

Introduction to IBM Quantum and Qiskit

Learning to run quantum circuits through domain-specific problem



Yuri Kobayashi

IBM Quantum

IBM Quantum

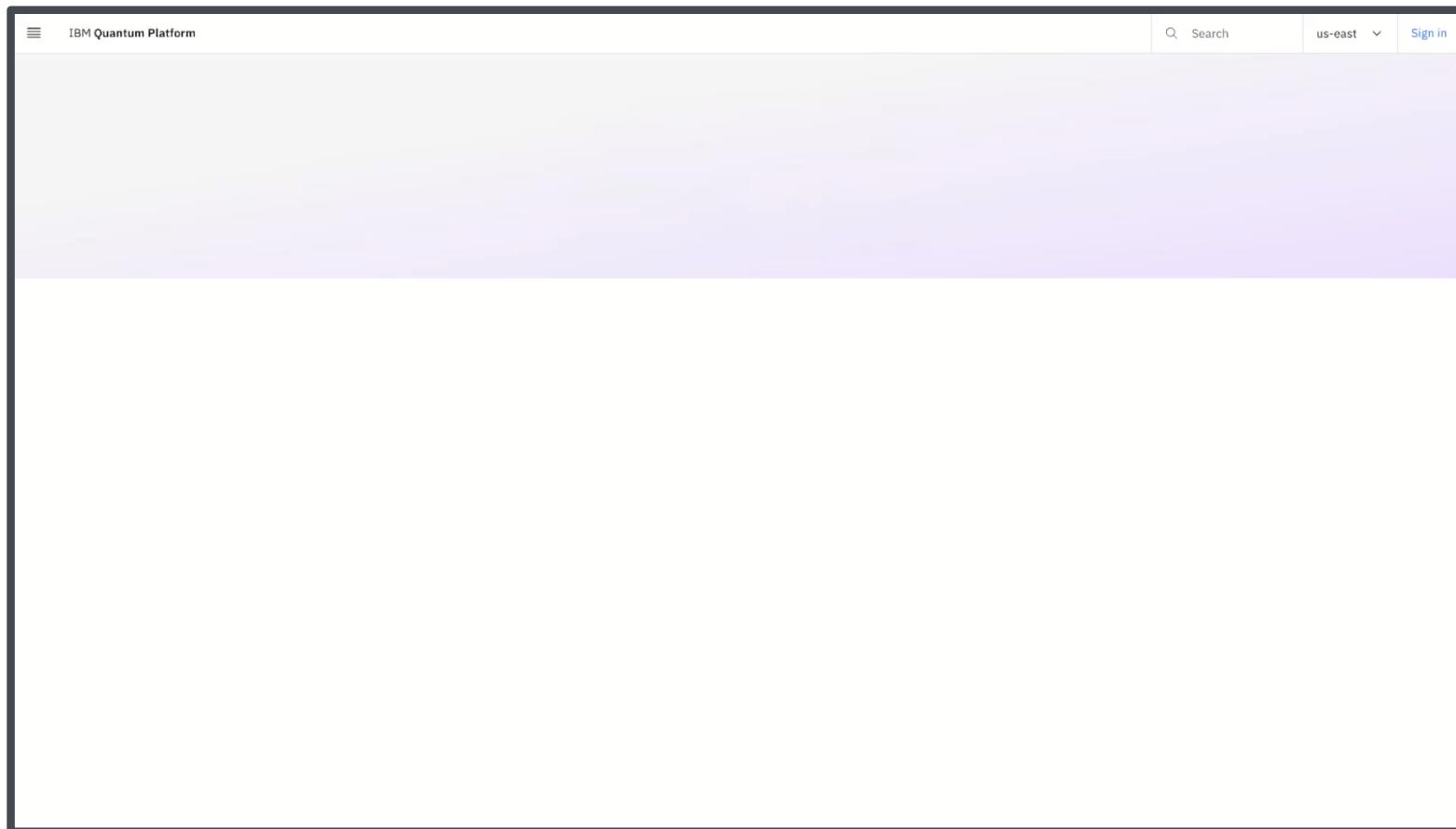
This session is meant for:

- Beginners of quantum computing
- Users who are new to IBM Quantum systems and Qiskit
- Looking to get started quickly by solving a practical domain specific problem.

Agenda

1. IBM Quantum Platform now on IBM Cloud (since July 2025)
2. Verifying your Account
3. Support Center (how to create cases to get issues resolved)
4. Navigating the platform
5. Qiskit – users' top choice quantum SDK
6. Qiskit Runtime Primitives
7. Qiskit Patterns – framework for running domain specific problems
8. Promising quantum computational areas
9. Installing Qiskit (traditional way vs fast way)
10. Getting up to speed with Qiskit (through a domain specific problem)
 - Taking Hamiltonian Simulation as an example
 - Qiskit coding tutorial: 1-D Transverse Ising Model
11. Summary

IBM Quantum Platform on IBM Cloud – July 2025



<https://quantum.cloud.ibm.com/>

Verify your account

You should have received an invitation to join an account in IBM Cloud.

Sender: IBM Cloud <no-reply@cloud.ibm.com>

Subject: Account: Action required: You are invited to join an account in IBM Cloud

Please check with QII Hub admin in case you have not received an invitation or if your join now link has expired.



Action required: You are invited to join an account in IBM Cloud

Source ID: [REDACTED]

Type: Account

Update Time: 9 Jul 2025, 4:58 PM UTC

[View notification \(in subscribed accounts\)](#):

Hi [REDACTED]

Fran Cabrera invited you to join the following IBM Cloud account:
[REDACTED]
(Account ID: [REDACTED])

[Join now.](#)

The link to join the account expires in 30 days.

Thank you,
IBM Cloud

In case of any issues, please contact [IBM Cloud support](#).

IBM Cloud Support

IBM Quantum

Dedicated Technical Support

To request assistance from the [IBM Cloud Support Center](#), please create a support case and select "Qiskit Runtime" in the topic field. A member of the IBM Quantum support team will respond to your inquiry.

The screenshot shows the IBM Cloud Support Center interface. At the top, there's a navigation bar with 'IBM Cloud' and a search bar. Below the navigation bar, the page title is 'サポート・センター / Case の作成'. The main area has tabs for 'カテゴリ' (Topic), 'トピック' (Topic), '詳細' (Details), and 'レビュー' (Review). A dropdown menu under 'トピック' is open, showing 'Qiskit Runtime' selected. A note says '直面している問題との関係性が最も高いトピックとサブトピックを選択します。' Below this, a section asks '問題に最も関連性のあるカテゴリーを教えてください' with four options: 'すべてのトピック', 'リソース', 'アカウント', and '請求および使用量'. A blue callout box highlights the 'Qiskit Runtime' selection with the text 'You can create your inquiry in English or Japanese (日本語 ok)'. The bottom right corner of the screenshot has the number '7'.

Sign in to IBM Quantum Platform

The screenshot shows the IBM Quantum Platform homepage at quantum.cloud.ibm.com. The top navigation bar includes a menu icon, the 'IBM Quantum Platform' logo, a search bar, a location dropdown set to 'us-east', and a 'Sign in' button. A blue arrow points from the text 'IBM id (the e-mail address you used to create your account.)' to the 'Sign in' button. The main content area features the platform's logo and tagline: 'Get access to IBM quantum computers, Qiskit documentation, and learning resources all in one place.' Below this is a search bar labeled 'Search for resources'. The 'Quantum information science' section is highlighted with a dark background and white text. Other sections include 'Optimization', 'Hamiltonian simulation', and 'Machine learning'. The 'Quantum information science' section contains a 'Basic' category with three items: 'Hello world' (Tutorial), 'Circuit library' (Docs), and 'Primitive inputs and outputs' (Docs). It also contains an 'Advanced' category with two items: 'Combine error mitigation options with the Estimator primitive' (Tutorial) and 'Utility-scale error mitigation with probabilistic error amplification' (Tutorial). At the bottom right, the page number '8' is visible.

IBM Quantum Platform

Get access to IBM quantum computers, Qiskit documentation, and learning resources all in one place.

Search for resources

IBM id (the e-mail address you used to create your account.)

quantum.cloud.ibm.com

Quantum information science

Conduct research with Qiskit at the level of quantum circuits and gates, leveraging tools for optimization, visualization, and execution to turn foundational ideas into experiments.

Learn

[Basics of quantum information course →](#)

Explore the research

[Disambiguating Pauli noise in quantum computers ↗](#)

[Order from chaos with adaptive circuits on quantum hardware ↗](#)

Basic

- [Hello world](#)
Tutorial • Run your first quantum circuit and view the results
- [Circuit library](#)
Docs • Explore ready-to-use quantum circuits and components
- [Primitive inputs and outputs](#)
Docs • Learn how to define inputs and interpret results for primitives

Advanced

- [Combine error mitigation options with the Estimator primitive](#)
Tutorial • Improve circuit performance with error mitigation techniques
- [Utility-scale error mitigation with probabilistic error amplification](#)
Tutorial • Apply probabilistic error amplification for better accuracy

8

Navigating the IBM Quantum Platform

(and successfully logged in)

Make sure you are selecting the correct Account

The screenshot shows the IBM Quantum Platform dashboard. At the top, there's a navigation bar with a three-bar menu icon, the text "IBM Quantum Platform", a search bar, a dropdown for "us-east", and a user profile icon. A callout arrow points to the account selection dropdown with the text "Make sure you are selecting the correct Account". Below the header, a purple banner says "YURI KOEYANEN". A callout arrow points to the three-bar menu icon with the text "Click on the three bar to see all menus". The main area has a blue header "My recent workloads" with columns: ID, Status, Instance, Created (sorted), QPU, and Usage. A large pink box contains the text "Your recent circuit executions (workloads) will be shown here.". To the left, there's a sidebar with "Instances" and a "View all" button. A callout arrow points to this sidebar with the text "You should see your instance here.". At the bottom, there's a section titled "QPUs" with a "View all" button and a count of "5". It lists five QPUs: "ibm_kingston" (156q | Heron r2), "ibm_pittsburgh" (156q | Heron r3), "ibm_fez" (156q | Heron r2), "ibm_marrakesh" (156q | Heron r2), and "ibm_torino" (133q | Heron r1). A callout arrow points to this section with the text "Available quantum systems will be shown here."

ID	Status	Instance	Created ↓	QPU	Usage

QPUs	View all
5	

IBM Quantum Systems

IBM Quantum

As of December 15, 2025, the following systems are currently available via IBM Cloud to QII Members in Japan who have premium access.

Quantum Devices	Qubits	Processor	Description	Chip image	Details
ibm_pittsburgh	156	Heron r3			
ibm_kingston	156	Heron r2	The Heron series can accurately execute up to 5,000 two-qubit gate operations, achieving nearly twice the scale of gate operations demonstrated in the 2023 utility-scale experiment using Eagle.		Details of each system such as calibration data, 2Q error, T1, T2, etc. are available in the IBM Quantum Platform
ibm_marrakesh	156	Heron r2	.		
ibm_fez	156	Heron r2	.		
ibm_torino	133	Heron r1			

* This list is as of Dec 15, 2025. The quantum systems available for premium users may change over time as new devices will be made ready and replace the old.

Navigating the platform

Make sure you are selecting the correct Account

The screenshot shows the IBM Quantum Platform interface. On the left, a vertical sidebar lists various menu items: Home, Instances, Workloads, Compute resources, Analytics, Access management, Functions (selected), Preview, Documentation, Tutorials, API references, Learning, and Composer. An arrow points to the three-bar icon at the top of this sidebar with the text "Click on the three bar to see all menus". At the top right, there is a search bar, a dropdown for "us-east", and a user profile icon. A callout bubble points to the account selection dropdown with the text "Make sure you are selecting the correct Account".

In the center, a purple header bar says "IBM Quantum Platform". Below it, a section titled "My recent workloads" displays a table with columns: ID, Status, Instance, Created (sorted by descending time), QPU, and Usage. A "View all" button is at the top right of this table. A large pink box contains the text "Your recent circuit executions (workloads) will be shown here."

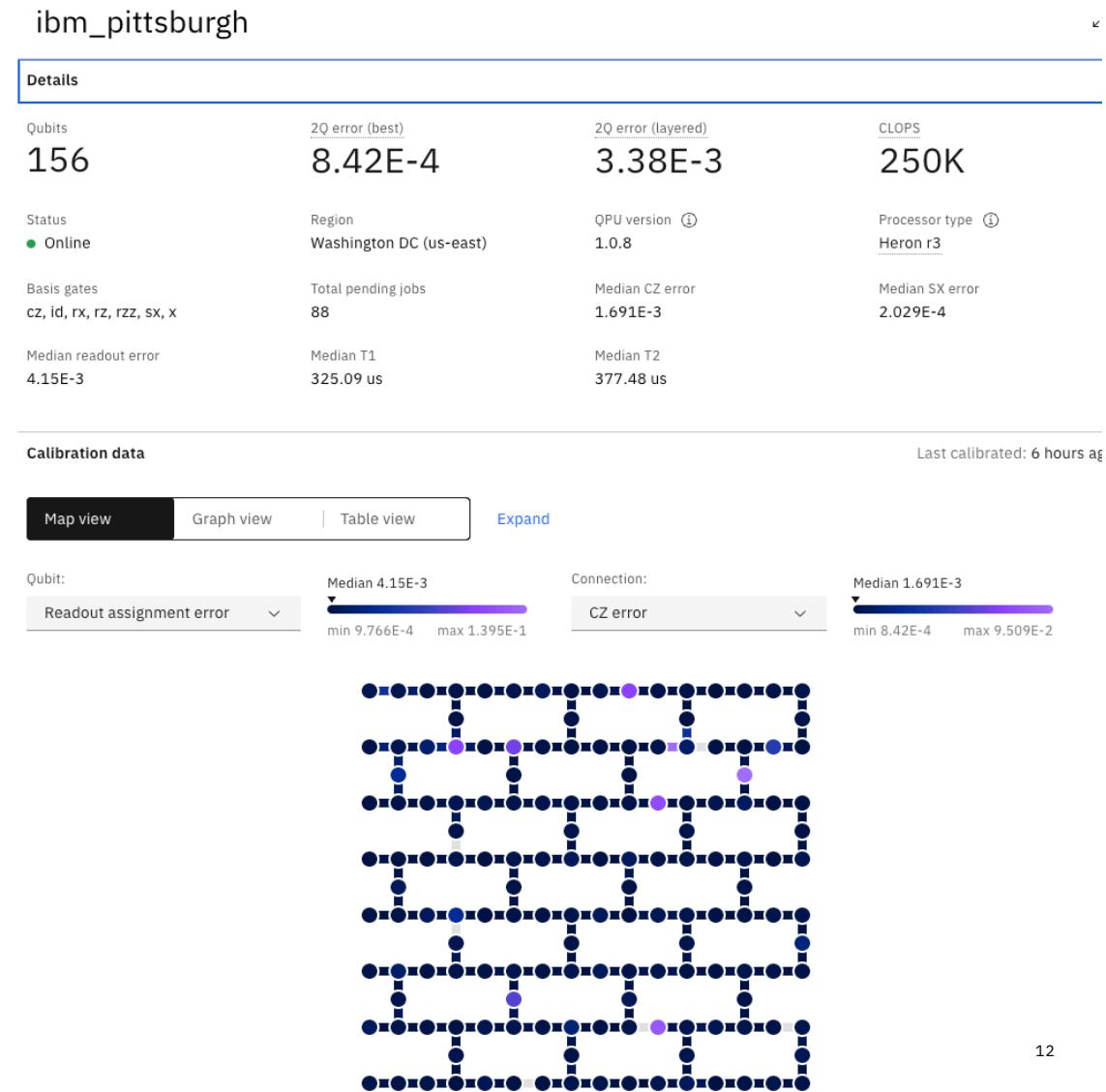
At the bottom, a section titled "QPUs" shows a table with six entries. The first entry has a blue border and contains the number "5". The table columns are: QPUs, View all, and five system entries: ibm_kingston (156q | Heron r2), ibm_pittsburgh (156q | Heron r3), ibm_fez (156q | Heron r2), ibm_marrakesh (156q | Heron r2), and ibm_torino (133q | Heron r1). A callout bubble points to the QPUs section with the text "Available quantum systems will be shown here."

Quantum system details

<https://quantum.cloud.ibm.com/computers>

For every accessible quantum computer, IBM Quantum discloses the qubit connectivity map, calibration data, error rates (2Q Error), and coherence times such as T1 and T2.

The latest Heron r3 processor is IBM's highest-performing quantum processor to date, with two-qubit error rates improved tenfold compared to the previous generation



Qiskit – most preferred and performant quantum SDK

The screenshot shows the Qiskit website homepage. At the top, there's a navigation bar with links for Quantum, Hardware, Qiskit (which is underlined), Product, Research, Blog, Community, Resources, and Sign in to Platform. Below the navigation is a banner with a purple button that says "Qiskit v2.2 is here →". The main content area features the Qiskit logo and a brief description: "Qiskit is the world's most popular and performant software stack for quantum computing and algorithms research. Build, optimize, and execute quantum workloads at scale." Below this is a "Get started" button and a "pip install qiskit" button. A large code editor window displays Python code for a molecular orbital calculation. A tooltip over the code says "Map classical inputs to a quantum problem". At the bottom, there are numbered links: 01 About, 02 Parts of Qiskit, 03 Applied tutorials, 04 Qiskit ecosystem, 05 Community resources, 06 Qiskit advocates, and 07 Latest updates.

Open-source quantum software
for large-scale execution

Speed

**129x faster in wall
clock time**

(Qiskit 2.2.0 vs. [Tket](#)
2.6.0)

**83x faster mean
transpilation time**

(Qiskit 2.2.0 vs. [Tket](#)
2.6.0)

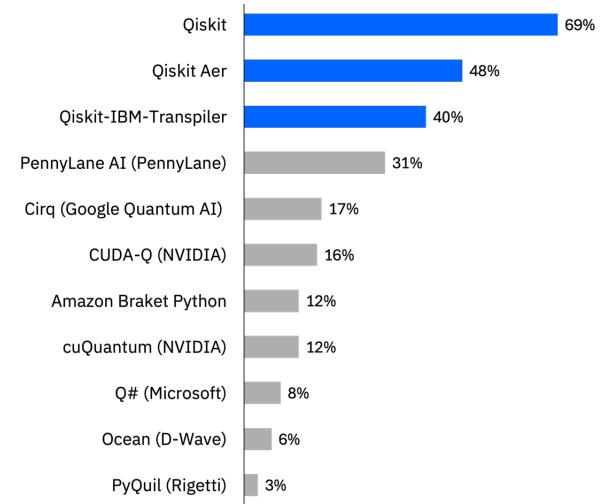
Quality

**29% fewer 2Q
gates**

(Qiskit 2.2.0 vs. [Tket](#)
2.6.0)

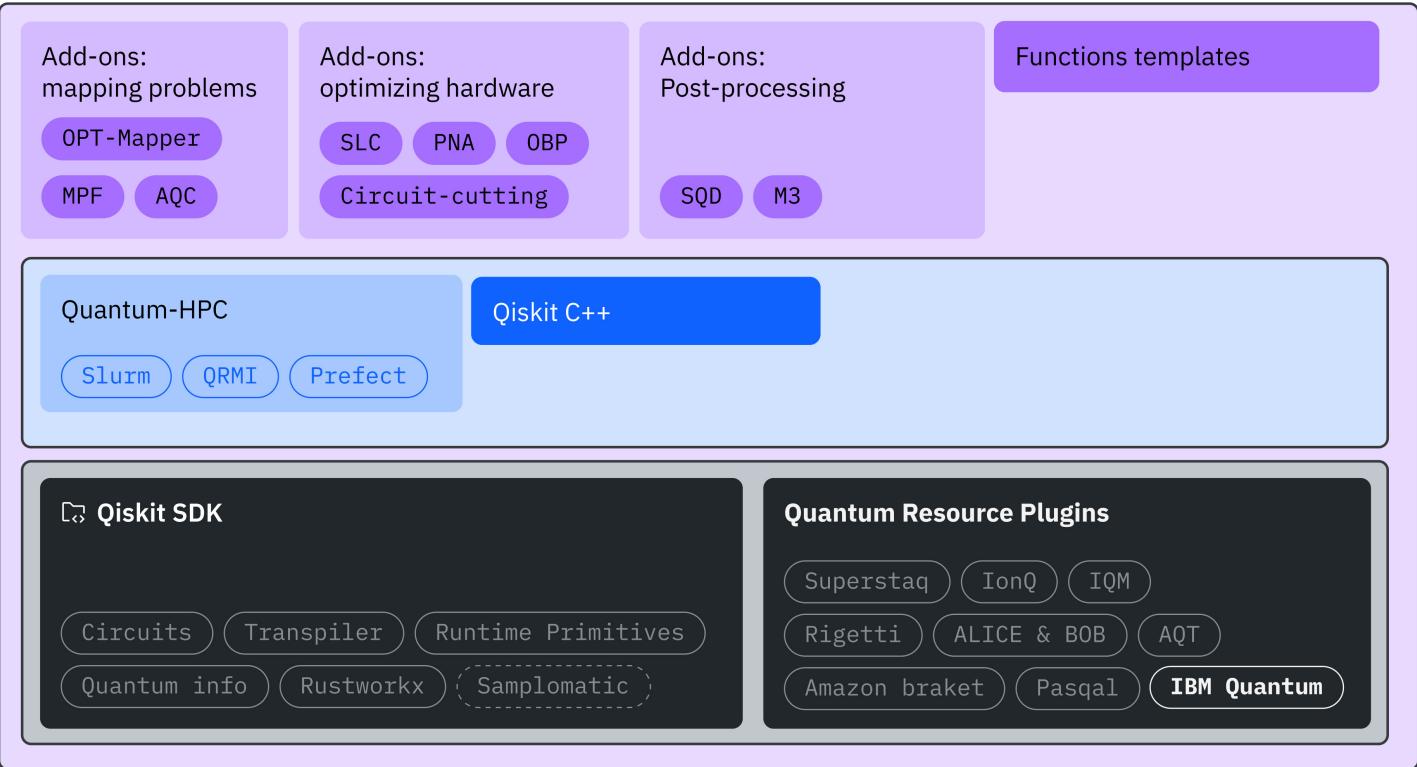
**Only SDK not to
fail any tests**

Full-stack Development Platforms, Compilers, and Simulators Adoption

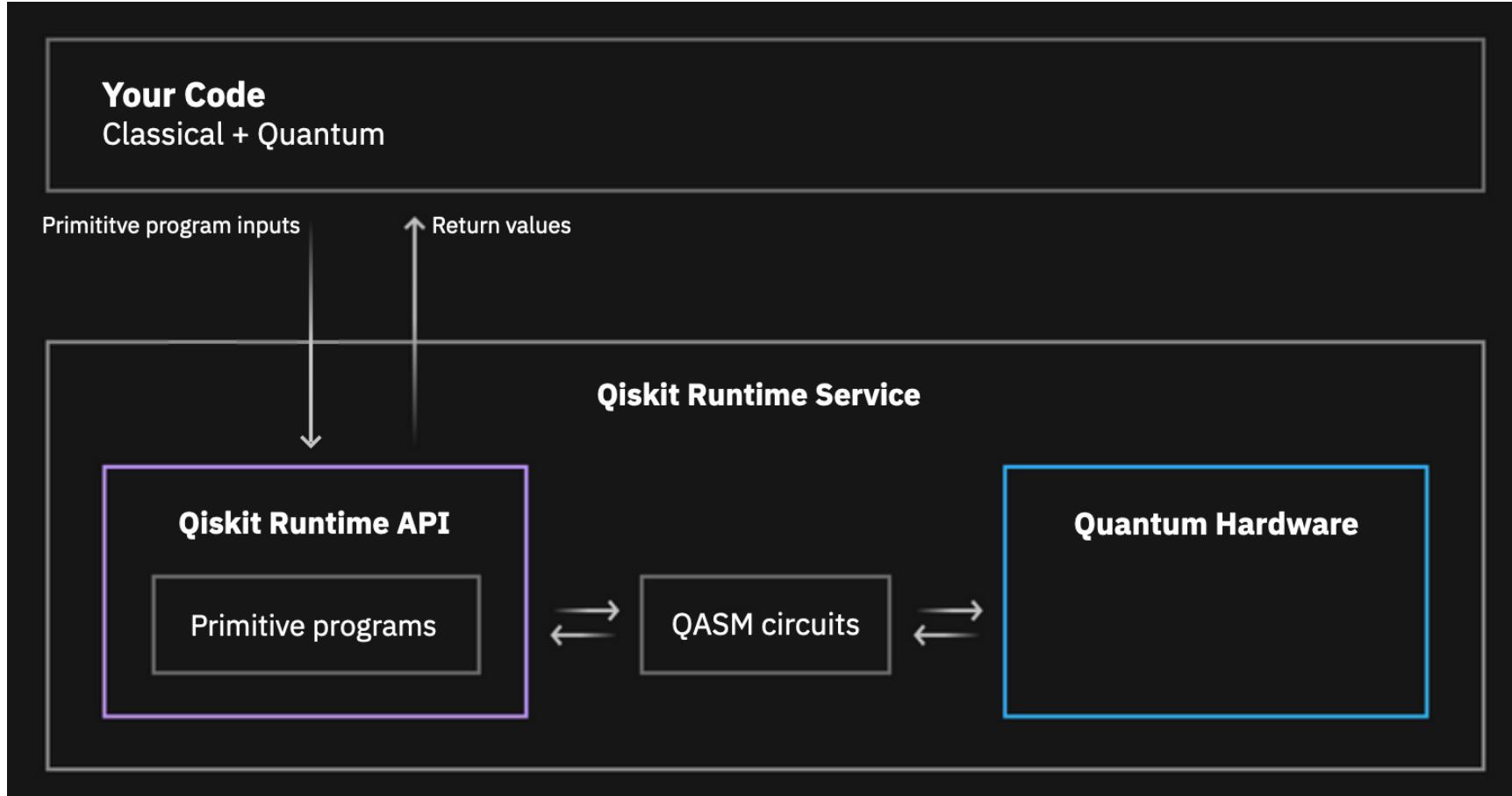


Qiskit – Open ecosystem for quantum computing

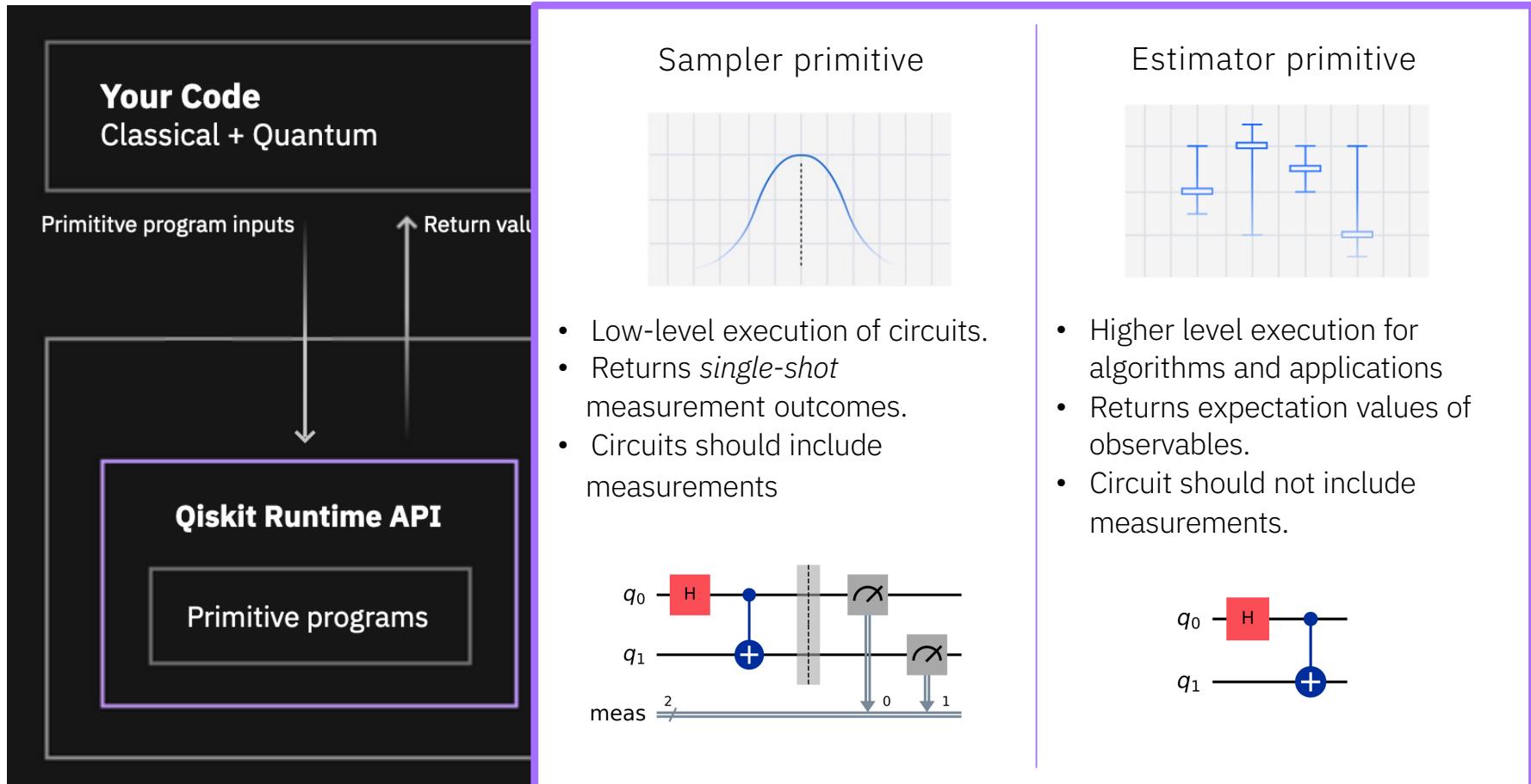
Qiskit for:
■ Algorithm research



Qiskit Runtime



Qiskit Primitives



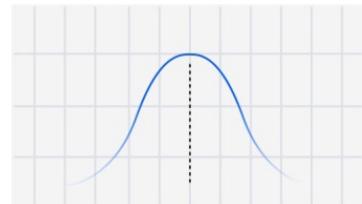
Sampler V2

- The Sampler primitive is a low-level primitive for the execution of circuits and obtaining raw measurement outcomes for each shot.
- The Sampler primitive API is defined by its run method

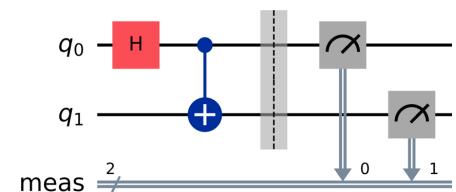
```
Sampler.run(  
    pubs: list[SamplerPubLike],  
    shots: int | None = None,  
) -> Job[SamplerResult]
```

- It can accept a list of 1 or more input programs represented as sampler Primitive Unified Blocs (pubs)
- Optional specification of the number of shots to run the pubs for

Sampler primitive



- Low-level execution of circuits
- Returns *single-shot* measurement outcomes
- Circuits should include measurements



Sampler Pubs

Primitive Unified Blocs

The Primitive Unified Bloc (pub) program input for the sampler is in general a tuple

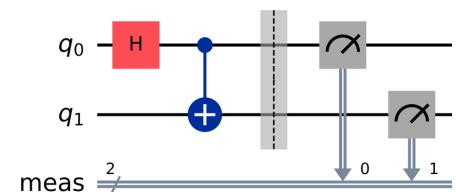
```
pub = (circuit [required],  
       parameter_values [optional],  
       shots [optional])
```

- **Circuit**: An ISA `QuantumCircuit` containing 1 or more `ClassicalRegister` and `measure` instructions.
- **Parameter Values**: A tensor (ND-array) of sets of parameter values to evaluate a parametric circuit with.
- **Shots**: The number of samples or repetitions to measure the circuit for each set of parameter values.

Sampler primitive



- Low-level execution of circuits
- Returns *single-shot* measurement outcomes
- Circuits should include measurements



Estimator V2

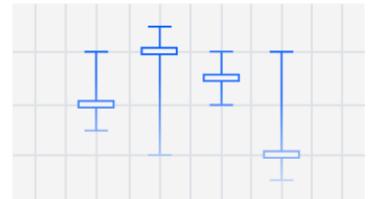
The `Estimator` primitives is a higher-level primitive than the Sampler.

- Used for evaluating estimates of *expectation values* on the state prepared by a circuit
- The `Estimator` primitive API is defined by its run method

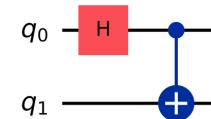
```
Estimator.run(  
    pubs: list[EstimatorPub],  
    precision: float | None = None,  
) -> Job[EstimatorResult]
```

- It can accept a list of 1 or more input programs represented as estimator pubs
- Optional specification of the desired *precision* of expectation value estimates.

Estimator primitive



- Higher level execution for algorithms and applications
- Returns expectation values of observables.
- Circuit should not include measurements.



Estimator Pubs

Primitive Unified Blocs

The Primitive Unified Bloc (pub) program input for the estimator is in general a tuple

```
pub = (circuit [required],  
       observables [required],  
       parameter_values [optional],  
       precision [optional],)
```

Allowed pub-like inputs

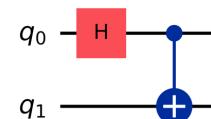
Because of the optional arguments the following are all valid input pubs for the sampler

1. (parametric_isa_circuit, isa_observables, parameter_values, precision)
2. (parametric_isa_circuit, isa_observables, parameter_values)
3. (non_parametric_isa_circuit, isa_observables, None, precision)
4. (non_parametric_isa_circuit, isa_observables)

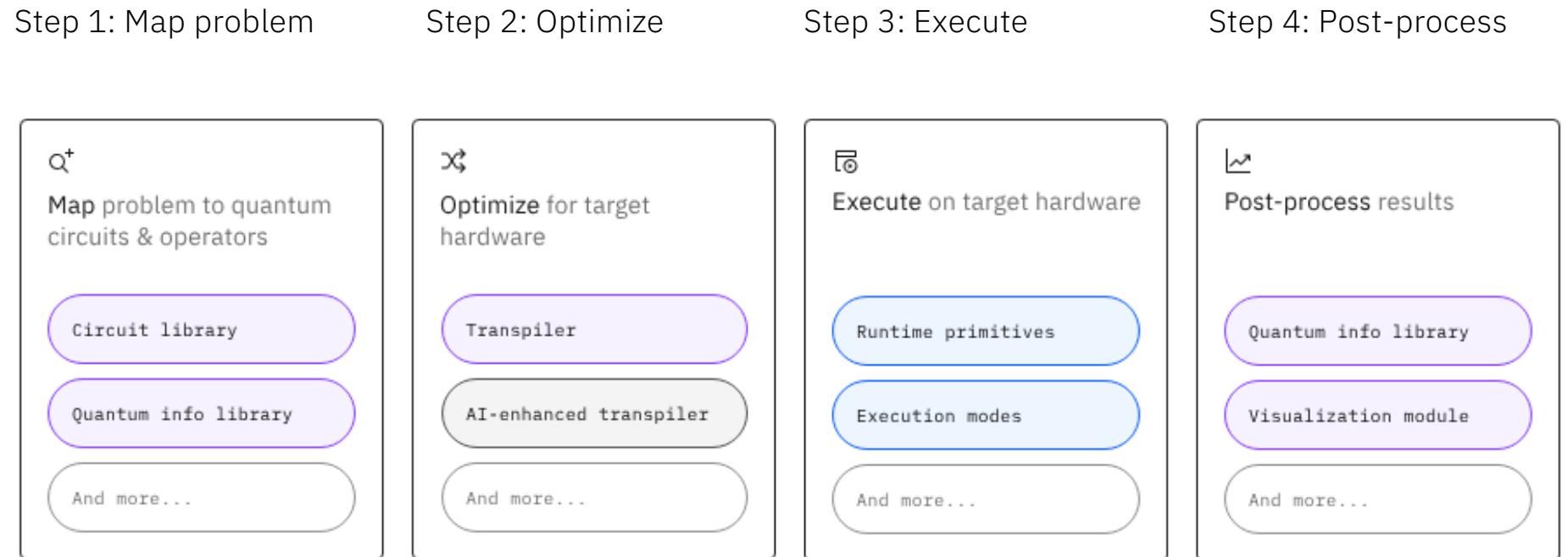
Estimator primitive



- Higher level execution for algorithms and applications
- Returns expectation values of observables.
- Circuit should not include measurements.

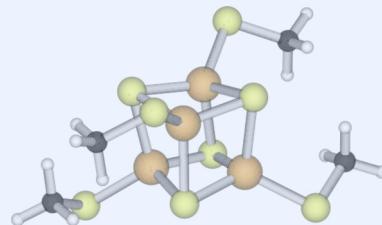


Qiskit Patterns – framework for running domain specific problems

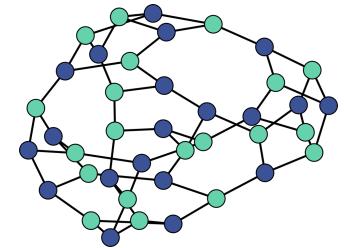


Domain specific problems: Promising quantum computational areas

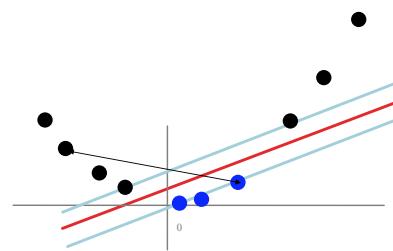
Hamiltonian
simulation



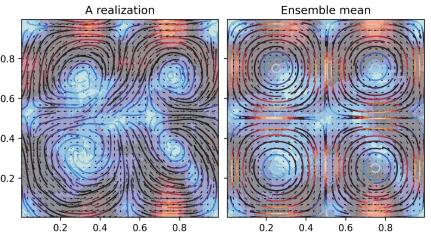
Optimization



Machine learning



Differential equations



Install and set up Qiskit 2.x (macOS)

- Reference URL : <https://docs.quantum.ibm.com/guides/install-qiskit>(For non-macOS users, please refer this.)
- For those who are using Qiskit prior to 1.x. It is strongly recommended to start a new virtual environment to install Qiskit 2.x.
- An extremely fast Python installer uv (10–100x faster than pip) is also available. Please see next slide to utilize uv.

1. Create a new virtual environment, using Python 3.8 or later.

```
python3 -m venv venv
```

2. Activate the environment.

```
source venv/bin/activate
```

3. Install Qiskit.

```
pip install qiskit
```

4. Install the necessary packages one-by-one.

```
pip install qiskit-ibm-runtime  
pip install qiskit[visualization]  
pip install jupyter  
pip install qiskit-aer
```

zsh users will need to put '*qiskit[visualization]*' in single quotes.

5. With the following command, you can launch Jupyter notebook and start using Qiskit.

```
jupyter notebook
```

6. When switching back to your global environment or to another virtual environment, you can deactivate the environment with the following command.

```
deactivate
```

A guide for the impatient - Install and set up Qiskit 2.x (macOS) lightning fast

- Reference URL : <https://docs.quantum.ibm.com/guides/install-qiskit>(For non-macOS users, please refer this.)
- For those who are using Qiskit prior to 1.x. It is strongly recommended to start a new virtual environment to install Qiskit 2.x.
- An extremely fast Python installer [uv](#) (10–100x faster than pip) is also available. Please see instructions below to use uv

1. Install [uv](#), a Python installer for saving time.

```
python3 -m pip install uv
```

2. Create a new virtual environment

```
uv venv venv
```

#python will be installed by default

3. Activate the environment.

```
source venv/bin/activate
```

4. Install Qiskit.

```
uv pip install qiskit
```

5. Install the necessary packages below one-by-one.

```
uv pip install qiskit-ibm-runtime
```

```
uv pip install qiskit[visualization]
```

```
uv pip install qiskit-aer
```

```
uv pip install jupyter
```

5. With the following command, you can launch Jupyter notebook and start using Qiskit.

```
jupyter notebook
```

6. When switching back to your global environment or to another virtual environment, you can deactivate the environment with the following command.

```
deactivate
```

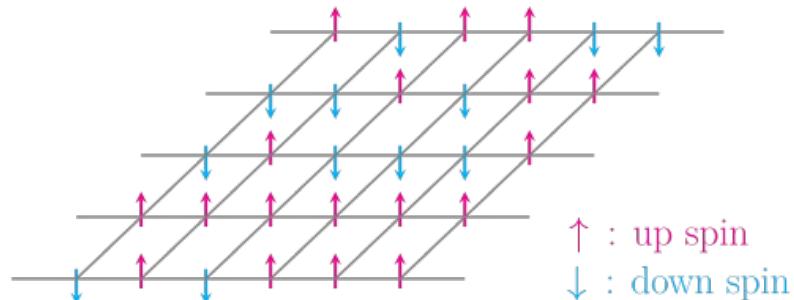
Getting up to speed with Qiskit:
Taking Time-Evolution Simulation as an example

Getting up to speed with Qiskit through simulating a 1-D Transverse Ising Model

Ising Models magnetic dipole moments of atomic "spins" that can be in one of two states (+1 or -1). The spins are arranged in a graph, usually a lattice allowing each spin to interact with its neighbors.

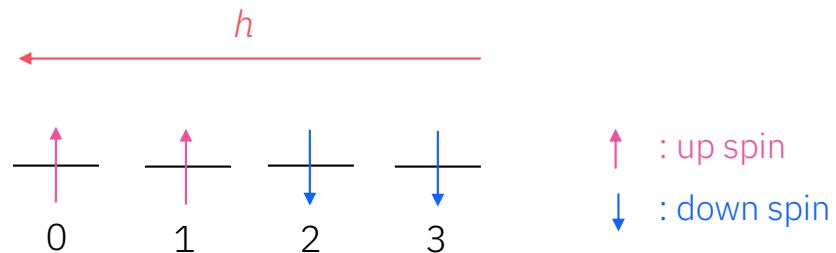
In today's exercise, we will look at a 1-D case.

2-D Ising Model



↑ : up spin
↓ : down spin

→ 1-D Ising Model



↑ : up spin
↓ : down spin

Background information for mapping our problem

Hamiltonian Simulation for Electronic Structure Problems

The goal: solve the Schrödinger equation for $\Psi(t)$

Time-dependent Schrödinger equation

$$\hat{H}\Psi(t) = i\hbar \frac{\partial}{\partial t} \Psi(t)$$

\hat{H} Hamiltonian
corresponds to the total energy
of the system

Ψ Wavefunction
tells us about the characteristics of
a particle system (e.g., energy, position,
momentum, spin)

The goal is to compute the wave function that satisfies the above

$$|\Psi(t)\rangle = e^{-i\hat{H}t} |\Psi(0)\rangle$$

The wave function contains vital information about a quantum system

$$\hat{E}$$

$$\hat{x}$$

$$\hat{p}$$

$$\hat{L}$$

energy

position

momentum

orbital angular momentum ~ Spin

Hamiltonian in general

Hamiltonian of a quantum system is an operator representing the total energy of the system

$$\hat{H} = \hat{T} + \hat{V}$$

The equation $\hat{H} = \hat{T} + \hat{V}$ is displayed. Below the equation, two horizontal lines extend from the '+' sign to the left and right respectively. From the left line, a diagonal line points down to the text 'Kinetic Energy'. From the right line, a diagonal line points down to the text 'Potential Energy'.

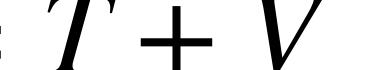
Important in many fields

- Quantum chemistry (material science), Condensed matter physics, High-energy physics
- Optimization problems where the cost function is defined as a Hamiltonian

Spin Hamiltonian

In Hamiltonian Simulations, these energies can come from spin 1/2 interactions and external interactions

$$\hat{H} = \hat{T} + \hat{V}$$



 KE: spin interactions PE: e.g., interactions between electron and Nuclei
 interaction between Nuclei-Nuclei
 external magnetic field

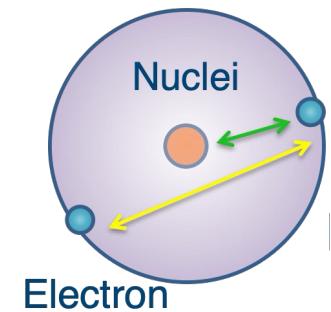
Electronic structure Hamiltonian of a molecule

The molecular Hamiltonian is

$$\hat{\mathbf{H}} = \sum_i^{\text{electrons}} \frac{-\hbar^2}{2m_e} \nabla_i^2 + \sum_A^{\text{nuclei}} \frac{-\hbar^2}{2m_A} \nabla_A^2 + \sum_i^{\text{electrons}} \sum_A^{\text{nuclei}} \frac{-e^2 Z_A}{r_{iA}} + \sum_{i>j}^{\text{electrons}} \frac{e^2}{r_{ij}} + \sum_{A>B}^{\text{nuclei}} \frac{e^2 Z_A Z_B}{r_{AB}}$$

Kinetic energy of electrons Kinetic energy of the nuclei Electron-nuclei attraction Electron-electron repulsion Nuclei-nuclei repulsion

The Born-Oppenheimer approximation neglects the motion of the atomic nuclei



Second quantization

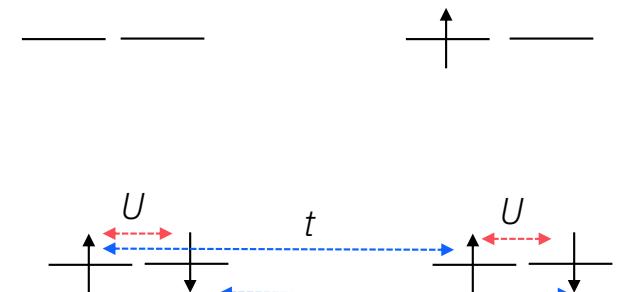
- Hubbard model

Describe conducting and insulating systems

$$H = -t \sum_{i,\sigma} \left(\hat{a}_{i,\sigma}^\dagger \hat{a}_{i+1,\sigma} + \hat{a}_{i+1,\sigma}^\dagger \hat{a}_{i,\sigma} \right) + U \sum_i \hat{n}_{i,\uparrow} \hat{n}_{i,\downarrow}$$

$$\hat{n}_{i,\sigma} = \hat{a}_{i,\sigma}^\dagger \hat{a}_{i,\sigma}$$

Creation operator Annihilation operator



- Quantum Chemistry Hamiltonian

$$\hat{H}_{ele}(\mathbf{r}; \mathbf{R}) = - \sum_i^{N_{ele}} \frac{1}{2} \nabla_i^2 - \sum_A^{N_{nuc}} \sum_i^{N_{ele}} \frac{Z_A}{r_{iA}} + \sum_{i>j}^{N_{ele}} \frac{1}{r_{ij}}$$

Complexity & Computational resources

Kinetic
energy of
electrons

Electron-
nucleus
attraction

Electron-
electron
repulsion

Second quantization: Hartree-Fock method

The Hartree–Fock method is a mean-field approximation typically used to solve the Schrödinger equation for a multi-electron atom or molecule as described in the Born–Oppenheimer approximation and is considered a good starting point for solving this problem.

This method approximates an N -body problem by N one-body problems where each electron evolves in the mean-field of the others.

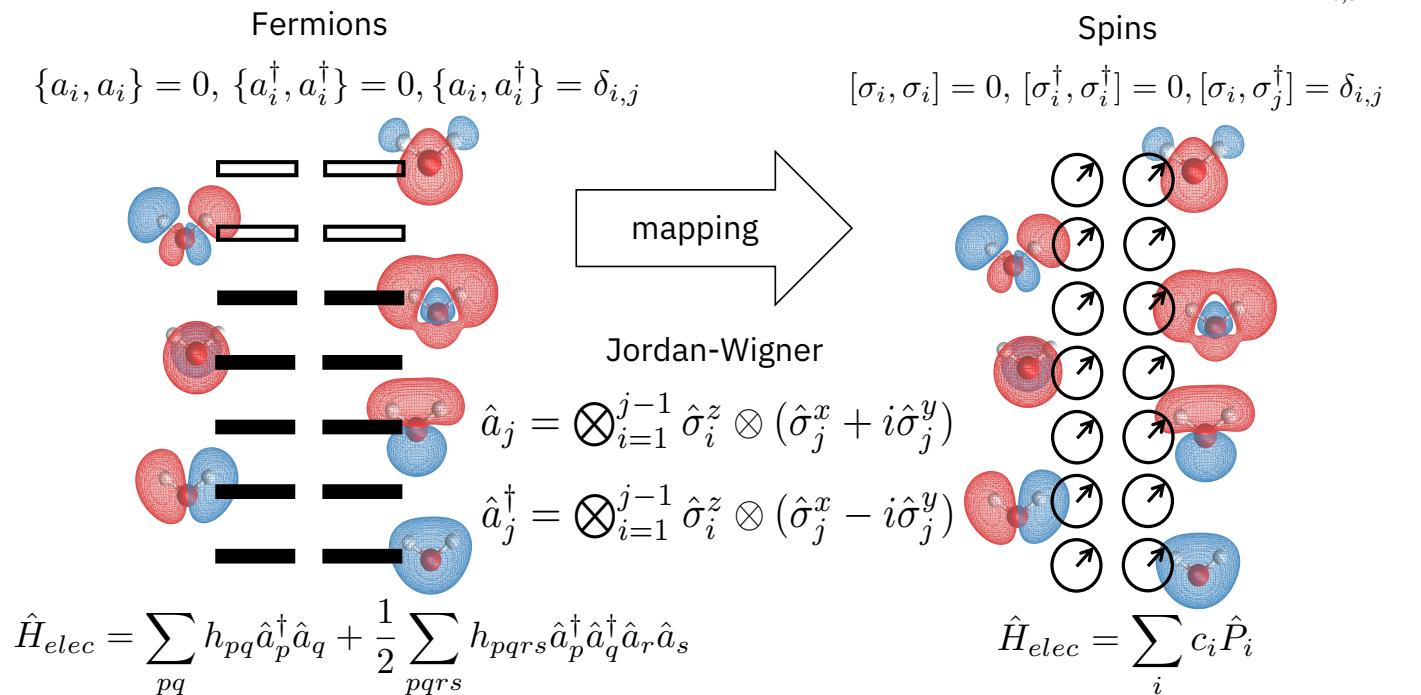
$$\hat{H}_{elec} = \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s$$

Mapping the Hamiltonian

Map the second-quantized Hamiltonian to qubits:

$$H = -t \sum_{i,\sigma} \left(\hat{a}_{i,\sigma}^\dagger \hat{a}_{i+1,\sigma} + \hat{a}_{i+1,\sigma}^\dagger \hat{a}_{i,\sigma} \right) + U \sum_i \hat{n}_{i,\uparrow} \hat{n}_{i,\downarrow}$$

Hubbard, Quantum chemistry



Pauli matrices

- Pauli matrices play a critical role in describing spin $\frac{1}{2}$. (Usually written as $\sigma_x, \sigma_y, \sigma_z$)
- Linear algebraic characteristics: Hermitian, traceless, and unitary
- They represent operators which gives us a projection of spin of electrons along the x, y, z, axis
- And the good news is that they can be implemented using quantum gates on a quantum computer!

Pauli Operators

Pauli x	$\sigma_1 = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli y	$\sigma_2 = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli z	$\sigma_3 = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

Rotational gates

$$R_x(\theta) = e^{-i\frac{\theta}{2}\sigma_x} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i \sin\left(\frac{\theta}{2}\right) \\ -i \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$R_y(\theta) = e^{-i\frac{\theta}{2}\sigma_y} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$R_z(\theta) = e^{-i\frac{\theta}{2}\sigma_z} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$$

Jordan–Wigner Mapping

Fermionic Hamiltonian

$$\hat{H}_M = \sum_{pq} h_{pq} a_p^\dagger a_q + \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s$$

Creation operator

$$a_p^\dagger = \frac{1}{2} (X_p - iY_p) \otimes Z_{p-1} \otimes \cdots \otimes Z_1$$

These our Pauli operators!

Jordan–Wigner mapping

Annihilation operator

$$a_q = \frac{1}{2} (X_q + iY_q) \otimes Z_{q-1} \otimes \cdots \otimes Z_1$$

$$\sigma_{X_i}, \sigma_{Y_i}, \sigma_{Z_i} = X_i, Y_i, Z_i$$

Hydrogen molecule (bond length=0.735 Angstrom, STO-3G basis set. 4 spin orbitals and 36 terms)

$$\begin{aligned}
H_f = & -1.26a_0^\dagger a_0 - 0.47a_1^\dagger a_1 - 1.26a_2^\dagger a_2 - 0.47a_3^\dagger a_3 \\
& + 0.34a_0^\dagger a_0^\dagger a_0 a_0 + 0.33a_0^\dagger a_1^\dagger a_1 a_0 + 0.34a_0^\dagger a_2^\dagger a_2 a_0 + 0.33a_0^\dagger a_3^\dagger a_3 a_0 + \cdots \\
& + 0.09a_0^\dagger a_2^\dagger a_3 a_1 + \cdots
\end{aligned}$$

Mapping the one-body term using Pauli operators

$$\sigma_X \sigma_Y = -\sigma_Y \sigma_X = i\sigma_Z$$

Use these relations

$$\sigma_{X_i}, \sigma_{Y_i}, \sigma_{Z_i} = X_i, Y_i, Z_i \quad \sigma_Y \sigma_Z = -\sigma_Z \sigma_Y = i\sigma_X$$

$$\sigma_Z \sigma_X = -\sigma_X \sigma_Z = i\sigma_Y$$

By using the above relations, we can describe the one-body terms with Pauli operators as shown below

$$\begin{aligned} a_3^\dagger a_3 &= \frac{1}{2} (X_3 - iY_3) \otimes Z_2 Z_1 Z_0 \times \frac{1}{2} (X_3 + iY_3) \otimes Z_2 Z_1 Z_0 \\ &= \frac{1}{4} (X_3 Z_2 Z_1 Z_0 - iY_3 Z_2 Z_1 Z_0) \times (X_3 Z_2 Z_1 Z_0 + iY_3 Z_2 Z_1 Z_0) = \frac{1}{4} (I + I + iX_3 Y_3 - iY_3 X_3) \\ &= \frac{1}{4} (I + I + iX_3 Y_3 - iY_3 X_3) = \frac{1}{2} (I + iX_3 Y_3) = \frac{1}{2} (I - Z_3) \end{aligned}$$

Mapping the two-body term using Pauli operators

$$\sigma_X \sigma_Y = -\sigma_Y \sigma_X = i\sigma_Z$$

Use these relations

$$\sigma_X^2 = \sigma_Y^2 = \sigma_Z^2$$

$$\sigma_Y \sigma_Z = -\sigma_Z \sigma_Y = i\sigma_X$$

$$\sigma_Z \sigma_X = -\sigma_X \sigma_Z = i\sigma_Y$$

By using the above relations, we can describe the two-body terms with Pauli operators as shown below

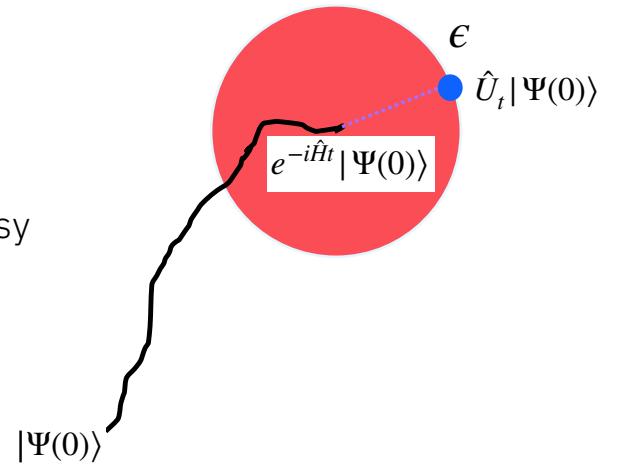
$$\begin{aligned} a_0^\dagger a_2^\dagger a_3 a_1 &= \frac{1}{2} (X_0 - iY_0) \times \frac{1}{2} (X_2 - iY_2) \otimes Z_1 Z_0 \times \frac{1}{2} (X_3 + iY_3) \otimes Z_2 Z_1 Z_0 \times \frac{1}{2} (X_1 + iY_1) \otimes Z_0 \\ &= \frac{1}{16} [-X_3 Y_2 X_1 Y_0 - iX_3 Y_2 Y_1 Y_0 - iY_3 Y_2 X_1 Y_0 + Y_3 Y_2 Y_1 Y_0 - iX_3 X_2 X_1 Y_0 + X_3 X_2 Y_1 Y_0 + Y_3 X_2 X_1 Y_0 + iY_3 X_2 Y_1 Y_0 \\ &\quad - iX_3 Y_2 X_1 X_0 + X_3 Y_2 Y_1 X_0 + Y_3 Y_2 X_1 X_0 + iY_3 Y_2 Y_1 X_0 + X_3 X_2 X_1 X_0 + iX_3 X_2 Y_1 X_0 + iY_3 X_2 X_1 X_0 - Y_3 X_2 Y_1 X_0] \end{aligned}$$

Approximation techniques

Compute wavefunction at time t

← computing this is not easy

$$|\Psi(t)\rangle = e^{-i\hat{H}t} |\Psi(0)\rangle$$



We apply an approximation

$$|\Psi(t + \Delta t)\rangle = e^{-i\hat{H}\Delta t} |\Psi(t)\rangle \approx \left(1 - iH\Delta t - \frac{\hat{H}^2\Delta t^2}{2} + \dots \right) |\Psi(t)\rangle$$

↑
Very small time slice

Trotterization

Let us focus on a simple Hamiltonian

$$\hat{H} = \hat{H}_1 + \hat{H}_2$$

Lie Product Formula (also known as the Trotter Formula)

$$e^{-it(H_1+H_2)} = \lim_{n \rightarrow \infty} \left(e^{-iH_1 \frac{t}{n}} e^{-iH_2 \frac{t}{n}} \right)^n$$

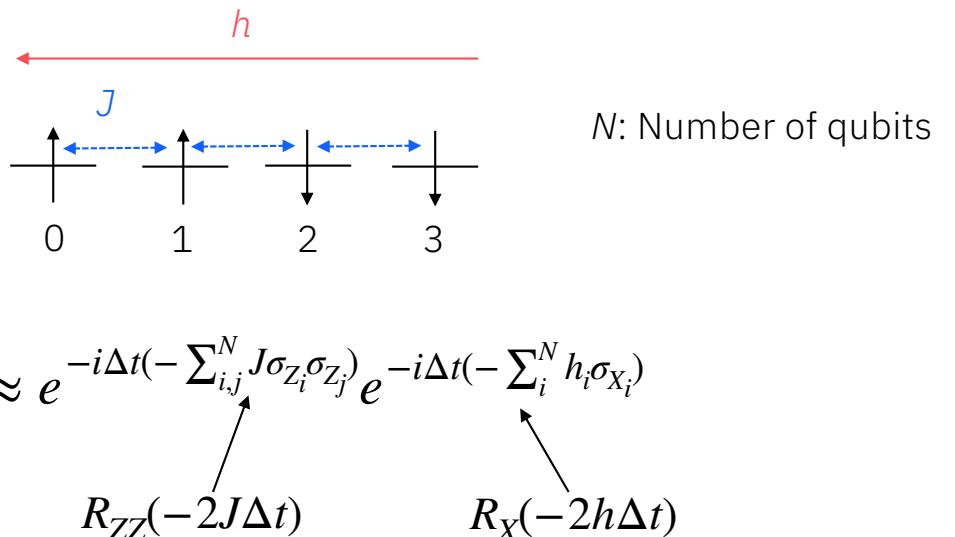
We will take “ n ” to be finite

$$e^{-i(\hat{H}_1+\hat{H}_2)\Delta t} = e^{-i\hat{H}_1\Delta t} e^{-i\hat{H}_2\Delta t}$$

This only holds when H_1 and H_2 commute, but this is often not the case

Example: Trotterization (first-order) Transverse Ising model

$$H = - \sum_{\langle i,j \rangle}^{N-1} J \sigma_{Z_i} \sigma_{Z_j} - \sum_i^N h_i \sigma_{X_i}$$



$$e^{-i\hat{H}\Delta t} = e^{-i\Delta t(-\sum_{i,j}^N J \sigma_{Z_i} \sigma_{Z_j} - \sum_i^N h_i \sigma_{X_i})} \approx e^{-i\Delta t(-\sum_{i,j}^N J \sigma_{Z_i} \sigma_{Z_j})} e^{-i\Delta t(-\sum_i^N h_i \sigma_{X_i})}$$

$$R_{ZZ}(-2J\Delta t)$$

$$R_X(-2h\Delta t)$$

$$R_{ZZ}(\theta) = e^{-i\frac{\theta}{2}\sigma_Z\sigma_Z} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 & 0 & 0 \\ 0 & e^{i\frac{\theta}{2}} & 0 & 0 \\ 0 & 0 & e^{i\frac{\theta}{2}} & 0 \\ 0 & 0 & 0 & e^{-i\frac{\theta}{2}} \end{pmatrix}$$

$$R_X(\theta) = e^{-i\frac{\theta}{2}\sigma_X} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i \sin\left(\frac{\theta}{2}\right) \\ -i \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

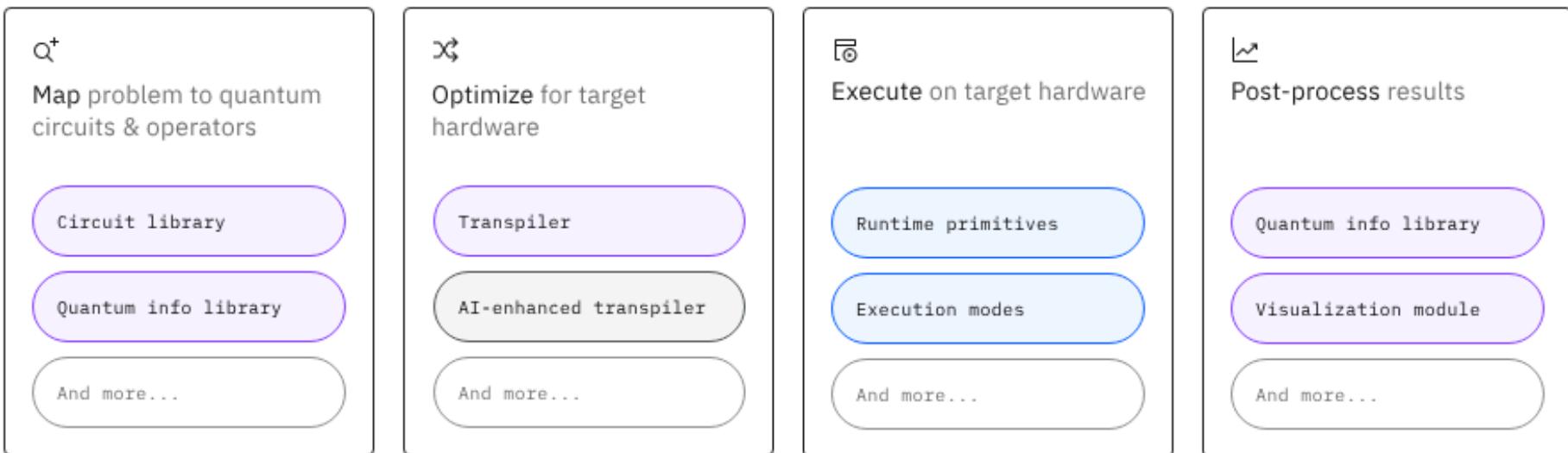
Qiskit Patterns

Step 1: Map problem

Step 2: Optimize

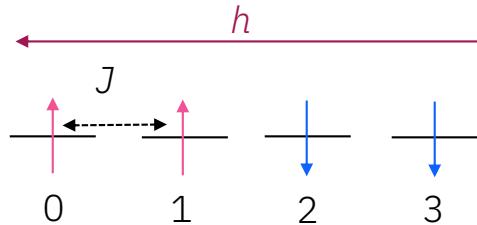
Step 3: Execute

Step 4: Post-process



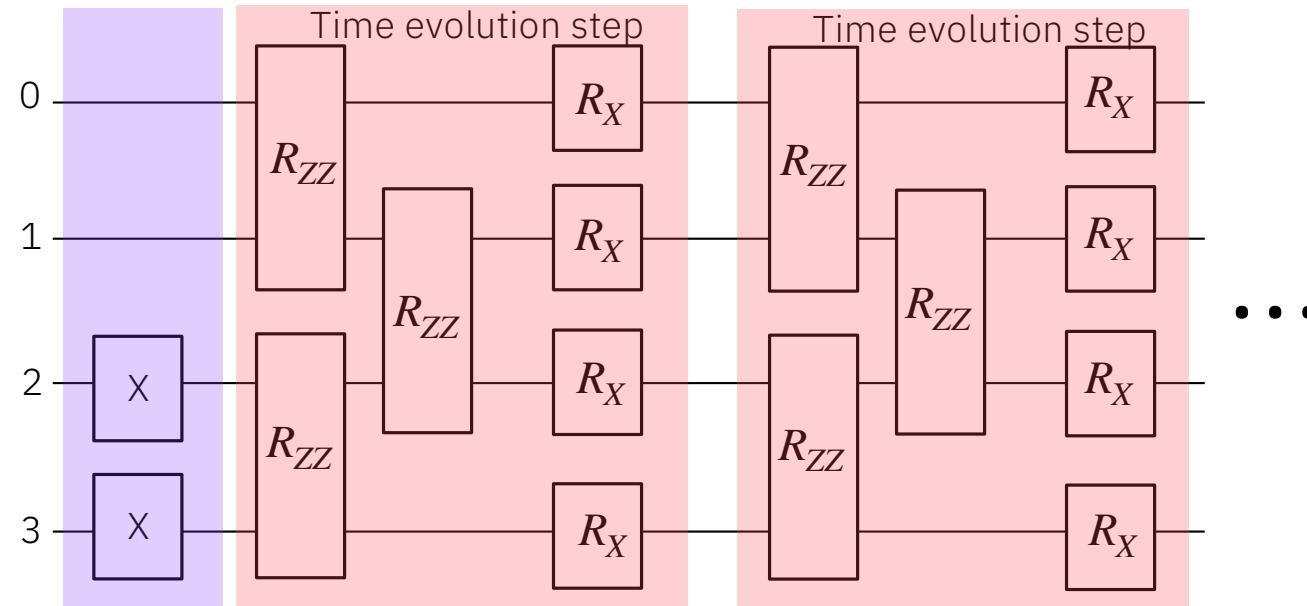
Step 1: Map problem to circuit – 1-D Transverse Ising Model with 1st order trotterization

$$H = - \sum_{\langle i,j \rangle}^{N-1} J \sigma_{Z_i} \sigma_{Z_j} - \sum_i^N h_i \sigma_{X_i}$$



Bit strings
0: up spin
1: down spin

$|0011\rangle$
 $|1100\rangle$



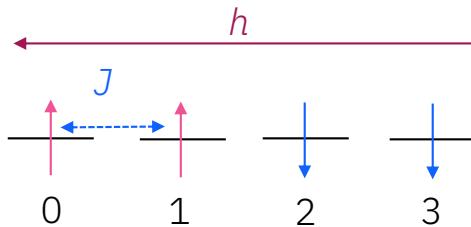
State preparation

By repeating this, we can get the wavefunction at time t

$$|\Psi(t)\rangle = e^{-i\hat{H}t} |\Psi(0)\rangle$$

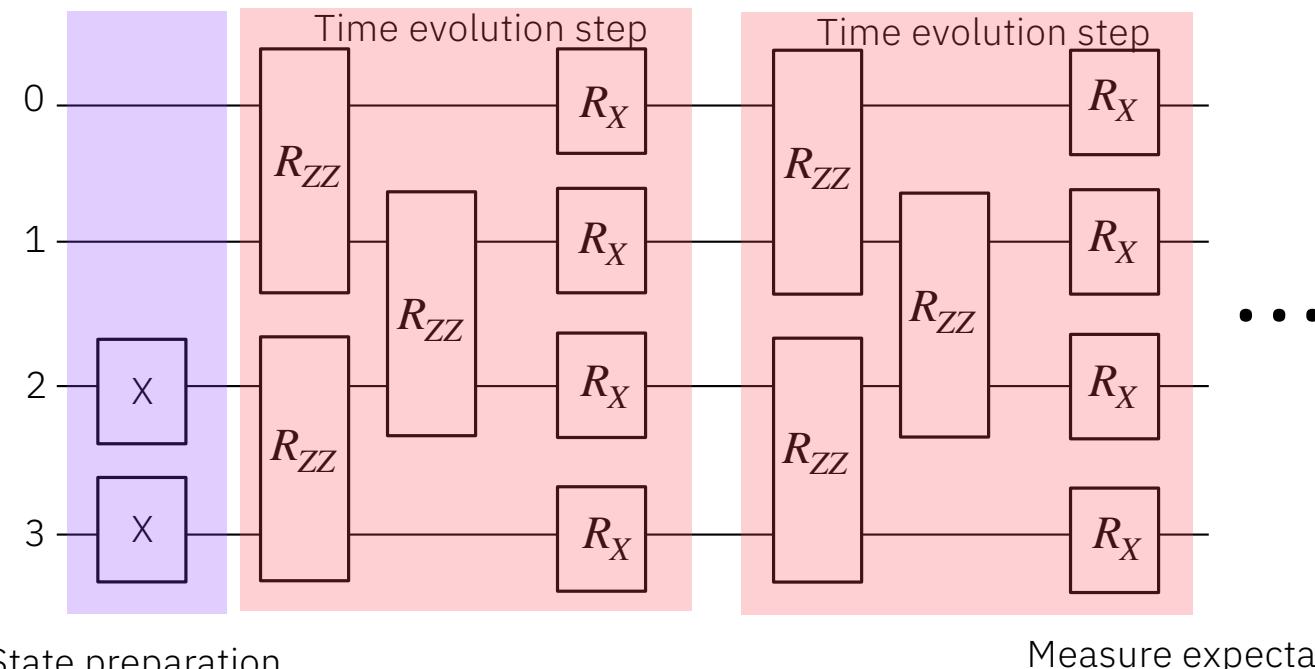
Measuring a physical observable

$$H = - \sum_{\langle i,j \rangle}^{N-1} J \sigma_{Z_i} \sigma_{Z_j} - \sum_i^N h_i \sigma_{X_i}$$



Magnetization

$$\sum_i^N Z_i / N$$



Suzuki-Trotter Formula (2nd order)

Hamiltonian (general form)

$$\hat{H} = \sum_{i=1}^L a_i P_i$$

Second-order Suzuki–Trotter formula

$$U_{ST2} = \prod_{j=1}^L e^{-ia_j P_j \frac{\tau}{2}} \prod_{j'=L}^1 e^{-ia_{j'} P_{j'} \frac{\tau}{2}}$$

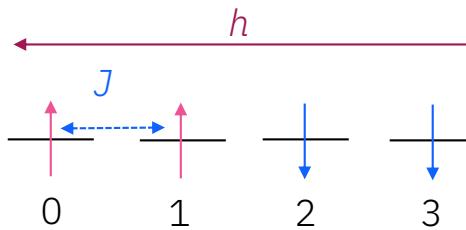
Again, let us focus on a simple Hamiltonian

$$\hat{H} = \hat{H}_1 + \hat{H}_2$$

$$\hat{U}_{ST2} = e^{-i\hat{H}_1 \frac{\Delta t}{2}} e^{-i\hat{H}_2 \Delta t} e^{-i\hat{H}_1 \frac{\Delta t}{2}}$$

Example: Trotterization (second-order) Transverse Ising model

$$H = - \sum_{\langle i,j \rangle}^{N-1} J \sigma_{Z_i} \sigma_{Z_j} - \sum_i^N h_i \sigma_{X_i}$$



$$e^{-i\hat{H}\Delta t} = e^{-i\Delta t(-\sum_{i,j}^N J \sigma_{Z_i} \sigma_{Z_j} - \sum_i^N h_i \sigma_{X_i})}$$

$$\approx e^{-i\frac{\Delta t}{2}(-J \sigma_{Z_0} \sigma_{Z_1})} e^{-i\frac{\Delta t}{2}(-J \sigma_{Z_1} \sigma_{Z_2})} e^{-i\frac{\Delta t}{2}(-J \sigma_{Z_2} \sigma_{Z_3})} e^{-i\frac{\Delta t}{2}(-h \sigma_{X_0})} e^{-i\frac{\Delta t}{2}(-h \sigma_{X_1})} e^{-i\frac{\Delta t}{2}(-h \sigma_{X_2})}$$

$$e^{-i\Delta t(-h \sigma_{X_3})}$$

$$e^{-i\frac{\Delta t}{2}(-h \sigma_{X_2})} e^{-i\frac{\Delta t}{2}(-h \sigma_{X_1})} e^{-i\frac{\Delta t}{2}(-h \sigma_{X_0})} e^{-i\frac{\Delta t}{2}(-J \sigma_{Z_2} \sigma_{Z_3})} e^{-i\frac{\Delta t}{2}(-J \sigma_{Z_1} \sigma_{Z_2})} e^{-i\frac{\Delta t}{2}(-J \sigma_{Z_0} \sigma_{Z_1})}$$

Error in Suzuki-Trotter Formula (2nd order)

$$\hat{U}_{\text{exact}} = e^{-i(\hat{H}_1 + \hat{H}_2)\Delta t}$$

The exact Taylor expansion truncated at the 3rd order

$$\hat{U}_{\text{exact}_3} = \mathbb{I} + (-i\Delta t)(\hat{H}_1 + \hat{H}_2) + \frac{(-i\Delta t)^2}{2} (\hat{H}_1 + \hat{H}_2)^2 + \frac{(-i\Delta t)^3}{6} (\hat{H}_1 + \hat{H}_2)^3$$

Error of the second-order Suzuki-Trotter

$$\|\hat{U}_{\text{exact}_3} - \hat{U}_{\text{ST2}_3}\| \leq \frac{1}{24} \|\hat{H}_2^2 \hat{H}_1 + \hat{H}_1 \hat{H}_2^2 + \hat{H}_1 \hat{H}_2 \hat{H}_1 + \hat{H}_2 \hat{H}_1 \hat{H}_2\| \Delta t^3$$

$$\hat{U}_{\text{ST2}} = e^{-i\hat{H}_1 \frac{\Delta t}{2}} e^{-i\hat{H}_2 \Delta t} e^{-i\hat{H}_1 \frac{\Delta t}{2}}$$

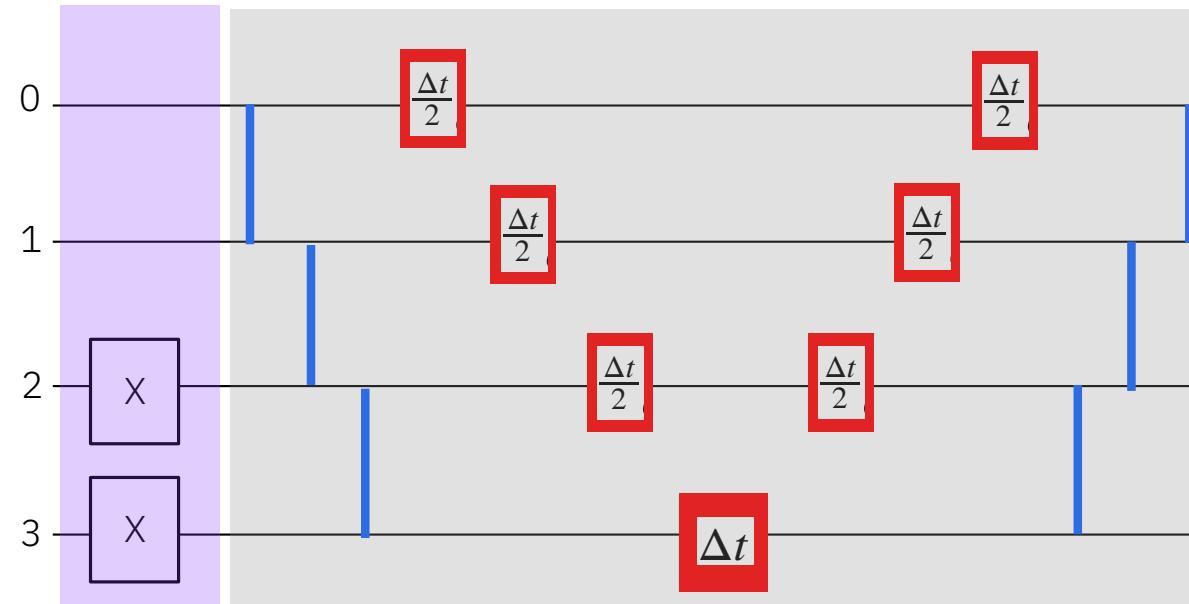
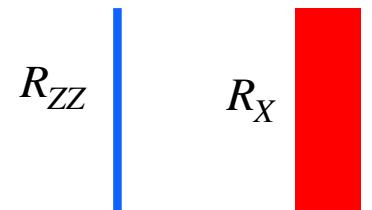
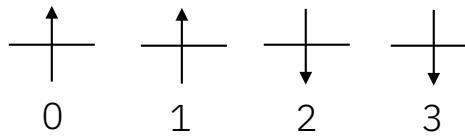
The Taylor expansion for each term (3rd order)
in the 2nd order Suzuki-Trotter Formula

$$\begin{aligned} \hat{U}_{\text{ST2}_3} &= \left[\mathbb{I} + (-i\Delta t/2)\hat{H}_1 + \frac{(-i\Delta t/2)^2}{2} (\hat{H}_1^2) + \frac{(-i\Delta t/2)^3}{6} (\hat{H}_1^3) \right] \\ &\quad \left[\mathbb{I} + (-i\Delta t)\hat{H}_2 + \frac{(-i\Delta t)^2}{2} (\hat{H}_2^2) + \frac{(-i\Delta t)^3}{6} (\hat{H}_2^3) \right] \\ &\quad \left[\mathbb{I} + (-i\Delta t/2)\hat{H}_1 + \frac{(-i\Delta t/2)^2}{2} (\hat{H}_1^2) + \frac{(-i\Delta t/2)^3}{6} (\hat{H}_1^3) \right] \\ \hat{U}_{\text{ST2}_3} &\approx \mathbb{I} + (-i\Delta t/2)(\hat{H}_1 + \hat{H}_2) + \frac{(-i\Delta t/2)^2}{2} (\hat{H}_1 + \hat{H}_2)^2 \\ &\quad + \frac{(-i\Delta t/2)^3}{6} \left[\hat{H}_1^3 + \frac{3}{2}(\hat{H}_1 \hat{H}_2^2 + \hat{H}_2^2 \hat{H}_1 + \hat{H}_1 \hat{H}_2 \hat{H}_1) + \frac{3}{2}(\hat{H}_2 \hat{H}_1^2 + \hat{H}_1^2 \hat{H}_2 + \hat{H}_2^3) \right] \end{aligned}$$

Example: Trotterization (second-order) Transverse Ising model

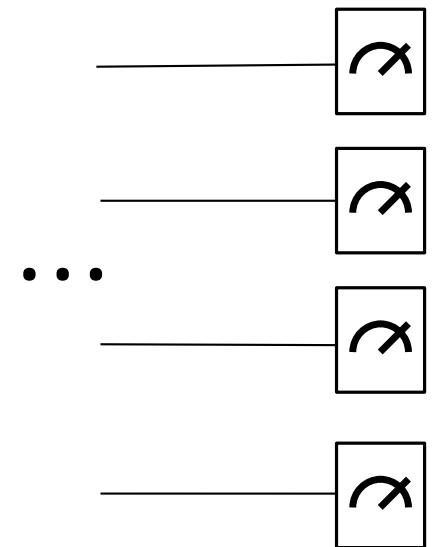
$$H = - \sum_{\langle i,j \rangle}^{N-1} J \sigma_{Z_i} \sigma_{Z_j} - \sum_i^N h_i \sigma_{X_i}$$

Time evolution step



State preparation

By repeating this, we can get the wavefunction of time t



Suzuki-Trotter recursion formula for higher (\mathcal{P}^{th}) order

Second-order Suzuki-Trotter formula $e^{-itH} \approx \hat{U}_{ST2}(t) = \prod_{j=1}^L e^{-ia_j P_j \frac{t}{2}} \prod_{j'=L+1}^{2L} e^{-ia'_{j'} P'_{j'} \frac{t}{2}}$

$$\mathcal{P} = 2k$$

Recursion relation $U_{ST(2k)}(t) = \left[U_{ST(2k-2)}(p_k t) \right]^2 U_{ST(2k-2)}((1 - 4p_k)t) \left[U_{ST(2k-2)}(p_k t) \right]^2$

$$p_k = 1 / \left(4 - 4^{\frac{1}{2k-1}} \right)$$

Fourth order Suzuki-Trotter $\hat{U}_{ST4}(t) = \left[\hat{U}_{ST2}(p_2 t) \right]^2 \hat{U}_{ST2}((1 - 4p_2)t) \left[\hat{U}_{ST2}(p_2 t) \right]^2$

$$p_2 = 1 / \left(4 - 4^{\frac{1}{2*2-1}} \right) = 1 / \left(4 - 4^{\frac{1}{3}} \right) \approx 0.4145$$

$$\hat{U}_{ST4}(\Delta t) = \hat{U}_{ST2}(0.4145\Delta t) \hat{U}_{ST2}(0.4145\Delta t) \hat{U}_{ST2}(-0.6579\Delta t) \hat{U}_{ST2}(0.4145\Delta t) \hat{U}_{ST2}(0.4145\Delta t)$$

Reasons to study product formulas (trotterization)

- The method is intuitive and easy to implement
- Number of qubits required is minimal (no auxiliary qubits required)
- The scaling of the gate depth against the error is not optimal
 - First-order trotter error: $O(t^2/\epsilon)$
 - Second-order trotter error: $O(t^{1.5}/\epsilon^{0.5})$

Qiskit Coding Session

Please have the Jupyter notebook '**121525-qiskit-hamiltonian-simulation.ipynb**' downloaded and ready

<https://github.com/veenaiyuri/riken-quantum-handson>

What we will demonstrate in this coding session:

1. Quantum simulation with a simulator

- Time evolution of an observable
- “Estimator” in Qiskit

2. Quantum simulation with a quantum hardware

- Time evolution of the wavefunction
- “Sampler” in Qiskit



Qiskit Circuit Library

Evolution of the transverse Ising model

Ising model on spin-1/2 particles:

$$H = \underbrace{-J \sum_{jk} Z_j Z_k}_{H_{ZZ}} + \underbrace{h \sum_j X_j}_{H_X}$$

The **SparsePauliOp** is used to implement the Hamiltonian

The **PauliEvolutionGate** implements $U(t) = e^{-itH}$

```
from qiskit.circuit.library import PauliEvolutionGate
from qiskit.quantum_info import Statevector, SparsePauliOp

def get_hamiltonian(nqubits, J, h, alpha):

    # List of Hamiltonian terms as 3-tuples containing
    # (1) the Pauli string,
    # (2) the qubit indices corresponding to the Pauli string,
    # (3) the coefficient.
    Hzz = [("ZZ", [i, i + 1], -J) for i in range(0, nqubits - 1)]
    Hz = [("Z", [i], -h * np.sin(alpha)) for i in range(0, nqubits)]
    Hx = [("X", [i], -h * np.cos(alpha)) for i in range(0, nqubits)]

    # We create the Hamiltonian as a SparsePauliOp, via the method
    # `from_sparse_list`, and multiply by the interaction term.
    hamiltonian = SparsePauliOp.from_sparse_list([*Hzz, *Hz, *Hx], num_qubits=nqubits)
    return hamiltonian.simplify()
```

Summary

Hamiltonian Simulations in the context of simulating electronic structures are one of the most promising areas for quantum computing to show advantage.

Solving the wave function is key to simulation and obtaining physical observables of interest

However, calculating the wave function exactly is very challenging and algorithms that can approximate the solution effectively is an active area of research

Trotterization is a widely used mathematical approach to simulate Hamiltonians on a quantum computer

While higher order trotterization theoretically should provide better accuracy of results, it increases circuit depth which is something to consider when computing on near-term quantum computers.

This should give us inspiration to learning more about error mitigation techniques which is the next topic that will be covered by our partners!

Topics we covered today

IBM Quantum

1. IBM Quantum Platform now on IBM Cloud (since July 2025)
2. Verifying your Account
3. Support Center (how to create cases to get issues resolved)
4. Navigating the platform
5. Qiskit – users' top choice quantum SDK
6. Qiskit Runtime Primitives
7. Qiskit Patterns – framework for running domain specific problems
8. Promising quantum computational areas
9. Installing Qiskit (traditional way vs fast way)
10. Getting up to speed with Qiskit (through a domain specific problem)
 - Taking Hamiltonian Simulation as an example
 - Qiskit coding tutorial: 1-D Transverse Ising Model
11. Summary

Visit IBM Quantum Learning to continue your quantum journey

IBM Quantum Platform

Learning Courses Modules

Search Sign in

Learn quantum computing

Start learning and applying quantum computing with Qiskit through our library of 10+ courses from leading experts.

[View all courses →](#)

Start with the fundamentals

Course: Basics of Quantum Information with John Watrous

[Start this course](#)



Start exploring Key techniques and applications

Course: Quantum Computing in Practice with Olivia Lanes

[Start this course](#)



Quantum Computing in Practice
Learn potential use cases and best practices for experimenting with quantum processors having 100+ qubits.
[New lesson](#) 5 hours

Quantum Diagonalization Algorithms
Explore quantum approaches to matrix diagonalization, including VQE, QKD, SKD and variations of these.
[New](#) 5 hours

Quantum Machine Learning
Learn to leverage the power of quantum computing in machine learning methods.
[New](#) 5 hours

Variational Algorithm Design
An overview of variational algorithms: hybrid classical quantum algorithms.
5 hours

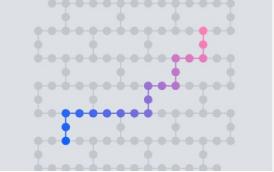
Teach quantum in your classroom

Explore a suite of instructional modules designed to help incorporate quantum computing into traditional STEM courses.

[View all modules →](#)



Quantum Chemistry with VQE
An introduction to VQE covering basic building blocks and applications.
5 hours



Utility-scale quantum computing
A collection of learning assets from a 14-lesson course on utility-scale quantum computing.
5 hours

