Introduction to Quantum Simulation

Kaelyn Ferris, PhD
Research Software Engineer
IBM Quantum

Outline

3 Review of Core Concepts

- Overview of Dirac notation
- Hamiltonians & Observables
- Eigenvalues & Eigenvectors
- Postulates of Quantum Mechanics

14 Quantum Simulation

- Eigenstate solvers
- Simulating dynamics
- Phase Transitions

28 Classical Approaches

32 Simulating Quantum Systems with QPUs

- Preparing quantum simulation algorithms
- Mapping to QPUs

35 Approaches to QPU-based Quantum Simulation

- Trotter Approximation
- Phase Estimation
- VQE
- SQD

Practical Example

Core Concept Review

Core Concept Review

Postulates of quantum mechanics

The entire formalism of quantum mechanics rests upon five basic postulates, which enable us to understand:

- *How a quantum state is described at any given time t*
- *How to calculate the various physical quantities from this quantum state*
- *How to follow the time evolution of a given quantum state*

The mathematical tools we describe later are what we will use to investigate how quantum systems evolve as well as how to characterize them

Core Concept Review

Postulates of quantum mechanics

Postulate 1 (State of a system):

The state of any physical system, at any time, is specified by a state vector $|\psi(t)\rangle$ in a Hilbert space: $\mathcal{H}$

The system is completely described by this state vector. And any superposition of state vectors is *also* a state vector.

Core Concept Review

Postulates of quantum mechanics

Postulate 1 (State of a system):

The

is sp

in a

The

vec

also

Postulate 2 (Observables & Operators):

For every physically measurable quantity: A (known as an observable), there is a corresponding Hermitian operator: $\hat{A}$ whose eigenvectors form a complete basis.

Core Concept Review

Postulates of quantum mechanics

Postulate 1 (State of a system):

The
is sp
in a
The
vec
also

Postulate 2

For
(kn
Her
wh

Postulate 3 (Measurements & eigenvalues):

The measurement of an observable, may be represented formally by the action of $\hat{A}$ on a state vector $|\psi(t)\rangle$

The only possible result of a measurement are one of the eigenvalues a_n of the operator $\hat{A}$

The resulting state immediately after is then:

$$|\psi_n\rangle\boxed{\langle\psi_n|\psi(t)\rangle}$$

Projection of $|\psi(t)\rangle$ onto eigenstate $|\psi_n\rangle$

Core Concept Review

Postulates of quantum mechanics

Postulate 1 (State of a system):

The
is sp

in a

The
vect
also

Postulate 2

For
(kno
Her
who

Postulate 3 (Measurements & eigenvalues):

The me
represe
on a sta

The onl
one of t

The res

Postulate 4 (Probabilistic outcome of measurement):

When making a measurement on a state $|\psi(t)\rangle$ the probability of obtaining one of the nondegenerate eigenvalues a_n is given by:

$$P_n(a_n) = \frac{|\langle\psi_n|\psi\rangle|^2}{\langle\psi|\psi\rangle}$$

Core Concept Review

Postulates of quantum mechanics

Postulate 1 (State of a system):

The

is sp

in a

The

vec

also

Postulate 2

For

(kno

Her

who

Postulate 3 (Measurements & eigenvalues):

The mea

represe

on a sta

The onl

one of t

The res

Postulate 4 (Probabilistic outcome of measurement):

W

th

n

Postulate 5 (Time evolution):

The time evolution of the state vector is governed by the time-dependent Schrödinger equation

$$i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle$$

Which has the general solution:

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$

Core Concept Review

Wavefunctions: *bras*, *kets*, and *state vectors*

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$\langle\psi| = \alpha^*\langle 0| + \beta^*\langle 1| = \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix}$$

Probability of measurement is based on *amplitude*

$$|\alpha|^2 = p_{|0\rangle} \qquad |\beta|^2 = p_{|1\rangle}$$

$$|\alpha|^2 + |\beta|^2 = 1 \qquad \{\alpha, \beta\} \in \mathbb{C}$$

Basis states are *orthogonal*

$$\langle 0|1\rangle = \langle 1|0\rangle = 0$$
$$\langle 0|0\rangle = \langle 1|1\rangle = 1$$

Core Concept Review

A *spanning set* (often called a basis) for a vector space is a set of vectors:

$$\{|v_1\rangle, |v_2\rangle, \ldots, |v_n\rangle\}$$

such that any vector in the space can be written as a linear combination

$$|\psi\rangle = \sum_i a_i |v_i\rangle$$

of vectors in that set

For example:

$$|v_1\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |v_2\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$|v\rangle = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = a_1|v_1\rangle + a_2|v_2\rangle$$

Linear operators also have a matrix representation (with input and output bases)

$$A|v_j\rangle = \sum_i A_{ij}|w_i\rangle$$

$$A = \begin{pmatrix} A_{00} & A_{01} & \ldots & A_{0N} \\ A_{10} & A_{11} & \ldots & A_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{N0} & A_{N1} & \ldots & A_{NN} \end{pmatrix}$$

Often times a system's Hamiltonian is written in this way

Core Concept Review

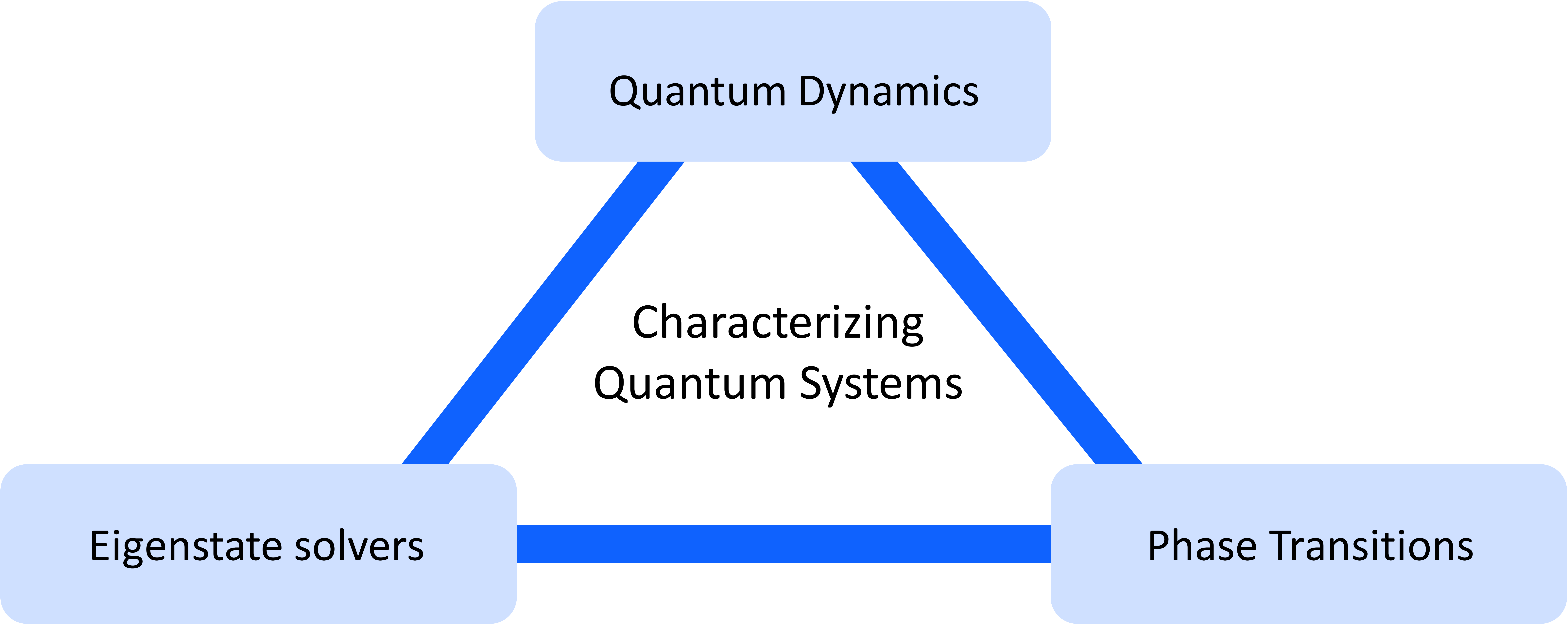
Notation for common operations

z^*	Complex conjugate $(1+i)^* = 1 - i$
$\vert\psi\rangle$	Vector (aka a *ket*)
$\langle\psi\vert$	Vector dual (aka a *bra*)
$\langle\phi\vert\psi\rangle$	*Inner product*
$\vert\phi\rangle \otimes \vert\psi\rangle$ $\vert\phi\rangle\vert\psi\rangle$	*Tensor product*

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^\dagger = \begin{pmatrix} a^* & c^* \\ b^* & d^* \end{pmatrix}$$

$A^\dagger = \left(A^T\right)^*$	*Hermitian conjugate (*adjoint*)*
$\langle\phi\vert A\vert\psi\rangle$	*Expectation value*

Quantum Simulation



Quantum Dynamics

Characterizing Quantum Systems

Eigenstate solvers

Phase Transitions

Quantum Simulation

Most straightforward approach, but quickly becomes intractable as system size grows

Eigenstate solvers

Quantum Simulation

$$\boxed{\lambda = \pm 1}$$

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

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

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

Eigenstate solvers

Quantum Simulation

$$\boxed{\lambda = \pm 1}$$

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

$$\psi_{z+} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \psi_{x+} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \psi_{y+} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}$$

$$\psi_{z-} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \psi_{x-} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad \psi_{y-} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}$$

Eigenstate solvers

Quantum Simulation

$$\boxed{\lambda = \pm 1}$$

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

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

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

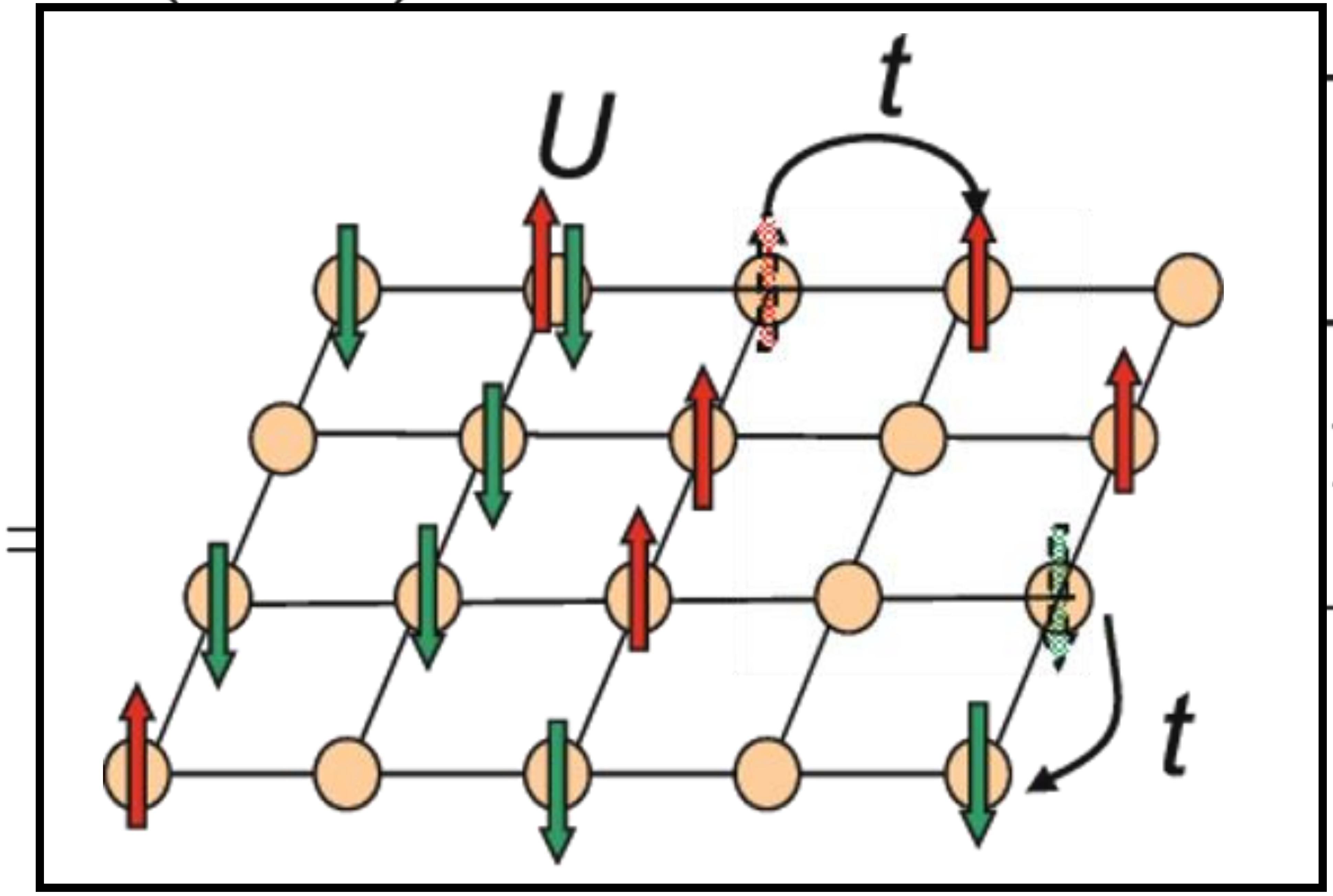
$$\psi_{z+} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\psi_{z-} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\psi_{x+} \quad \begin{bmatrix} 1 \\ i \end{bmatrix}$$

$$\psi_{x-} = \quad \begin{bmatrix} 1 \\ -i \end{bmatrix}$$

Eigenstate solvers

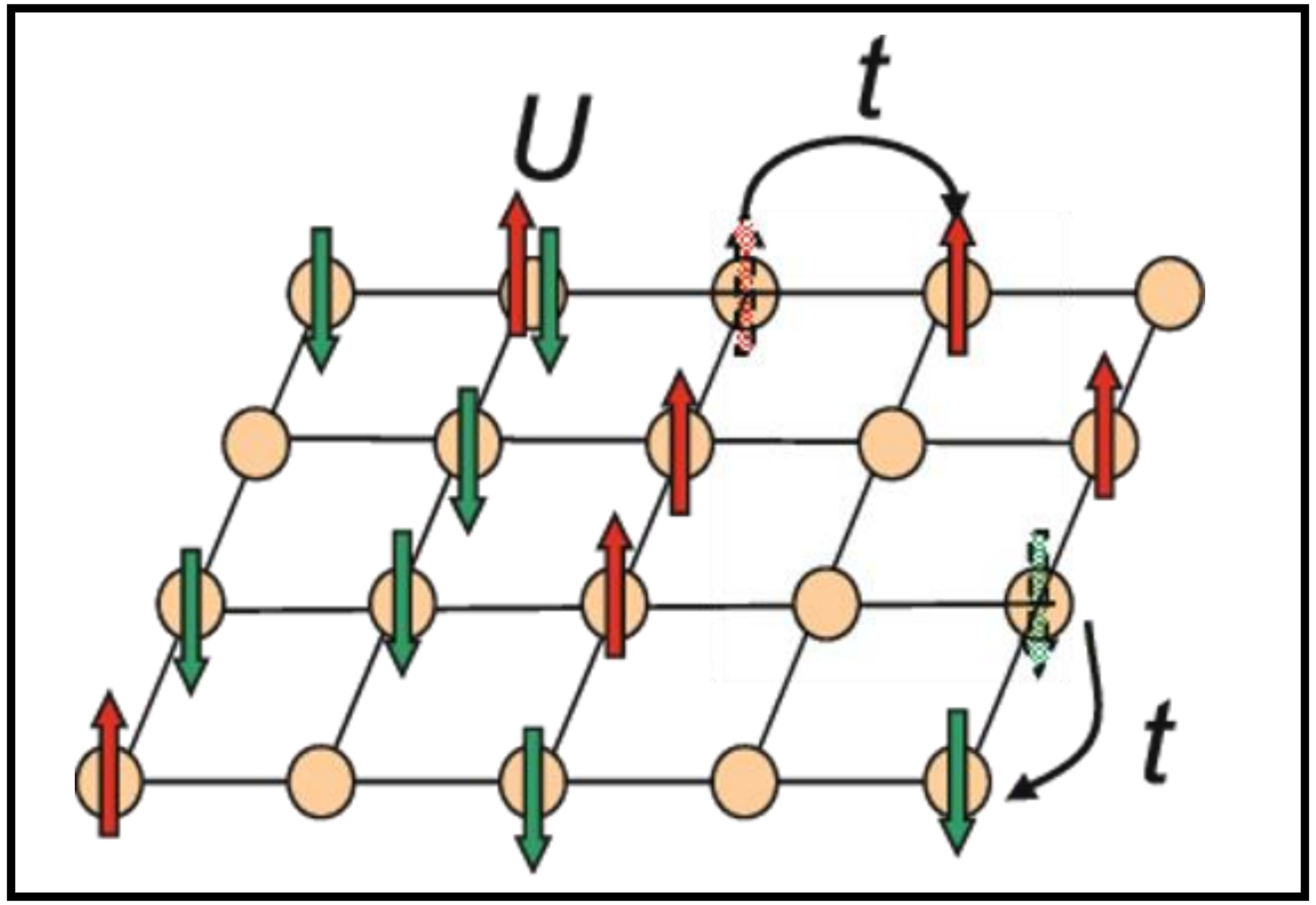


Susumu Yamada, et. al. *High Performance LOBPCG Method for Solving Multiple Eigenvalues of Hubbard Model: Efficiency of Communication Avoiding Neumann Expansion*

Quantum Simulation

$$H = \begin{pmatrix} H_{00} & H_{01} & \dots & H_{0N} \\ H_{10} & H_{11} & \dots & H_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ H_{N0} & H_{N1} & \dots & H_{NN} \end{pmatrix}$$

Eigenstate solvers

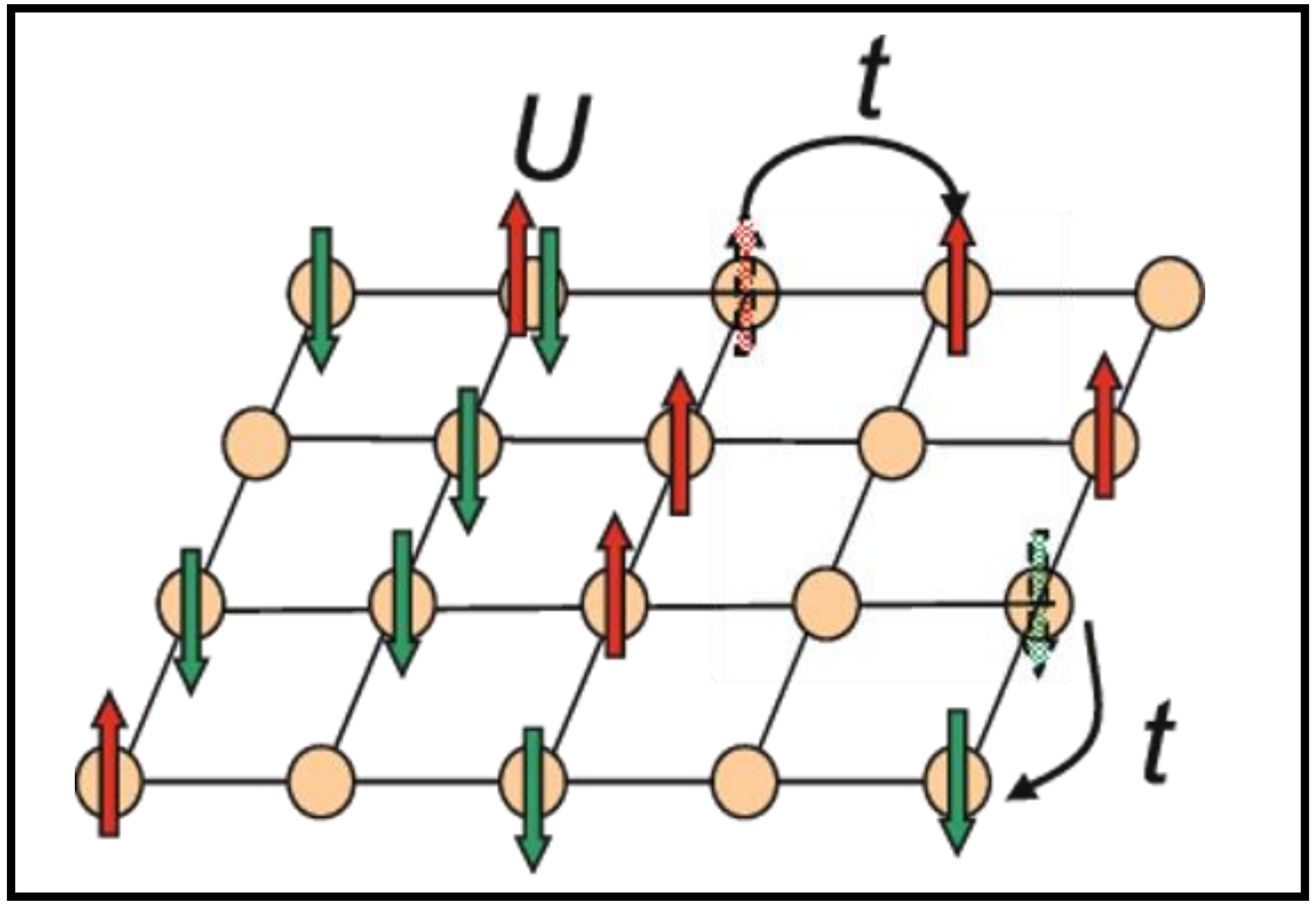


Susumu Yamada, et. al. *High Performance LOBPCG Method for Solving Multiple Eigenvalues of Hubbard Model: Efficiency of Communication Avoiding Neumann Expansion*

Quantum Simulation

$$H = \xcancel{\begin{pmatrix} H_{00} & H_{01} & \cdots & H_{0N} \\ H_{10} & H_{11} & \cdots & H_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ H_{N0} & H_{N1} & \cdots & H_{NN} \end{pmatrix}}$$

Eigenstate solvers



Susumu Yamada, et. al. *High Performance LOBPCG Method for Solving Multiple Eigenvalues of Hubbard Model: Efficiency of Communication Avoiding Neumann Expansion*

Quantum Simulation

Quantum Dynamics

Used to understand how quantum systems respond or change with time

Quantum Simulation

Quantum Dynamics

Used to understand how quantum systems respond or change with time

$$i\hbar\frac{d|\psi\rangle}{dt} = H|\psi\rangle$$

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$

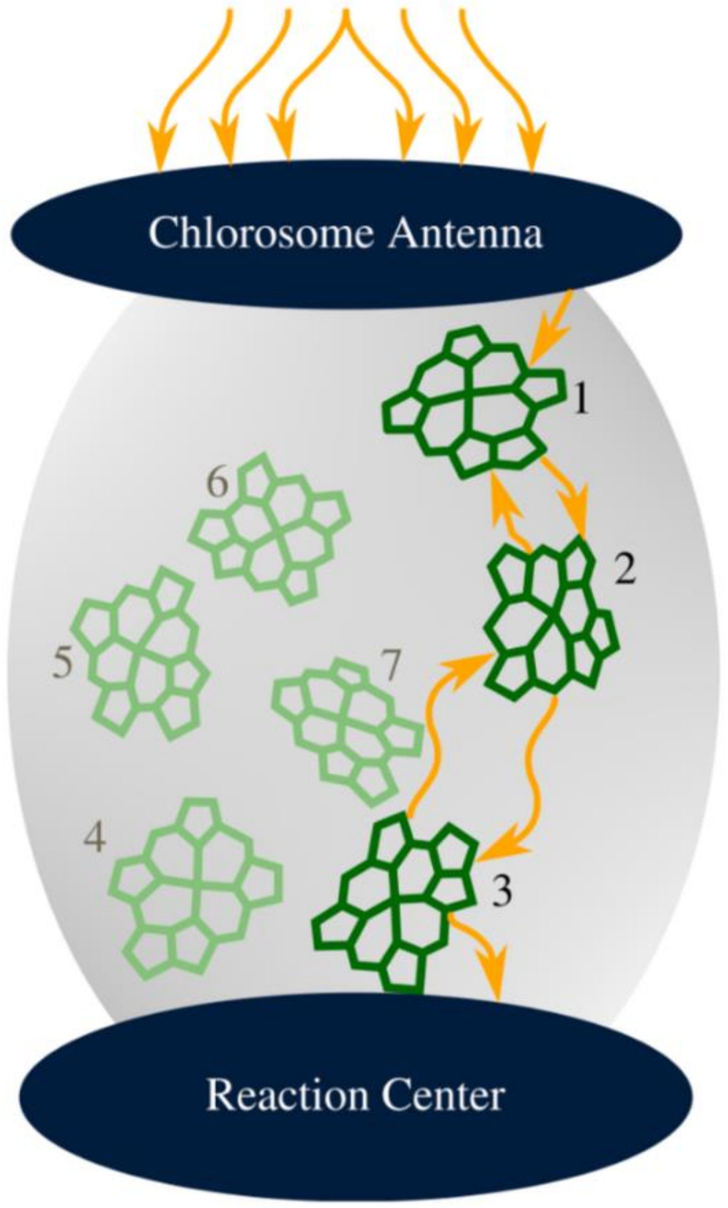
Quantum Simulation

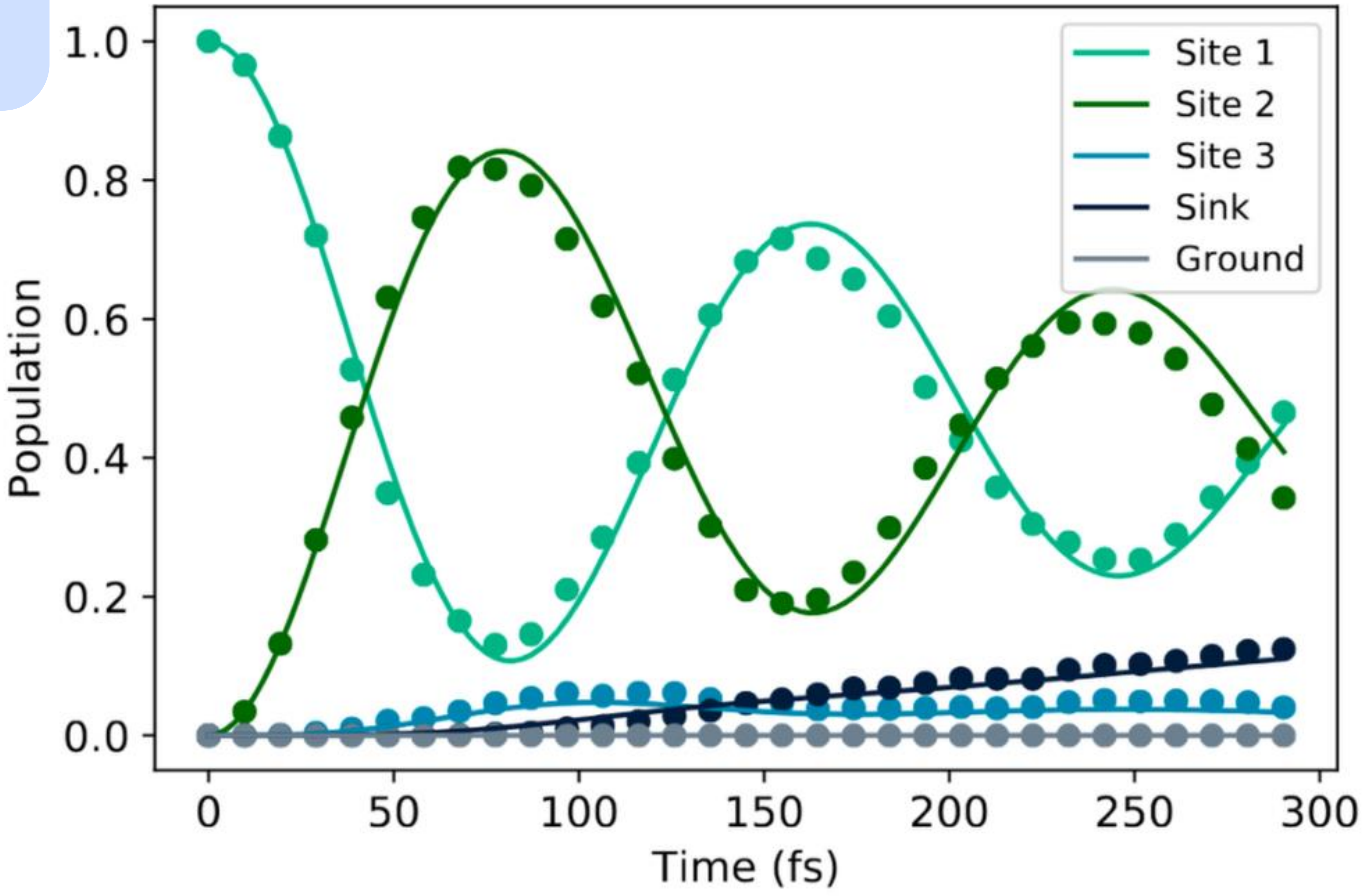
Quantum Dynamics

Used to understand how quantum systems respond or change with time

$$i\hbar\frac{d|\psi\rangle}{dt} = H|\psi\rangle$$

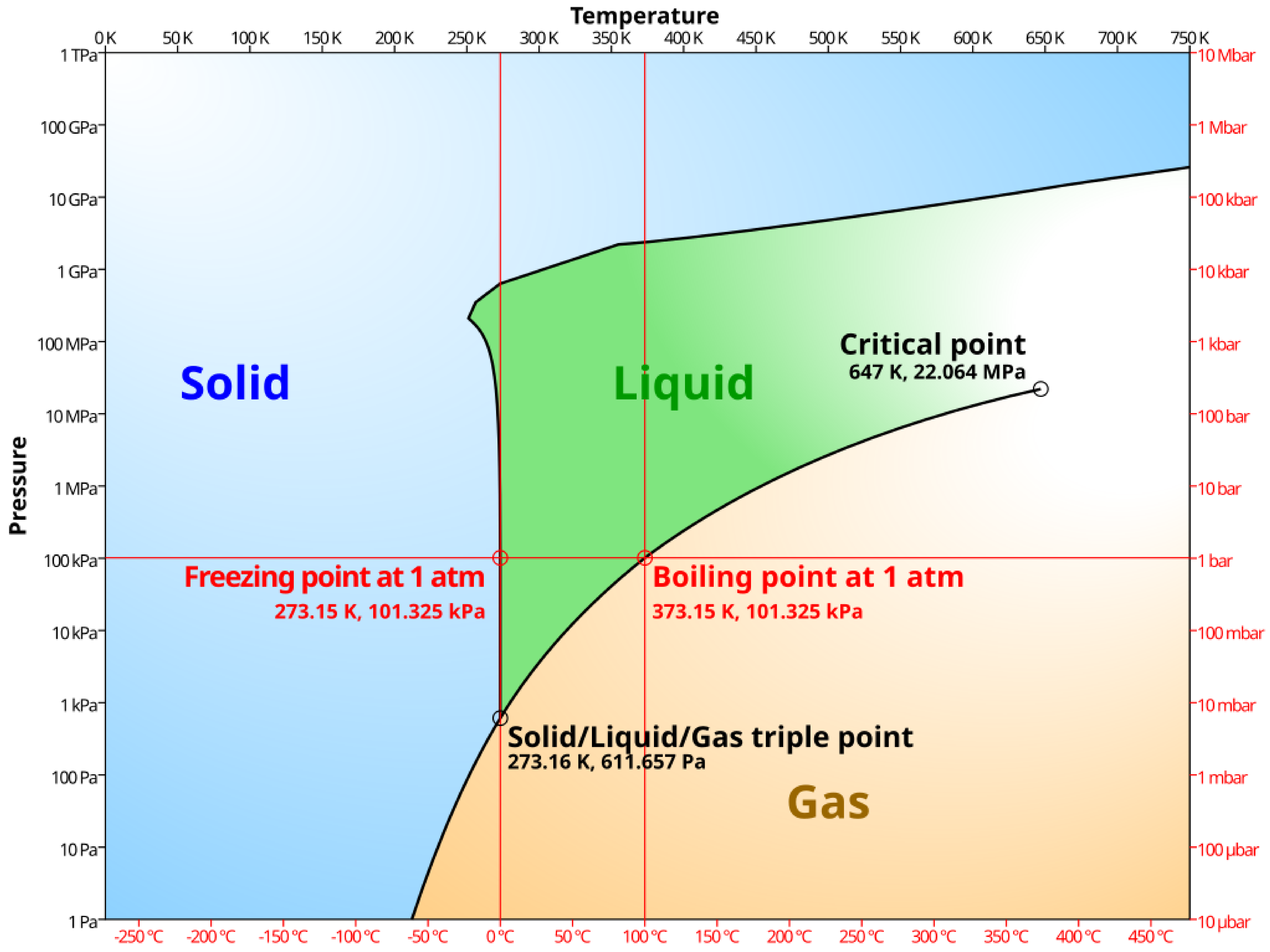
$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$





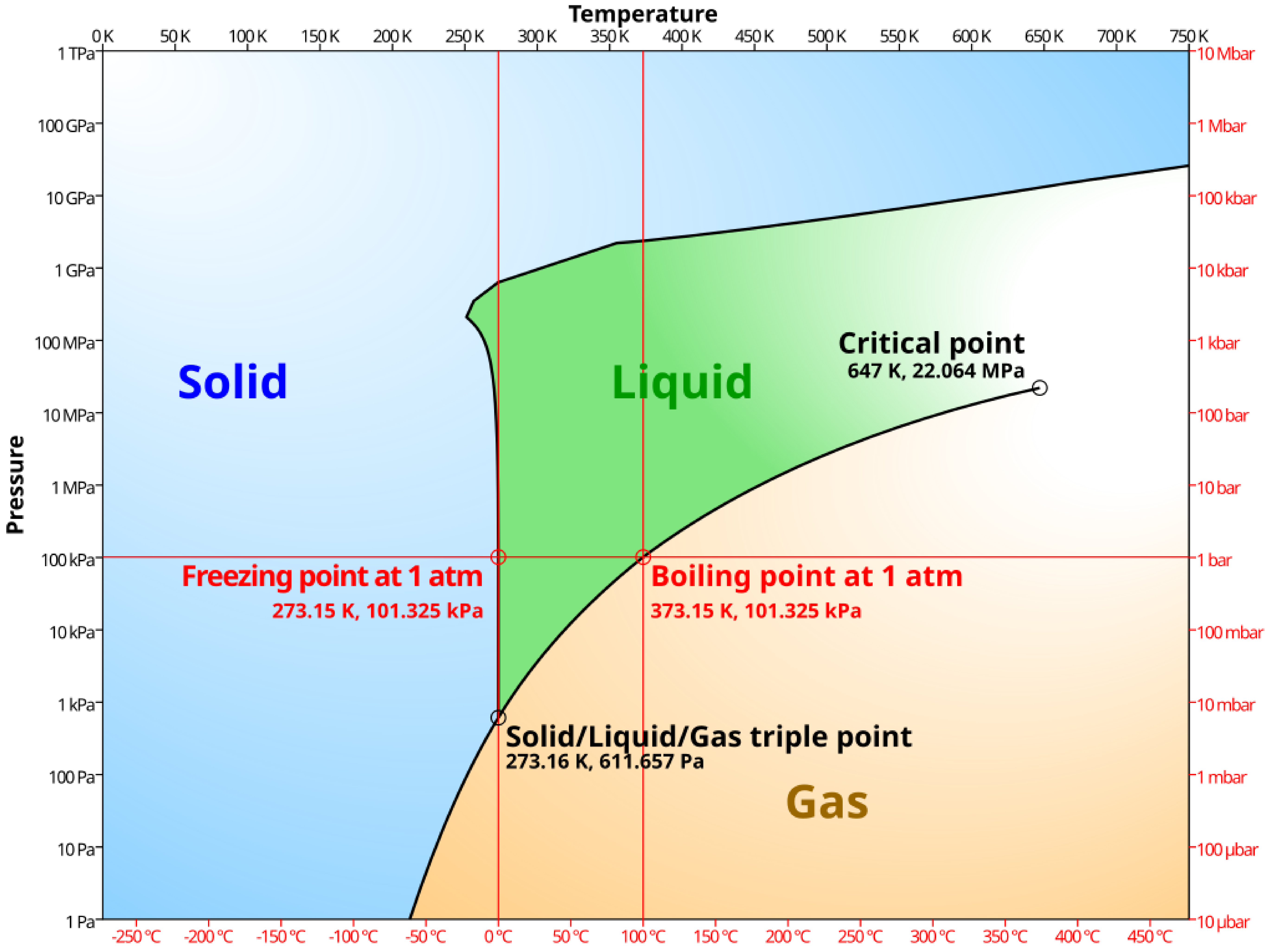
Quantum 6, 726 (2022)

Quantum Simulation



Phase Transitions

Quantum Simulation



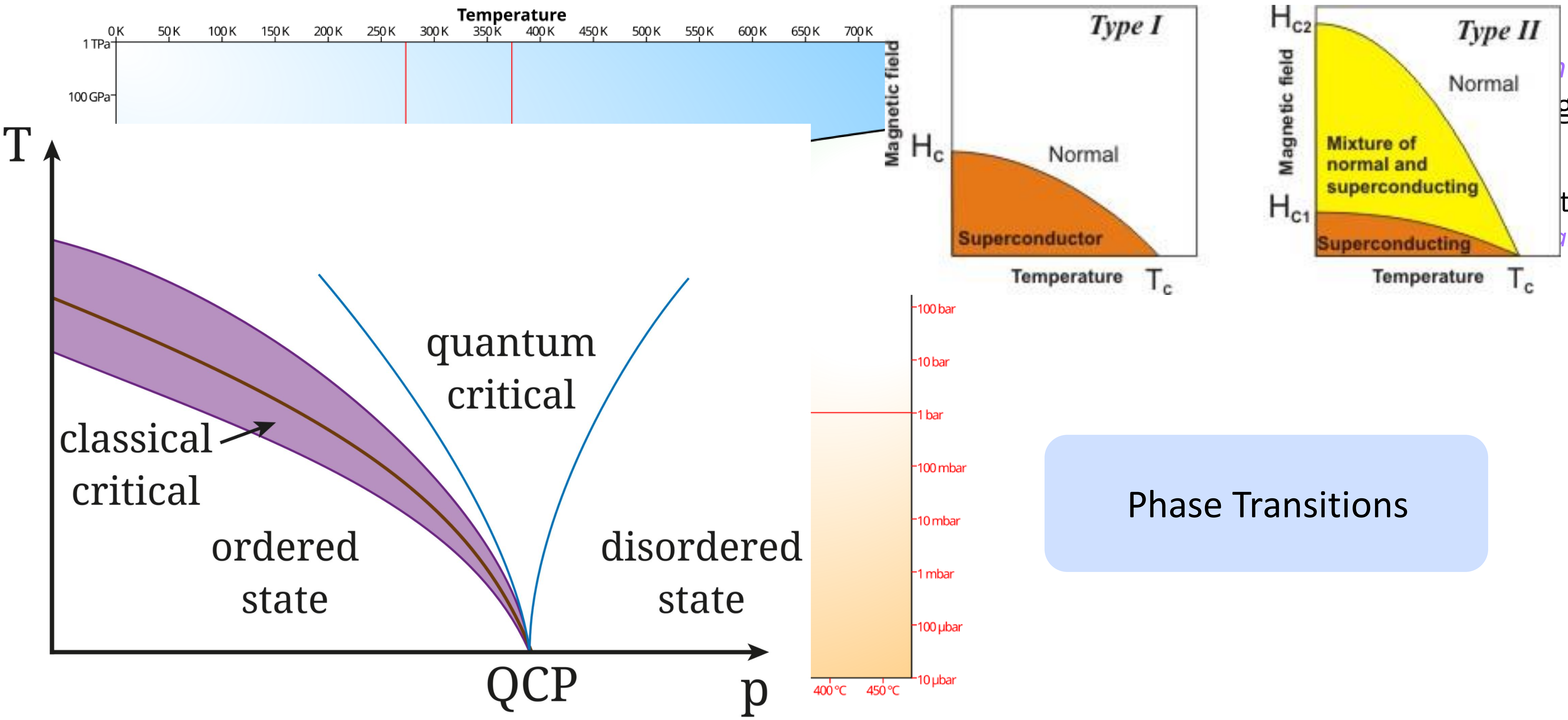
Unlike classical phase transitions, *quantum phase* transitions describe abrupt changes in the ground state of a system.

Typically induced by varying a magnetic field, temperature, or some other *order parameter*

Phase Transitions

Quantum Simulation

Phase transition from conductor to superconductor



ntum

ges

tic field,

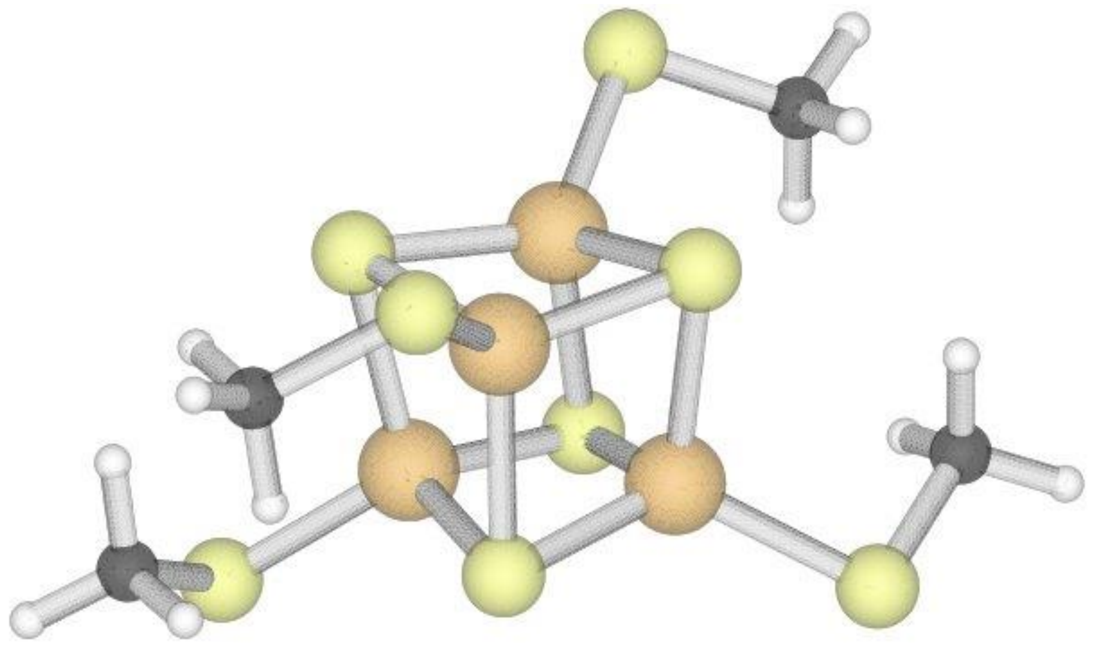
ameter

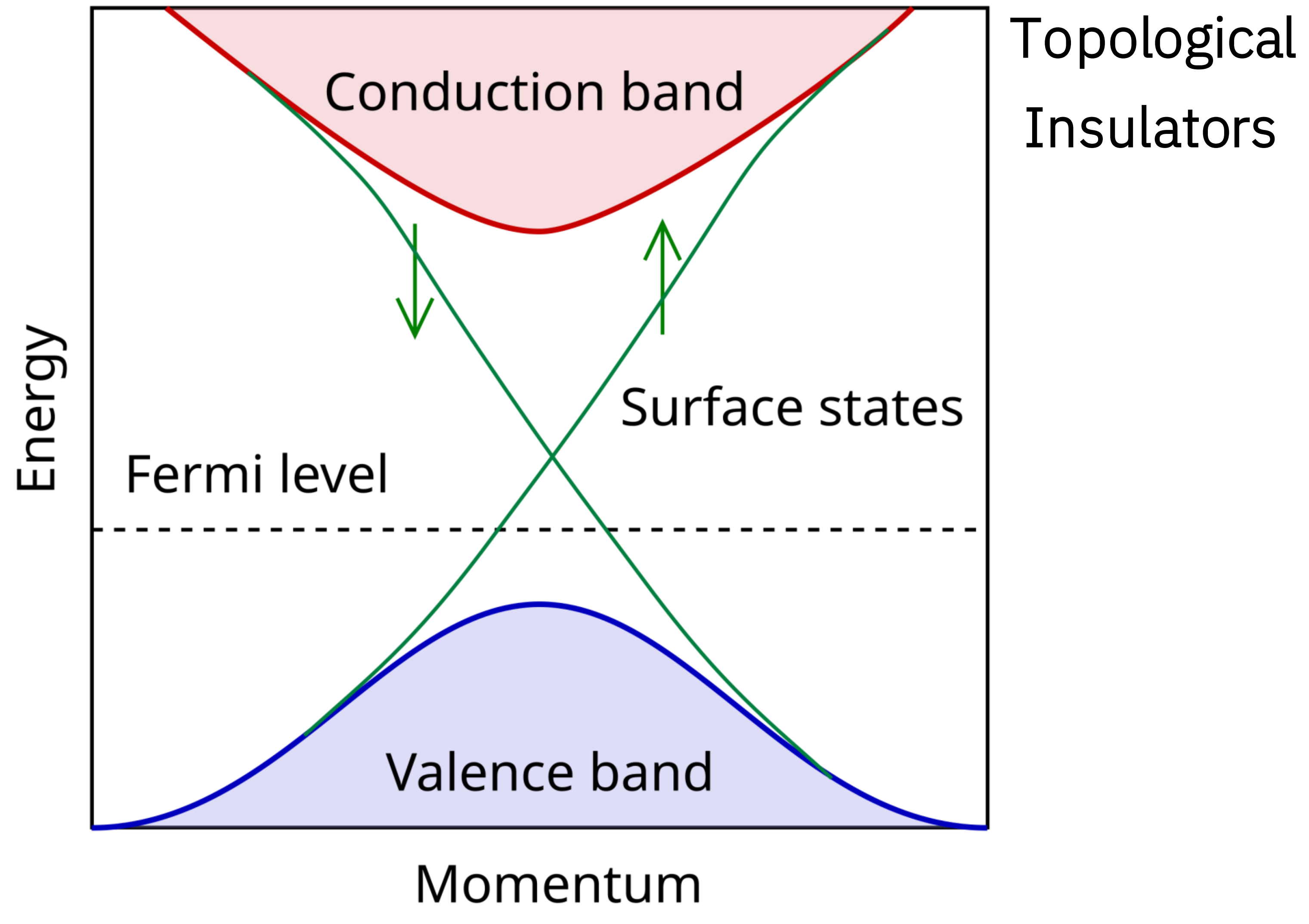
Phase Transitions

Classical Approaches

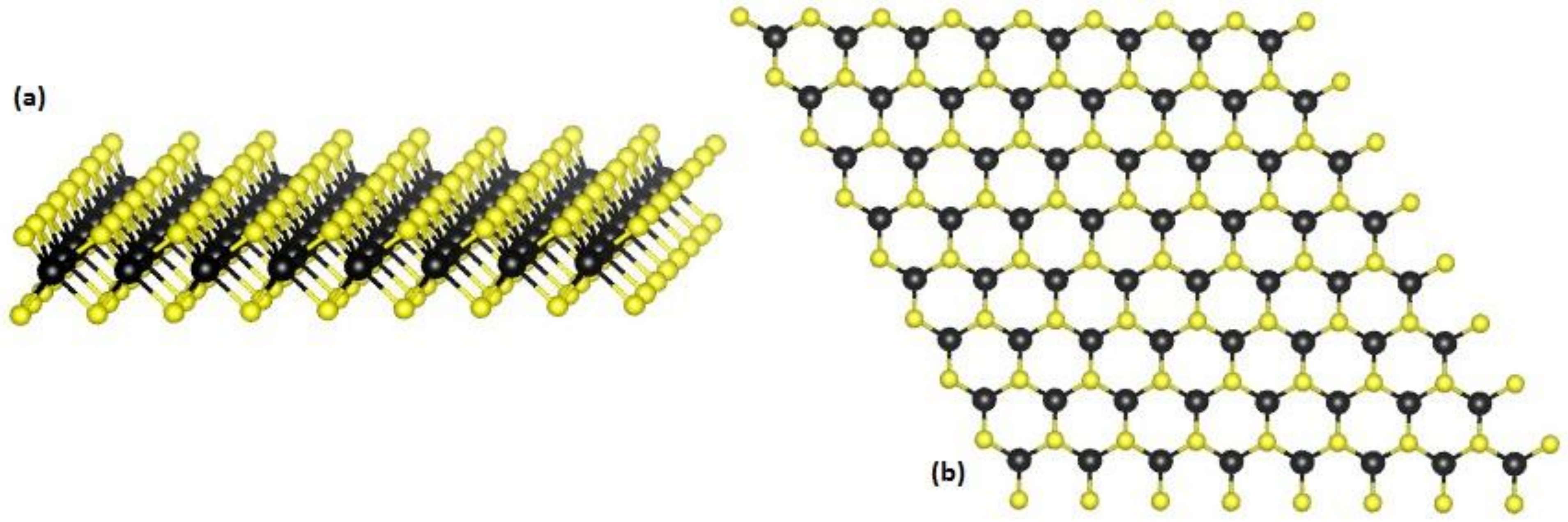
Classical Approaches to Quantum Simulation

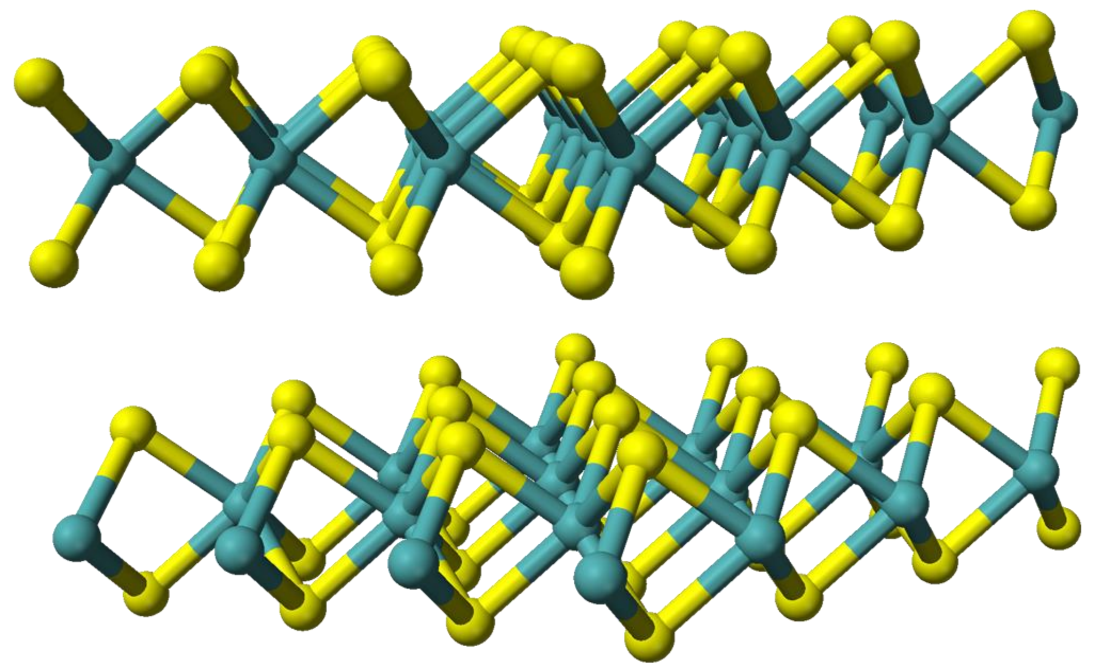
Most classical approaches to quantum simulation involve tradeoffs





So we make useful approximations!

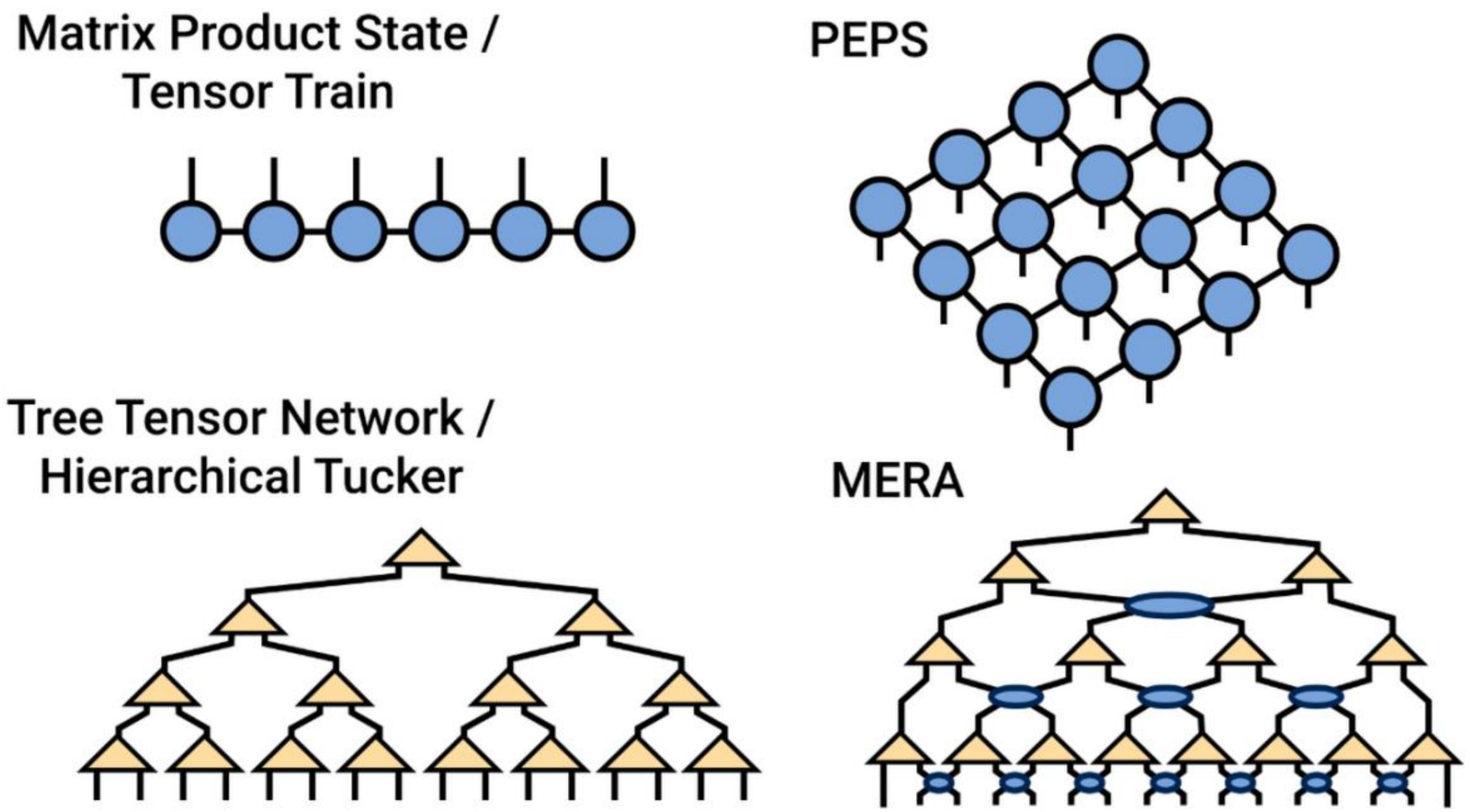


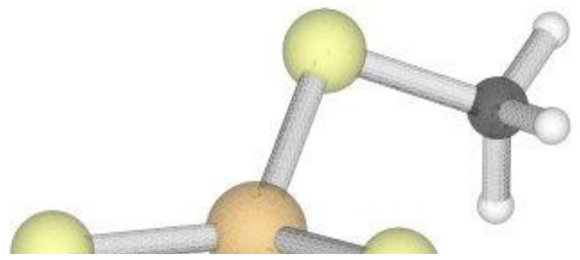


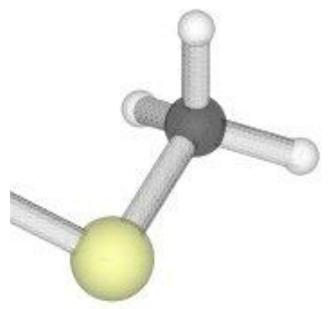
MoS2

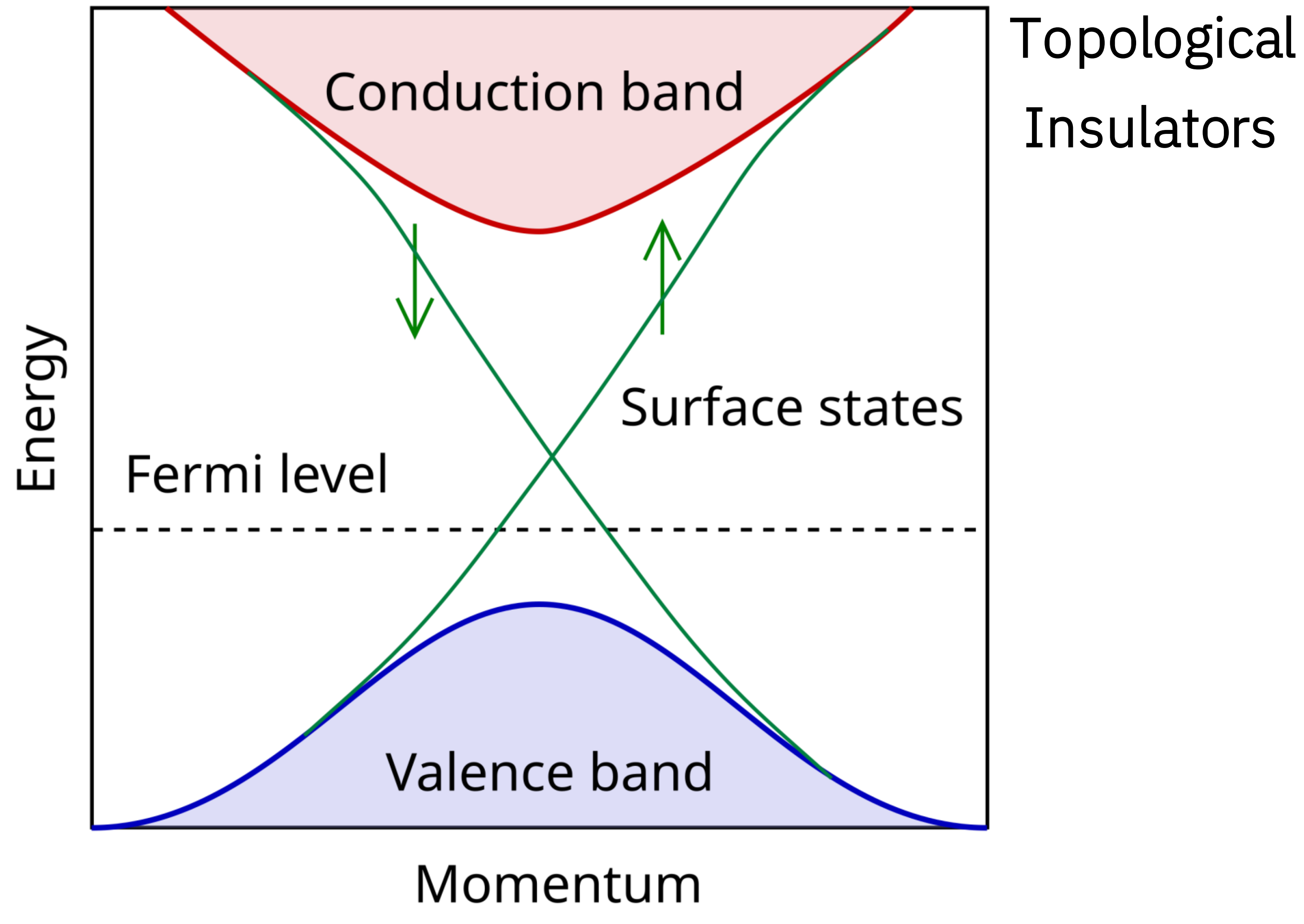
Classical Approaches to Quantum Simulation

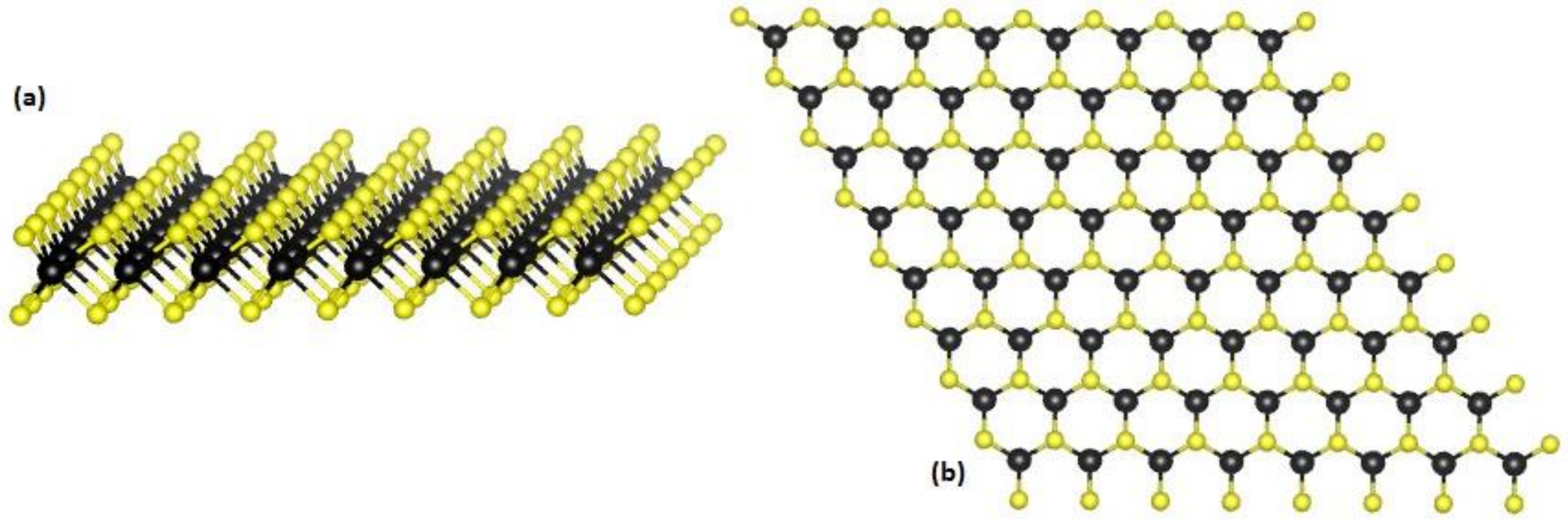
Tensor networks

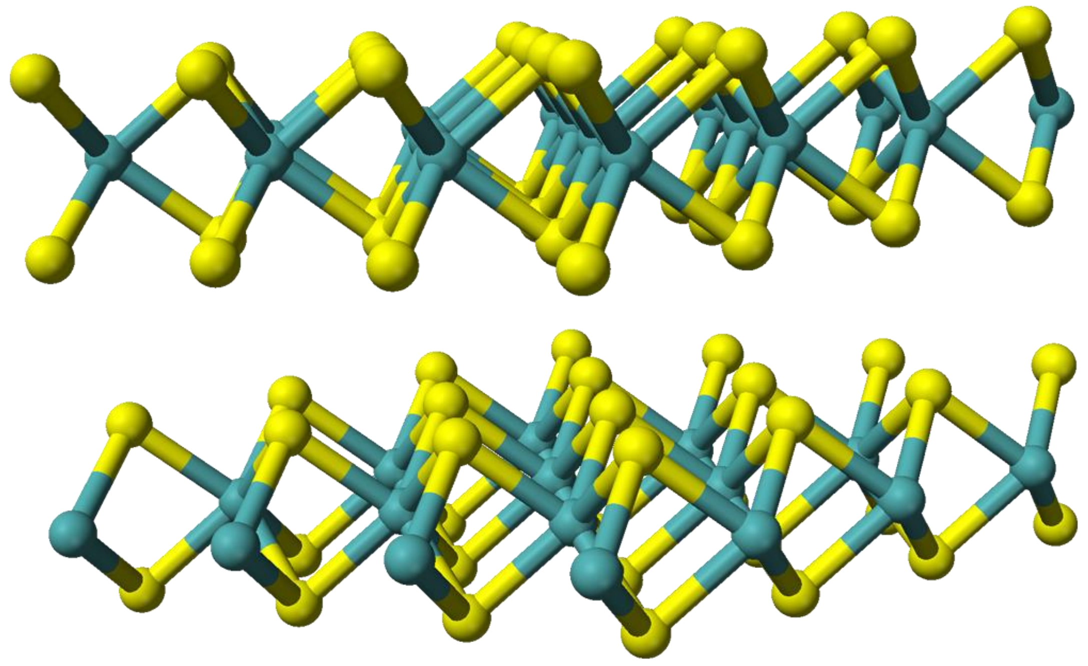








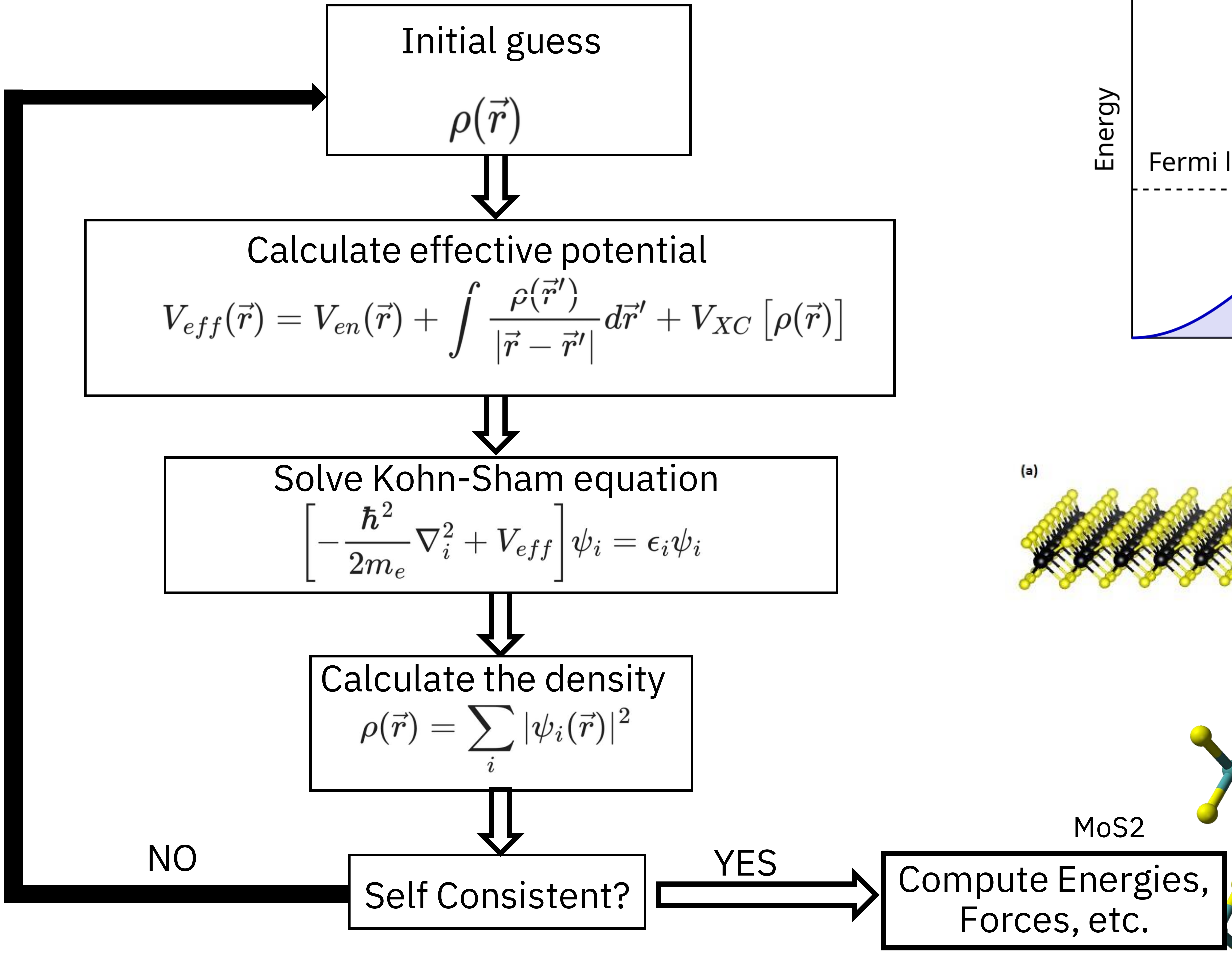




MoS2

Classical Approaches to Quantum Simulation

Density Functional Theory



Initial guess

$$\rho(\vec{r})$$

Calculate effective potential

$$V_{eff}(\vec{r}) = V_{en}(\vec{r}) + \int \frac{\rho(\vec{r}')}{|\vec{r}-\vec{r}'|} d\vec{r}' + V_{XC}\left[\rho(\vec{r})\right]$$

Solve Kohn-Sham equation

$$\left[-\frac{\hbar^2}{2m_e}\nabla_i^2 + V_{eff}\right]\psi_i = \epsilon_i\psi_i$$

Calculate the density

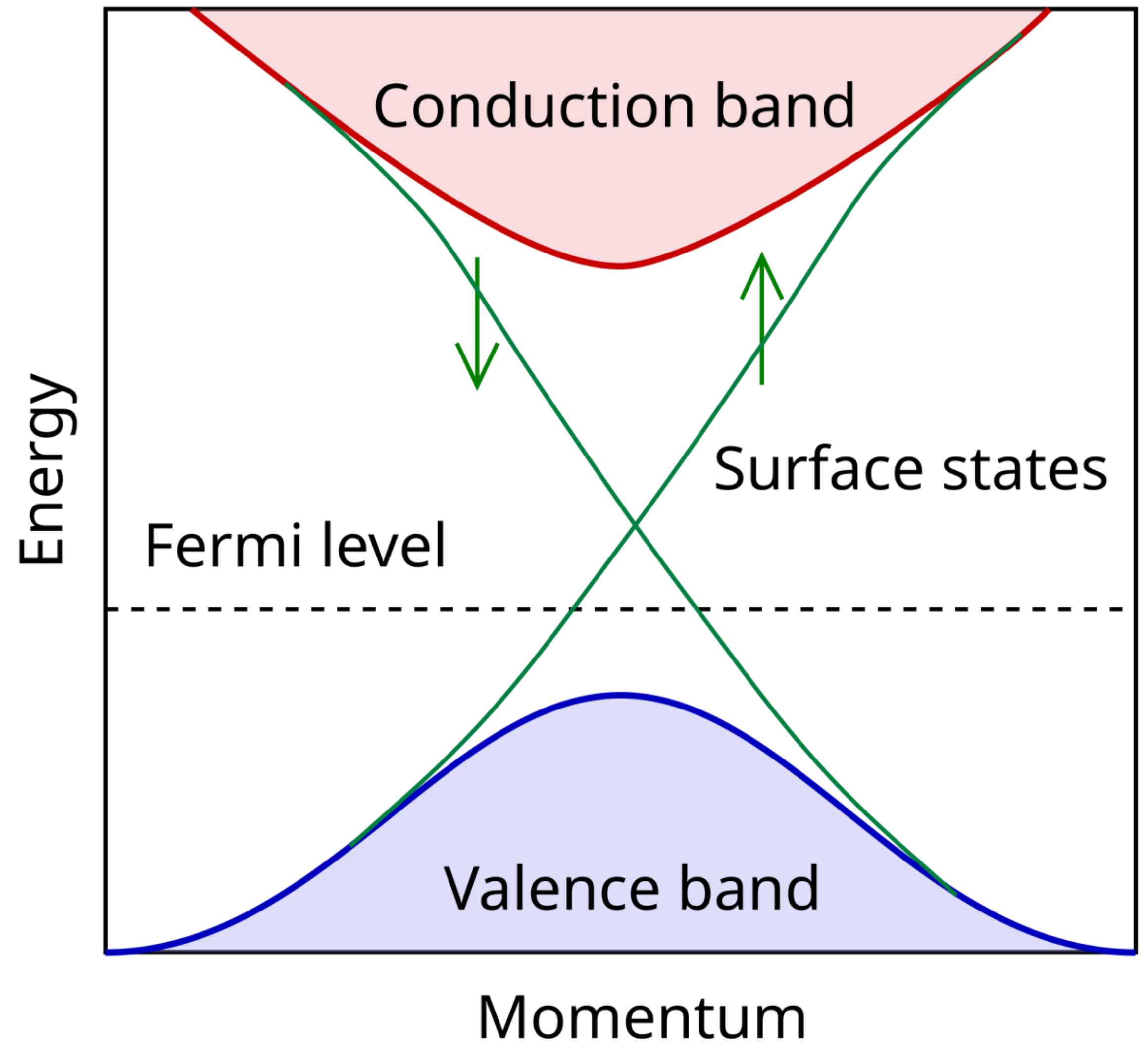
$$\rho(\vec{r}) = \sum_i |\psi_i(\vec{r})|^2$$

Self Consistent?

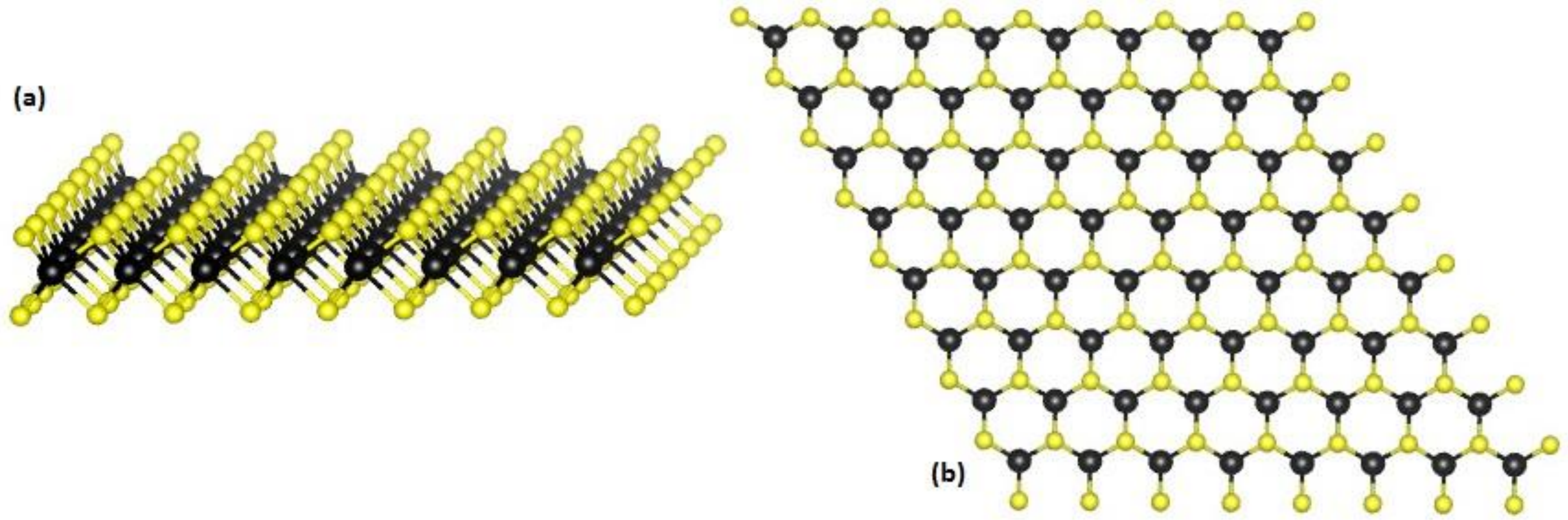
NO

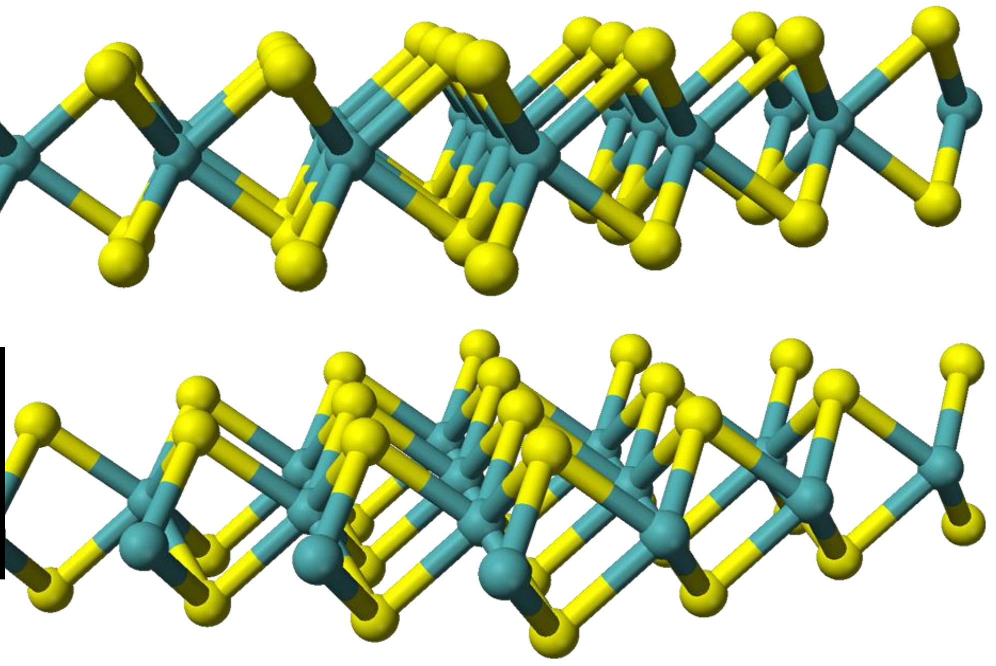
YES

Compute Energies, Forces, etc.



Topological Insulators





MoS2

IBM **Quantum**

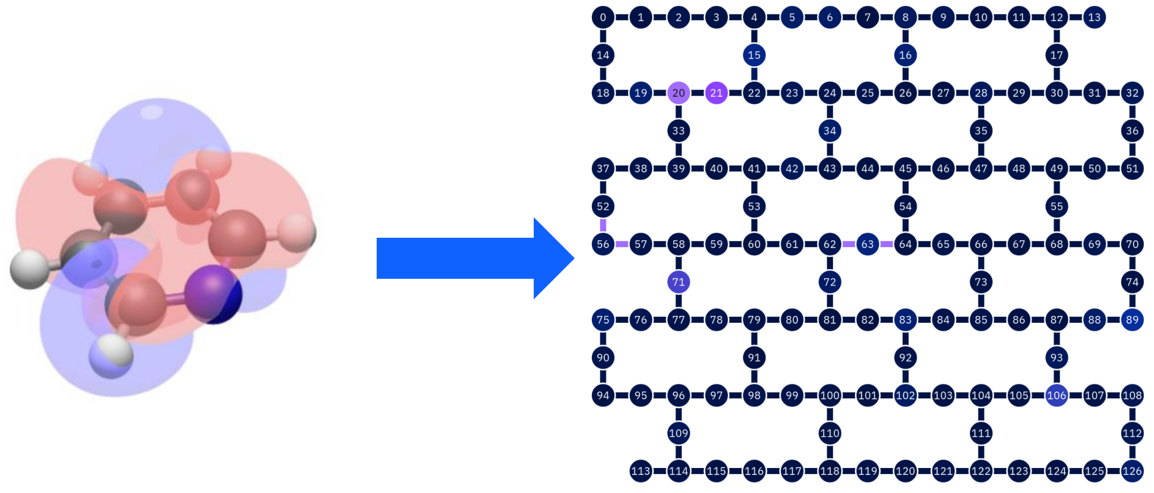
Simulating Quantum Systems

Anatomy of a Quantum Simulation Algorithm

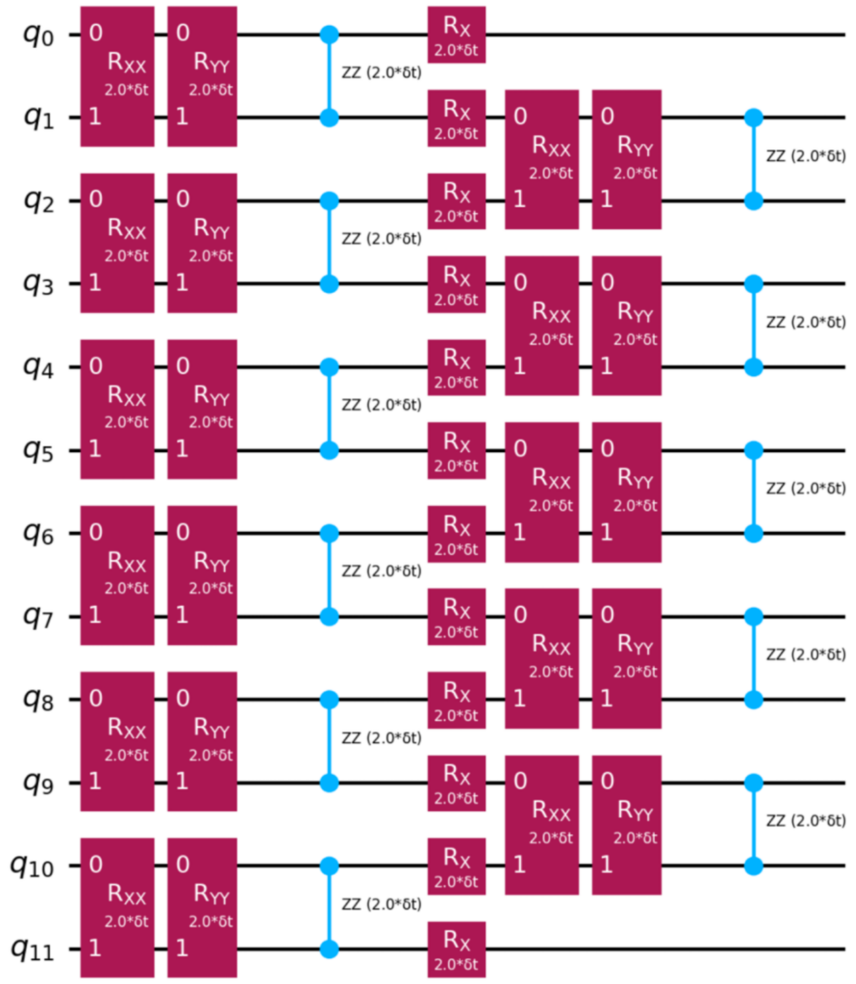
Choose a quantum system to simulate/characterize. Determine attributes such as the Hamiltonian and any important observables to measure.

$$H = \sum_i H_i$$

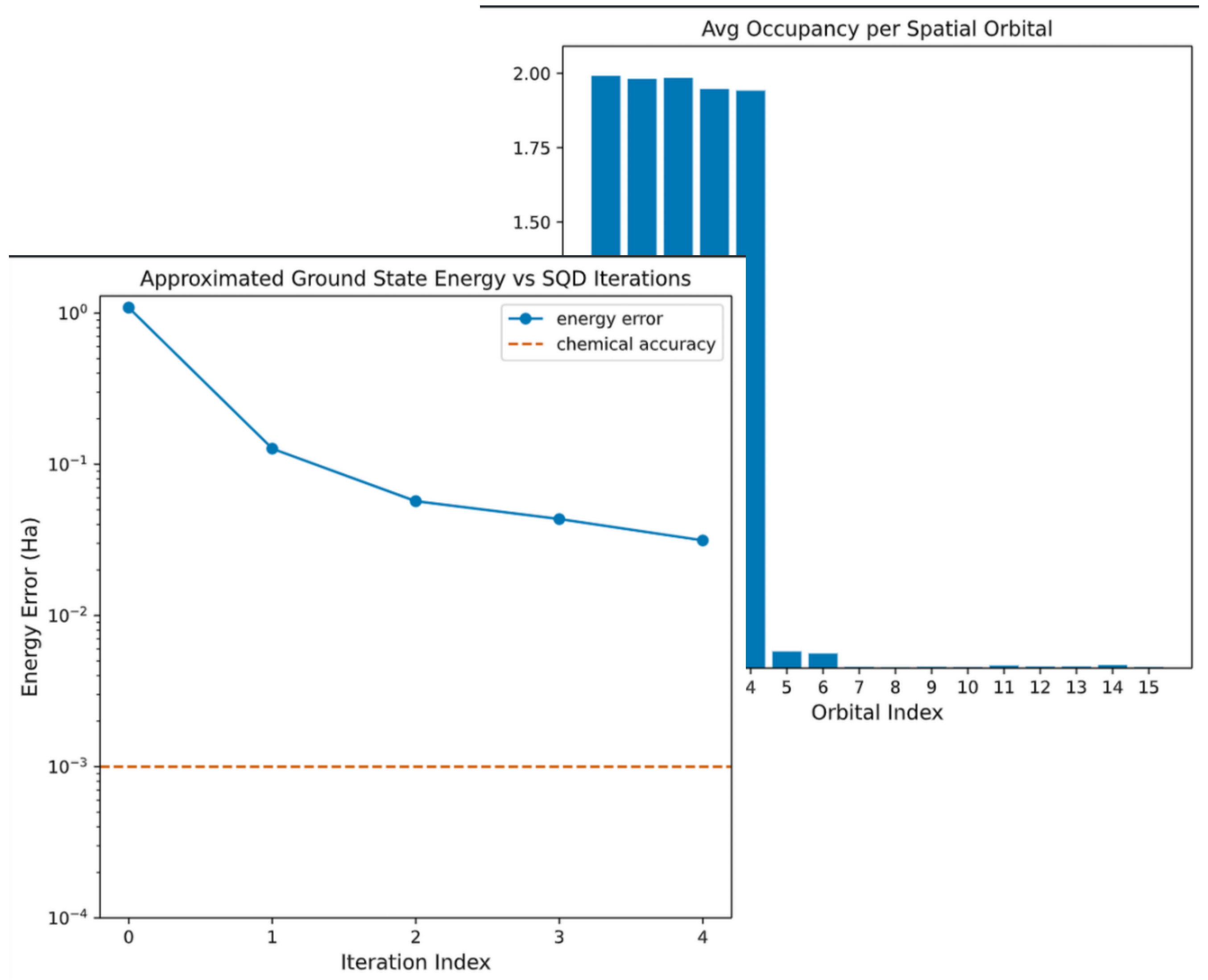
Map the Hamiltonian, and the chosen set of basis vectors to a quantum computing environment. This includes rewriting the Hamiltonian in terms of Pauli operators, as well as mapping the states to the computational basis.



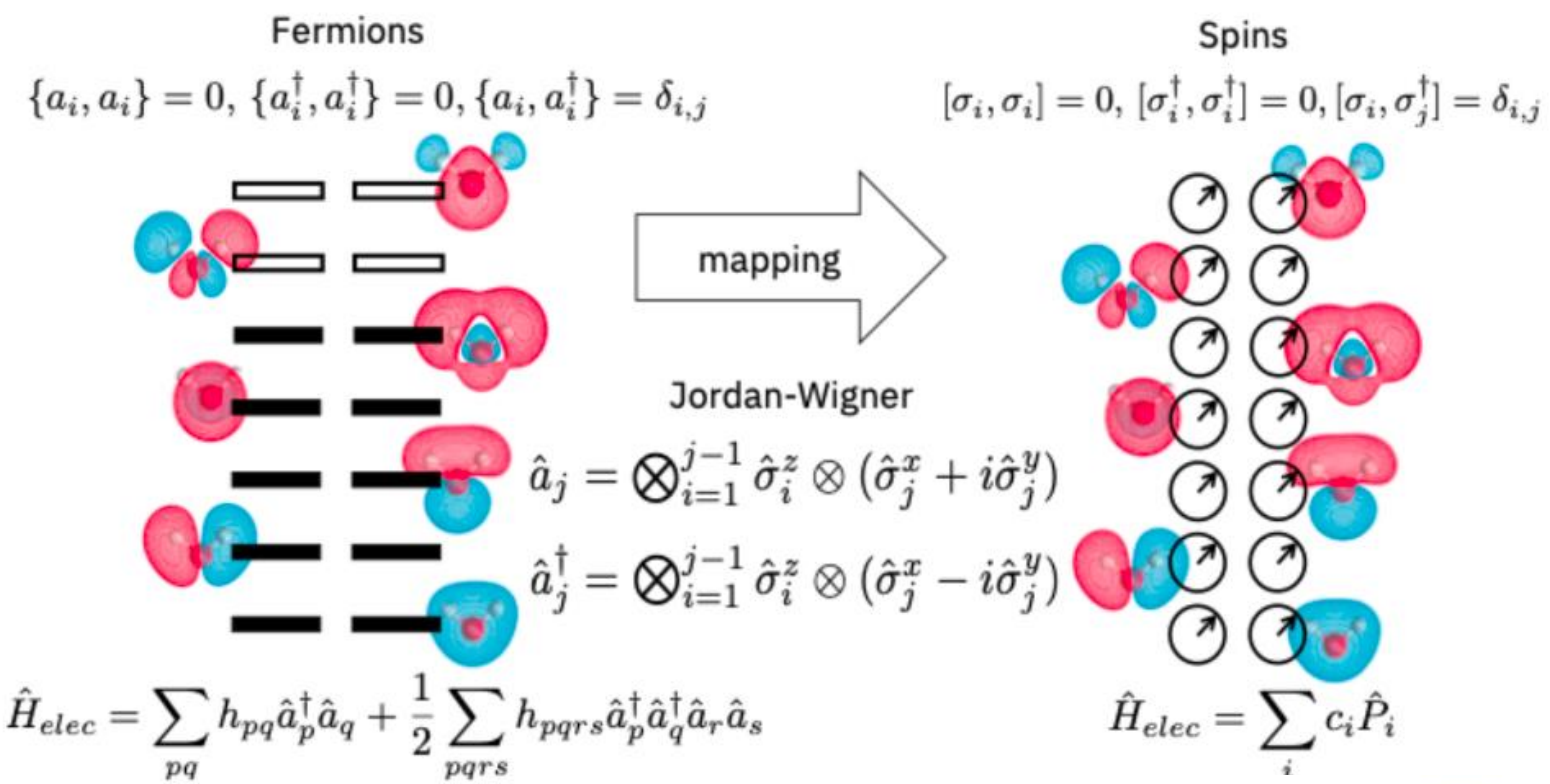
Prepare the quantum circuits which will be used for the simulation. This might include choosing an ansatz or creating a time evolution circuit.



Once the experiments have completed, run any necessary post-processing for error mitigation or analysis or raw data.



Mapping a problem to QPUs

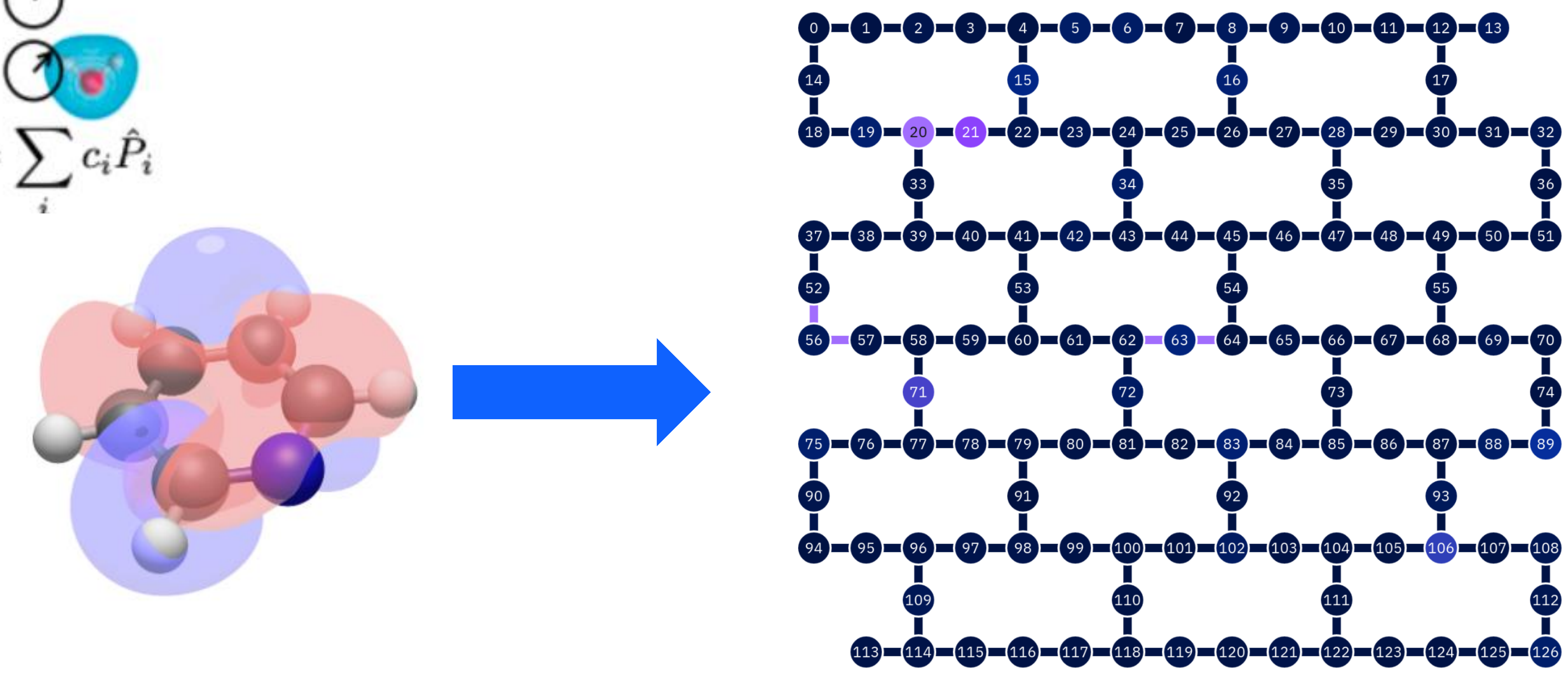

Fermions
$\{a_i, a_i\} = 0, \{a_i^\dagger, a_i^\dagger\} = 0, \{a_i, a_i^\dagger\} = \delta_{i,j}$
Spins
$[\sigma_i, \sigma_i] = 0, [\sigma_i^\dagger, \sigma_i^\dagger] = 0, [\sigma_i, \sigma_j^\dagger] = \delta_{i,j}$
mapping
Jordan-Wigner
$\hat{a}_j = \bigotimes_{i=1}^{j-1} \hat{\sigma}_i^z \otimes (\hat{\sigma}_j^x + i\hat{\sigma}_j^y)$
$\hat{a}_j^\dagger = \bigotimes_{i=1}^{j-1} \hat{\sigma}_i^z \otimes (\hat{\sigma}_j^x - i\hat{\sigma}_j^y)$
$\hat{H}_{elec} = \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s$
$\hat{H}_{elec} = \sum_i c_i \hat{P}_i$

Carefully consider the arrangement of fermionic modes

$$\hat{a}_j = \bigotimes_{i=1}^{j-1} \hat{\sigma}_i^z \otimes (\hat{\sigma}_j^x + i\hat{\sigma}_j^y)$$

$$\hat{a}_j^\dagger = \bigotimes_{i=1}^{j-1} \hat{\sigma}_i^z \otimes (\hat{\sigma}_j^x - i\hat{\sigma}_j^y)$$

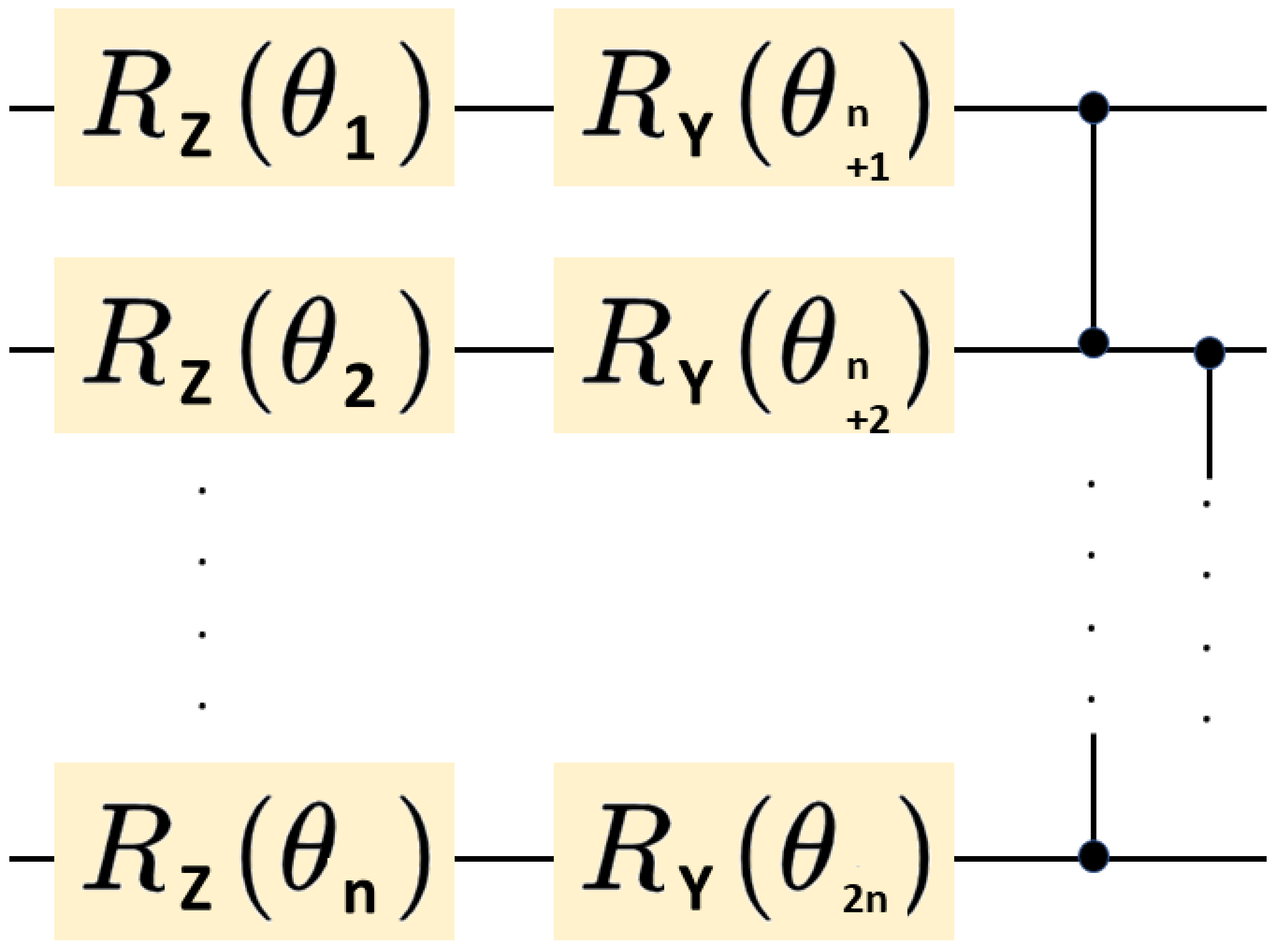
Most common is the Jordan-Wigner transform, but there are many others.

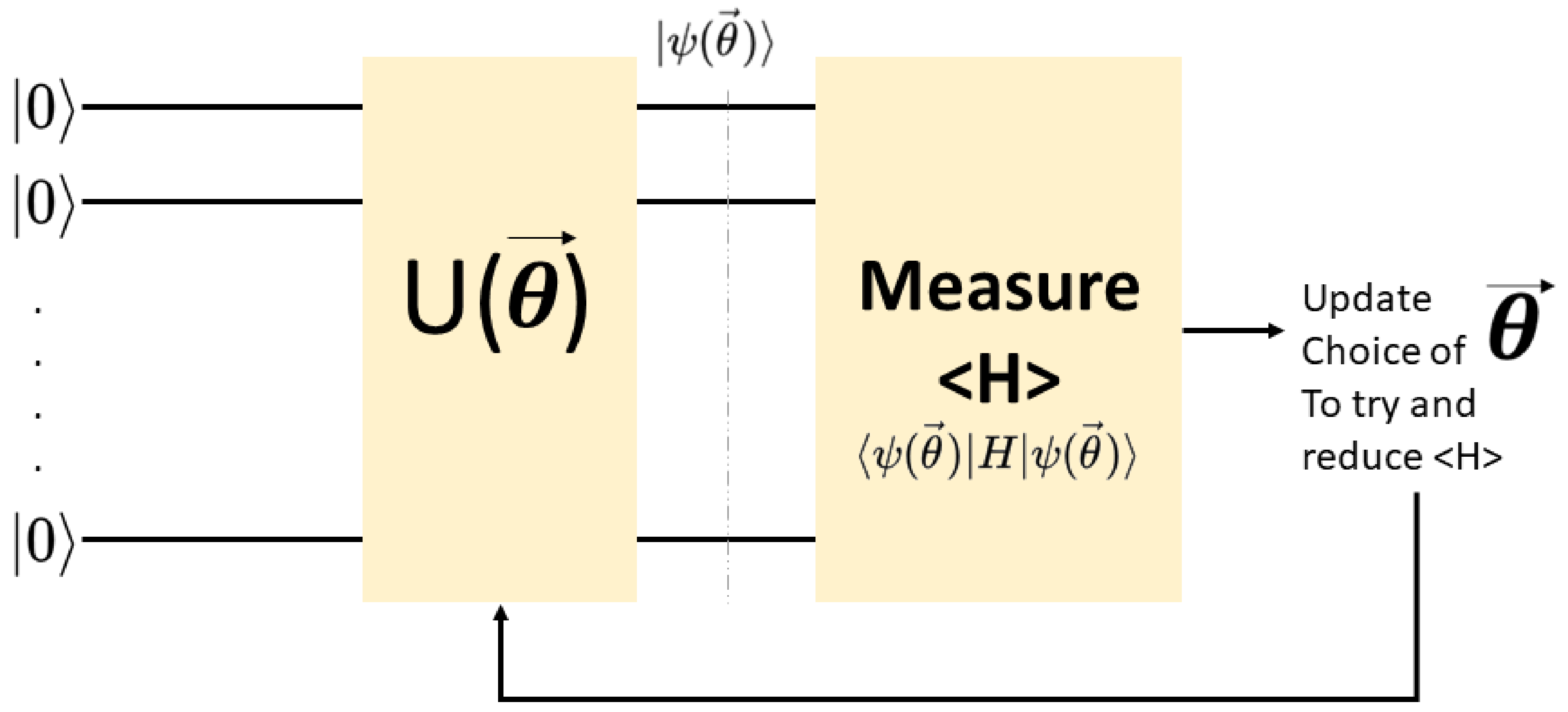


Simulating Quantum Systems

Variational Quantum Eigensolver (VQE)

$$U(\theta)|\psi_0\rangle = |\psi(\theta)\rangle$$



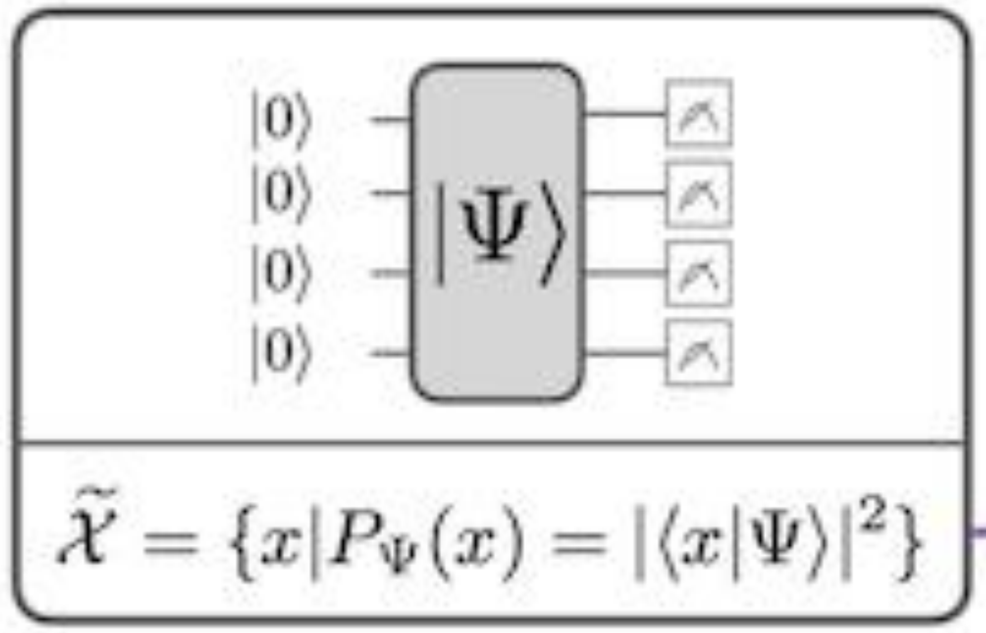


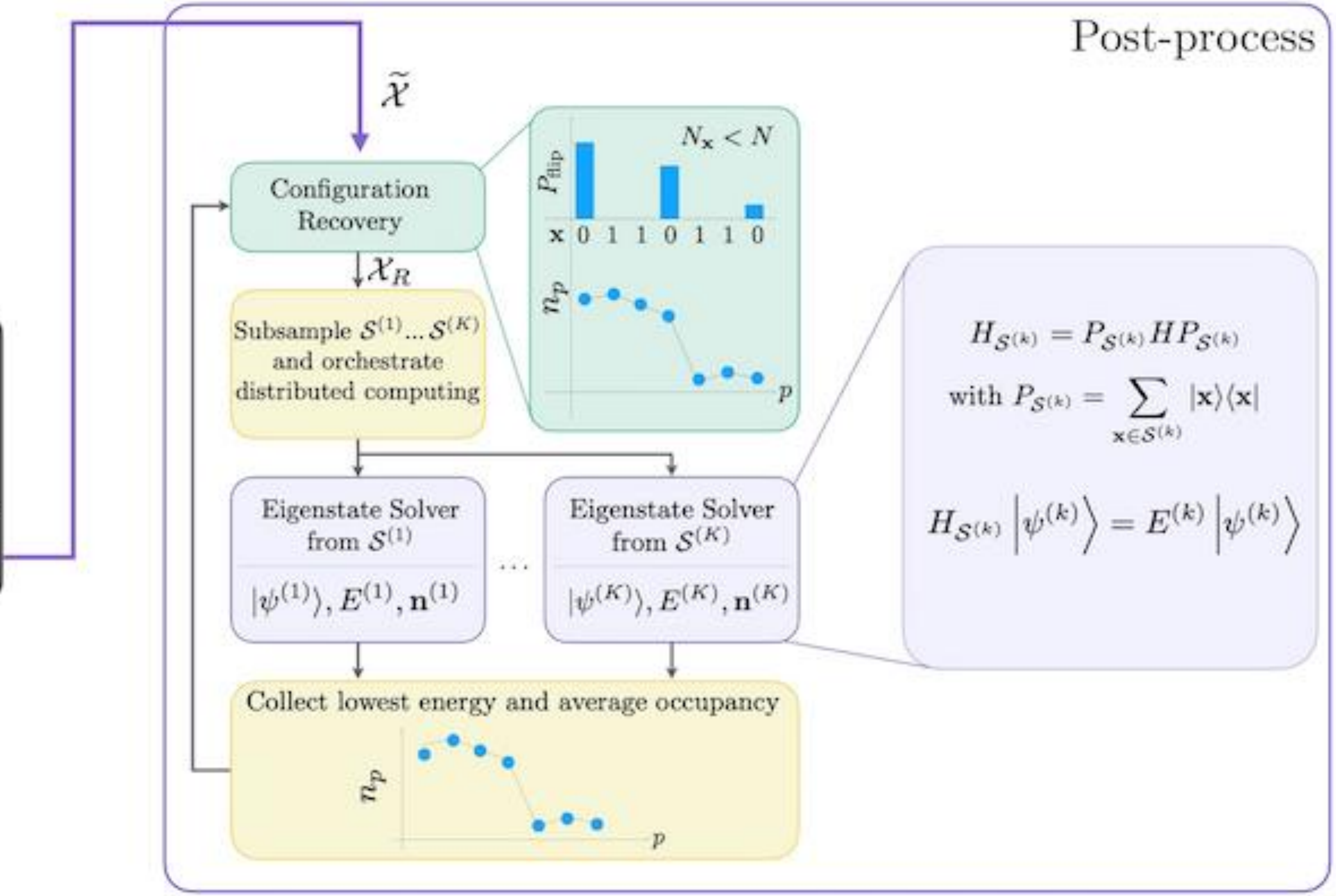
$|\psi(\vec{\theta})\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$U(\vec{\theta})$
Measure <H>
$\langle\psi(\vec{\theta})|H|\psi(\vec{\theta})\rangle$
Update Choice of $\vec{\theta}$ To try and reduce <H>

$$\lambda_{min} \leq \lambda_\theta \equiv \langle\psi(\theta)|H|\psi(\theta)\rangle$$

Simulating Quantum Systems

Sample-based Quantum Diagonalization


$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|\Psi\rangle$
$\widetilde{\mathcal{X}} = \{x | P_\Psi(x) = |\langle x|\Psi\rangle|^2\}$


Post-process
$\widetilde{\mathcal{X}}$
Configuration Recovery
$\mathcal{X}_R$
P_{flip}
$N_{\mathbf{x}} < N$
$\mathbf{x}$ 0 1 1 0 1 1 0
n_p
p
Subsample $\mathcal{S}^{(1)} \ldots \mathcal{S}^{(K)}$ and orchestrate distributed computing
Eigenstate Solver from $\mathcal{S}^{(1)}$
$|\psi^{(1)}\rangle, E^{(1)}, \mathbf{n}^{(1)}$
...
Eigenstate Solver from $\mathcal{S}^{(K)}$
$|\psi^{(K)}\rangle, E^{(K)}, \mathbf{n}^{(K)}$
$H_{\mathcal{S}^{(k)}} = P_{\mathcal{S}^{(k)}} H P_{\mathcal{S}^{(k)}}$
with $P_{\mathcal{S}^{(k)}} = \sum_{\mathbf{x} \in \mathcal{S}^{(k)}} |\mathbf{x}\rangle\langle\mathbf{x}|$
$H_{\mathcal{S}^{(k)}} \left|\psi^{(k)}\right\rangle = E^{(k)} \left|\psi^{(k)}\right\rangle$
Collect lowest energy and average occupancy
n_p
p

Simulating Quantum Systems

Approximating time evolution

Goal is to implement:

$$U(t) = e^{-iHt}$$

$$H = \sum_i H_i$$

However, things are complicated if not all terms in the Hamiltonian commute:

$$e^{-iHt} \neq \prod_i e^{-iH_i t}$$

$$\lim_{n\to\infty} \left(e^{iAt/n} e^{iBt/n}\right)^n = e^{i(A+B)t}$$

Trotter approximation

$$e^{i(A+B)\Delta t} = e^{iA\Delta t} e^{iB\Delta t} + \mathcal{O}(\Delta t^2)$$

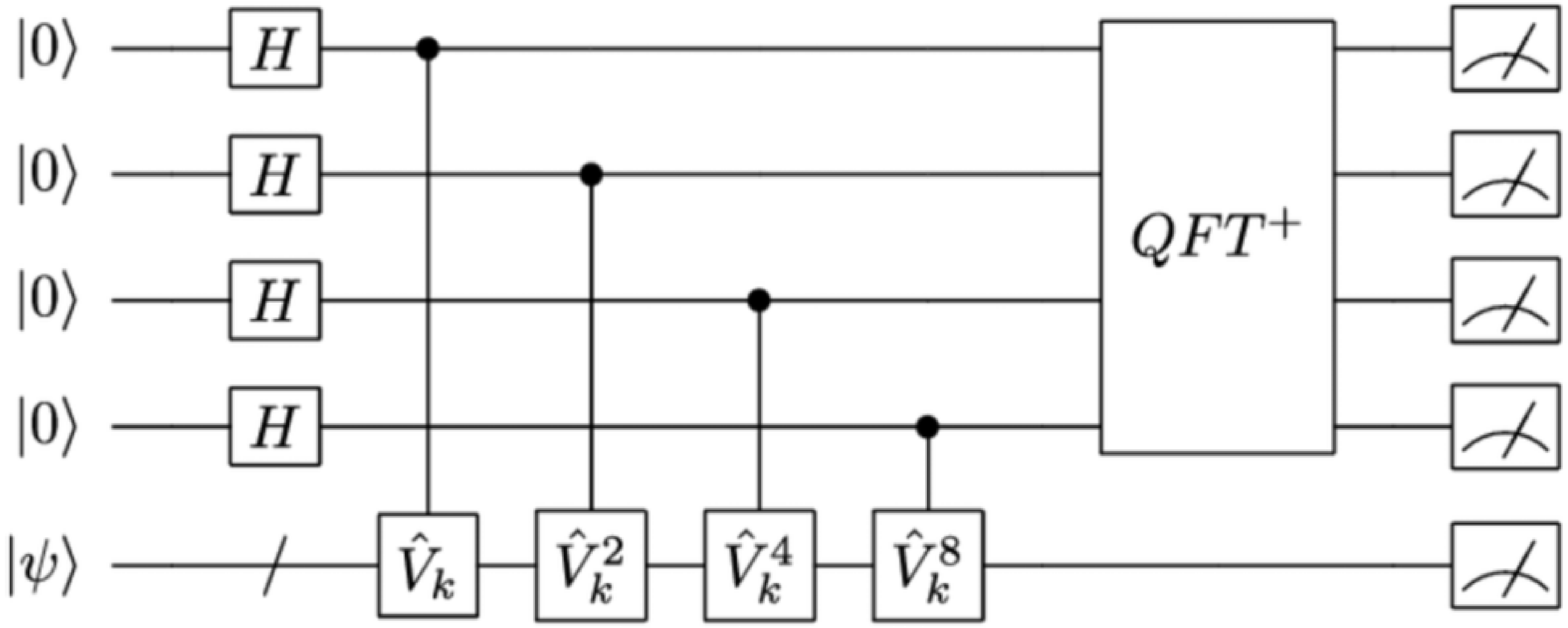
Trotter-Suzuki approximation

$$e^{i(A+B)\Delta t} = e^{iA\Delta t/2} e^{iB\Delta t} e^{iA\Delta t/2} + \mathcal{O}(\Delta t^3)$$

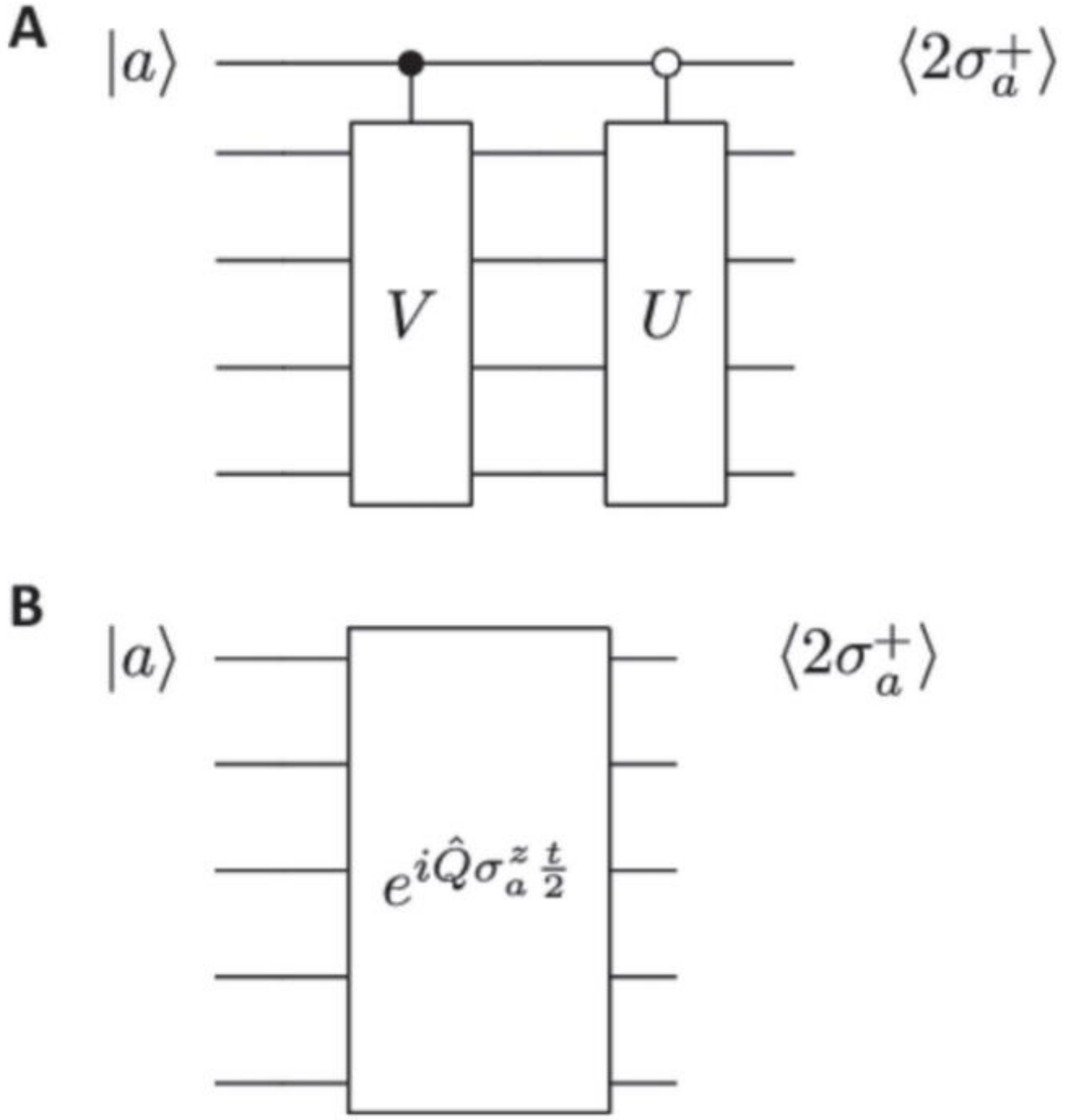
Simulating Quantum Systems

Phase Estimation of observables

Several different ways of doing this:
(QPE, iterative QPE with phase kickback, etc.)



Aspuru-Guzik, A. et. Al. *Science 309, 1704–1707*

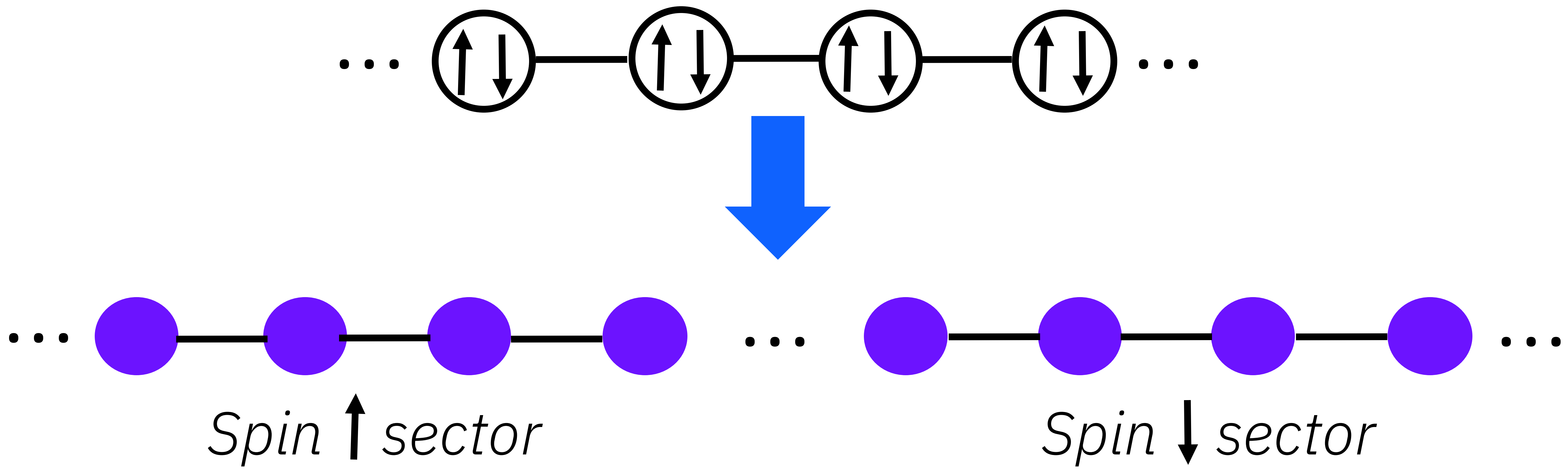


A
$|a\rangle$
$\langle 2\sigma_a^+\rangle$
V
U
B
$|a\rangle$
$\langle 2\sigma_a^+\rangle$
$e^{i\hat{Q}\sigma_a^z\frac{t}{2}}$

Practical Example

Practical Example

Time evolution of 1D Hubbard model



$$H = -t \sum_{i,\sigma} \left(c^{\dagger}_{i,\sigma} c_{i+1,\sigma} + c^{\dagger}_{i+1,\sigma} c_{i,\sigma} \right) + U \sum_{i} n_{i\uparrow} n_{i\downarrow}$$

"Hopping" energy

Repulsion energy

$$n_{i,\sigma} = c^{\dagger}_{i,\sigma} c_{i,\sigma}$$

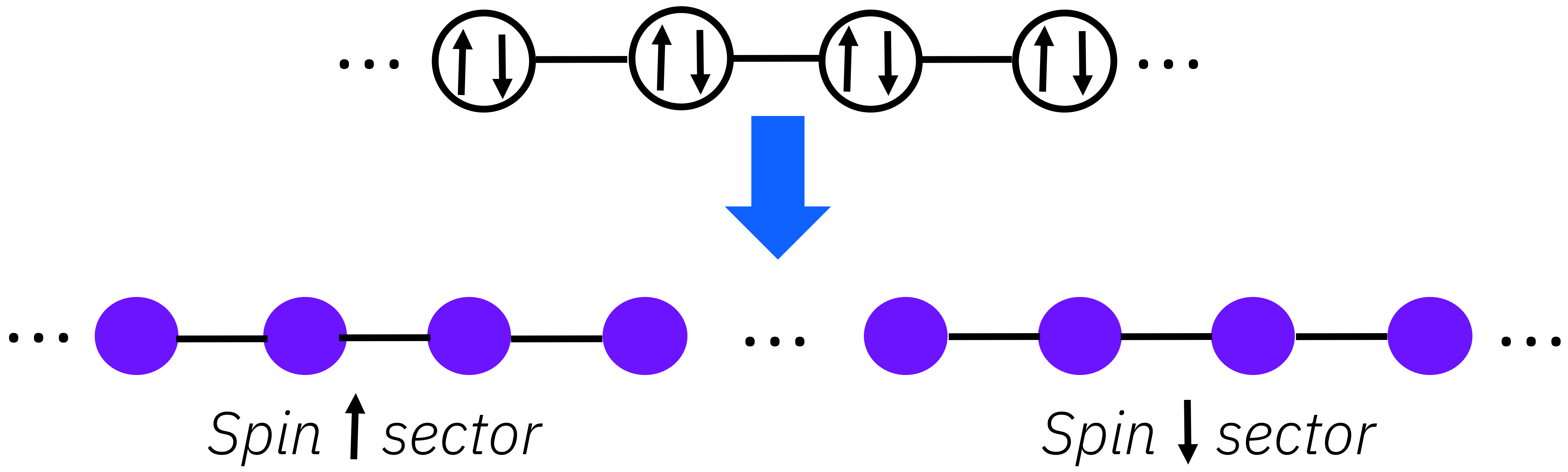
Number operator

Jordan Wigner Transform

$$c^{\dagger}_{i} = Z_0 \ldots Z_{i-1} \left(\frac{X_i - iY_i}{2} \right) I_{i+1} \ldots I_N$$

$$c_{i} = Z_0 \ldots Z_{i-1} \left(\frac{X_i + iY_i}{2} \right) I_{i+1} \ldots I_N$$

Practical Example



Time evolution of 1D Hubbard model

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

$$\boxed{c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i = (X_i X_{i+1} + Y_i Y_{i+1})}$$

$$\boxed{n_{i\uparrow} n_{i\downarrow} = c_{i,\uparrow}^\dagger c_{i,\downarrow} = \left(I_i^\uparrow - Z_i^\uparrow \right) \left(I_i^\downarrow - Z_i^\downarrow \right)}$$

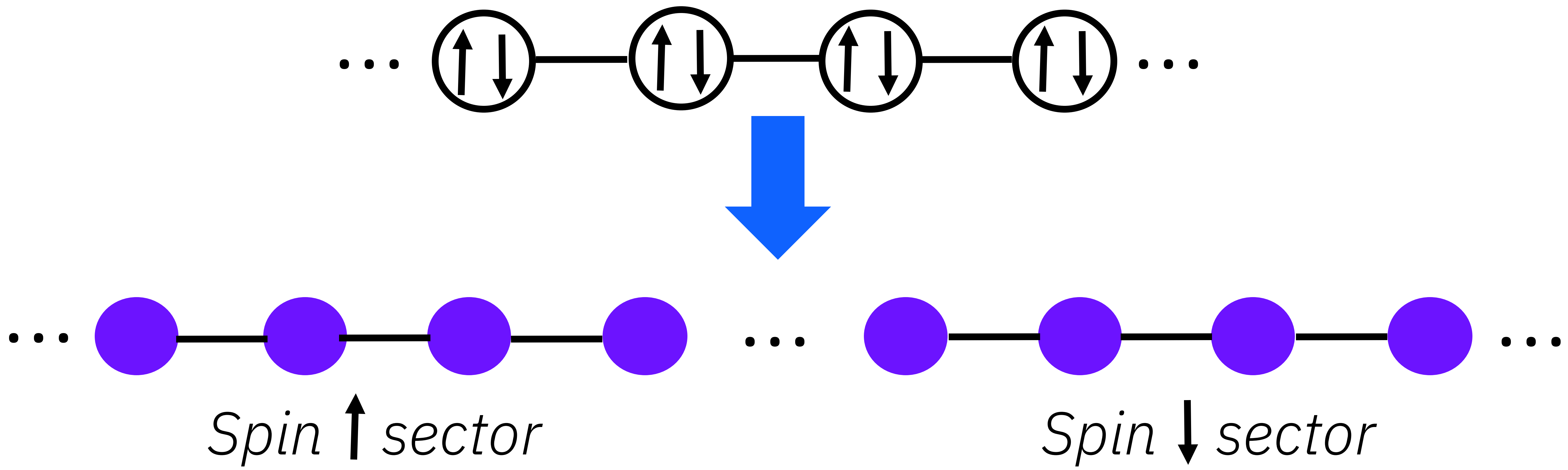
Jordan Wigner Transform

$$c_i^\dagger = Z_0 \ldots Z_{i-1} \left(\frac{X_i - iY_i}{2} \right) I_{i+1} \ldots I_N$$

$$c_i = Z_0 \ldots Z_{i-1} \left(\frac{X_i + iY_i}{2} \right) I_{i+1} \ldots I_N$$

Practical Example

Time evolution of 1D Hubbard model



$$H = -t\sum_{i,\sigma}\left(c^\dagger_{i,\sigma}c_{i+1,\sigma} + c^\dagger_{i+1,\sigma}c_{i,\sigma}\right) + U\sum n_{i\uparrow}n_{i\downarrow}$$

$$H = -t\sum_i\left(X_iX_{i+1} + Y_iY_{i+1}\right) + U\sum_i\left(I_i^\uparrow - Z_i^\uparrow\right)\left(I_i^\downarrow - Z_i^\downarrow\right)$$

$$c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i = \left(X_iX_{i+1} + Y_iY_{i+1}\right)$$

$$n_{i\uparrow}n_{i\downarrow} = c^\dagger_{i,\uparrow}c_{i,\downarrow} = \left(I_i^\uparrow - Z_i^\uparrow\right)\left(I_i^\downarrow - Z_i^\downarrow\right)$$

$$c_i^\dagger = Z_0 \ldots Z_{i-1}\left(\frac{X_i - iY_i}{2}\right)I_{i+1}\ldots I_N$$

$$c_i = Z_0 \ldots Z_{i-1}\left(\frac{X_i + iY_i}{2}\right)I_{i+1}\ldots I_N$$

Now let's code this up!