C|\psi_0\rangle\langle\psi_0|$$

valeur attendue

$$\langle\psi|H'|\psi\rangle = \langle\psi|H|\psi\rangle + C\langle\psi|\psi_0\rangle\langle\psi_0|\psi\rangle = \langle\psi|H|\psi\rangle + C|\langle\psi_0|\psi\rangle|^2$$

valeur estimée de  $H$  pour l'état fondamental

valeur non négative qui représente la distance ou l'overlap entre  $\psi_0$  et  $\psi$

Deux cas extrêmes  $|\psi\rangle = |\psi_0\rangle$ ;  $|\langle\psi_0|\psi\rangle|^2 = 1$  et  $|\psi\rangle \perp |\psi_0\rangle$ ;  $|\langle\psi_0|\psi\rangle|^2 = 0$

Il faut choisir  $C$  assez gros pour ne pas être dans l'état fondamental

# variational quantum Eigensolver

États excités?

$$H' = H + C|\psi_0\rangle\langle\psi_0|$$

valeur attendue

$$\langle\psi|H'|\psi\rangle = \langle\psi|H|\psi\rangle + C\langle\psi|\psi_0\rangle\langle\psi_0|\psi\rangle = \langle\psi|H|\psi\rangle + C|\langle\psi_0|\psi\rangle|^2$$

$\lambda_0$  est l'état d'énergie fondamental

$\lambda_1$  est l'état d'énergie du premier niveau excité

On peut ainsi déterminer tous les états excités:

$$\langle\psi|H''|\psi\rangle = \langle\psi|H'|\psi\rangle + C'\langle\psi|\lambda_1\rangle\langle\lambda_1|\psi\rangle$$

Problème?

# variational quantum Eigensolver

États excités?

$$H' = H + C|\psi_0\rangle\langle\psi_0|$$

valeur attendue

$$\langle\psi|H'|\psi\rangle = \langle\psi|H|\psi\rangle + C\langle\psi|\psi_0\rangle\langle\psi_0|\psi\rangle = \langle\psi|H|\psi\rangle + C|\langle\psi_0|\psi\rangle|^2$$

$\lambda_0$  est l'état d'énergie fondamental

$\lambda_1$  est l'état d'énergie du premier niveau excité

On peut ainsi déterminer tous les états excités:

$$\langle\psi|H''|\psi\rangle = \langle\psi|H'|\psi\rangle + C'\langle\psi|\lambda_1\rangle\langle\lambda_1|\psi\rangle$$

Problème?

$|\psi_0\rangle\langle\psi_0|$  n'est pas une somme d'opérateurs (Pauli gates)

Mais!

# variational Quantum Eigensolver

États excités?

$$H' = H + C|\psi_0\rangle\langle\psi_0|$$

valeur attendue

$$\langle\psi|H'|\psi\rangle = \langle\psi|H|\psi\rangle + C\langle\psi|\psi_0\rangle\langle\psi_0|\psi\rangle = \langle\psi|H|\psi\rangle + C|\langle\psi_0|\psi\rangle|^2$$

$\lambda_0$  est l'état d'énergie fondamental

$\lambda_1$  est l'état d'énergie du premier niveau excité

On peut ainsi déterminer tous les états excités:

$$\langle\psi|H''|\psi\rangle = \langle\psi|H'|\psi\rangle + C'\langle\psi|\lambda_1\rangle\langle\lambda_1|\psi\rangle$$

Problème?

$|\psi_0\rangle\langle\psi_0|$  n'est pas une somme d'opérateurs (Pauli gates)

Mais!

$$V(\theta_0)|0\rangle = |\psi_0\rangle$$

# variational quantum Eigensolver

États excités?

Problème?

$|\psi_0\rangle\langle\psi_0|$  n'est pas une somme d'opérateurs (Pauli gates)

Mais!

$$V(\theta_0)|0\rangle = |\psi_0\rangle$$

$$\langle\psi(\theta)|\psi_0\rangle\langle\psi_0|\psi(\theta)\rangle = |\langle\psi_0|\psi(\theta)\rangle|^2 = \underbrace{\left|\langle 0|V(\theta_0)^\dagger V(\theta)|0\rangle\right|^2}$$

Probabilité d'obtenir l'état  $|0\rangle$

# variational quantum Eigensolver

États excités?

Problème?

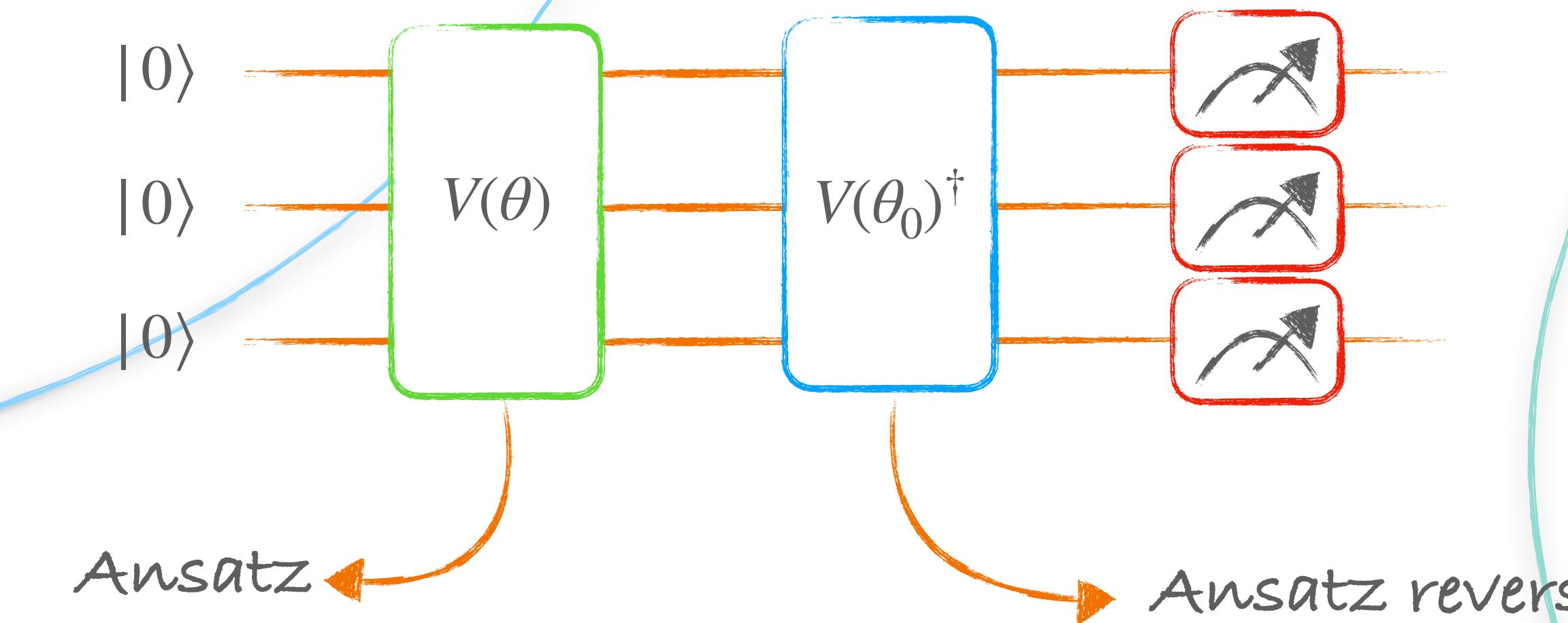
$|\psi_0\rangle\langle\psi_0|$  n'est pas une somme d'opérateurs (Pauli gates)

Mais!

$$V(\theta_0)|0\rangle = |\psi_0\rangle$$

$$\langle\psi(\theta)|\psi_0\rangle\langle\psi_0|\psi(\theta)\rangle = |\langle\psi_0|\psi(\theta)\rangle|^2 = \underbrace{|\langle 0|V(\theta_0)^\dagger V(\theta)|0\rangle|^2}$$

Probabilité d'obtenir l'état  $|0\rangle$



# variational quantum Eigensolver

États excités?

Problème?

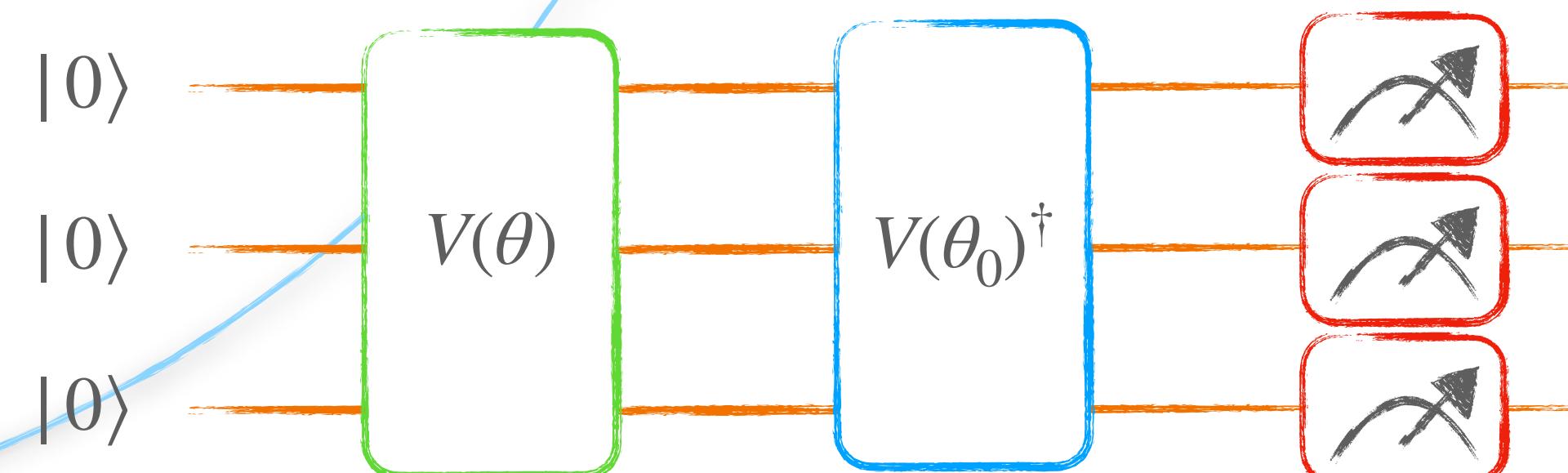
$|\psi_0\rangle\langle\psi_0|$  n'est pas une somme d'opérateurs (Pauli gates)

Mais!

$$V(\theta_0)|0\rangle = |\psi_0\rangle$$

$$\langle\psi(\theta)|\psi_0\rangle\langle\psi_0|\psi(\theta)\rangle = |\langle\psi_0|\psi(\theta)\rangle|^2 = \left|\langle 0 | V(\theta_0)^\dagger V(\theta) | 0 \rangle\right|^2$$

Probabilité d'obtenir l'état  $|0\rangle$



Comme les opérations sont réversibles:

$$\text{Si } \theta_0 = (a, b)$$

$$V(\theta_0) = X R_z(a) R_x(b) S$$

$$V(\theta_0)^\dagger = S^\dagger R_x(-b) R_z(-a) X$$

# variational Quantum Eigensolver

code!!

Molécule  $H_2$

```
import pennylane as qml  
from pennylane import numpy as np
```

```
seed = 42
```

```
np.random.seed(seed)
```

```
symbols = ["H", "H"]
```

```
coordinates = np.array([0., 0., -0.6991986158,  
0., 0., 0.6991986158])
```

```
H, qubits = qml.qchem.molecular_hamiltonian(symbols, coordinates)
```

```
print("Qubit Hamiltonian:")
```

```
print(H)
```

# variational Quantum Eigensolver

code!!

Molécule  $H_2$

```
print("Qubit Hamiltonian:")
print(H)
```

Qubit Hamiltonian:

$$\begin{aligned} & (-0.22343155727069636) [Z_2] \\ & + (-0.22343155727069636) [Z_3] \\ & + (-0.09706620778648187) [I_0] \\ & + (0.1714128349818368) [Z_0] \\ & + (0.1714128349818368) [Z_1] \\ & + (0.12062523781171192) [Z_0 Z_2] \\ & + (0.12062523781171192) [Z_1 Z_3] \\ & + (0.1659278524199735) [Z_0 Z_3] \\ & + (0.1659278524199735) [Z_1 Z_2] \\ & + (0.16868898461465948) [Z_0 Z_1] \\ & + (0.1744128778003259) [Z_2 Z_3] \\ & + (-0.045302614608261585) [Y_0 Y_1 X_2 X_3] \\ & + (-0.045302614608261585) [X_0 X_1 Y_2 Y_3] \\ & + (0.045302614608261585) [Y_0 X_1 X_2 Y_3] \\ & + (0.045302614608261585) [X_0 Y_1 Y_2 X_3] \end{aligned}$$

# Variational Quantum Eigensolver

code!!

Molécule  $H_2$

nqubits = 4

```
def efficientSU2(theta):
    # first part of the ansatz
    for i in range(nqubits):
        qml.RY(theta[i], wires = i)
        qml.RZ(theta[i+nqubits], wires = i)
    # entanglement part - linear entanglement
    for i in range(nqubits-1):
        qml.CNOT(wires = [i, i+1])
    for i in range(nqubits):
        qml.RY(theta[i+2*nqubits], wires = i)
        qml.RZ(theta[i+3*nqubits], wires = i)
```

```
dev = qml.device('lightning.qubit', wires=nqubits)
@qml.qnode(dev)
def energy(param):
    efficientSU2(param)
    return qml.expval(H)

from scipy.optimize import minimize
theta = np.array(np.random.random(4*nqubits),
                 requires_grad = True)
result = minimize(energy, x0 = theta)
print(f'Optimal parameters: \n{result.x}')
print(f'Energy: {result.fun}')
```

# Variational Quantum Eigensolver

Code!!

Molécule  $H_2$

```
from scipy.optimize import minimize
theta = np.array(np.random.random(4*nqubits),
                 requires_grad = True)
result = minimize(energy, x0 = theta)
print(f'Optimal parameters: \n{result.x}')
print(f'Energy: {result.fun}')
```

Optimal parameters:

[ -2.25577888e-01 -7.30356359e-06 3.14159756e+00 8.06725274e-05  
-8.81755671e-02 -2.76576671e-01 -2.20170721e-02 -4.83456259e-01  
3.14158937e+00 3.14158997e+00 -3.14158301e+00 3.14168535e+00  
1.46600136e-01 5.14321684e-01 8.67669767e-01 -1.18579256e-01]

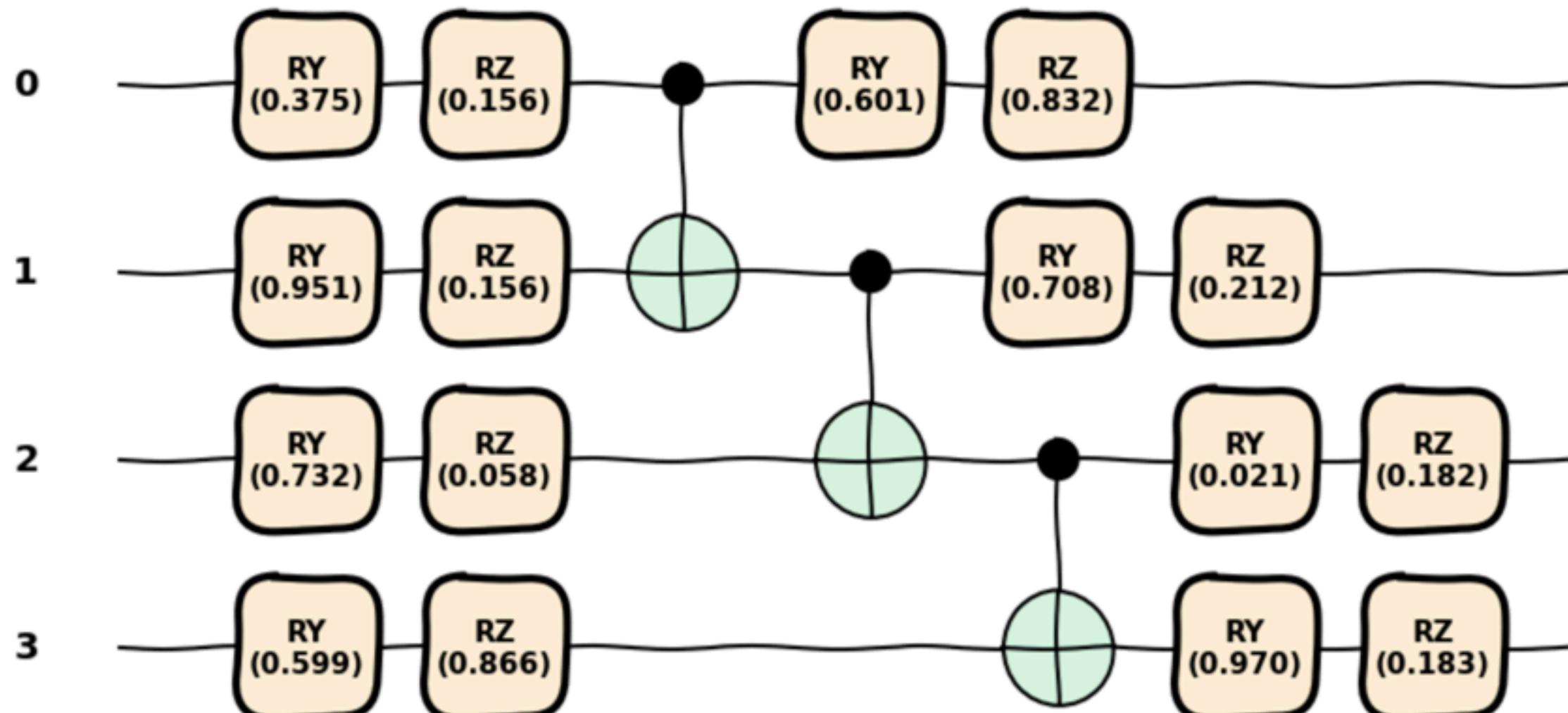
Energy: -1.1372838346118395

# variational quantum Eigensolver

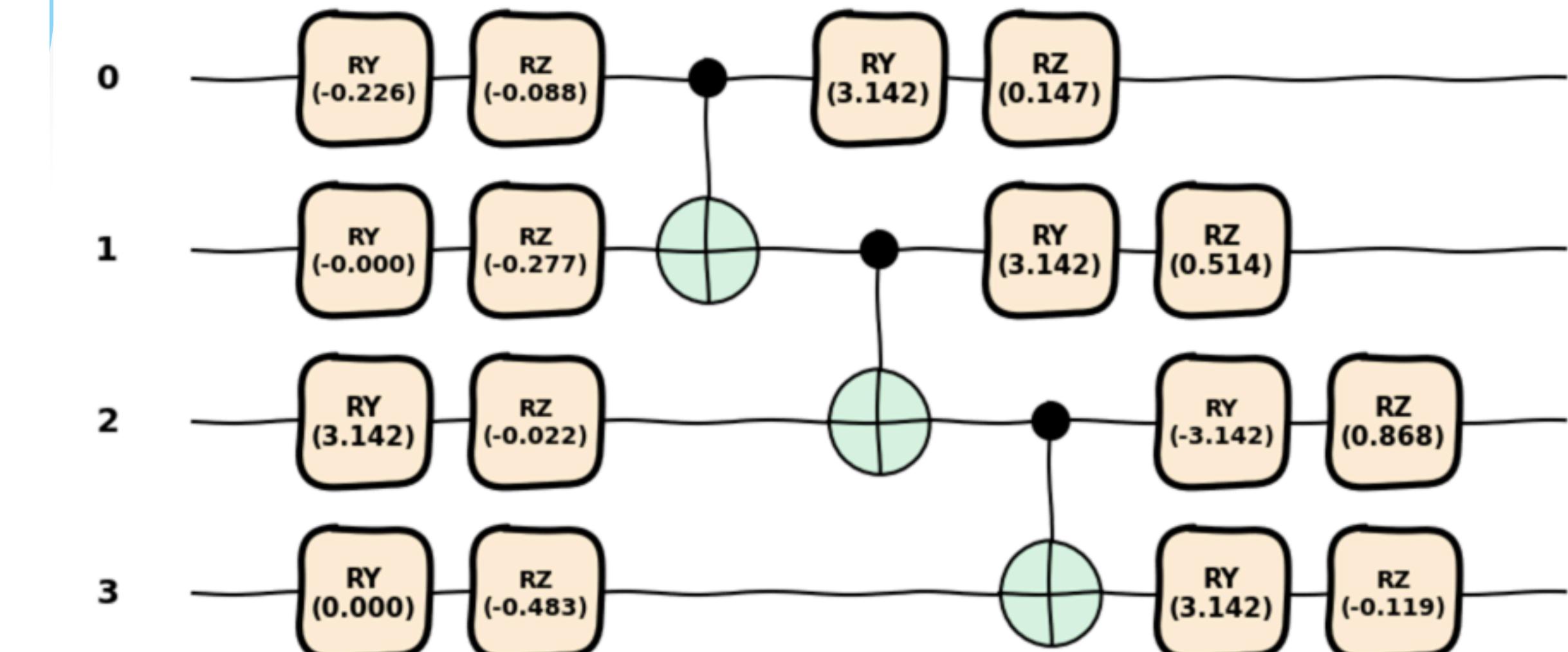
code!!

Molécule  $H_2$

Initialisation



Paramètres Optimaux



Énergie: -1.1372838346118395

# Variational Quantum Classifier

Difference avec un VQE?

VQE



Estimation de l'eigenvalue la plus basse  
d'un Hamiltonien

VQC



Mesure de la valeur attendue en Z, extraction  
du signe pour déterminer la classe via la  
projection sur un qubit - map Bloch Sphere

Fonction de cout associée au label comme la  
distance entre le Label et la donnée réelle

# Références

---

- [1] Combarro E. F. & González-Castillo S., 2023, A Practical Guide to Quantum Machine Learning and Quantum Optimization, Chapitres 3, 5
- [2] <https://quantumcomputing.stackexchange.com/questions/17555/whats-the-role-of-mixer-in-qaoa>
- [3] Zhao & Juan, Exploiting anticommutation in Hamiltonian simulation, Quantum 5, 534 (2021)
- [4] He, et al., Alignment between initial state and mixer improves QAOA performance for constrained optimization, npj Quantum Inf 9, 121 (2023). <https://doi.org/10.1038/s41534-023-00787-5>
- [5] Blekots et al., 2023, A Review on Quantum Approximate Optimization Algorithm and its variants, <https://arxiv.org/pdf/2306.09198.pdf>
- [6] Michał Stęchły, Variational Quantum Algorithms – How do they work?, Musty Thoughts, 2020
- [7] Schulz. M & Petruccione F., 2021, Machine Learning with Quantum Computers, Chap 5
- [8] Michał Stęchły, VQAs - challenges and state of research, Musty Thoughts, 2021