Outils de programmation quantique avec Qiskit

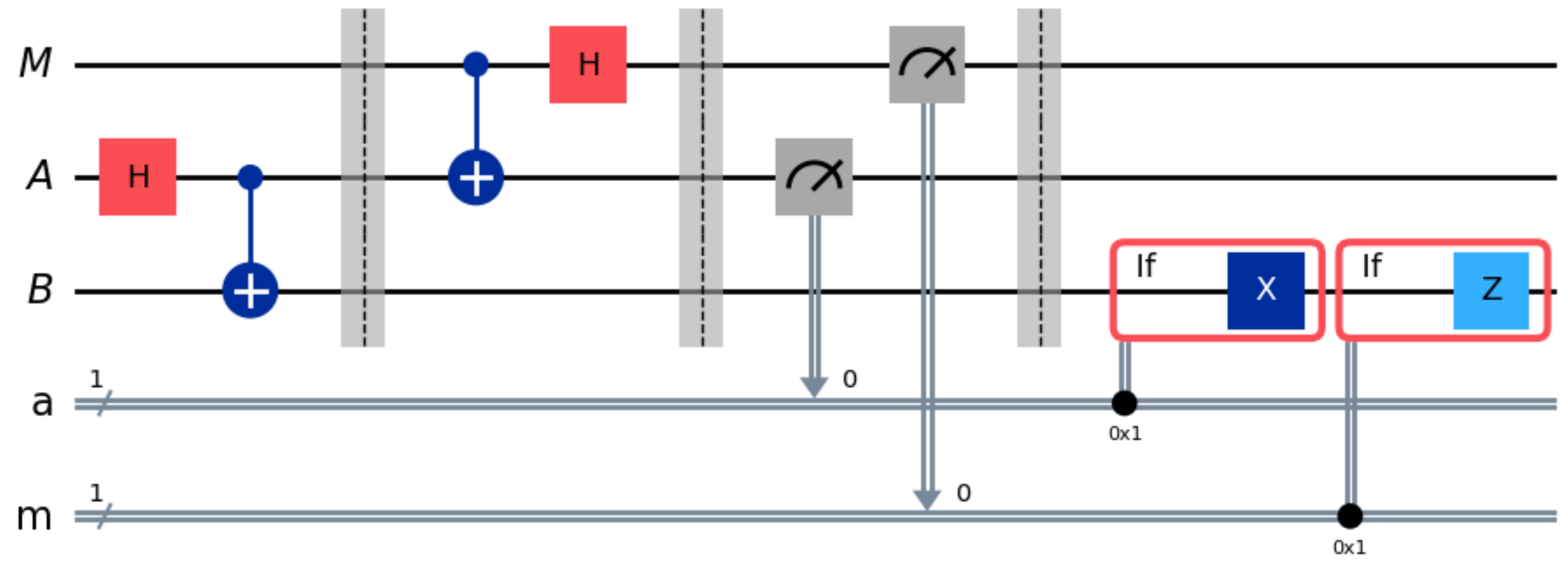
Jean Frederic Laprade
6 mai 2025
jean-frederic.laprade@usherbrooke.ca

UNIVERSITÉ DE SHERBROOKE

Introduction

Récapitulatif

- Un état quantique à N qubits est décrit par un vecteur unitaire complexe dans un espace de 2^N dimensions, $|\psi\rangle = \sum_i \alpha_i |i\rangle$.
- Les opérations sur les qubits sont représentées par des matrices unitaires.
- Un algorithme quantique est une séquence d'opérations sur les qubits et s'écrit sous la forme d'un circuit quantique.



Introduction

Objectifs

Quand on veut exécuter des circuits sur un processeur quantique, on doit garder en tête que

1. la machine n'est pas sur le coin du bureau, **il faut y accéder** ;

2. on ne peut pas faire abstraction de **l'architecture de la machine** - types de qubits, connectivité, opérations natives ;

3. les **calculs sont bruités**, il faut user de stratégies pour obtenir un signal utile.

L'objectif de cet atelier est de vous familiariser avec les outils Qiskit pour exécuter des circuits quantiques sur les ordinateurs d'IBM.

Introduction

Plan

1. Observables et mesure
2. Primitives Qiskit
3. Environnement IBM Quantum
4. Soumettre des tâches

Observables et mesure

La mesure

Règle Born

Pour un état $|\psi\rangle = \sum_{i=0}^{2^N-1} \alpha_i |i\rangle,$

la probabilité de mesurer l'état $|i\rangle$ est $|\alpha_i|^2$.

La mesure

Règle Born

Pour un état $|\psi\rangle = \sum_{i=0}^{2^N-1} \alpha_i |i\rangle,$

la probabilité de mesurer l'état $|i\rangle$ est $|\alpha_i|^2$.

Questions

1. Quels sont les mesures possibles et avec quelles probabilités pour l'état

$$|\psi\rangle = \sqrt{\frac{1}{2}}|00\rangle + \sqrt{\frac{1}{4}}|01\rangle + \sqrt{\frac{3}{20}}|10\rangle + \sqrt{\frac{1}{10}}|11\rangle$$

La mesure

Règle Born

Pour un état $|\psi\rangle = \sum_{i=0}^{2^N-1} \alpha_i |i\rangle,$

la probabilité de mesurer l'état $|i\rangle$ est $|\alpha_i|^2$.

Questions

1. Quels sont les mesures possibles et avec quelles probabilités pour l'état

$$|\psi\rangle = \sqrt{\frac{1}{2}}|00\rangle + \sqrt{\frac{1}{4}}|01\rangle + \sqrt{\frac{3}{20}}|10\rangle + \sqrt{\frac{1}{10}}|11\rangle$$

2. Pouvez-vous imaginer le circuit qui prépare cet état ?

La mesure

Circuit

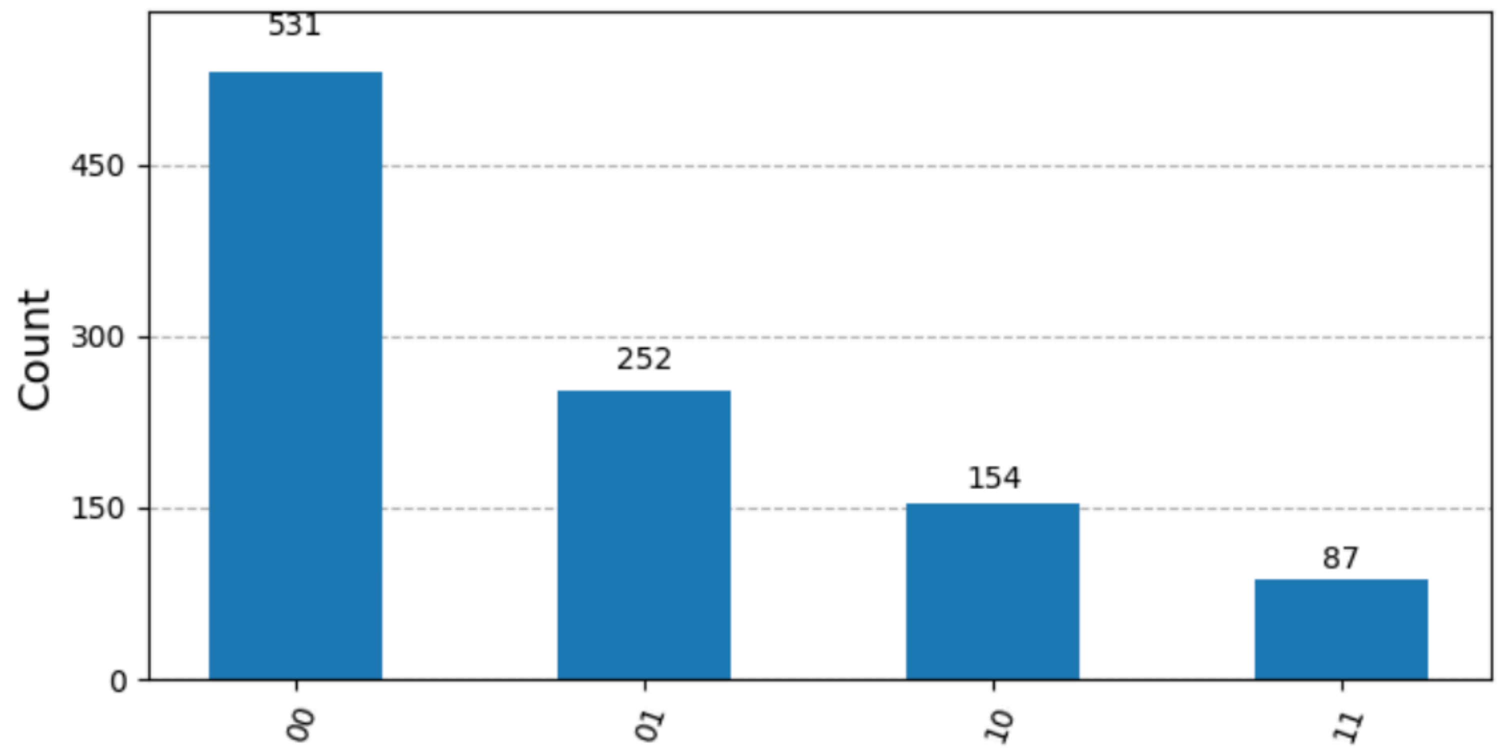
$$|\psi\rangle = \sqrt{\frac{1}{2}}|00\rangle + \sqrt{\frac{1}{4}}|01\rangle + \sqrt{\frac{3}{20}}|10\rangle + \sqrt{\frac{1}{10}}|11\rangle$$

```python
qc = QuantumCircuit(2)
qc.ry(2 * np.arccos(np.sqrt(0.65)), 0)
qc.x(0)
qc.cry(2 * np.arccos(np.sqrt(0.50/0.65)), 0, 1)
qc.x(0)
qc.cry(2 * np.arccos(np.sqrt(0.25/0.35)), 0, 1)
qc.measure_all()

simulator = AerSimulator()

result = simulator.run(qc).result()
plot_histogram(result.get_counts(qc), figsize=(8,4))
```

✓ 0.0s Python



La mesure

Observable

- Le seul moyen d'extraire de l'information d'un système quantique est d'effectuer une mesure.

La mesure

Observable

- Le seul moyen d'extraire de l'information d'un système quantique est d'effectuer une mesure.

- Un **observable** M est une quantité qui peut être mesurée. Par exemple, l'énergie ou la magnétisation.

La mesure

Observable

- Le seul moyen d'extraire de l'information d'un système quantique est d'effectuer une mesure.
- Un **observable** M est une quantité qui peut être mesurée. Par exemple, l'énergie ou la magnétisation.
- Mathématiquement, un **observable** M est une matrice hermitienne sur l'espace d'états des qubits, c'est-à-dire que $M = (M*)^T \equiv M^\dagger$.

La mesure

Observable

- Le seul moyen d'extraire de l'information d'un système quantique est d'effectuer une mesure.

- Un **observable** M est une quantité qui peut être mesurée. Par exemple, l'énergie ou la magnétisation.

- Mathématiquement, un **observable** M est une matrice hermitienne sur l'espace d'états des qubits, c'est-à-dire que $M = (M^*)^T \equiv M^\dagger$.

- Une matrice hermitienne est diagonalisable, c-à-d qu'il existe une base orthonormale $\{\,|m\rangle\}$ telle que $M = \sum_m \lambda_m |m\rangle\langle m|$ où λ_m est la valeur propre associée au vecteur propre $|m\rangle$. $P_m \doteq |m\rangle\langle m|$ est appelé un **projecteur**.

La mesure

Matrices de Pauli

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Les matrices de Pauli sont des opérateurs…

- hermitiens (*donc des observables!*) : $X = X^\dagger$, $Y = Y^\dagger$ et $Z = Z^\dagger$
- unitaires (*donc des portes quantiques!*) : $XX^\dagger = YY^\dagger = ZZ^\dagger = I$

La mesure

Pauli Z

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

Vecteur propre	Valeur propre
$\|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\lambda_0 = +1$
$\|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\lambda_1 = -1$

La mesure

Pauli Z

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

Diagonalisation

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Q^{-1} X Q$$

Vecteur propre	Valeur propre
$\|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\lambda_0 = +1$
$\|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\lambda_1 = -1$

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

$$\begin{aligned} Z &= (+1)|0\rangle\langle 0| + (-1)|1\rangle\langle 1| \\ &= (+1)P_0 + (-1)P_1 \end{aligned}$$

La mesure

Pauli X

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

Vecteur propre | Valeur propre

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \lambda_+ = +1$$

$$|-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad \lambda_- = -1$$

La mesure

Pauli X

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

Diagonalisation

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Q^{-1} X Q$$

Vecteur propre

Valeur propre

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\lambda_+ = +1$$

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

$$|-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\lambda_- = -1$$

$$\begin{aligned} X &= (+1)|+\rangle\langle+| + (-1)|-\rangle\langle-| \\ &= (+1)P_+ + (-1)P_- \end{aligned}$$

La mesure

Pauli Y

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Vecteur propre | Valeur propre

$$|+i\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ i \end{bmatrix} \qquad \lambda_{+i} = +1$$

$$|-i\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -i \end{bmatrix} \qquad \lambda_{-i} = -1$$

La mesure
Pauli Y

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Vecteur propre

Valeur propre

$$|+i\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}$$

$$\lambda_{+i} = +1$$

$$|-i\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}$$

$$\lambda_{-i} = -1$$

Diagonalisation

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Q^{-1} Y Q$$

$$Q = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}$$

$$\begin{aligned} Y &= (+1)|+i\rangle \langle +i| + (-1)|-i\rangle \langle -i| \\ &= (+1)P_{+i} + (-1)P_{-i} \end{aligned}$$

La mesure

Mesurer des observables

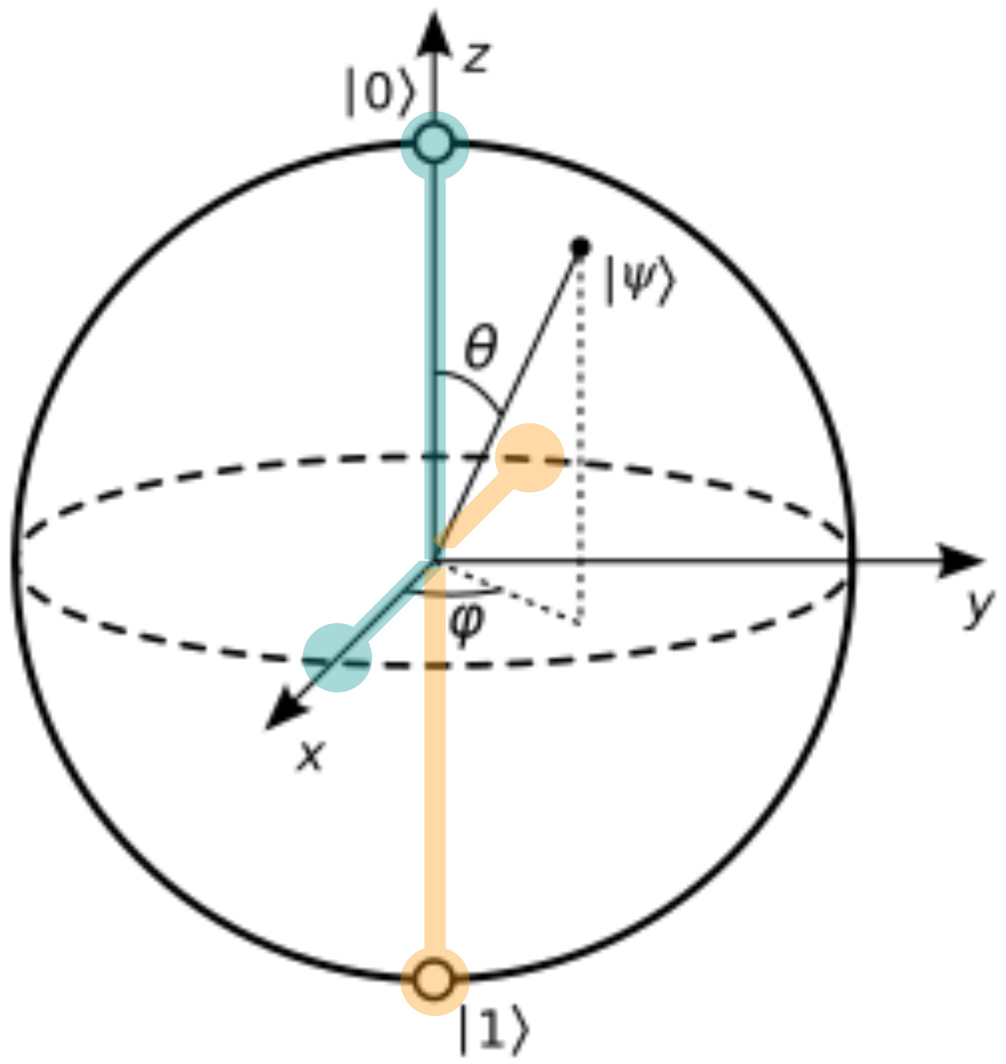
- Les vecteurs propres de Z correspondent à la base computationnelle.
  On peut donc mesurer Z directement !

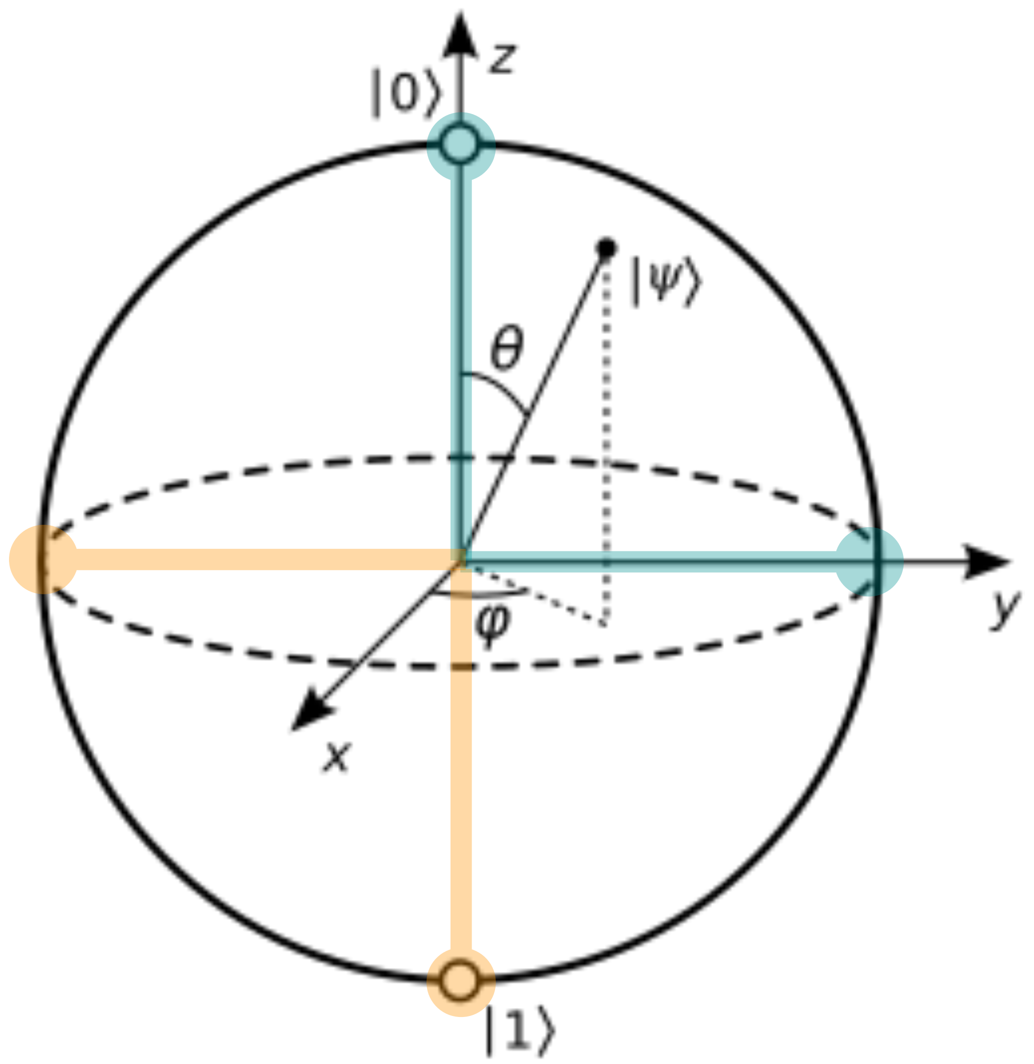
La mesure

Mesurer des observables

- Les vecteurs propres de Z correspondent à la base computationnelle. On peut donc mesurer Z directement !

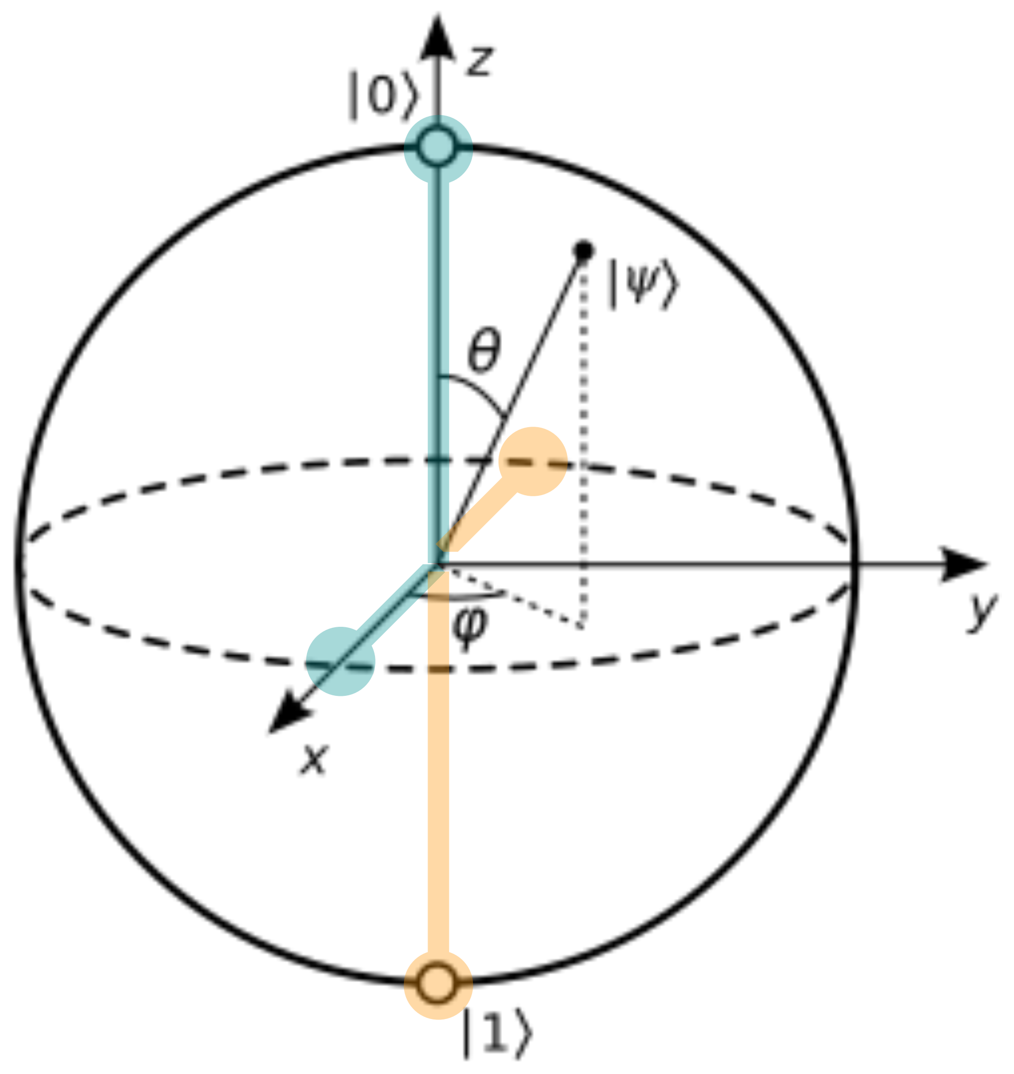
- Comment mesurer X et Y ?





La mesure

Diagonaliser X



Diagonalisation

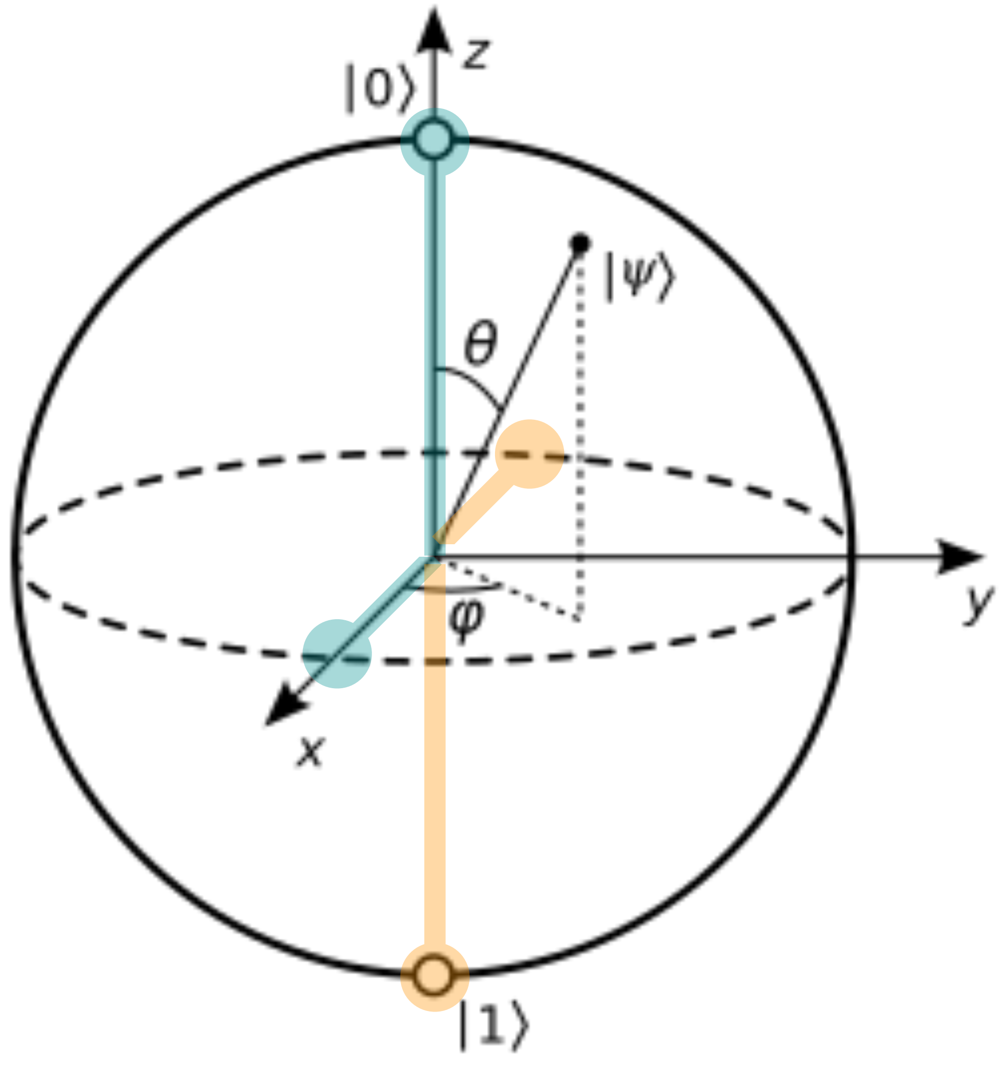
$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Q^{-1} X Q$$

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

Quelle porte quantique permet de diagonaliser l'observable X ?

La mesure

Diagonaliser X

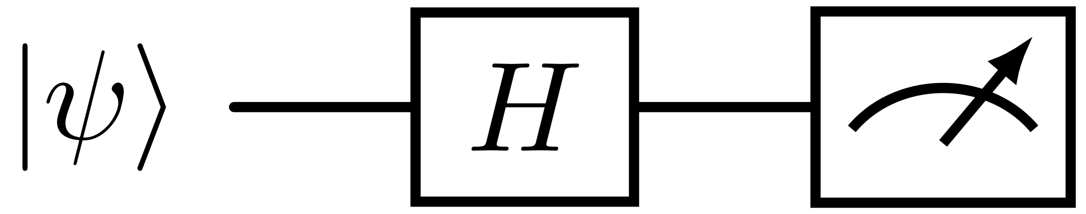


Diagonalisation

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Q^{-1} X Q$$

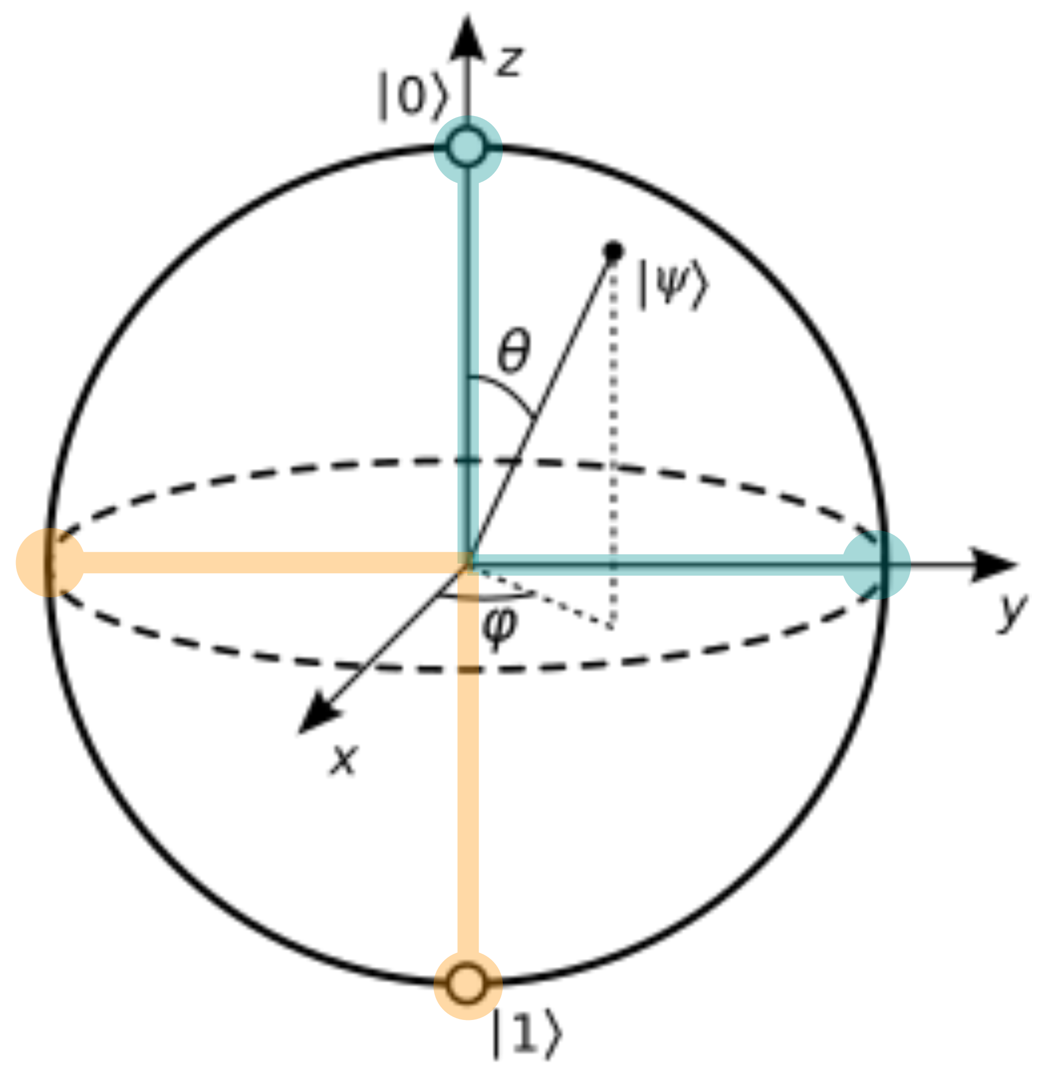
$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Quelle porte quantique permet de diagonaliser l'observable X ?



La mesure

Diagonaliser Y



Diagonalisation

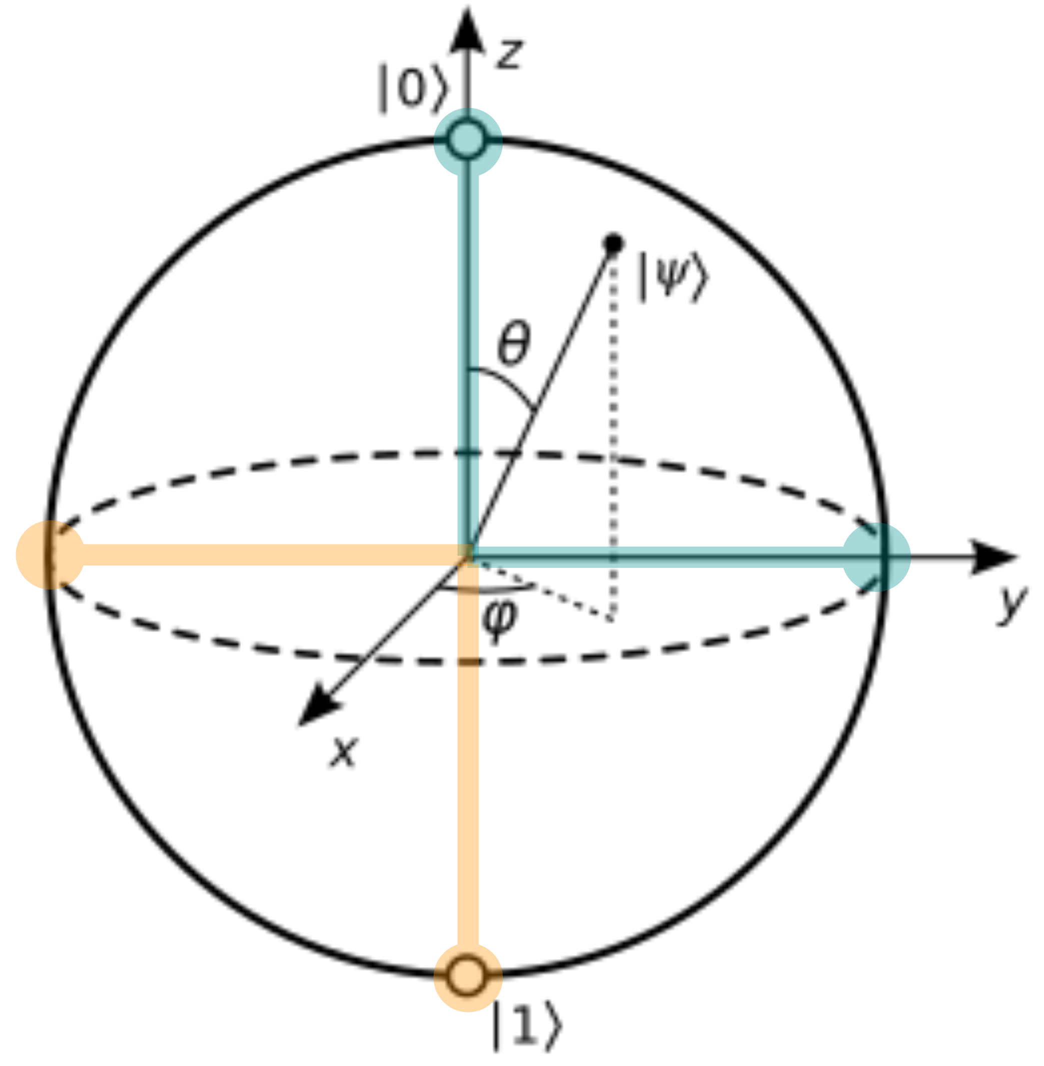
$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Q^{-1} Y Q$$

$$Q = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}$$

Quelle porte quantique permet de diagonaliser l'observable Y ?

La mesure

Diagonaliser Y

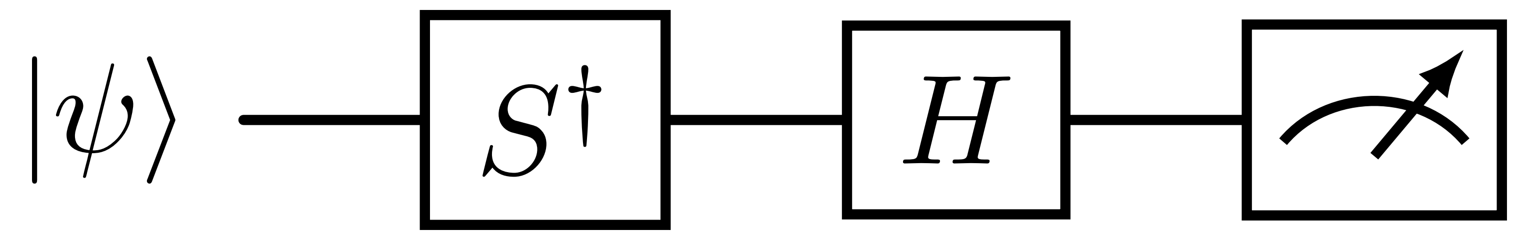


Diagonalisation

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Q^{-1} Y Q$$

$$Q = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}$$

Quelle porte quantique permet de diagonaliser l'observable Y ?



où S est une porte qui effectue une rotation de $\pi/2$ autour de l'axe Z dans le sens anti-horaire.

La mesure

Base des matrices de Pauli

- Les matrices de Pauli, augmentées de la matrice identité I, sont une base pour les opérateurs hermitiens 2×2.
- Pour un observable $M \in \mathbb{C}^{2\times2}$, il existe $\alpha_i \in \mathbb{R}$ tels que

$$M = \alpha_0 I + \alpha_1 X + \alpha_2 Y + \alpha_3 Z$$

- Il suffit donc de savoir mesurer X, Y et Z pour évaluer n'importe quel observable !

La mesure

Les chaînes de Pauli

- Quand on a un système de qubits, on représente les bases de mesure sur chaque qubit à l'aide d'une chaîne de caractères. Par exemple

$$I_3 Z_2 Y_1 X_0$$

- Lorsque le contexte fait en sorte qu'il n'y a pas de confusion possible, on peut laisser tomber les indices identifiant les qubits

$$IZYX$$

où on prend la convention de Qiskit dans laquelle le qubit de plus petit poids est à droite.

La mesure

Base des chaînes de Pauli

Les chaînes de Pauli constituent également une base pour les observables sur plusieurs qubits.

```python
from qiskit.quantum_info import SparsePauliOp

sum_paulis = SparsePauliOp(data=['IIZ', 'IZI', 'ZII', 'XYZ'],
                           coeffs=[0.1, 0.2, 0.3, 0.4])
sum_paulis
```

✓ 0.0s Python

```
SparsePauliOp(['IIZ', 'IZI', 'ZII', 'XYZ'],
              coeffs=[0.1+0.j, 0.2+0.j, 0.3+0.j, 0.4+0.j])
```

La classe SparsePauliOp permet de définir des observables par des sommes de chaînes de Pauli pondérées.

La mesure

Hamiltonien et somme de chaînes de Pauli

Modèle de Ising

$$H_{\text{Ising}} = \sum_{<i,j>} J_{ij} \hat{Z}_i \hat{Z}_j + \sum_{i=0}^{N-1} h_i \hat{X}_i$$

Modèle de Heisenberg

$$H_{\text{Heisenberg}} = \sum_{<i,j>} J_X \hat{X}_i \hat{X}_j + J_Y \hat{Y}_i \hat{Y}_j + J_Z \hat{Z}_i \hat{Z}_j$$

D’autres Hamiltoniens sont exprimés de façon plus compacte par des opérateurs qui ne sont pas les opérateurs de Pauli, p.e. molécules.

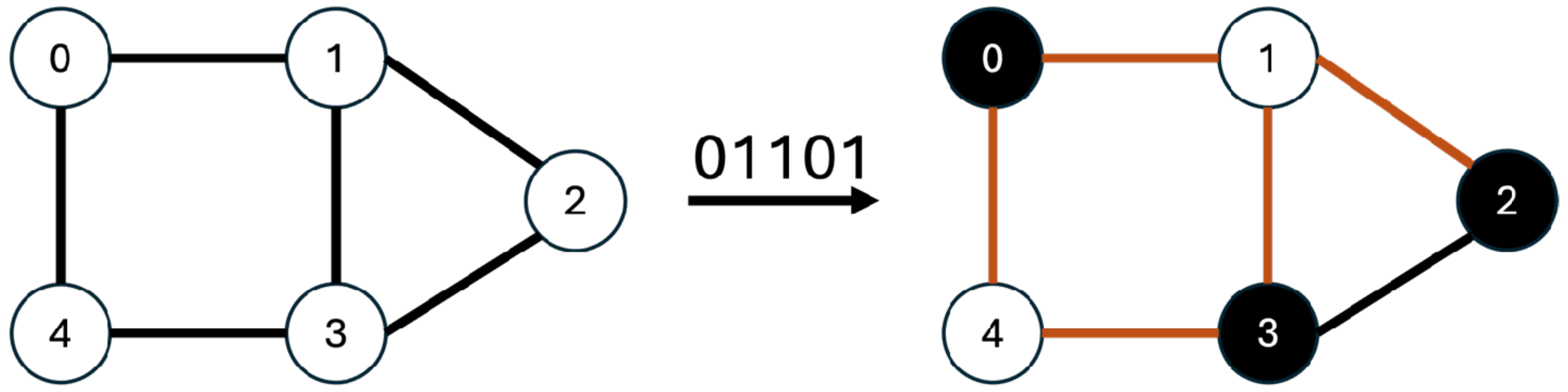
Il faut alors exprimer ces opérateurs sur la base de chaînes de Pauli avant de pouvoir faire des calculs sur l’ordinateur quantique.

La mesure

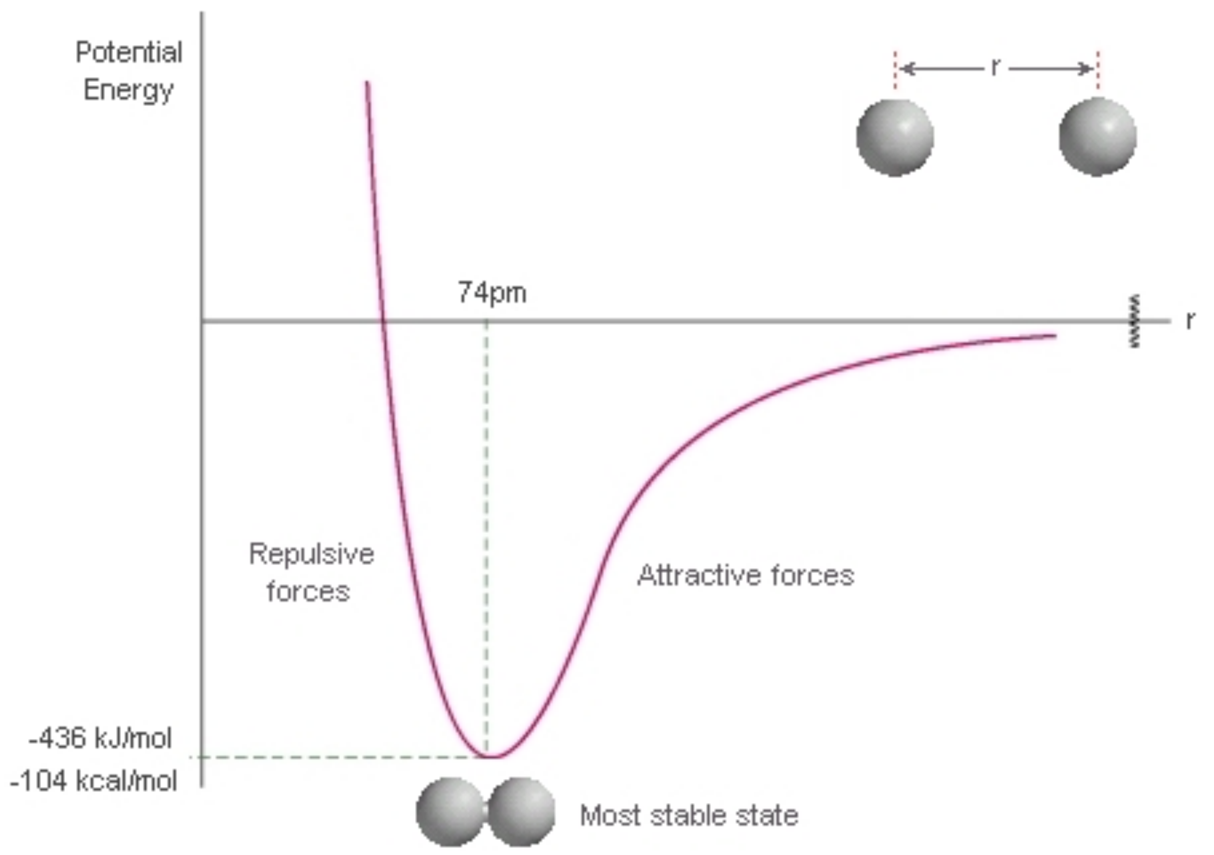
Pourquoi mesurer, quoi mesurer?

Deux tâches typiques sur un processeur quantique :

1. Mesurer dans la base computationnelle



2. Estimer une valeur moyenne, par exemple l'énergie d'une molécule



Source: Ian Hunt, Organic Chemistry etex

Qiskit offre des interface pour ces 2 cas de figure.

La mesure

Valeur moyenne d'un observable

$$\mathbb{E}_M = \sum_m \lambda_m \mathcal{P}(m)$$

La valeur moyenne d'une variable aléatoire M est la somme des valeurs possibles, pondérées par la probabilité de voir cette valeur se réaliser.

La mesure

Valeur moyenne d'un observable

$$\mathbb{E}_M = \sum_m \lambda_m \mathcal{P}(m)$$
$$= \sum_m \lambda_m \langle\psi| P_m |\psi\rangle$$

Ici $P_m = |m\rangle\langle m|$ est le projecteur et la probabilité de voir la mesure m se réaliser est donnée par $\langle\psi|P_m|\psi\rangle$.
Pour s'en convaincre, prenons l'état $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

$$\langle\psi|P_0|\psi\rangle = \langle\psi|(|0\rangle\langle 0|)|\psi\rangle = \alpha^*\alpha = \mathcal{P}(0)$$

$$\langle\psi|P_1|\psi\rangle = \langle\psi|(|1\rangle\langle 1|)|\psi\rangle = \beta^*\beta = \mathcal{P}(1)$$

La mesure

Valeur moyenne d'un observable

$$\begin{aligned}\mathbb{E}_M &= \sum_m \lambda_m \mathcal{P}(m) \\ &= \sum_m \lambda_m \langle\psi| P_m |\psi\rangle \\ &= \langle\psi| \left(\sum_m \lambda_m P_m\right) |\psi\rangle\end{aligned}$$

Par linéarité

La mesure

Valeur moyenne d'un observable

$$
\begin{aligned}
\mathbb{E}_M &= \sum_m \lambda_m \mathcal{P}(m) \\
&= \sum_m \lambda_m \langle\psi| P_m |\psi\rangle \\
&= \langle\psi| \left(\sum_m \lambda_m P_m \right) |\psi\rangle \\
&= \langle\psi| M |\psi\rangle
\end{aligned}
$$

Par définition

La mesure

Valeur moyenne d'un observable

$$
\begin{aligned}
\mathbb{E}_M &= \sum_m \lambda_m \mathcal{P}(m) \\
&= \sum_m \lambda_m \langle\psi| P_m |\psi\rangle \\
&= \langle\psi| \left(\sum_m \lambda_m P_m \right) |\psi\rangle \\
&= \langle\psi| M |\psi\rangle
\end{aligned}
$$

La valeur moyenne d'un observable est obtenu en prenant en "sandwich" l'opérateur hermitien entre le *bra* et le *ket* de l'état quantique.

Primitives Qiskit

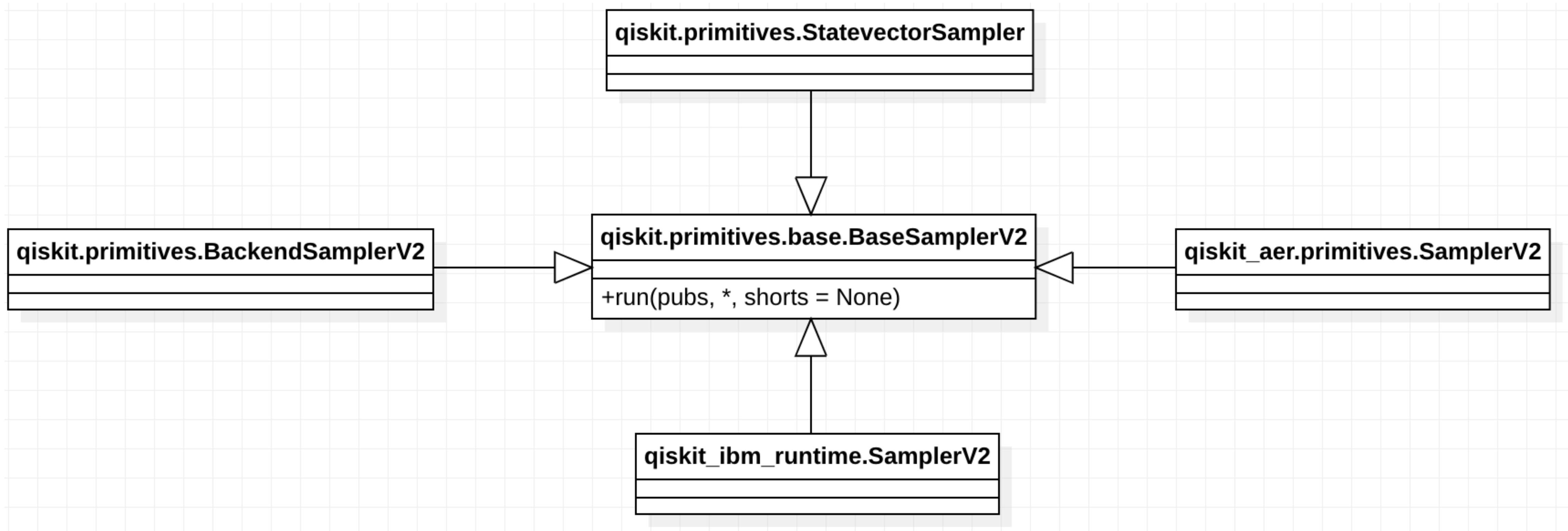
Primitives Qiskit

Définition

Les primitives Qiskit sont des modules de calcul qui permettent d'abstraire les tâches d'échantillonnage d'un état quantique et de calcul de valeur moyenne.

Deux types de primitives :

1. Sampler (BaseSamplerV2)



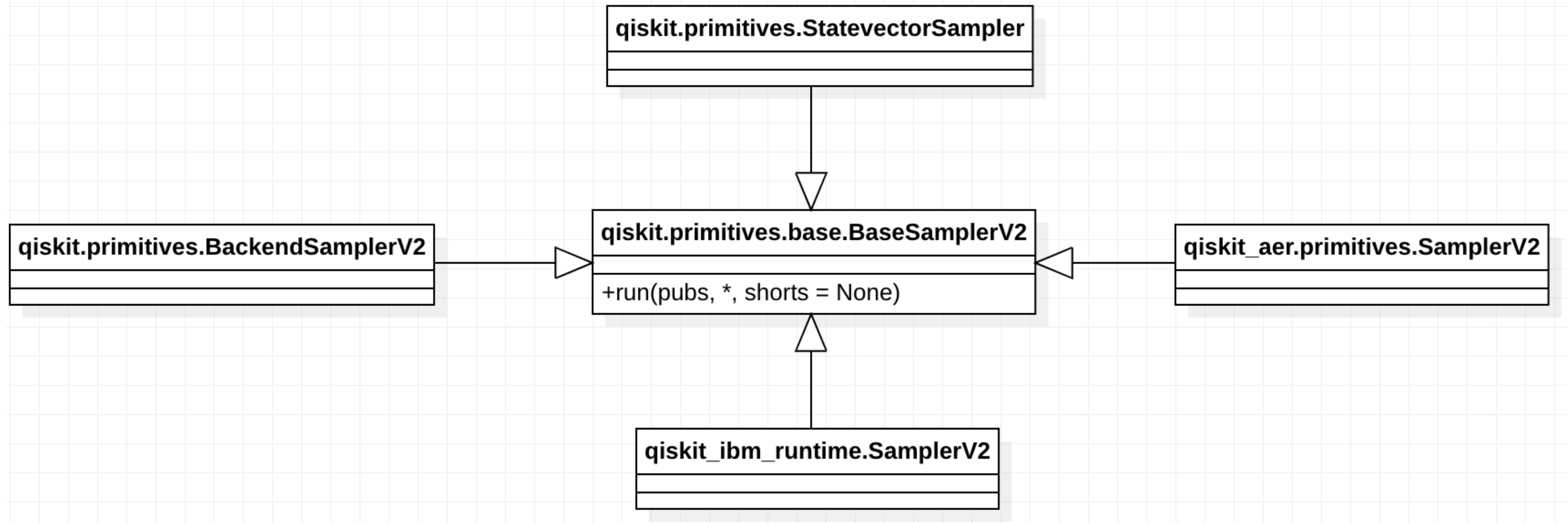
Primitives Qiskit

Définition

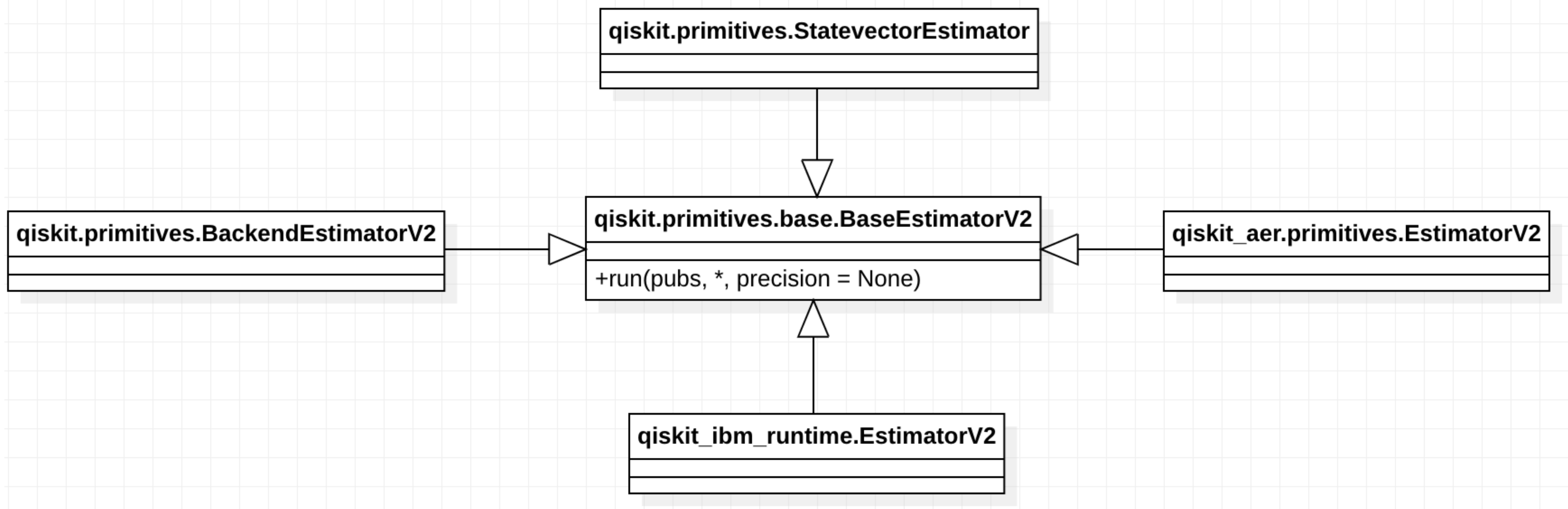
Les primitives Qiskit sont des modules de calcul qui permettent d’abstraire les tâches d’échantillonnage d’un état quantique et de calcul de valeur moyenne.

Deux types de primitives :

1. Sampler (BaseSamplerV2)



2. Estimator (BaseEstimatorV2)



Primitives Qiskit

Primitive unified blocs (*PUBs*)

- Le SamplerV2 et l'EstimatorV2 prennent en entrée un ou plusieurs *PUB**.
- Un *PUB* est un tuple qui contient un circuit et les données diffusées à ce circuit, qui peuvent être des observables et des paramètres multiples. Chaque *PUB* renvoie un résultat.

SamplerV2

Format d'un PUB (*<circuit>*,
*<valeurs des paramètres>**,
*<shots>**)

EstimatorV2

Format d'un PUB (*<circuit>*,
<observables>,
*<valeurs des paramètres>**,
*<shots>**)

* primitive unified blocs

Primitives Qiskit

Sampler

```
from qiskit.primitives import StatevectorSampler
```

Implémentation de BaseSamplerV2 en utilisant une simulation complète du vecteur d'état.

```
from qiskit.primitives import BackendSamplerV2
```

Implémentation de BaseSamplerV2 qui encapsule un objet BackendV2.

L’objet BackendV2 peut être une référence à

- ~~un processeur quantique~~,
- un FakeBackend
  (voir *qiskit_ibm_runtime.fake_provider.backends*)

```
from qiskit_aer.primitives import SamplerV2
```

Implémentation de BaseSamplerV2 avec une méthode from_backend().

```
from qiskit_ibm_runtime import SamplerV2
```

Implémentation de BaseSamplerV2 optimisée pour l’environnement IBM Runtime.

Primitives Qiskit

Estimator

```
from qiskit.primitives import StatevectorEstimator
```

Implémentation de BaseEstimatorV2 en utilisant une simulation complète du vecteur d'état.

```
from qiskit.primitives import BackendEstimatorV2
```

Implémentation de BaseEstimatorV2 qui encapsule un objet BackendV2. Ne permet pas de faire de la mitigation d'erreurs.

L'objet BackendV2 peut être une référence à

- ~~un processeur quantique~~,
- un FakeBackend
  (voir *qiskit_ibm_runtime.fake_provider.backends*)

```
from qiskit_aer.primitives import EstimatorV2
```

Implémentation de BaseEstimatorV2 avec une méthode from_backend().

```
from qiskit_ibm_runtime import EstimatorV2
```

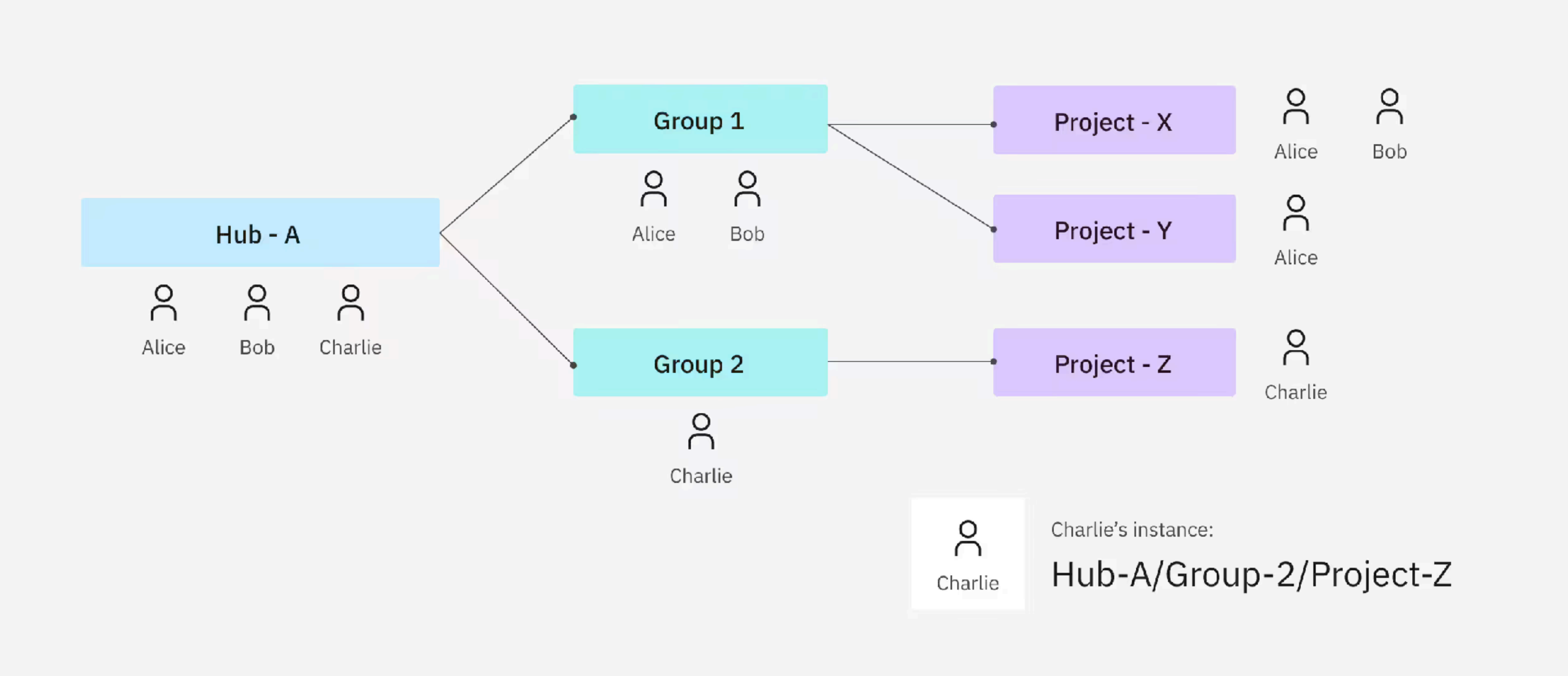
Implémentation de BaseEstimatorV2 optimisée pour l'environnement IBM Runtime.

Environnement IBM Quantum

Environnement IBM Quantum

Instance

- Structure hiérarchique: *hub/group/project*



- Exemples
  - *ibm-q/open/main*
  - *pinq-quebec-hub/ecole-dt/main*

Source: https://docs.quantum.ibm.com/guides/instances

Environnement IBM Quantum

Processeurs quantiques

ibm_sherbrooke

Details

Qubits	2Q error (best)	2Q error (layered)	CLOPS
127	2.94e-3	1.52e-2	150K

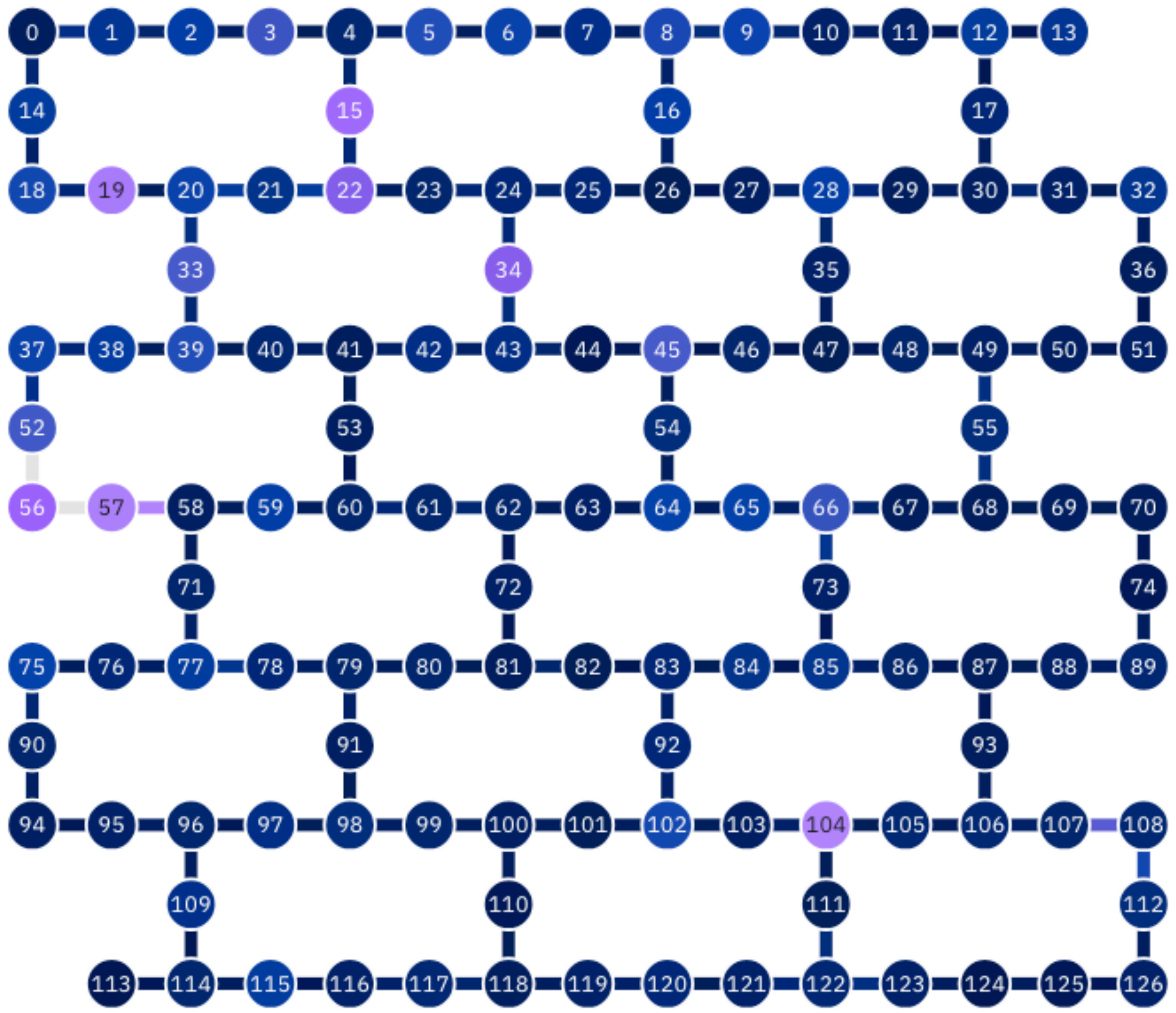
Status: Online	Region: us-east	Processor type: Eagle r3	Version: 1.6.88
Total pending workloads: 7 jobs	Your instance usage: 1 job	Basis gates: ECR, ID, RZ, SX, X	Median ECR error: 7.305e-3
Median SX error: 2.297e-4	Median readout error: 1.611e-2	Median T1: 276.69 us	Median T2: 174.7 us

Calibration data Last calibrated: 43 minutes ago

Map view | Graph view | Table view | Expand map view

Qubit: Readout assignment error — Median 1.611e-2 — min 2.441e-3 max 1.682e-1

Connection: ECR error — Median 7.305e-3 — min 2.936e-3 max 1.264e-1



Environnement IBM Quantum

Accès avec Qiskit

```python
from qiskit_ibm_runtime import QiskitRuntimeService

service = QiskitRuntimeService(channel='ibm_quantum',
                               instance="pinq-quebec-hub/ecole-dt/main")
backends = service.backends()
backends
```

✓ 13.3s Python

```
[<IBMBackend('ibm_quebec')>]
```

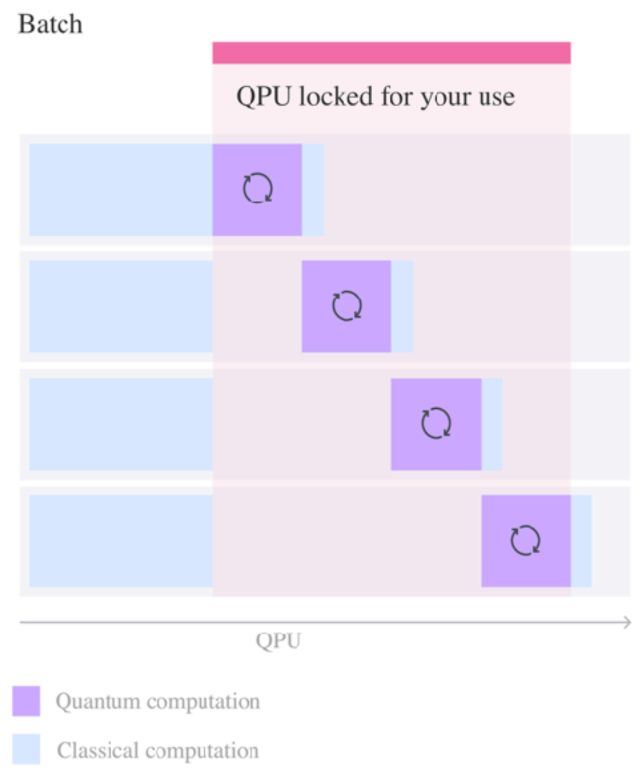
Environnement IBM Quantum

Modes d’exécution

Job

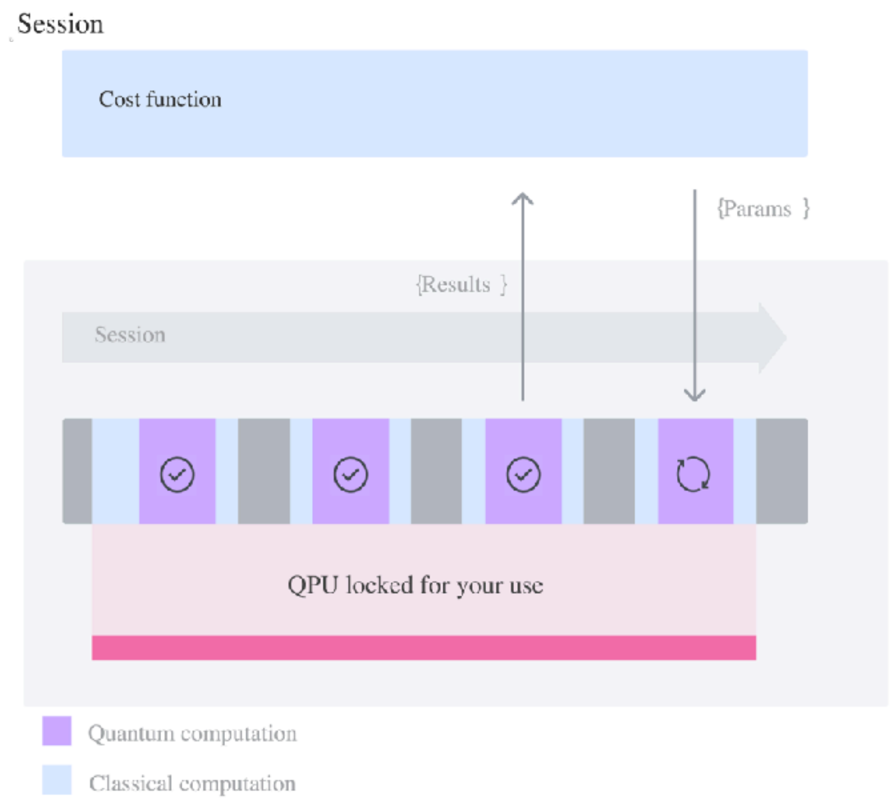
- Pas de contexte fourni à l’exécution des tâches

Batch



- Pré-traitement classique fait en parallèle
- Calcul quantique effectué en succession
- Pas d’ordonnancement des tâches garanti
- Pas d’accès exclusif au processeur
- Des calibrations du processeur peuvent survenir

Session



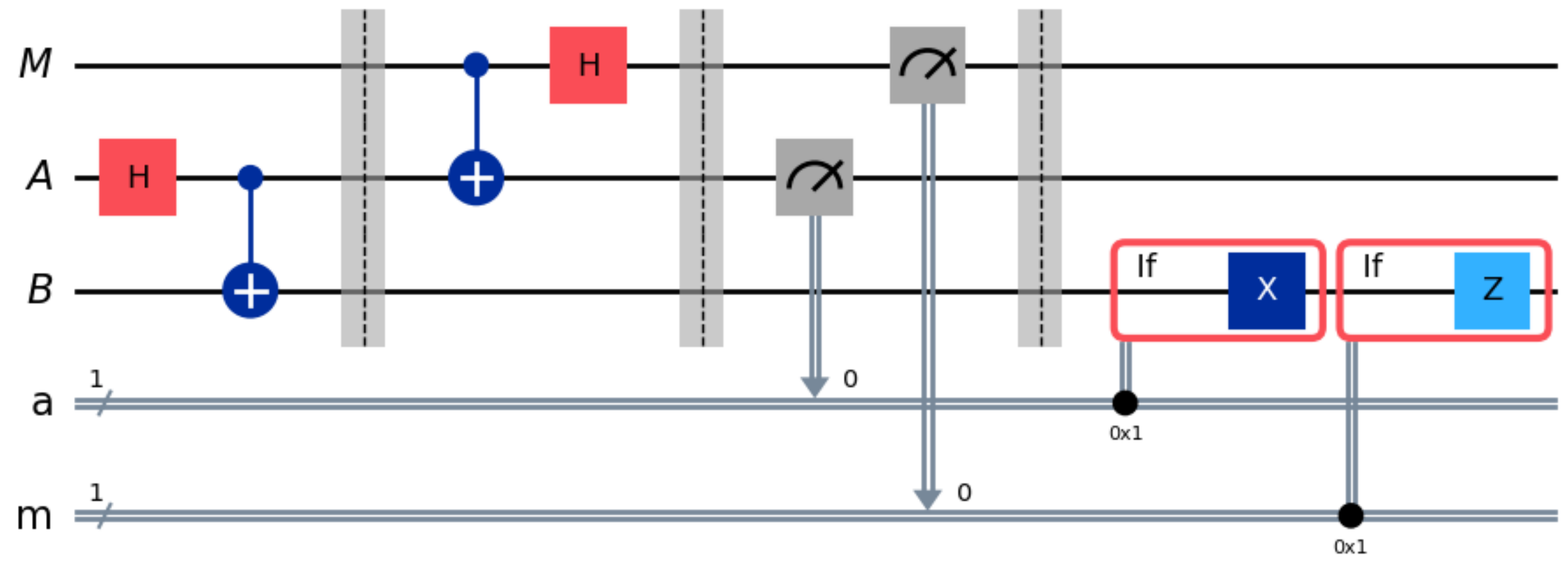
- Accès exclusif au processeur
- Une nécessité pour les algorithmes “variationnels”
- Permet la communication avec le client en cours d’exécution
- Pas disponible avec l’accès “Open”

Source: https://docs.quantum.ibm.com/guides/execution-modes

Soumettre des tâches

Soumettre des tâches

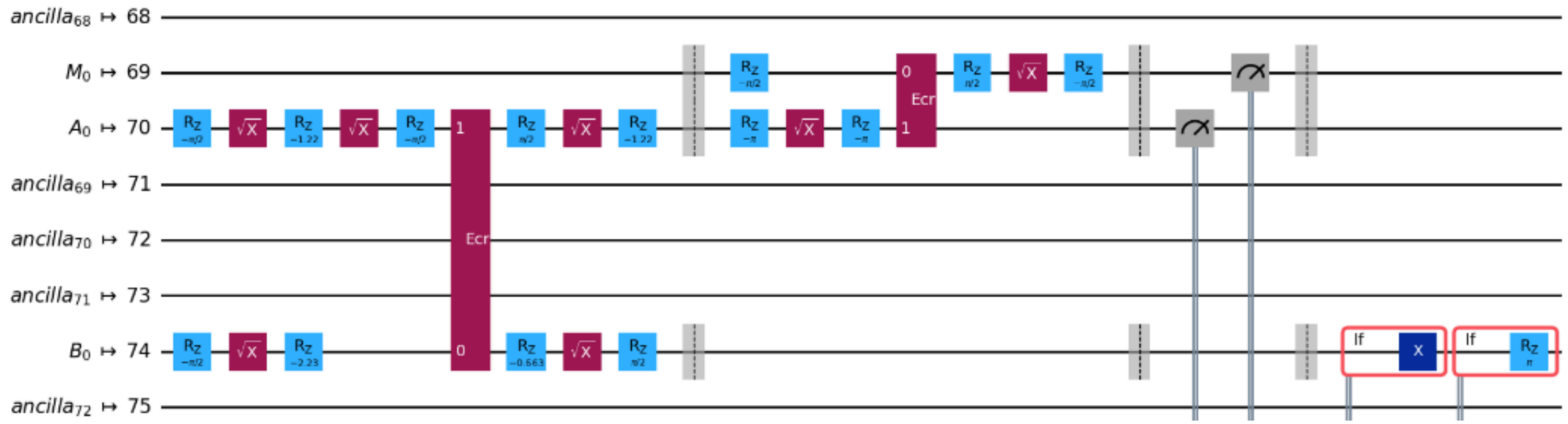
Transpilation des circuits




```python
from qiskit import transpile

teleportation_sherby = transpile(teleportation, sherby)
teleportation_sherby.draw()
```

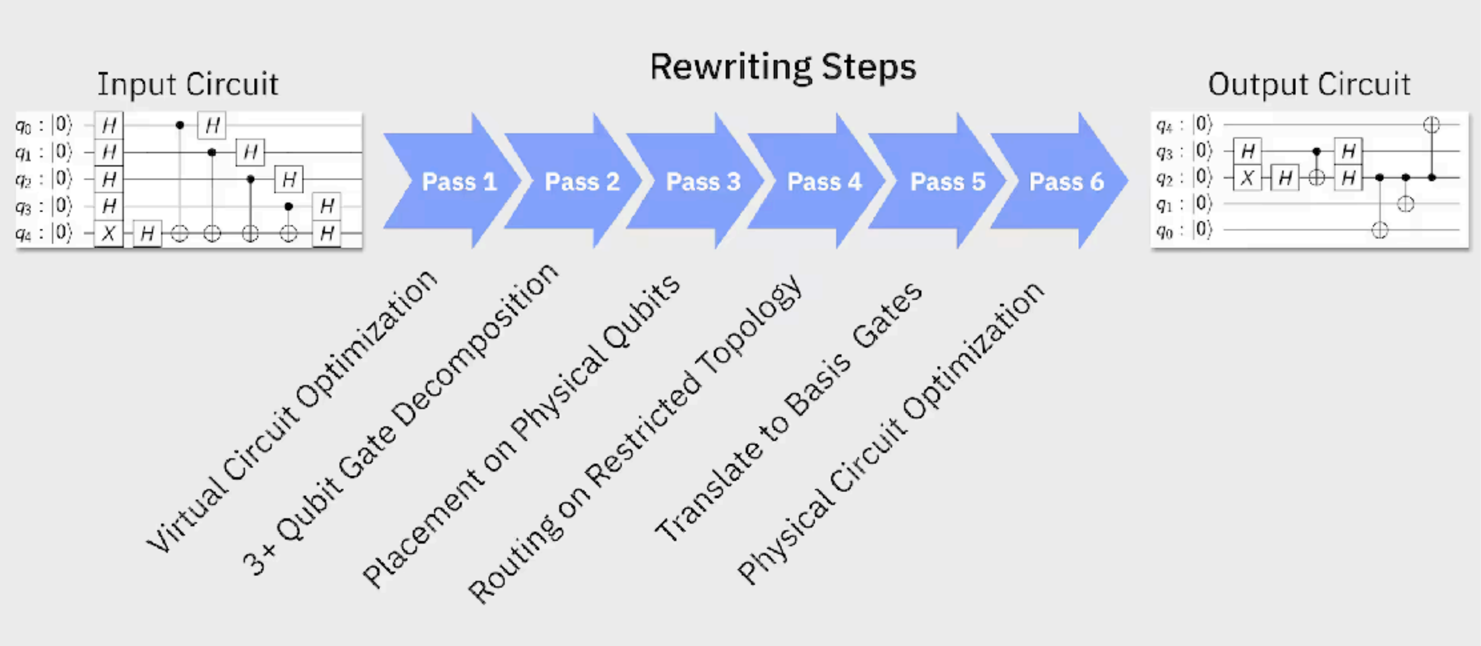
✓ 1.1s Python



La *transpilation* est le processus de **réécriture** d'un circuit pour qu'il corresponde à la **topologie** d'un dispositif quantique spécifique et/ou pour **optimiser** le circuit en vue de son exécution sur les systèmes quantiques **bruyants actuels**.

Soumettre des tâches

Transpilation des circuits et des observables




```python
from qiskit.transpiler import generate_preset_pass_manager

pm = generate_preset_pass_manager(optimization_level=1, backend=qc_qc)

qc = QuantumCircuit(2)
qc.h(0)
qc.cx(0, 1)

obs = ['XX', 'YY', 'ZZ']

pubs = []
isa_circuit = pm.run(qc)
isa_obs = [SparsePauliOp(o).apply_layout(isa_circuit.layout) for o in obs]
pubs.append((isa_circuit, isa_obs))
```

✓ 0.0s Python

La fonction transpile() compte 28 paramètres 😵‍💫

On peut utiliser un *PassManager* par défaut.
C’est aussi la méthode à privilégier lorsqu’on travaille avec des observables.

Source: https://docs.quantum.ibm.com/api/qiskit/transpiler

Des questions?

jean-frederic.laprade@usherbrooke.ca

Soumettre des tâches

Mitigation et suppression d'erreurs

Références

- https://docs.quantum.ibm.com/guides/error-mitigation-and-suppression-techniques
- https://docs.quantum.ibm.com/guides/configure-error-mitigation
- https://docs.quantum.ibm.com/guides/configure-error-suppression