

Hamiltonian Systems on Quantum Computers: Spectra, Dynamics, and Limitations[†]

Emanuele Mendicelli

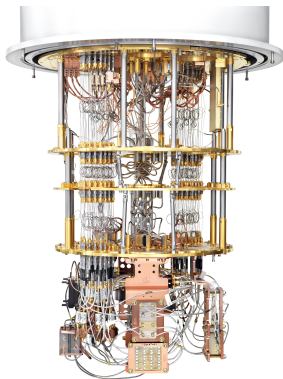
Department of Theoretical Physics, Jožef Stefan Institute, Ljubljana, Slovenia



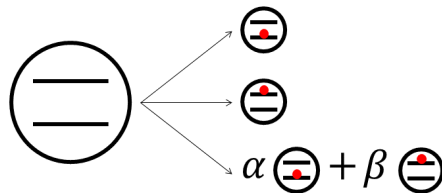
BRDA 2026: Challenges in particle physics and cosmology, Medana, October 16th

[†]Based on [[arXiv:2103.08661](https://arxiv.org/abs/2103.08661)], [[arXiv:2205.09247](https://arxiv.org/abs/2205.09247)], [[arXiv:2510.26506](https://arxiv.org/abs/2510.26506)] and [[arXiv:2505.02553](https://arxiv.org/abs/2505.02553)]

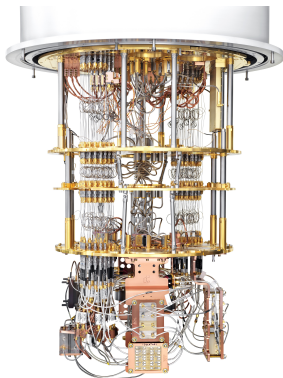
Quantum Hardware and Qubits



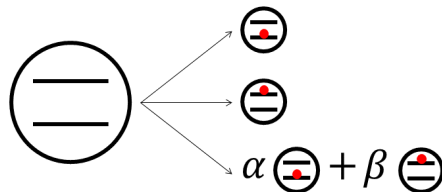
- Quantum superposition
- Entanglement



Quantum Hardware and Qubits



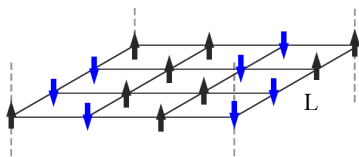
- Quantum superposition
- Entanglement



- Thanks to qubits the Hamiltonian formalism can be computationally feasible!

Why do we like qubits? Classical Vs quantum resources

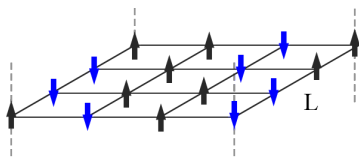
Let's consider a simple 3D spin model system with N spin-1/2



Hamiltonian $2^N \times 2^N$
(Hilbert-space grows exponentially!)

Why do we like qubits? Classical Vs quantum resources

Let's consider a simple 3D spin model system with N spin-1/2

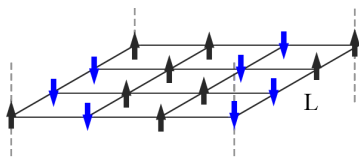


Hamiltonian $2^N \times 2^N$
(Hilbert-space grows exponentially!)

Classically we need 2^N resources $\longrightarrow$ N qubits

Why do we like qubits? Classical Vs quantum resources

Let's consider a simple 3D spin model system with N spin-1/2



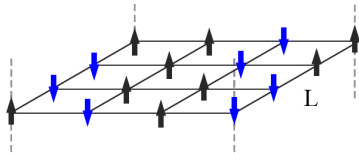
Hamiltonian $2^N \times 2^N$
(Hilbert-space grows exponentially!)

Classically we need 2^N resources $\longrightarrow$ N qubits

Classical statistical methods on importance-sampling Monte Carlo methods suffer from the **sign problem**:

Why do we like qubits? Classical Vs quantum resources

Let's consider a simple 3D spin model system with N spin-1/2



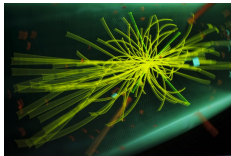
Hamiltonian $2^N \times 2^N$
(Hilbert-space grows exponentially!)

Classically we need 2^N resources $\longrightarrow$ N qubits

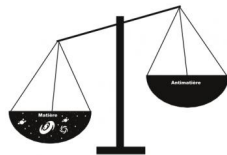
Classical statistical methods on importance-sampling Monte Carlo methods suffer from the **sign problem**:

Real-time evolution

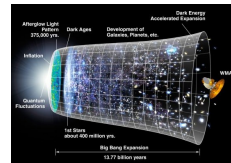
nonzero chemical potential



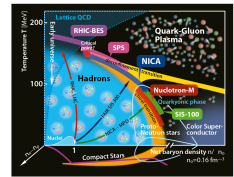
propagations/collisions



Imbalance matter-antimatter



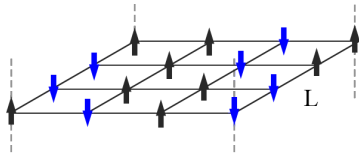
Early Universe



QCD phase diagram

Why do we like qubits? Classical Vs quantum resources

Let's consider a simple 3D spin model system with N spin-1/2



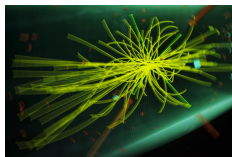
Hamiltonian $2^N \times 2^N$
(Hilbert-space grows exponentially!)

Classically we need 2^N resources $\longrightarrow$ N qubits

Classical statistical methods on importance-sampling Monte Carlo methods suffer from the **sign problem**:

Real-time evolution

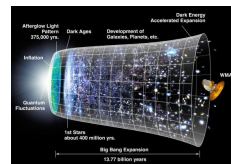
nonzero chemical potential



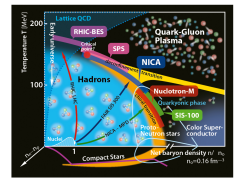
propagations/collisions



Imbalance matter-antimatter



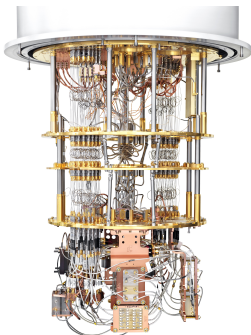
Early Universe



QCD phase diagram

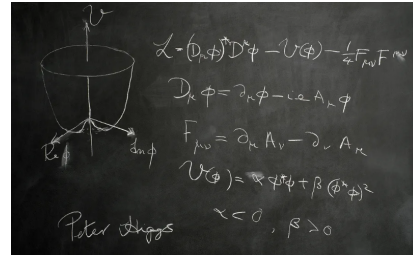
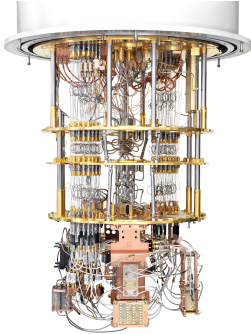
- Hamiltonian simulations/calculations on quantum computer are free of the sign problem!

Challenges in using quantum computers: Hardware & Formalism



- Few qubits
- Low qubit connectivity
- Noisy gates
- No error correction $\Rightarrow$ **Error mitigation!**

Challenges in using quantum computers: Hardware & Formalism



- Few qubits
- Low qubit connectivity
- Noisy gates
- No error correction $\Rightarrow$ **Error mitigation!**

- Hamiltonian formulation
- Representation of the Hamiltonian
- Scalability of the continuum limit
- Scalability of the thermodynamic limit

Encoding challenges

Presentation Plan

- Part I: $SU(2)$ pure gauge lattice theory on a quantum hardware
- Part II: D-Wave superconducting quantum annealer
- Part III: IBM superconducting gate-based universal quantum computer
- Part IV: Supersymmetric Quantum Mechanics on a quantum simulator
- Conclusions

Part I:
 $SU(2)$ Lattice Gauge Theory

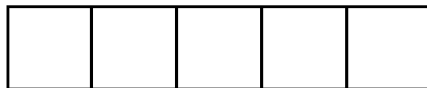
Hamiltonian for SU(2) pure gauge lattice theory

$$\mathcal{L} = -\frac{1}{4}F_{\mu\nu}(x)F^{\mu\nu}(x)$$

$\Downarrow$

$$\hat{H} = \frac{g^2}{2} \left(\sum_{i=\text{links}} \hat{E}_i^2 - 2x \sum_{i=\text{plaquettes}} \hat{\square}_i \right)$$

We used a single-row lattice with few plaquettes:

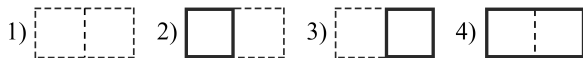


2 and 5 plaquettes with closed boundary conditions.

- g is the gauge coupling and $x \equiv 2/g^4$.
- $\hat{E}_i^2$ is the chromoelectric field for the i th lattice link.
(Returns the chromoelectric energy stored on the lattice)
- $\hat{\square}_i$ is the plaquette operator trace of the product of four gauge link operators of the i th plaquette.
(Adds or subtracts energy flux on the i th plaquette)

States of the theory on 2 plaquettes

- The physical states can be obtained by applying the plaquette operators on the vacuum, state 1), and by respecting the SU(2) Gauss's law at each node:
- Fixing the maximum energy flux to $j_{\max} = 1/2$, there are 4 states:



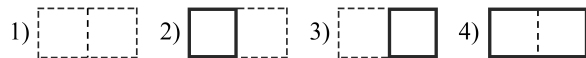
..... J=0

———— J=1/2

———— J=1

States of the theory on 2 plaquettes

- The physical states can be obtained by applying the plaquette operators on the vacuum, state 1), and by respecting the SU(2) Gauss's law at each node:
- Fixing the maximum energy flux to $j_{max} = 1/2$, there are 4 states:

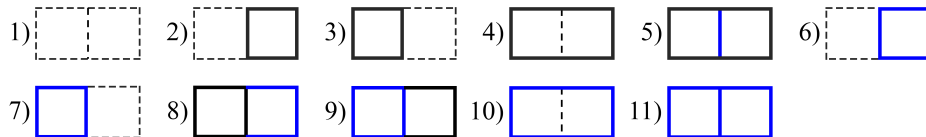


----- J=0

———— J=1/2

———— J=1

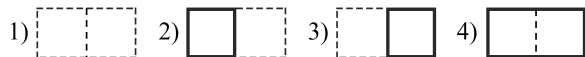
- While for $j_{max} = 1$, there are 11 states:



- The same process can be used for larger lattices and gauge truncation (j_{max}).

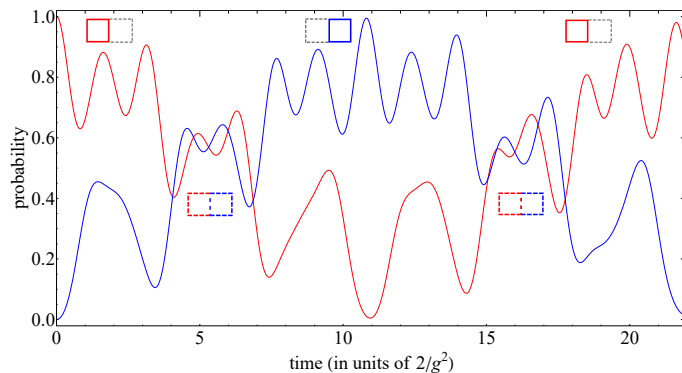
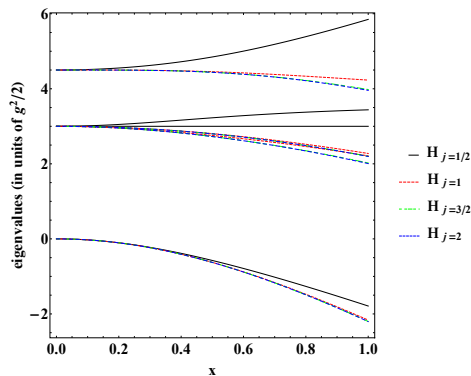
Representing the Hamiltonian for 2 plaquettes

- Let's use these states to represent the Hamiltonian:



$$\Rightarrow H = \frac{g^2}{2} \begin{pmatrix} 0 & -2x & -2x & 0 \\ -2x & 3 & 0 & -x \\ -2x & 0 & 3 & -x \\ 0 & -x & -x & \frac{9}{2} \end{pmatrix}$$

- The energy spectrum and the time evolution can be calculated on a classical computer:



- Can we do this simple kind of calculations on a quantum hardware?

Part II: D-Wave superconducting quantum annealer

How D-Wave works in a nutshell

The D-Wave hardware uses the Adiabatic theorem to find the ground state of any matrix that can be rewritten as a quadratic unconstrained binary optimization (QUBO):

$$H(q)_{QUBO} = \sum_{i=1}^N h_i q_i^z + \sum_{i=1}^N \sum_{j=i+1}^N J_{ij} q_i^z q_j^z$$

where q_i could be 0 or 1, while h_i and J_{ij} are real value coefficients chosen by the user.

How D-Wave works in a nutshell

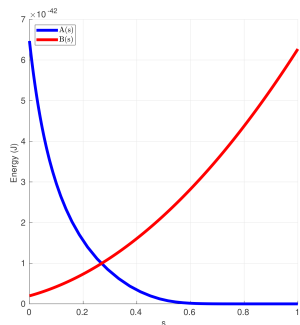
The D-Wave hardware uses the Adiabatic theorem to find the ground state of any matrix that can be rewritten as a quadratic unconstrained binary optimization (QUBO):

$$H(q)_{QUBO} = \sum_{i=1}^N h_i q_i^z + \sum_{i=1}^N \sum_{j=i+1}^N J_{ij} q_i^z q_j^z$$

where q_i could be 0 or 1, while h_i and J_{ij} are real value coefficients chosen by the user.

The Adiabatic theorem guarantees that: if the Hamiltonian is varied slowly enough, the ground state will stay close to the Hamiltonian instantaneous ground state at each time t .*

$$H(q, s) = \underbrace{A(s) \left[\sum_{i=1}^N h_i q_i^x \right]}_{\text{DW initial H}} + \underbrace{B(s) \left[\sum_{i=1}^N q_i^z + \sum_{i=1}^N \sum_{j=i+1}^N J_{ij} q_i^z q_j^z \right]}_{\text{User's entered Hamiltonian}}$$



*[J. Roland and N. J. Cerf, Phys. Rev. A 65, 042308, [doi: 10.1103/PhysRevA.65.042308](https://doi.org/10.1103/PhysRevA.65.042308)

How to use DW for a non Ising Hamiltonian?

Use the quantum annealer eigensolver (QAE)* that implements the variational method on DW and can find the ground state of any Hamiltonian using the following functional:

$$F \equiv \langle \psi | H | \psi \rangle - \lambda \langle \psi | \psi \rangle$$

λ is tuned by the user to find the minimum eigenvalue of H .

K qubits are used for the fixed-point representation of real numbers.

*[A. Teplukhin, B. K. Kendrick, and D. Babikov, Calculation of Molecular Vibrational Spectra on a Quantum Annealer, J. Chem. Theory and Comp. 15, 4555(2019), [doi: 10.1021/acs.jctc.9b00402](https://doi.org/10.1021/acs.jctc.9b00402)

How to use DW for a non Ising Hamiltonian?

Use the quantum annealer eigensolver (QAE)* that implements the variational method on DW and can find the ground state of any Hamiltonian using the following functional:

$$F \equiv \langle \psi | H | \psi \rangle - \lambda \langle \psi | \psi \rangle$$

λ is tuned by the user to find the minimum eigenvalue of H.

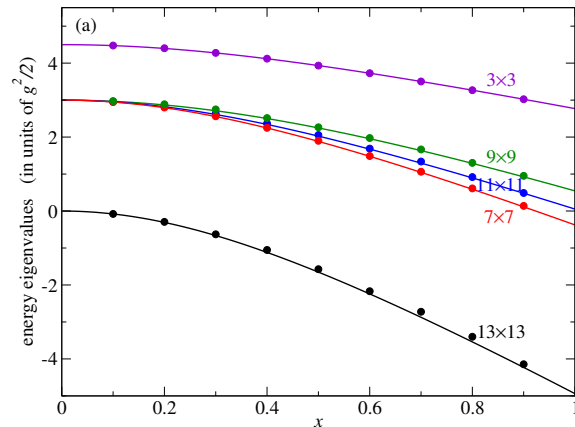
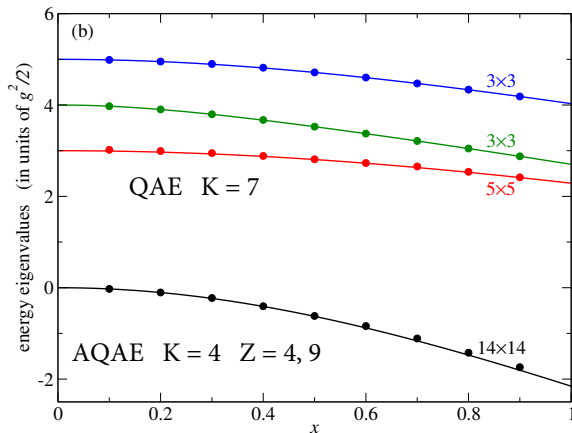
K qubits are used for the fixed-point representation of real numbers.

Our algorithm Adaptive QAE, for higher precision:

- ① Fix K & find λ_{opt}
- ② Tune the Chain Strength.
- ③ Accept or repeat with a larger K by using the last-step minimum as input.

*[A. Teplukhin, B. K. Kendrick, and D. Babikov, Calculation of Molecular Vibrational Spectra on a Quantum Annealer, J. Chem. Theory and Comp. 15, 4555(2019), [doi: 10.1021/acs.jctc.9b00402](https://doi.org/10.1021/acs.jctc.9b00402)

Several eigenvalues of a 2 and 6 plaquettes lattice



• 2 plaquettes $j_{max} = 1$.

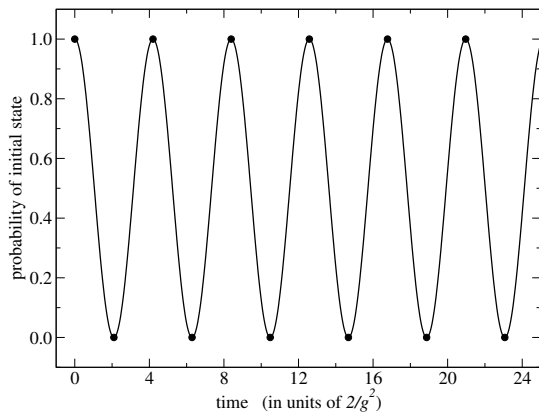
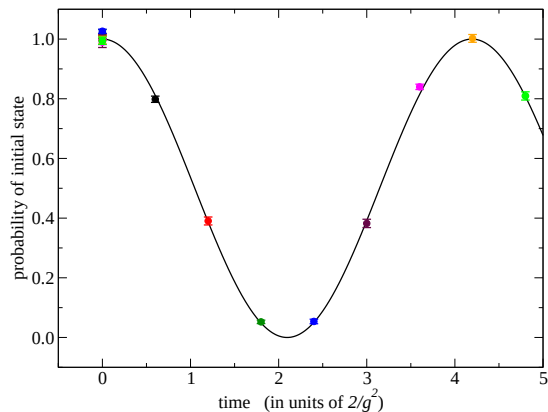
• 6 plaquettes $j_{max} = 1/2$.

- Solid lines are exact classical calculations, while points are D-Wave results.
- The Hamiltonian was block-diagonalized and each block submitted separately.

Time Evolution as a ground-state eigenvalue problem

- Find the ground-state **history state** of the **Feynman-Kitaev clock Hamiltonian**

$$|1_2^1 1_2^2\rangle = \left[\begin{array}{|c|c|} \hline \text{ } & \text{ } \\ \hline \end{array} \right] \leftrightarrow \frac{1}{\sqrt{2}} \left(\left[\begin{array}{|c|c|} \hline \text{ } & \text{ } \\ \hline \end{array} \right] + \left[\begin{array}{|c|c|} \hline \text{ } & \text{ } \\ \hline \end{array} \right] \right) = \frac{1}{\sqrt{2}} (|2_2^2 2_1^1\rangle + |2_1^1 2_2^2\rangle)$$



- 8 different 2 time steps.

- 12 time steps at once.

Part III:
IBM superconducting gate-based
universal quantum computer

IBM gate-based quantum computer

Some of the most common gates:

Gate	Symbol	Matrix Representation		
Z-gate		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	RY-gate	 $\begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$
Hadamard-gate		$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	CNOT-gate	 $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$

- The common notation for the qubit state is: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

IBM gate-based quantum computer

Some of the most common gates:

Gate	Symbol	Matrix Representation		
Z-gate		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	RY-gate	 $\begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$
Hadamard-gate		$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	CNOT-gate	 $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$

- The common notation for the qubit state is: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- The Hadamard is a superposition gate : $H|0\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$ and $H|1\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$

IBM gate-based quantum computer

Some of the most common gates:

Gate	Symbol	Matrix Representation		
Z-gate		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	RY-gate	 $\begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$
Hadamard-gate		$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	CNOT-gate	 $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$

- The common notation for the qubit state is: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- The Hadamard is a superposition gate : $H|0\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$ and $H|1\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$
- The CNOT is an entangling gate, applies a X on the second qubit when the first is $|1\rangle$:

$$\mathbf{CNOT}|0\rangle|0\rangle = |0\rangle|0\rangle \text{ and } \mathbf{CNOT}|1\rangle|0\rangle = |1\rangle|1\rangle$$

An example of using a IBM hardware: Create and measure a Bell state

- Let's create the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ starting from two qubits both in $|0\rangle$.

$$|0\rangle \otimes |0\rangle \rightarrow CNOT(H|0\rangle \otimes |0\rangle) = CNOT(1/\sqrt{2}(|0\rangle + |1\rangle) \otimes |0\rangle) = 1/\sqrt{2}(|00\rangle + |11\rangle)$$

An example of using a IBM hardware: Create and measure a Bell state

- Let's create the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ starting from two qubits both in $|0\rangle$.

$$|0\rangle \otimes |0\rangle \rightarrow CNOT(H|0\rangle \otimes |0\rangle) = CNOT(1/\sqrt{2}(|0\rangle + |1\rangle) \otimes |0\rangle) = 1/\sqrt{2}(|00\rangle + |11\rangle)$$

```

qr = QuantumRegister(2)           # Ask 2 qubits
cr = ClassicalRegister(2)          # Ask 2 bits
circuit = QuantumCircuit(qr,cr)    # Create the circuit

#=====

circuit.h(qr[0])                   # Hadamar gate acts on q_0
circuit.cx(qr[0],qr[1])            # CNOT acts on q_0 and q_1

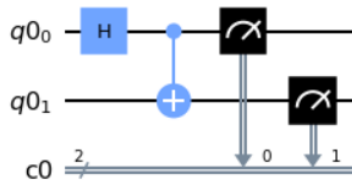
circuit.measure(qr, cr)            # Measure the qubits

#=====

backend = provider.get_backend('ibm_perth') # Choose the Q. hardware (Perth)
job=execute(circuit,backend,shots=10000)    # Execute 10k experiments
job.result().get_counts()                 # Get the counts on the z base

plot_histogram(job.result().get_counts())   # Print the histogram

```



An example of using a IBM hardware: Create and measure a Bell state

- Let's create the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ starting from two qubits both in $|0\rangle$.

$$|0\rangle \otimes |0\rangle \rightarrow CNOT(H|0\rangle \otimes |0\rangle) = CNOT(1/\sqrt{2}(|0\rangle + |1\rangle) \otimes |0\rangle) = 1/\sqrt{2}(|00\rangle + |11\rangle)$$

```

qr = QuantumRegister(2)           # Ask 2 qubits
cr = ClassicalRegister(2)          # Ask 2 bits
circuit = QuantumCircuit(qr,cr)    # Create the circuit

#=====

circuit.h(qr[0])                   # Hadamar gate acts on q_0
circuit.cx(qr[0],qr[1])            # CNOT acts on q_0 and q_1

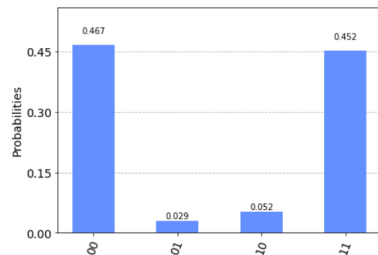
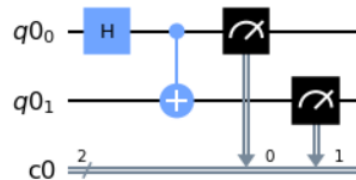
circuit.measure(qr, cr)            # Measure the qubits

#=====

backend = provider.get_backend('ibm_perth') # Choose the Q. hardware (Perth)
job=execute(circuit,backend,shots=10000)    # Execute 10k experiments
job.result().get_counts()                  # Get the counts on the z base

plot_histogram(job.result().get_counts())    # Print the histogram

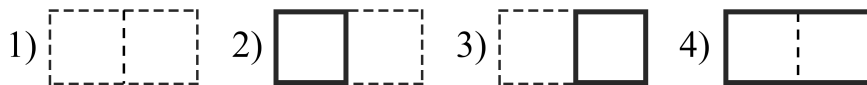
```



The system returns the counts in each of basis elements: ['00' : 4666, '01' : 294, '10' : 517, '11' : 4523]

Encoding the 2 plaquettes theory on the hardware

- A 2 plaquettes lattice with $j_{max} = 1/2$ has 4 states:

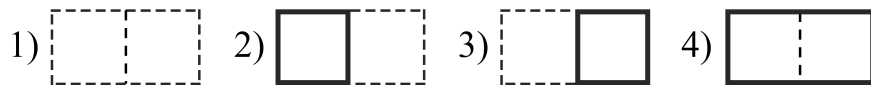


- The states can be represented using two qubits, one for each plaquette:

$$1) \longrightarrow |0\rangle|0\rangle, \quad 2) \longrightarrow |0\rangle|1\rangle, \quad 3) \longrightarrow |1\rangle|0\rangle, \quad 4) \longrightarrow |1\rangle|1\rangle$$

Encoding the 2 plaquettes theory on the hardware

- A 2 plaquettes lattice with $j_{max} = 1/2$ has 4 states:



- The states can be represented using two qubits, one for each plaquette:

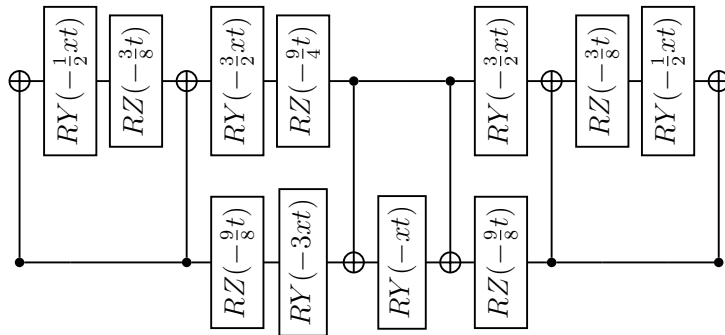
$$1) \longrightarrow |0\rangle|0\rangle, \quad 2) \longrightarrow |0\rangle|1\rangle, \quad 3) \longrightarrow |1\rangle|0\rangle, \quad 4) \longrightarrow |1\rangle|1\rangle$$

- The Hamiltonian representation can be rewritten in gates as:

$$\begin{aligned} \frac{2}{g^2}H &= \begin{pmatrix} 0 & -2x & -2x & 0 \\ -2x & 3 & 0 & -x \\ -2x & 0 & 3 & -x \\ 0 & -x & -x & \frac{9}{2} \end{pmatrix} = \\ &= \frac{3}{8}(7I_1 \otimes I_0 - 3I_1 \otimes Z_0 - Z_1 \otimes Z_0 - 3Z_1 \otimes I_0) - \frac{x}{2}(3I_1 + Z_1) \otimes X_0 - \frac{x}{2}X_1 \otimes (3I_0 + Z_0) \end{aligned}$$

Time evolution circuit 2-plaquette case

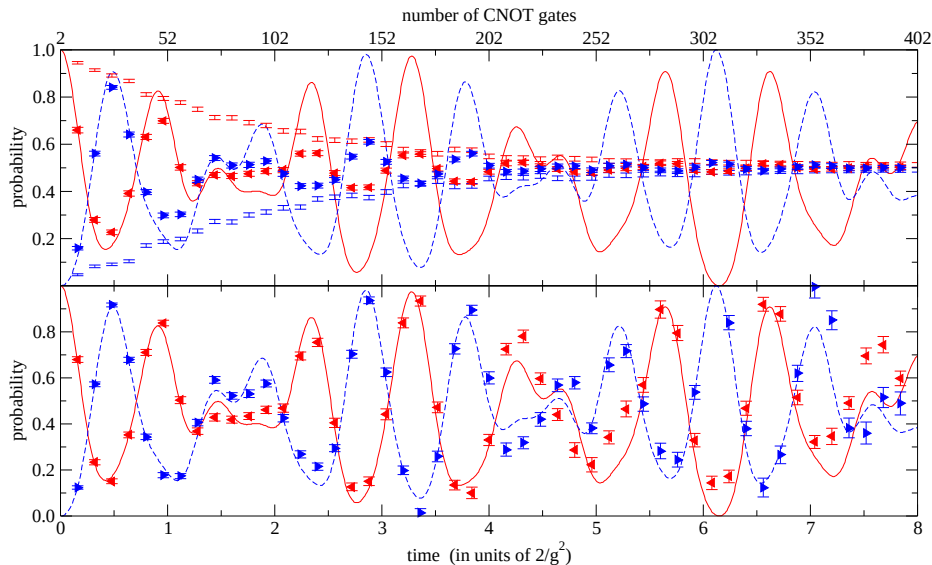
- $\exp(-iHt)$ circuit for one single second-order Suzuki-Trotter step for the 2-plaquettes lattice:



To obtain the time evolution of the system we need to:

- Prepare an **initial state**
- **Reach time** $N dt \rightarrow$ Apply N times the time evolution circuit with time step dt

Results time evolution 2 plaquettes case



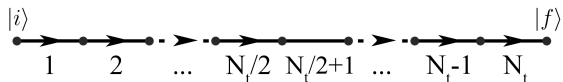
- Initial state: left plaquette "on" $j = 1/2$.
- Upper panel "raw data" after measurement error mitigation and randomized compiling (Pauli-twirling)
- Lower panel: final data, using our approach called "Self-mitigation"

- solid curve exact probability left plaquette; blue dashed curve exact probability right plaquette
- triangles are physics data; error-bar-only mitigation data, both from `ibm_lagos`

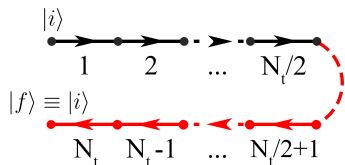
Error mitigation technique: Self-mitigation [1/2]

- The idea is simple: use an extra circuit with a priori known result to estimate hardware errors.
- Self-mitigation uses an error estimation circuit identical to the physics circuit:

Physics circuit: $|f\rangle = (e^{-iHdt})^{N_t} |i\rangle$



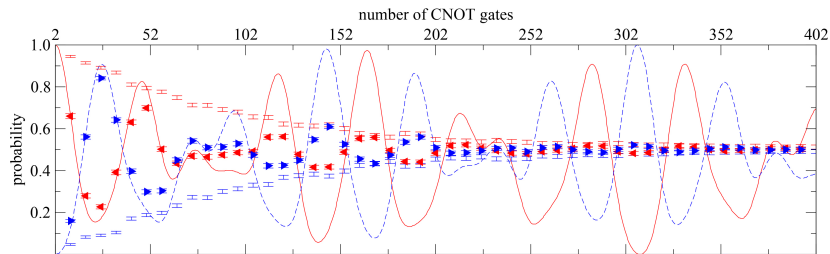
Error estimation circuit: $|f\rangle = (e^{iHdt})^{N_t/2} (e^{-iHdt})^{N_t/2} |i\rangle = |i\rangle$



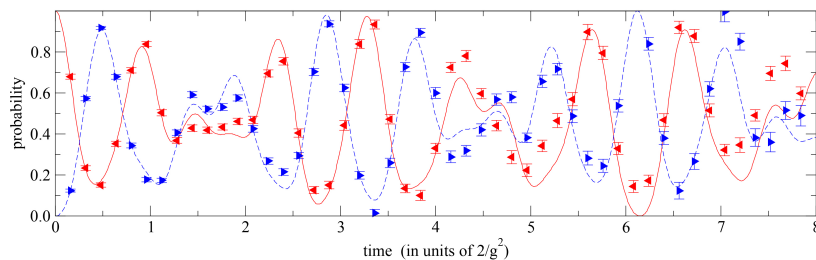
time
→

- The hardware error can be estimate by measuring how far the final state of the error estimation circuit deviates from the initial state.

Error mitigation technique: Self-mitigation [2/2]

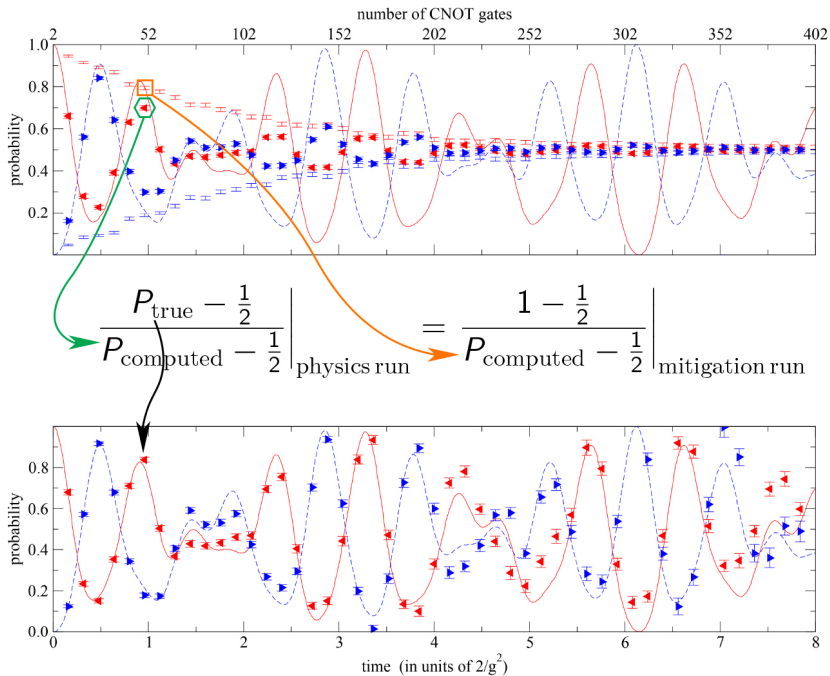


$$\left| \frac{P_{\text{true}} - \frac{1}{2}}{P_{\text{computed}} - \frac{1}{2}} \right|_{\text{physics run}} = \left| \frac{P_{\text{true}} - \frac{1}{2}}{P_{\text{computed}} - \frac{1}{2}} \right|_{\text{mitigation run}}$$



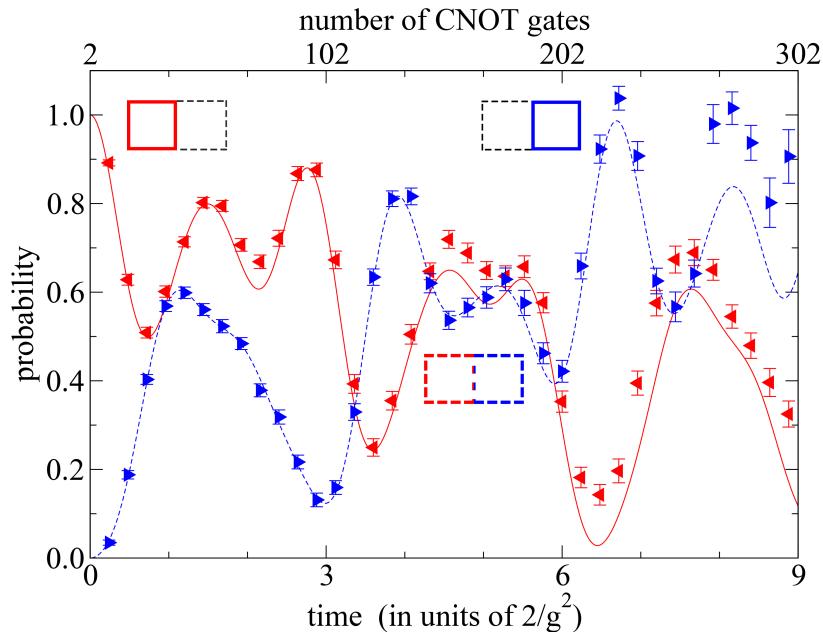
- On a perfect hardware the mitigation circuit returns a probability of 1 for the left plaquette and 0 for the right one.
- red error bar without symbols
mitigation data left plaquette $j = 1/2$
from the `ibm_lagos`.
- blue error bar without symbols
mitigation data right plaquette $j = 1/2$
from the `ibm_lagos`.
- red (blue) triangles left (right)
plaquette $j = 1/2$ data from the
`ibm_lagos`.

Error mitigation technique: Self-mitigation [2/2]



- On a perfect hardware the mitigation circuit returns a probability of 1 for the left plaquette and 0 for the right one.
- red error bar without symbols
mitigation data left plaquette $j = 1/2$
from the `ibm_lagos`.
- blue error bar without symbols
mitigation data right plaquette $j = 1/2$
from the `ibm_lagos`.
- red (blue) triangles left (right)
plaquette $j = 1/2$ data from the
`ibm_lagos`.

An example of dynamic process in real-time: A traveling excitation



- Initial state: left plaquette "on"
 $j = 1/2$.
- $x = 0.8$, $dt = 0.12$, 2 Trotter steps.
- red solid curve exact probability left
plaquette has $j = 1/2$,
 $P[2 \rightarrow 2] + P[2 \rightarrow 4]$
- blue dashed curve exact probability
right plaquette has $j = 1/2$,
 $P[2 \rightarrow 3] + P[2 \rightarrow 4]$
- red and blue triangles are calculations
on the `ibm_lagos`.

Part IV:
Supersymmetric Quantum Mechanics

Supersymmetric Quantum Mechanics

- Single-site model with one fermion and one boson interacting

$$H = \frac{1}{2} \left(\hat{p}^2 + [W'(\hat{q})]^2 - W''(\hat{q}) [\hat{b}^\dagger, \hat{b}] \right)$$

$$\begin{array}{c} \varphi \\ \bullet \\ \psi \end{array}$$

Supersymmetric Quantum Mechanics

- Single-site model with one fermion and one boson interacting

$$H = \frac{1}{2} (\hat{p}^2 + [W'(\hat{q})]^2 - W''(\hat{q}) [\hat{b}^\dagger, \hat{b}])$$

$$\begin{array}{c} \varphi \\ \bullet \\ \psi \end{array}$$

Harmonic Oscillator

$$W(\hat{q}) = \frac{1}{2} m \hat{q}^2$$

[Supersymmetric]

Double Well

$$W(\hat{q}) = \frac{1}{2} m \hat{q}^2 + g \left(\frac{1}{3} \hat{q}^3 + \mu^2 \hat{q} \right)$$

[Spontaneously breaking]

Anharmonic Oscillator

$$W(\hat{q}) = \frac{1}{2} m \hat{q}^2 + \frac{1}{4} g \hat{q}^4$$

[Supersymmetric]

Supersymmetric Quantum Mechanics

- Single-site model with one fermion and one boson interacting

$$H = \frac{1}{2} (\hat{p}^2 + [W'(\hat{q})]^2 - W''(\hat{q}) [\hat{b}^\dagger, \hat{b}])$$

$$\begin{array}{c} \varphi \\ \bullet \\ \psi \end{array}$$

Harmonic Oscillator

$$W(\hat{q}) = \frac{1}{2} m \hat{q}^2$$

[Supersymmetric]

Double Well

$$W(\hat{q}) = \frac{1}{2} m \hat{q}^2 + g \left(\frac{1}{3} \hat{q}^3 + \mu^2 \hat{q} \right)$$

[Spontaneously breaking]

Anharmonic Oscillator

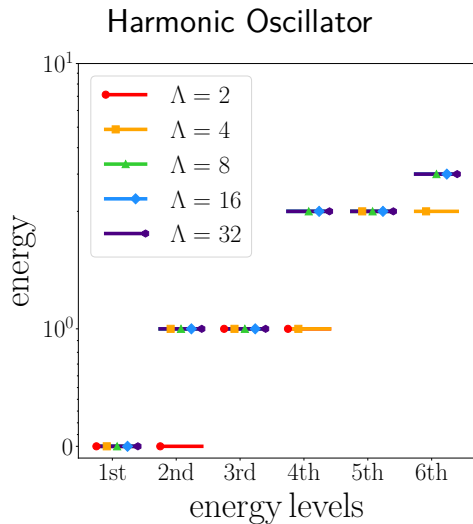
$$W(\hat{q}) = \frac{1}{2} m \hat{q}^2 + \frac{1}{4} g \hat{q}^4$$

[Supersymmetric]

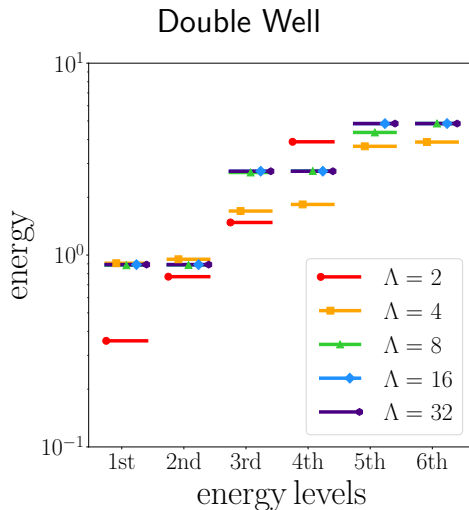
- Fermionic annihilation and creating operators: $\hat{b}|1\rangle = |0\rangle$, $\hat{b}|0\rangle = 0$; $\hat{b}^\dagger|1\rangle = 0$, $\hat{b}^\dagger|0\rangle = |1\rangle$

- Supersymmetry preserved: $E_0 = 0$, singlet ground state and paired excited states
- Supersymmetry breaking: $E_0 \neq 0$, all states are paired

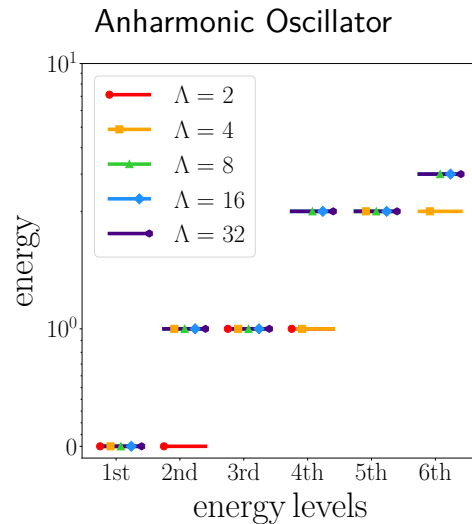
- Supersymmetry preserved: $E_0 = 0$, singlet ground state and paired excited states
- Supersymmetry breaking: $E_0 \neq 0$, all states are paired



[Supersymmetric]



[Spontaneously breaking]



[Supersymmetric]

Encoding the theory in qubits

- Fermionic operators $\implies$ Jordan-Wigner transformation $\hat{b} = \frac{1}{2}(X + iY) \rightarrow [\hat{b}^\dagger, \hat{b}] = -Z$
1 qubit per fermion

Encoding the theory in qubits

- Fermionic operators $\implies$ Jordan-Wigner transformation $\hat{b} = \frac{1}{2}(X + iY) \rightarrow [\hat{b}^\dagger, \hat{b}] = -Z$
1 qubit per fermion

- Bosonic operators $\implies$ Fock basis, $\hat{q}$ and $\hat{p}$ from the harmonic oscillator truncated to $\Lambda = 2^B$ modes:
B qubits per boson

Encoding the theory in qubits

- Fermionic operators $\implies$ Jordan-Wigner transformation $\hat{b} = \frac{1}{2}(X + iY) \rightarrow [\hat{b}^\dagger, \hat{b}] = -Z$
1 qubit per fermion
-

- Bosonic operators $\implies$ Fock basis, $\hat{q}$ and $\hat{p}$ from the harmonic oscillator truncated to $\Lambda = 2^B$ modes:
B qubits per boson

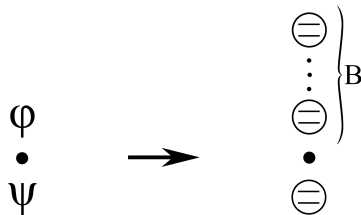
$$\hat{q} \doteq \frac{1}{\sqrt{2m}} \begin{pmatrix} 0 & \sqrt{1} & 0 & \dots & 0 \\ \sqrt{1} & 0 & \sqrt{2} & \dots & 0 \\ 0 & \sqrt{2} & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & 0 & \sqrt{\Lambda-1} \\ 0 & 0 & \dots & \sqrt{\Lambda-1} & 0 \end{pmatrix} \quad \hat{p} \doteq i\sqrt{\frac{m}{2}} \begin{pmatrix} 0 & -\sqrt{1} & 0 & \dots & 0 \\ \sqrt{1} & 0 & -\sqrt{2} & \dots & 0 \\ 0 & \sqrt{2} & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & 0 & -\sqrt{\Lambda-1} \\ 0 & 0 & \dots & \sqrt{\Lambda-1} & 0 \end{pmatrix}$$

Encoding the theory in qubits

- Fermionic operators $\implies$ Jordan-Wigner transformation $\hat{b} = \frac{1}{2}(X + iY) \rightarrow [\hat{b}^\dagger, \hat{b}] = -Z$
1 qubit per fermion

- Bosonic operators $\implies$ Fock basis, $\hat{q}$ and $\hat{p}$ from the harmonic oscillator truncated to $\Lambda = 2^B$ modes:
B qubits per boson

$$\hat{q} \doteq \frac{1}{\sqrt{2m}} \begin{pmatrix} 0 & \sqrt{1} & 0 & \dots & 0 \\ \sqrt{1} & 0 & \sqrt{2} & \dots & 0 \\ 0 & \sqrt{2} & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & 0 & \sqrt{\Lambda-1} \\ 0 & 0 & \dots & \sqrt{\Lambda-1} & 0 \end{pmatrix} \quad \hat{p} \doteq i\sqrt{\frac{m}{2}} \begin{pmatrix} 0 & -\sqrt{1} & 0 & \dots & 0 \\ \sqrt{1} & 0 & -\sqrt{2} & \dots & 0 \\ 0 & \sqrt{2} & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & 0 & -\sqrt{\Lambda-1} \\ 0 & 0 & \dots & \sqrt{\Lambda-1} & 0 \end{pmatrix}$$



$2\Lambda = 2^{B+1}$ classical resources

$(B + 1)$ qubits

Measure the ground state energy

$$E_0 = 0$$

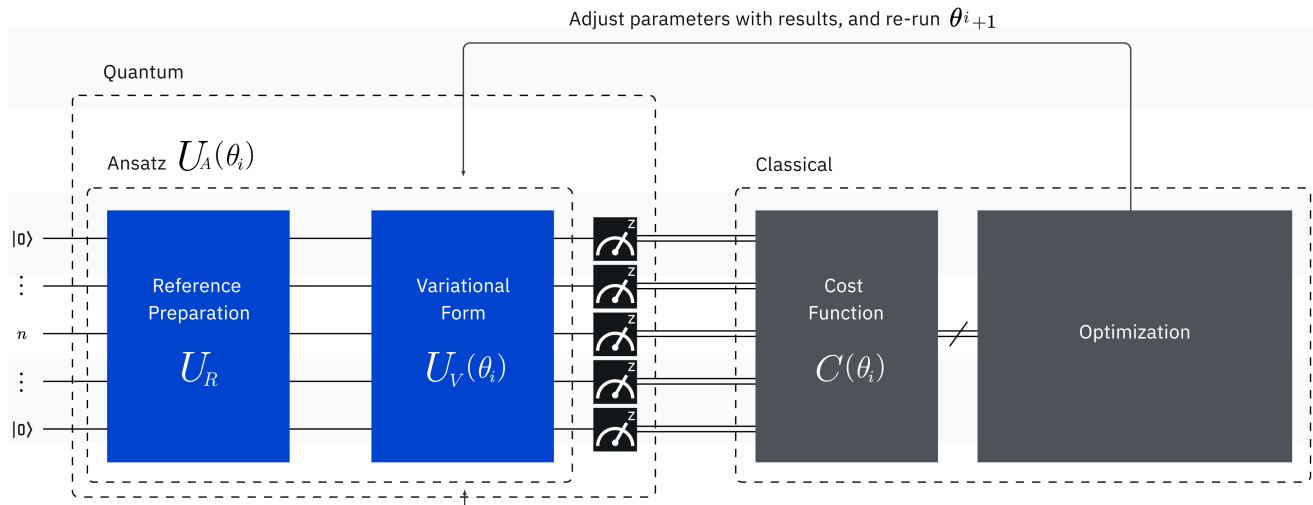
Supersymmetric

$$E_0 > 0$$

Supersymmetry breaking

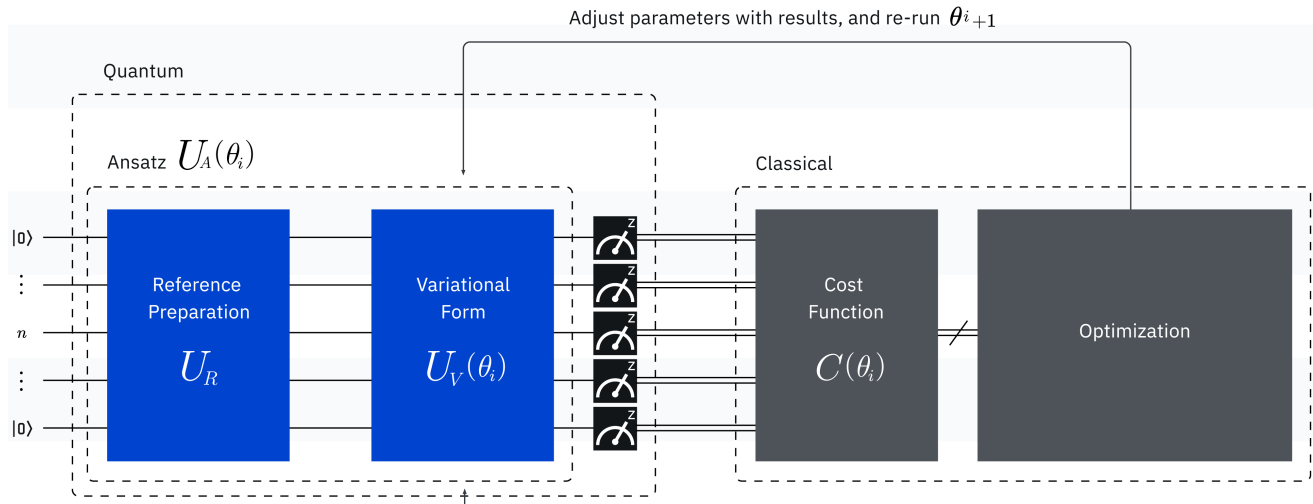
Variational Quantum Eigensolver (VQE)

- $|0\rangle \Rightarrow \text{Ansatz} \Rightarrow |\psi(\theta_i)\rangle \Rightarrow \text{Measurements} \Rightarrow \langle\psi(\theta_i)|H|\psi(\theta_i)\rangle \Rightarrow \text{C. Optimizer} \Rightarrow \text{new } (\theta_i) \Rightarrow \text{Re-run}$



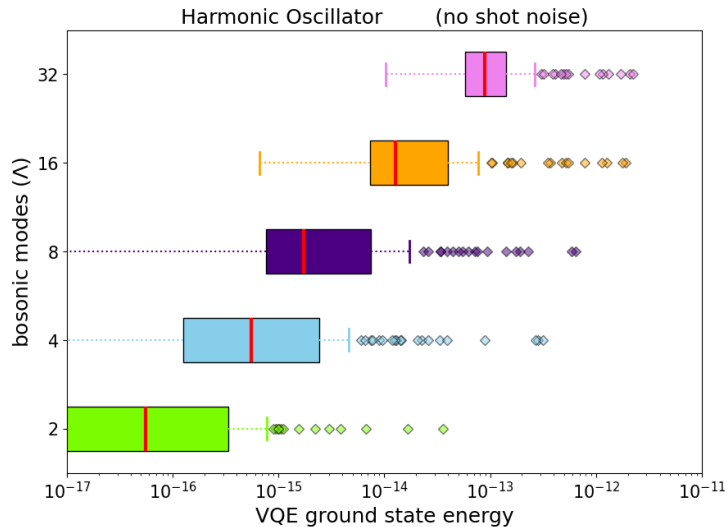
Variational Quantum Eigensolver (VQE)

- $|0\rangle \Rightarrow$ Ansatz $\Rightarrow |\psi(\theta_i)\rangle \Rightarrow$ Measurements $\Rightarrow \langle \psi(\theta_i) | H | \psi(\theta_i) \rangle \Rightarrow$ C. Optimizer $\Rightarrow$ new $(\theta_i) \Rightarrow$ Re-run



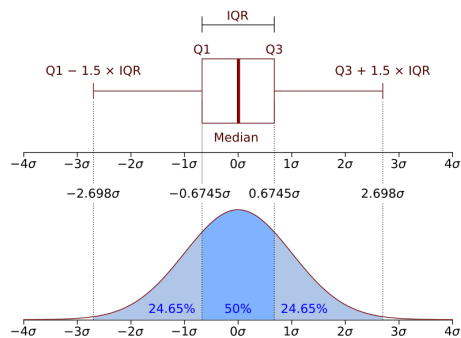
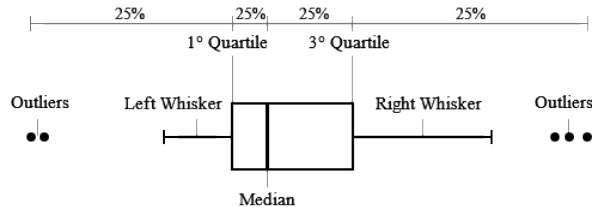
- Guess an ansatz expressive enough with only a few gates
- Optimizer: global vs. local & gradient-based vs. gradient-free

VQE results and box-and-whisker plot



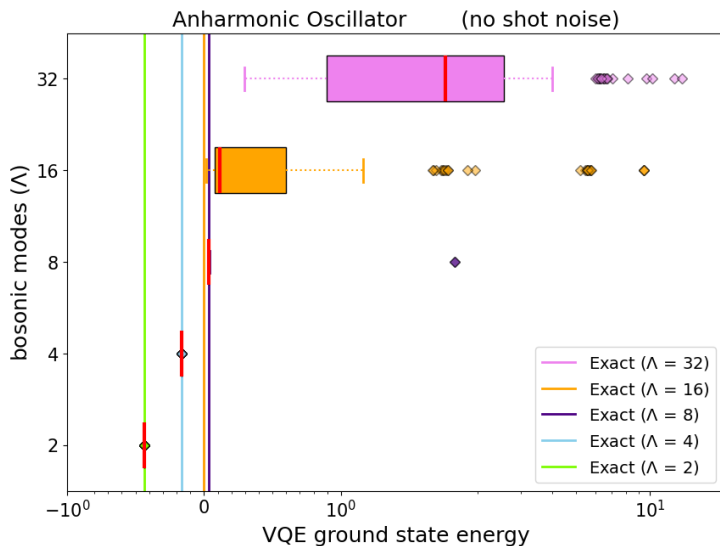
- Preserved supersymmetry, VQE agrees!

Key elements of a box-and-whisker plot

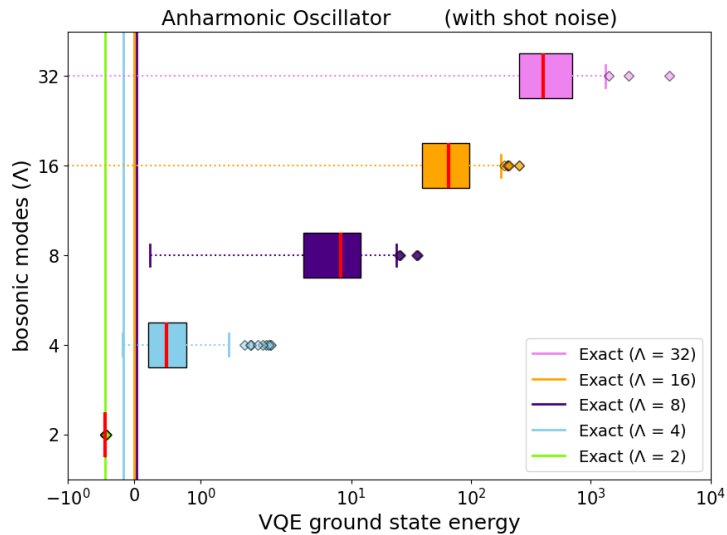


box plot for Gaussian distribution

VQE: measuring the ground state



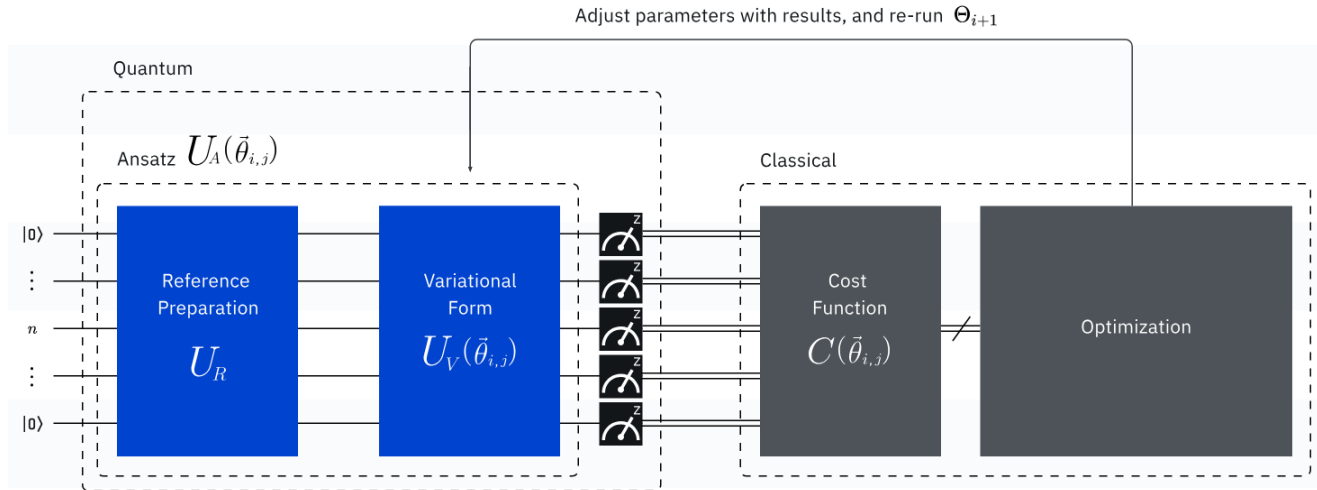
- Preserved supersymmetry,
VQE disagrees $\Lambda \geq 16$



- Preserved supersymmetry,
VQE disagrees $\Lambda \geq 4$

Variational Quantum Deflation (VQD)

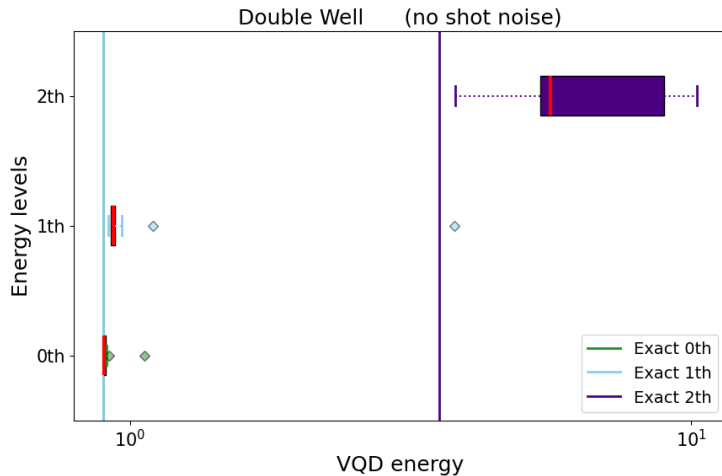
- VQD is a multi-step VQE where previously identified states are penalized



$$C_k(\vec{\theta}) = \langle \psi(\vec{\theta}) | \vec{H} | \psi(\vec{\theta}) \rangle + \sum_{j=0}^{k-1} \beta_j |\langle \psi(\vec{\theta}) | \psi(\vec{\theta}^j) \rangle|^2$$

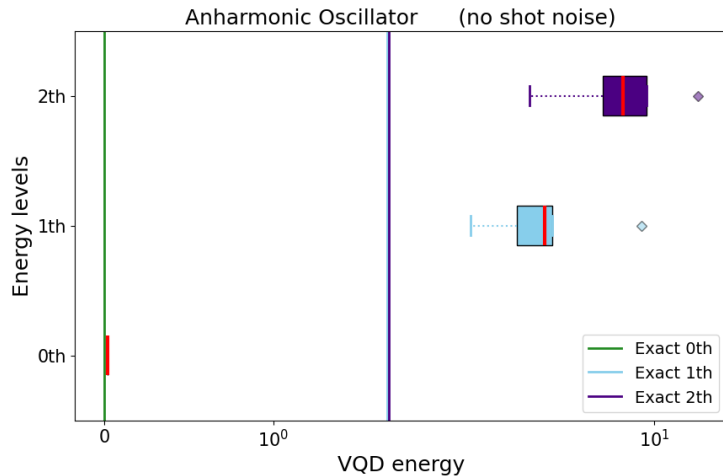
VQD results - exploring pairing structures

- Supersymmetry breaking: **paired states**



- Supersymmetry breaking, VQD agrees!

- Preserved supersymmetry: **single ground state**



- Preserved supersymmetry, VQD agrees!

The Challenge of Encoding a Hamiltonian

The Challenge of Encoding a Hamiltonian

$$\hat{H} = \frac{g^2}{2} \left(\sum_{i=\text{links}} \hat{E}_i^2 - 2 \times \sum_{i=\text{plaquettes}} \hat{\square}_i \right)$$



Gate decomposition

The Challenge of Encoding a Hamiltonian

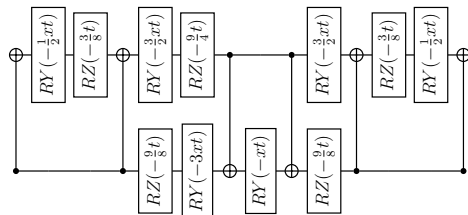
$$\hat{H} = \frac{g^2}{2} \left(\sum_{i=\text{links}} \hat{E}_i^2 - 2 \sum_{i=\text{plaquettes}} \hat{\square}_i \right)$$



Gate decomposition



$$H = \frac{3}{8} (7 - 3Z_0 - Z_0Z_1 - 3Z_1) - \frac{x}{2} (3 + Z_1)X_0 - \frac{x}{2} (3 + Z_0)X_1$$



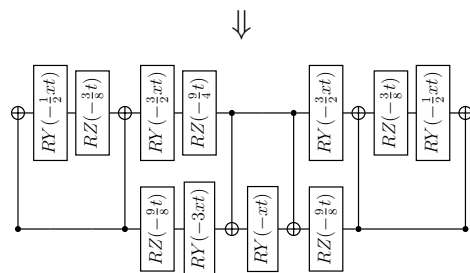
The Challenge of Encoding a Hamiltonian

$$\hat{H} = \frac{g^2}{2} \left(\sum_{i=\text{links}} \hat{E}_i^2 - 2 \sum_{i=\text{plaquettes}} \hat{\square}_i \right)$$



Gate decomposition

$$H = \frac{3}{8} (7 - 3Z_0 - Z_0Z_1 - 3Z_1) - \frac{x}{2} (3 + Z_1)X_0 - \frac{x}{2} (3 + Z_0)X_1$$



- Spin-like systems $H \longrightarrow$ quantum circuit (Ok)

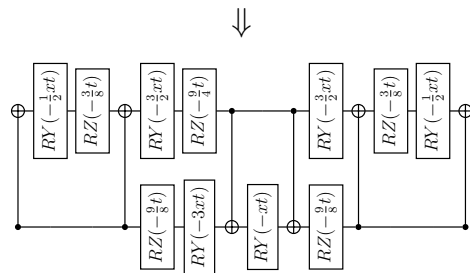
The Challenge of Encoding a Hamiltonian

$$\hat{H} = \frac{g^2}{2} \left(\sum_{i=\text{links}} \hat{E}_i^2 - 2 \sum_{i=\text{plaquettes}} \hat{\square}_i \right)$$



Gate decomposition

$$H = \frac{3}{8} (7 - 3Z_0 - Z_0Z_1 - 3Z_1) - \frac{x}{2} (3 + Z_1)X_0 - \frac{x}{2} (3 + Z_0)X_1$$



- Spin-like systems $H \longrightarrow$ quantum circuit (Ok)
- No spin-like systems:
 - 1 Matrix representation (**scales exponentially**)
 - 2 Convert matrix to quantum gates, Pauli decomposition (worst-case $\mathcal{O}(4^Q)$ [[arXiv:2310.13421](https://arxiv.org/abs/2310.13421)])

Conclusions

- Quantum hardware has revived interest in the Hamiltonian formulation.
- Quantum simulation of Non-Abelian LGT and interacting Fermion-Boson system are possible
- Simple error mitigation techniques like self-mitigation can extend the usability of NISQ hardware.
- **There is the need for formalisms that can make quantum advantage tangible.**

- Quantum hardware has revived interest in the Hamiltonian formulation.
- Quantum simulation of Non-Abelian LGT and interacting Fermion-Boson system are possible
- Simple error mitigation techniques like self-mitigation can extend the usability of NISQ hardware.
- **There is the need for formalisms that can make quantum advantage tangible.**
→ Orbifold Lattice approach for quantum computing of $SU(N)$ LGT [[arXiv:2604.15132](https://arxiv.org/abs/2604.15132)]

Thank you all for your time!

Time for Questions!

Please feel free to get in touch with any questions or curiosities!



Emanuele Mendicelli (he/him)

Department of Theoretical Physics
Jožef Stefan Institute
Ljubljana, Slovenia

`Emanuele.Mendicelli@ijs.si`

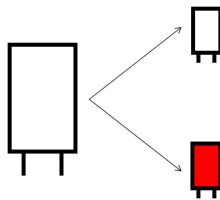
Alternative contact:

`emanuelemendicelli@hotmail.it`

Backup Slides

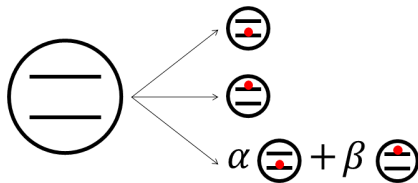
Why do we like qubits? qubit Vs bit

- Bit:



A bit can only be in one state at a time

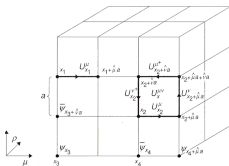
- Qubit:



A qubit can be in a superposition of basis states $\longrightarrow$ advantage in **Time Evolution** and **Variational method search**

Lattice Quantum Chromodynamics (LQCD) in a nutshell

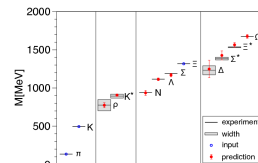
- The interaction between quarks and gluons has many phenomena that cannot be studied analytically.
- Lattice gauge theory is a theoretical approach in which theoretical models are directly formulated to be studied through simulations on supercomputers using the importance sampling Monte Carlo method:



Discrete spacetime



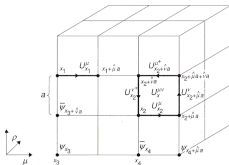
Classical Supercomputer



Predict particle masses

Lattice Quantum Chromodynamics (LQCD) in a nutshell

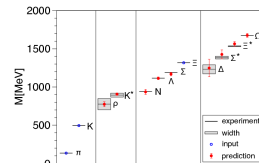
- The interaction between quarks and gluons has many phenomena that cannot be studied analytically.
- Lattice gauge theory is a theoretical approach in which theoretical models are directly formulated to be studied through simulations on supercomputers using the importance sampling Monte Carlo method:



Discrete spacetime



Classical Supercomputer

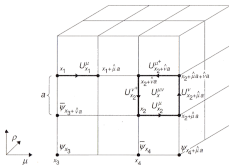


Predict particle masses

$$\langle O \rangle = \frac{\int D[C] O[C] e^{-S[C]}}{\int D[C] e^{-S[C]}} \Rightarrow \text{(Importance Sampling MC)} \approx \frac{1}{N} \sum_{n=1}^N O[C_n] \pm \mathcal{O}\left(\frac{\sigma_O}{\sqrt{N}}\right)$$

Lattice Quantum Chromodynamics (LQCD) in a nutshell

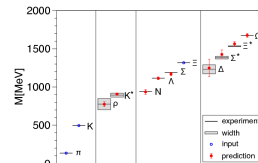
- The interaction between quarks and gluons has many phenomena that cannot be studied analytically.
- Lattice gauge theory is a theoretical approach in which theoretical models are directly formulated to be studied through simulations on supercomputers using the importance sampling Monte Carlo method:



Discrete spacetime



Classical Supercomputer



Predict particle masses

$$\langle O \rangle = \frac{\int D[C] O[C] e^{-S[C]}}{\int D[C] e^{-S[C]}} \Rightarrow \text{(Importance Sampling MC)} \approx \frac{1}{N} \sum_{n=1}^N O[C_n] \pm \mathcal{O}\left(\frac{\sigma_O}{\sqrt{N}}\right)$$

- Everything works like heaven unless the action S is negative or imaginary!
In those cases we have the **"Sign problem"**, the Monte Carlo breaks down and all hell breaks loose!!!

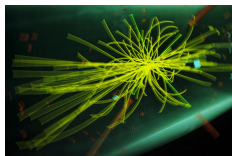
(Sign problem $\Rightarrow$ The numerical calculation requires an exponentially large amount of resources!)

Why do we like quantum computers? Quantum computer Vs Sign problem

Since the late '70s many phenomena are inaccessible to LQCD simulations due to the infamous sign problem:

Real-time evolution

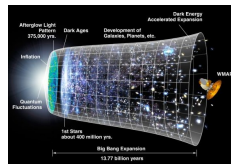
nonzero chemical potential



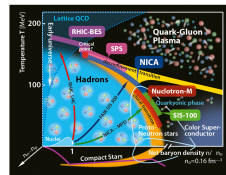
propagations/collisions



Imbalance matter-antimatter



Early Universe



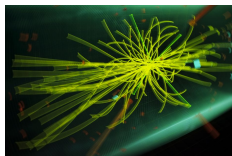
QCD phase diagram

Why do we like quantum computers? Quantum computer Vs Sign problem

Since the late '70s many phenomena are inaccessible to LQCD simulations due to the infamous sign problem:

Real-time evolution

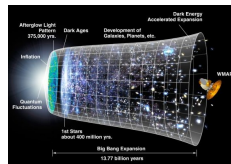
nonzero chemical potential



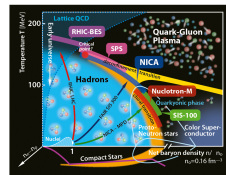
propagations/collisions



Imbalance matter-antimatter



Early Universe



QCD phase diagram

- **Hamiltonian simulations/calculations on quantum computer are free of the sign problem!**

Backup Slides - Section I
 $SU(2)$ Lattice Gauge theory

SU(2) pure gauge lattice theory, more about the operators

Multiplicities used for the basis states, $E \equiv 2j_E + 1$.

$$\hat{H} = \frac{g^2}{2} \left(\sum_{i=\text{links}} \hat{E}_i^2 - 2x \sum_{i=\text{plaquettes}} \hat{\square}_i \right)$$

$$\begin{array}{|c|c|} \hline j_B & j_D \\ \hline j_E & j_F \\ \hline j_A & j_C \\ \hline \end{array} = \left| E_A^B F_C^D G \right\rangle$$

g is the coupling constant and $x \equiv 2/g^4$.

$\hat{E}_i^2$ is the chromoelectric field for the i th lattice link.

$$\langle \psi_{\text{final}} | \sum_i \hat{E}_i^2 | \psi_{\text{initial}} \rangle = \sum_{i=A}^L j_i(j_i + 1) \delta_{\text{final,initial}}$$

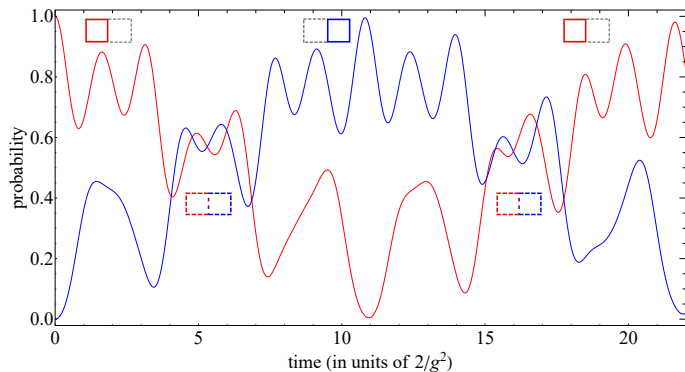
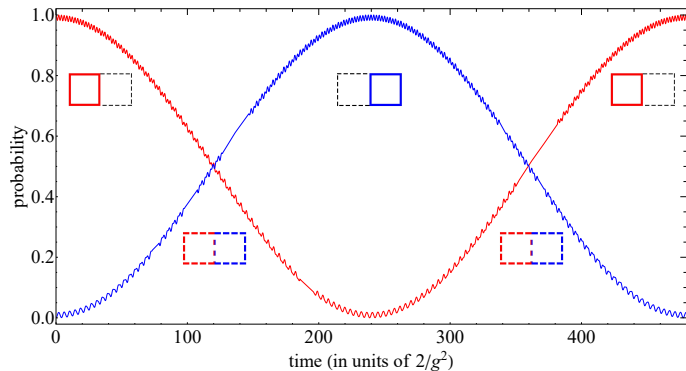
$\hat{\square}_i$ is the plaquette operator trace of the product of four gauge link operators of the i th plaquette.

$$\begin{aligned} \langle \psi_{\text{final}} | \square_1 | \psi_{\text{initial}} \rangle &= (-1)^{j_A+j_B+j_C+j_D+2j_E+2j_F+2j_I+2j_J} \\ &\quad \sqrt{2j_E+1} \sqrt{2j_E+1} \sqrt{2j_J+1} \sqrt{2j_J+1} \sqrt{2j_F+1} \sqrt{2j_F+1} \sqrt{2j_I+1} \sqrt{2j_I+1} \\ &\quad \left\{ \begin{array}{ccc} j_A & j_E & j_I \\ \frac{1}{2} & J_I & J_E \end{array} \right\} \left\{ \begin{array}{ccc} j_B & j_F & j_I \\ \frac{1}{2} & J_I & J_F \end{array} \right\} \left\{ \begin{array}{ccc} j_C & j_E & j_J \\ \frac{1}{2} & J_J & J_E \end{array} \right\} \left\{ \begin{array}{ccc} j_D & j_F & j_J \\ \frac{1}{2} & J_J & J_F \end{array} \right\} \end{aligned}$$

where j_i and J_i are the links in $|\psi_{\text{initial}}\rangle$ and $|\psi_{\text{final}}\rangle$, respectively.

The two different regimes $x < 1$ and $x > 1$ for the time evolution [1/2]

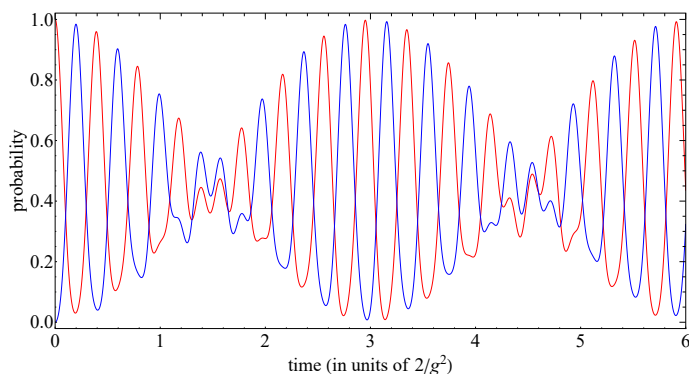
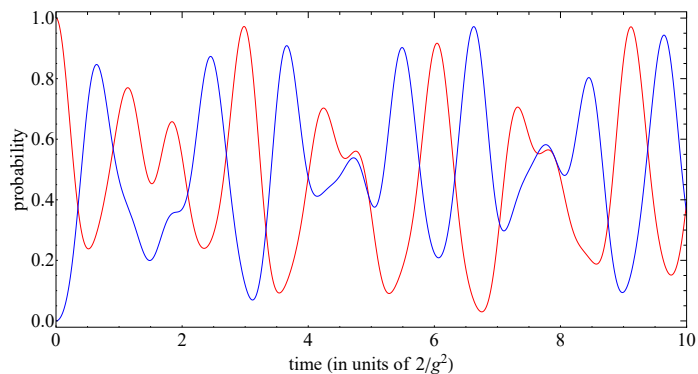
- For small x , the chromomagnetic contribution is negligible, therefore at low energy the dominant states are weakly coupled chromoelectric eigenstates. The single-plaquette states move across the lattice.



- The single-plaquette propagation time is larger for small x , diverging for $x = 0$, where the Hamiltonian is diagonal, containing only the chromoelectric term. The single-plaquettes are eigenstates, therefore they are constant in time.
- In the right figure, increasing x lets higher frequencies appear as oscillations superimposed on the single-plaquette transition.

The two different regimes $x < 1$ and $x > 1$ for the time evolution [2/2]

- For x larger than 1 at larger energy, the chromomagnetic contribution dominates mixing the single-plaquette states. This is evident by the presence of many higher frequencies superimposed on the single-plaquette transitions.



- The larger x is the more high frequencies are present as evident by moving from the left figure at $x = 1.5$ to the right figure at $x = 5.0$

The nightmare... of using compact variables

$$\begin{aligned}
\Box_1 |\psi_{\text{initial}}\rangle = & \sum_{M_E} \sum_{M'_E} \sum_{M_J} \sum_{M'_J} \sum_{M_F} \sum_{M'_F} \sum_{M_I} \sum_{M'_I} \sum_{m_A} \sum_{m'_A} \sum_{m_B} \sum_{m'_B} \sum_{m_C} \sum_{m'_C} \dots \sum_{m_L} \sum_{m'_L} \sum_{J_F} \sum_{J_E} \sum_{J_I} \sum_{J_J} \sum_{s_1} \sum_{s_2} \sum_{s_6} \sum_{s_5} \\
& (-1)^{s_1+s_2+s_6+s_5} (-1)^{-2j_E-2j_J-2j_F-2j_I+M_E+M'_E+M_J+M'_J+M_F+M'_F+M_I+M'_I} \\
& \sqrt{2j_E+1} \sqrt{2J_E+1} \sqrt{2j_J+1} \sqrt{2J_J+1} \sqrt{2j_F+1} \sqrt{2J_F+1} \sqrt{2j_I+1} \sqrt{2J_I+1} \\
& \begin{pmatrix} j_A & j_E & j_I \\ m'_A & m_E & m_I \end{pmatrix} \begin{pmatrix} j_C & j_E & j_J \\ m_C & m'_E & m_J \end{pmatrix} \begin{pmatrix} j_C & j_G & j_K \\ m'_C & m_G & m'_K \end{pmatrix} \begin{pmatrix} j_A & j_G & j_L \\ m_A & m'_G & m_L \end{pmatrix} \\
& \begin{pmatrix} j_B & j_F & j_I \\ m'_B & m_F & m'_I \end{pmatrix} \begin{pmatrix} j_D & j_F & j_J \\ m_D & m'_F & m'_J \end{pmatrix} \begin{pmatrix} j_D & j_H & j_K \\ m'_D & m_H & m'_K \end{pmatrix} \begin{pmatrix} j_B & j_H & j_L \\ m_B & m'_H & m'_L \end{pmatrix} \\
& \begin{pmatrix} j_E & \frac{1}{2} & J_E \\ m_E & -s_1 & -M_E \end{pmatrix} \begin{pmatrix} j_E & \frac{1}{2} & J_E \\ m'_E & s_2 & -M'_E \end{pmatrix} \begin{pmatrix} j_J & \frac{1}{2} & J_J \\ m_J & -s_2 & -M_J \end{pmatrix} \begin{pmatrix} j_J & \frac{1}{2} & J_J \\ m'_J & s_6 & -M'_J \end{pmatrix} \\
& \begin{pmatrix} j_F & \frac{1}{2} & J_F \\ m_F & s_5 & -M_F \end{pmatrix} \begin{pmatrix} j_F & \frac{1}{2} & J_F \\ m'_F & -s_6 & -M'_F \end{pmatrix} \begin{pmatrix} j_I & \frac{1}{2} & J_I \\ m_I & s_1 & -M_I \end{pmatrix} \begin{pmatrix} j_I & \frac{1}{2} & J_I \\ m'_I & -s_5 & -M'_I \end{pmatrix} \\
& |j_A, m_A, m'_A\rangle |j_B, m_B, m'_B\rangle |j_C, m_C, m'_C\rangle |j_D, m_D, m'_D\rangle |J_E, M_E, M'_E\rangle |J_F, M_F, M'_F\rangle \\
& |j_G, m_G, m'_G\rangle |j_H, m_H, m'_H\rangle |J_I, M_I, M'_I\rangle |J_J, M_J, M'_J\rangle |j_K, m_K, m'_K\rangle |j_L, m_L, m'_L\rangle . \tag{A15}
\end{aligned}$$

Applying a final state to that result allows all sums to be performed and the answer simplifies to

$$\begin{aligned}
\langle \psi_{\text{final}} | \Box_1 |\psi_{\text{initial}}\rangle = & (-1)^{j_A+j_B+j_C+j_D+2J_E+2J_F+2j_I+2j_J} \\
& \sqrt{2j_E+1} \sqrt{2J_E+1} \sqrt{2j_J+1} \sqrt{2J_J+1} \sqrt{2j_F+1} \sqrt{2J_F+1} \sqrt{2j_I+1} \sqrt{2J_I+1} \\
& \left\{ \begin{matrix} j_A & j_E & j_I \\ \frac{1}{2} & J_I & J_E \end{matrix} \right\} \left\{ \begin{matrix} j_B & j_F & j_I \\ \frac{1}{2} & J_I & J_F \end{matrix} \right\} \left\{ \begin{matrix} j_C & j_E & j_J \\ \frac{1}{2} & J_J & J_E \end{matrix} \right\} \left\{ \begin{matrix} j_D & j_F & j_J \\ \frac{1}{2} & J_J & J_F \end{matrix} \right\} \tag{A16}
\end{aligned}$$

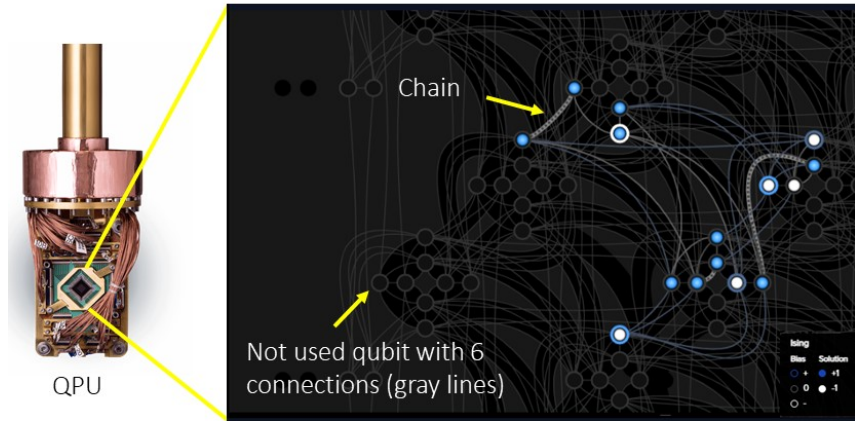
[arXiv:2205.09247]

Backup Slides - Section II

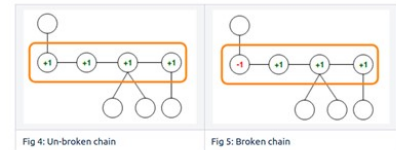
D-Wave

D-Wave Chain Strength

- ▶ The Chain is a set of qubits needed to represent the same variable.
- ▶ Problem embedded on the hardware



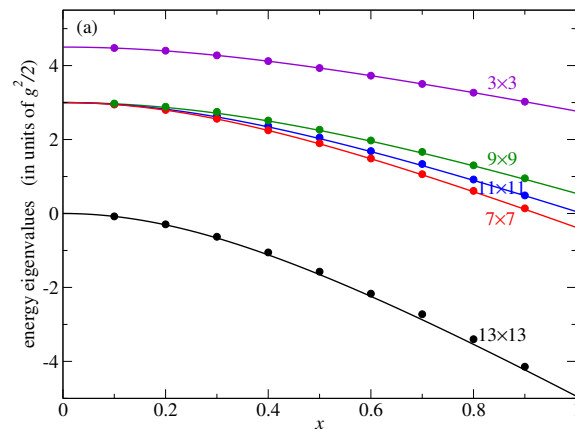
- ▶ Chain Broken \ Un-Broken



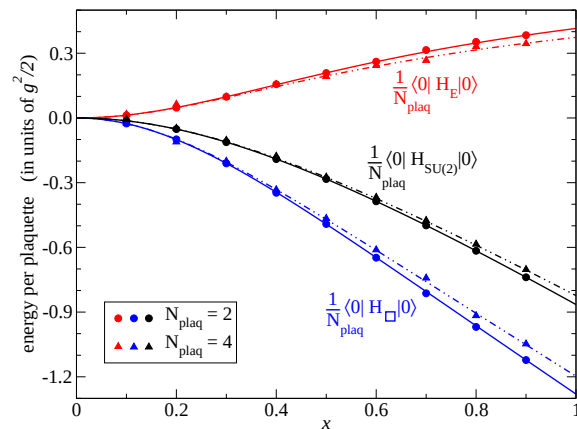
[<https://support.dwavesys.com/>]

- ▶ How to choose the chain value:
- ▶ The numerical value of the chain should be big enough to avoid that the chains break easily and small enough to not change the physics of the problem.

Several eigenvalues from D-Wave, lattice size effects



- 6 plaquettes $j_{\max} = 1$.
Solid lines are exact classical calculations.
Points are D-Wave results.



- $j_{\max} = 1/2$ lattice size effects.
Solid, dashed and dotted curves are classical calculations for $j_{\max} = \frac{1}{2}$, 1 and $\frac{3}{2}$ respectively.

The Quantum Annealer Eigensolver (QAE)*

- QAE introduces an extra Lagrange multiplier λ , to avoid appearance of the null vector as a trivial minimum:

$$\langle \psi | H | \psi \rangle \rightarrow F \equiv \langle \psi | H | \psi \rangle - \lambda \langle \psi | \psi \rangle$$

- Uses a fixed point representation, K qubits are used to represent a real number. Therefore for each value of K there is a set of allowed values for the solution:

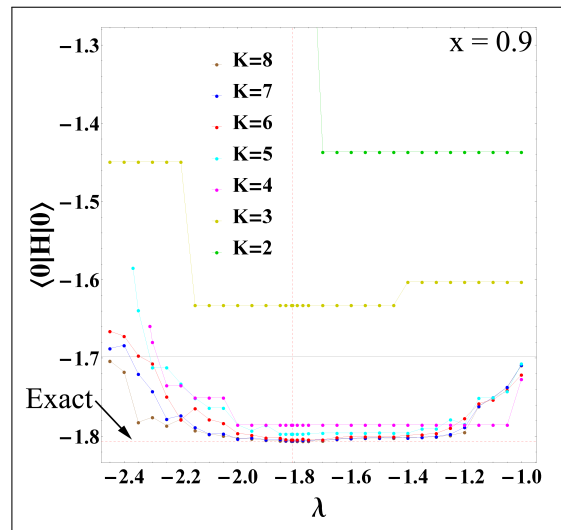
$$\begin{array}{cccccccccccccccc} (K=1) & & & & & & & -1 & 0 \\ (K=2) & & & & & & -1 & -\frac{1}{2} & 0 & \frac{1}{2} \\ (K=4) & -1 & -\frac{7}{8} & -\frac{3}{4} & -\frac{5}{8} & -\frac{1}{2} & -\frac{3}{8} & -\frac{1}{4} & -\frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{4} & \frac{3}{8} & \frac{1}{2} & \frac{5}{8} & \frac{3}{4} & \frac{7}{8} \end{array}$$

[* A. Teplukhin, B. K. Kendrick, and D. Babikov, *J. Chem. Theory and Comp.* 15, 4555(2019), doi: [10.1021/acs.jctc.9b00402](https://doi.org/10.1021/acs.jctc.9b00402)

Algorithm Steps

Steps to extract the ground state for a single value of the gauge coupling

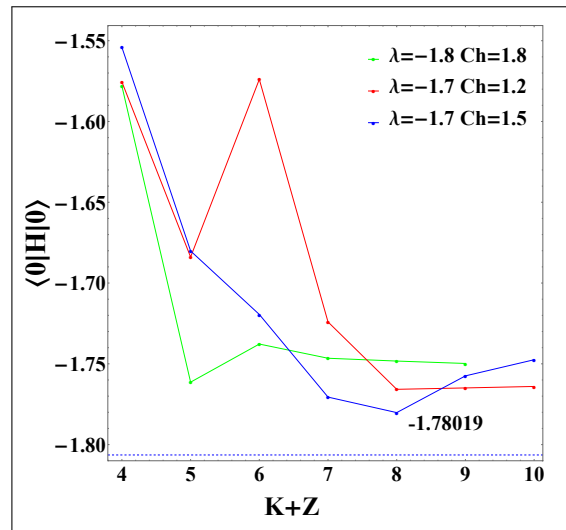
- 1 Find the optimum λ and the number of qubits needed (K) on the quantum simulator/hardware.
- Using $K = 2$ and 3 is not enough
- For $K = 6, 7$ and 8 there is a range of optimum values



Algorithm Steps

Steps to extract the ground state for a single value of the gauge coupling

2. Tune the chain strength
3. Accept or re-do it increasing the number of qubits (K)

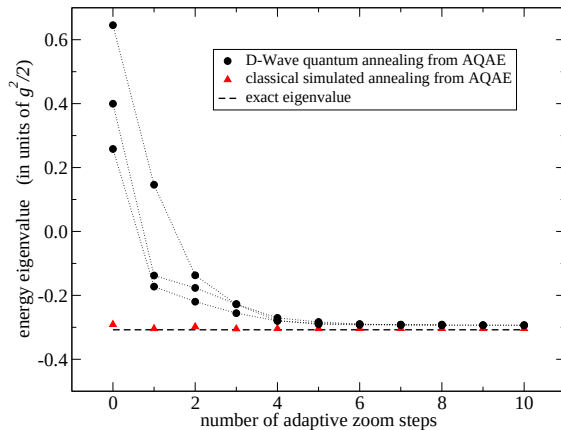


The Adaptive QAE algorithm (AQAE)*

- Run the QAE several times (Z) and use as a starting state (vector) the one previously found.
- The search for the new vector is centered on the previous one and now its allowed values are more finely spaced.
- Therefore it converges faster with fewer qubits.

[* A PYTHON implementation of the AQAE algorithm, [AQAE algorithm](#)]

Convergence of the AQAE algorithm



- Three runs of the AQAE with $K = 4$ showing fast convergence with consecutive adaptive steps for different initial starting point.

Stoquastic Hamiltonian

- A Hamiltonian is stoquastic if in the computational base all its off-diagonal elements are real and nonpositive. (Otherwise Sign problem. *)
- The reason why D-Wave has a hard time dealing with nonstoquastics is the absence of additional interaction such as $\sigma_i^x \sigma_j^x$ or $\sigma_i^y \sigma_j^y$ in its QUBO Hamiltonian. Their presence could potentially give rise to annealing-based universal quantum computers.(**)

In our case this problem could be eliminated using a specific value for the time step ϵ :

$$\epsilon = \pi/\omega$$

*[E. Y. Loh, Jr., J. E. Gubernatis, R. T. Scalettar, S. R. White, D. J. Scalapino, and R. L. Sugar Phys. Rev. B 41, 9301, [10.1103/PhysRevB.41.9301](https://doi.org/10.1103/PhysRevB.41.9301)]

** [I. Ozfidan et al. Phys. Rev. Applied 13, 034037, [doi:10.1103/PhysRevApplied.13.034037](https://doi.org/10.1103/PhysRevApplied.13.034037)]

Time Evolution as a ground-state eigenvalue problem

We would like to solve the Schrödinger equation ($i\partial_t |\Psi(t)\rangle = H |\Psi(t)\rangle$) without using gates.

$$|\Psi(t_f)\rangle = U(t_f, t_0) |\Psi(t_0)\rangle = U(t_f, t_f - 1) \dots U(t_2, t_1) U(t_1, t_0) |\Psi(t_0)\rangle$$

Lets divide the time evolution into N steps of size ϵ , and keep track of each step with a state:

$$\begin{array}{c|c|c|c|c} |0\rangle & |1\rangle & \dots & |t + \epsilon\rangle & \dots & |t_f - 1\rangle \\ \hline |\Psi(t_0)\rangle & U(t_1, t_0) |\Psi(t_0)\rangle & \dots & U(t + \epsilon, t - \epsilon) |\Psi(t - \epsilon)\rangle & \dots & U(t_f, t_{n-1}) |\Psi(t_{n-1})\rangle \end{array}$$

Lets create a connection between the clock states $\{|t\rangle\}$ and the corresponding states of the physical system $|\Psi_t\rangle$. We can entangle each corresponding couple and then sum all couples in this way:

$$|\Theta\rangle = \frac{1}{\sqrt{N}} \left(|\Psi(t_0)\rangle \otimes |0\rangle + \dots + |\Psi(t_f - 1)\rangle \otimes |t_f - 1\rangle \right) = \frac{1}{\sqrt{N}} \sum_{t=t_0}^{t_f-1} |\Psi(t)\rangle \otimes |t\rangle \quad (1)$$

This state contains the entire time evolution from t_0 to t_f , hence the name history state.

* [McClean, Jarrod R. and Parkhill, John A. and Aspuru-Guzik, Alán. Proc. Natl. Acad. Sci. 2013 110 (41) E3901-E3909, [doi:10.1073/pnas.1308069110](https://doi.org/10.1073/pnas.1308069110)]

Time Evolution as a ground-state eigenvalue problem

- This approach to the time evolution problem goes under the name "Kitaev-Feynman clock states" *.

$$|\Theta\rangle = \frac{1}{\sqrt{T}} \left(|\Psi(t_0)\rangle \otimes |0\rangle + |\Psi(t_1)\rangle \otimes |1\rangle + \dots + |\Psi(T-1)\rangle \otimes |T-1\rangle \right) = \frac{1}{\sqrt{T}} \sum_t |\Psi(t)\rangle \otimes |t\rangle$$

The history state contains the entire time evolution from t_0 to t_f .

The Hamiltonian which maps the sequence of unitary time operators $\{U_i\}$ to the the sequence of clock states $\{|t\rangle\}$ and has the history state as eigenvector with eigenvalue 0 is:

$$\mathcal{H}_c = C_0 + \frac{1}{2} \sum_t \left(I \otimes |t\rangle \langle t| - U_t \otimes |t+\epsilon\rangle \langle t| - U_t^\dagger \otimes |t\rangle \langle t+\epsilon| + I \otimes |t+\epsilon\rangle \langle t+\epsilon| \right)$$

where $U_t = e^{-i\epsilon H_t}$ is the time evolution operator from time t to time $t+\epsilon$

C_0 is used to specify the initial state at time $t=0$.

- We transformed H in purely imaginary so that H_c is real.

* [McClean, Jarrod R. and Parkhill, John A. and Aspuru-Guzik, Alán. Proc. Natl. Acad. Sci. 2013 110 (41) E3901-E3909, [doi:10.1073/pnas.1308069110](https://doi.org/10.1073/pnas.1308069110)]

Backup Slides - Section III

IBM

Writing the time evolution operator in gates

The time evolution operator $\exp(-iHt)$ can be approximated for small time-step dt using the second-order Suzuki-Trotter expansion*:

$$e^{-iHt} = e^{-i\sum_{j=1}^m H_j t} = \left(\prod_{j=1}^m e^{-iH_j dt/2} \prod_{j=m}^1 e^{-iH_j dt/2} \right)^{N_t} + O(m^3 t N_t dt^3)$$

Writing the time evolution operator in gates

The time evolution operator $\exp(-iHt)$ can be approximated for small time-step dt using the second-order Suzuki-Trotter expansion*:

$$e^{-iHt} = e^{-i\sum_{j=1}^m H_j t} = \left(\prod_{j=1}^m e^{-iH_j dt/2} \prod_{j=m}^1 e^{-iH_j dt/2} \right)^{N_t} + O(m^3 t N_t dt^3)$$

For the 2 plaquettes lattice Hamiltonian we obtain:

$$\begin{aligned} e^{-iHt} \approx & e^{i(xt/4)Z_1 Y_0} e^{i(3t/16)Z_0 Z_1} e^{i(3xt/4)Y_0} e^{i(9t/16)Z_1} \times \\ & e^{i(9t/16)Z_0} e^{i(3xt/4)Y_1} e^{i(xt/4)Z_0 Y_1} e^{i(xt/4)Z_0 Y_1} \times \\ & e^{i(3xt/4)Y_1} e^{i(9t/16)Z_0} e^{i(9t/16)Z_1} e^{i(3xt/4)Y_0} \times \\ & e^{i(3t/16)Z_0 Z_1} e^{i(xt/4)Z_1 Y_0} \end{aligned}$$

Fundamental gate identities:

- $e^{-i\theta Z_j} = RZ_j(2\theta)$
- $e^{-i\theta Y_j} = RY_j(2\theta)$
- $e^{-i\theta Z_j Z_k} = CX_{jk} RZ_k(2\theta) CX_{jk}$
- $e^{-i\theta Z_j Y_k} = CX_{jk} RY_k(2\theta) CX_{jk}$

$$\begin{aligned} & CX_{10} RY_0(-\frac{1}{2}xt) RZ_0(-\frac{3}{8}t) CX_{10} RY_0(-\frac{3}{2}xt) RZ_0(-\frac{9}{4}t) RZ_1(-\frac{9}{3}t) RY_1(-3xt) CX_{01} RY_1(-xt) CX_{01} \\ & RY_0(-\frac{3}{2}xt) RZ_1(-\frac{9}{8}t) CX_{10} RZ_0(-\frac{3}{8}t) RY_0(-\frac{1}{2}xt) CX_{10} \end{aligned}$$

IBM superconducting gate-based universal quantum computer

- Each hardware has its own native gates that form a universal base. [Solovay-Kitaev theorem]
- In encoding the theory on a hardware one has to consider the error rate of its gates and the geometry of the qubit's connectivity:
 - Gates error rate:

Gates	Error
RZ, Z, X, Y	$\sim (0.01 - 0.05)\%$
Readout	$\sim (0.5 - 3.0)\%$
CNOT	$\sim (0.5 - 2.5)\%$

IBM superconducting gate-based universal quantum computer

- Each hardware has its own native gates that form a universal base. [Solovay-Kitaev theorem]
- In encoding the theory on a hardware one has to consider the error rate of its gates and the geometry of the qubit's connectivity:

- Gates error rate:

- Hardware and qubit connectivity geometry:

Gates	Error
RZ, Z, X, Y	$\sim (0.01 - 0.05)\%$
Readout	$\sim (0.5 - 3.0)\%$
CNOT	$\sim (0.5 - 2.5)\%$

ibmq-manila 5 qubits



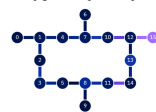
ibmq-belem 5 qubits



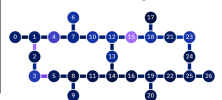
ibmq-lagos 7 qubits



ibmq-guadalupe 16 qubits



ibmq-hanoi 27 qubits



New systems with 127, 133, and 156 qubits.

IBM superconducting gate-based universal quantum computer

- Each hardware has its own native gates that form a universal base. [Solovay-Kitaev theorem]
- In encoding the theory on a hardware one has to consider the error rate of its gates and the geometry of the qubit's connectivity:

- Gates error rate:

- Hardware and qubit connectivity geometry:

Gates	Error
RZ, Z, X, Y	$\sim (0.01 - 0.05)\%$
Readout	$\sim (0.5 - 3.0)\%$
CNOT	$\sim (0.5 - 2.5)\%$

ibmq-manila 5 qubits



ibmq-belem 5 qubits



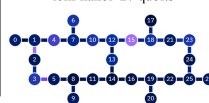
ibmq-lagos 7 qubits



ibmq-guadalupe 16 qubits



ibmq-hanoi 27 qubits



New systems with 127, 133, and 156 qubits.

- Keep the number of CNOT as low as possible!
- For LGT theories the error rate should be reduced and the qubit connectivity increased.

Going toward a larger lattice: The 5 plaquettes case with $j_{max} = 1/2$

From the N plaquettes Hamiltonian written using gates, the 5 plaquettes case is obtained with $N=5$:

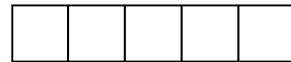
$$H = \frac{g^2}{2}(h_E + h_B),$$

$$h_E = \frac{3}{8}(3N + 1) - \frac{9}{8}(Z_0 + Z_{N-1}) - \frac{3}{4} \sum_{n=1}^{N-2} Z_n$$

$$- \frac{3}{8} \sum_{n=0}^{N-2} Z_n Z_{n+1},$$

$$h_B = -\frac{\chi}{2}(3 + Z_1)X_0 - \frac{\chi}{2}(3 + Z_{N-2})X_{N-1} \\ - \frac{\chi}{8} \sum_{n=1}^{N-2} (9 + 3Z_{n-1} + 3Z_{n+1} + Z_{n-1}Z_{n+1})X_n.$$

5 plaquettes lattice:

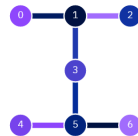


Initial state used:



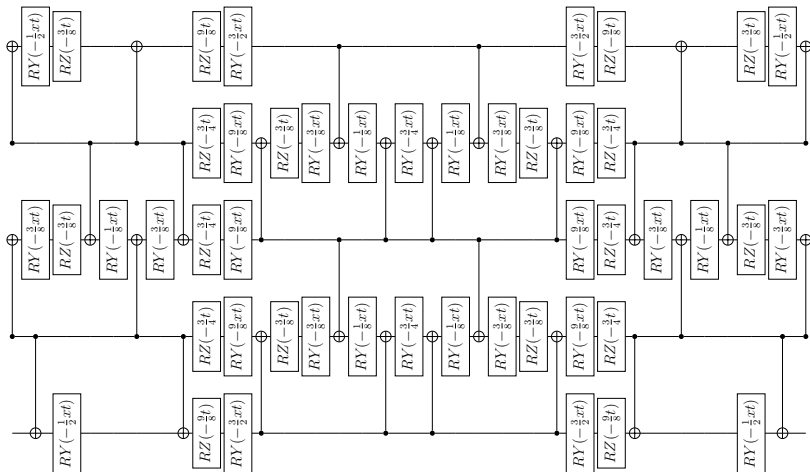
Dashed line $j = 0$ solid tick line $j = 1/2$

- A 7 qubits hardware such as `ibm_lagos` has only 5 qubits chain with nearest-neighbor connectivity, therefore 5 plaquettes is the current largest lattice.



Time evolution circuit for the 5 plaquettes case

- The corresponding time evolution circuit of a single second order Suzuki-Trotter step for the 5-plaquettes lattice is:



The code makes some optimization:

- The edge CNOTs cancel out, leaving 22 CNOTs per Trotter step.
- The edge rotation gates are combined in one gate.

Measurement protocol

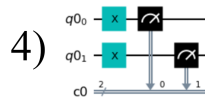
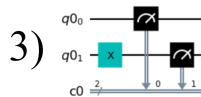
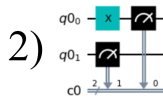
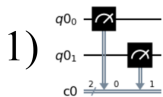
- On the hardware `ibm_lagos` 300 circuits with 10^4 hits can be submitted as a single list.
- Submit the 2^n circuits for the mitigation of measurement error.
(4 circuits for the 2 plaquettes case), (32 circuits for the 5 plaquettes case)
- Submit the randomized compiling circuits for the error-estimation circuits and the physics circuits
(148 error-estimation circuits and 148 physics circuits for 2 plaquettes case)
(134 error-estimation circuits and 134 physics circuits for 5 plaquettes case)
- Collect all the measurements
- Apply the measurement-error calibration matrix to the error-estimation and physics circuits results
- Use the self-mitigation equation to mitigate the hardware error on the physics result:

$$\left. \frac{P_{\text{true}} - \frac{1}{2}}{P_{\text{computed}} - \frac{1}{2}} \right|_{\text{physics run}} = \left. \frac{P_{\text{true}} - \frac{1}{2}}{P_{\text{computed}} - \frac{1}{2}} \right|_{\text{mitigation run}}$$

- Calculate the error of the error-estimation and physics results as the sum in quadrature of the statistical error from the 10^4 hits and the 1480 bootstrap samples.

Error mitigation technique: Mitigation of measurement error

- Working with n qubits there are 2^n possible states
- Each state should be recreated with a circuit and measured to construct the $2^n \times 2^n$ calibration matrix, whose entries are the probabilities that a particular states once it is measured has a superposition with another state.
- In the case of a 2 plaquette lattice we use 2 qubits, therefore there are 4 mitigation circuits:



A calibration matrix looks like :

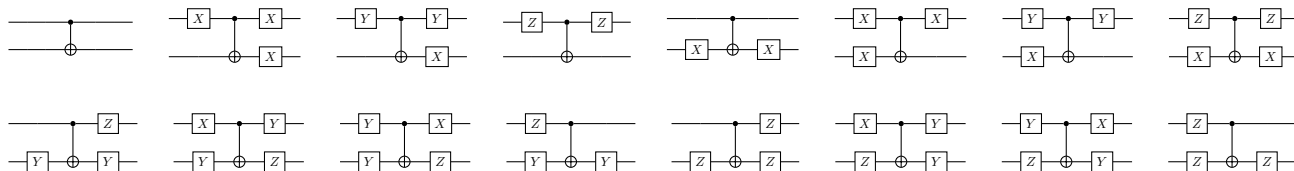
$$\begin{pmatrix} 0.9865 & 0.0131 & 0.0064 & 0.0003 \\ 0.0084 & 0.9817 & 0.0002 & 0.008 \\ 0.0049 & 0.0001 & 0.9832 & 0.0125 \\ 0.0002 & 0.0051 & 0.0102 & 0.9792 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

- No hardware errors $\Rightarrow$ the matrix is diagonal otherwise there are off-diagonal elements.

- The calibration matrix is then applied to the measurements of the physics circuit by fitting for the most correct "rotated" state.

Error mitigation technique: Randomized compiling* (Pauli-twirling)

- Randomized compiling is a technique to transform the CNOT coherent noise into incoherent noise.
- Each CNOT in the physics circuit is randomly substituted with one of the following 16 identities:



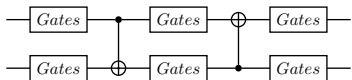
- The circuit should be run a sufficient number of times to randomly access a large part of identity combinations for each CNOT gate present in the physics circuit.

*[Joel J. Wallman and Joseph Emerson, Phys. Rev. A **94** 052325 (2016), [doi: 10.1103/PhysRevA.94.052325](https://doi.org/10.1103/PhysRevA.94.052325)

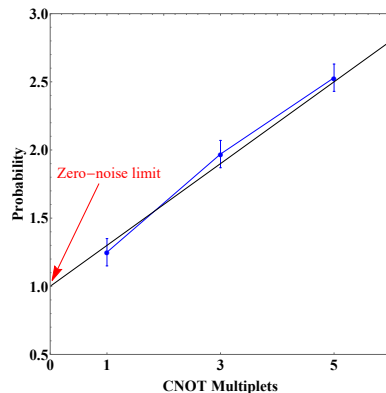
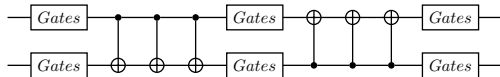
Error mitigation technique: Zero-noise extrapolation *

The method consists of studying the circuit noise by creating copies of the original circuits where the noise is artificially increased by replacing each CNOT gate by odd multiplets (triplet, quintet etc.) of CNOTS.

The original circuit:



A copy of the original circuit with each CNOT gate replaced by a CNOT triplet:



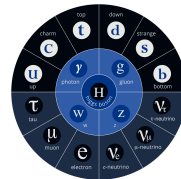
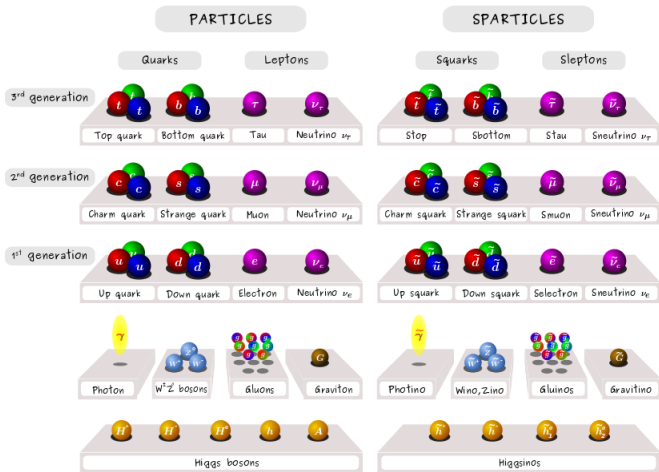
A possible linear extrapolation of the zero-noise limit.

- The zero-noise limit is extracted by fitting the circuits measurements with a function of the CNOT multiplets.

Backup Slides - Section IV

Supersymmetric Quantum Mechanics

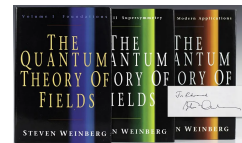
Why is supersymmetry such an interesting theoretical idea?



Extension of the Standard Model



Duality with Quantum gravity



Better knowledge of QFT

- Predicted **sparticles** have not been observed yet $\implies$ supersymmetry must be spontaneously broken!

Essential facts about supersymmetric models

- A supercharge Q introduces a symmetry between fermion and boson

$$Q|fermion\rangle = |boson\rangle \quad Q|boson\rangle = |fermion\rangle$$

- The Hamiltonian is $\sim Q^2$
- The energy spectrum has peculiar characteristics for broken/preserved supersymmetry:

Preserved symmetry

$$E_0 = \langle \Psi | H | \Psi \rangle = 0$$

Single ground state
Paired excited states

Spontaneously breaking

$$E_0 = \langle \Psi | H | \Psi \rangle > 0$$

All the states are paired

Essential facts about supersymmetric models

- A supercharge Q introduces a symmetry between fermion and boson

$$Q|fermion\rangle = |boson\rangle \quad Q|boson\rangle = |fermion\rangle$$

- The Hamiltonian is $\sim Q^2$
- The energy spectrum has peculiar characteristics for broken/preserved supersymmetry:

Preserved symmetry

$$E_0 = \langle \Psi | H | \Psi \rangle = 0$$

Single ground state

Paired excited states

Spontaneously breaking

$$E_0 = \langle \Psi | H | \Psi \rangle > 0$$

All the states are paired

- The Witten index expresses a condition necessary but insufficient for supersymmetry breaking:

$$\mathcal{W}_{\text{index}} = \text{Tr}[(-1)^F e^{-iHt}] = 0 \quad \implies \quad \mathcal{Z} = 0 \quad \implies \quad \text{severe Sign problem!}$$

Representing the Hamiltonian as a sum of Pauli strings

$$H(\hat{p}, \hat{q}) = \frac{1}{2} (\hat{p}^2 + [W'(\hat{q})]^2 + W''(\hat{q})Z) \longrightarrow H = \sum_i c_i P_i$$

Representing the Hamiltonian as a sum of Pauli strings

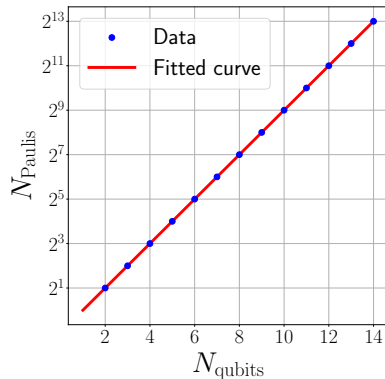
$$H(\hat{p}, \hat{q}) = \frac{1}{2} (\hat{p}^2 + [W'(\hat{q})]^2 + W''(\hat{q})Z) \longrightarrow H = \sum_i c_i P_i$$

Representing the Hamiltonian as a sum of Pauli strings

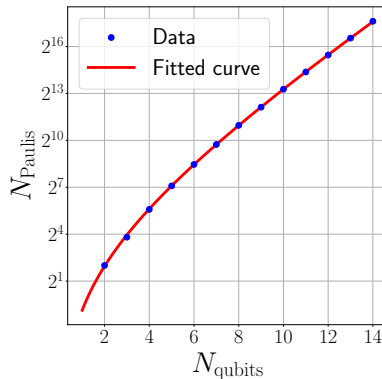
$$H(\hat{p}, \hat{q}) = \frac{1}{2} (\hat{p}^2 + [W'(\hat{q})]^2 + W''(\hat{q})Z) \longrightarrow H = \sum_i c_i P_i$$

- The number of Pauli strings grows exponentially with the number of qubits!

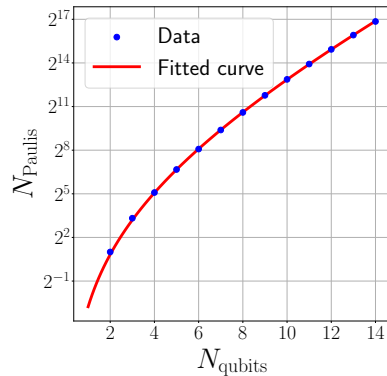
Harmonic Oscillator



Double Well



Anharmonic Oscillator

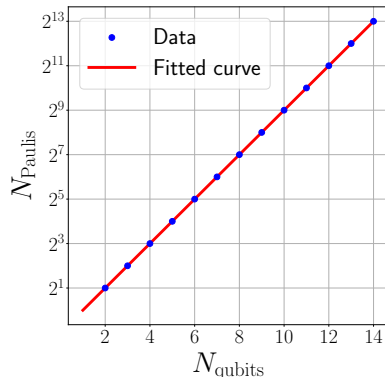


Representing the Hamiltonian as a sum of Pauli strings

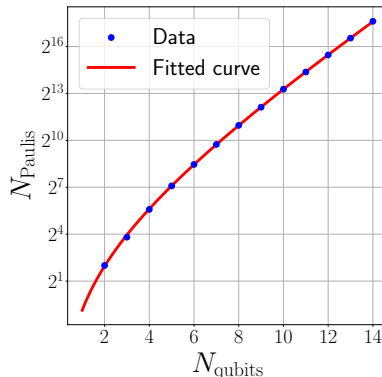
$$H(\hat{p}, \hat{q}) = \frac{1}{2} (\hat{p}^2 + [W'(\hat{q})]^2 + W''(\hat{q})Z) \longrightarrow H = \sum_i c_i P_i$$

- The number of Pauli strings grows exponentially with the number of qubits!

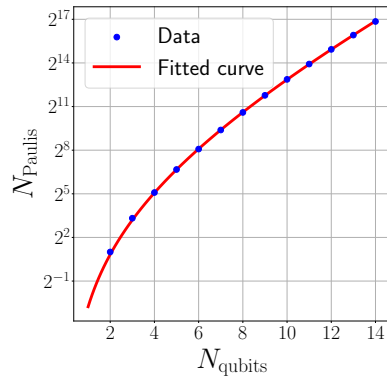
Harmonic Oscillator



Double Well



Anharmonic Oscillator



Possible solutions:

- Use position or momentum basis and connect them with a Quantum Fourier Transform
- Trapped ion hardware, ion modes for boson and ion spin for fermion

Lessons learned from these experimentations

Ansatz

- Are there enough parameters, or too many?
- Is there enough entanglement, or too much?
- Extrapolate from a small-system ansatz

Classical optimizer

- Local optimizers tend to get stuck
- Global optimizers are not too expensive
- Hardware noise helps overcoming barren plateaus or local minima

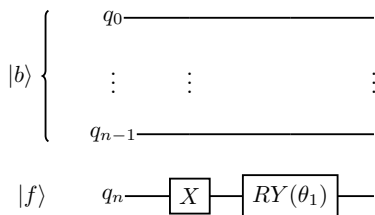
The Adaptive VQE algorithm

A simplified variant of ADAPT-VQE algorithm to construct an ansatz:

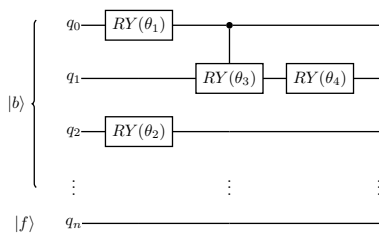
- 1 Define an initial ansatz $|\psi_{\text{ansatz}}\rangle$
- 2 Select an operator pool $\{\hat{O}\} = \{RY, RZ, CRY\}$
- 3 $\forall \hat{O}_i$ measure the gradient $\nabla \langle \psi_{\text{ansatz}} | \hat{O}_i(\theta)^\dagger \hat{H} \hat{O}_i(\theta) | \psi_{\text{ansatz}} \rangle$
- 4 Append $\hat{O}_i$ with largest gradient to the ansatz
- 5 VQE to measure energy using the ansatz
- 6 If $\Delta E < \text{threshold}$ stop, else return to step 3

Full Adaptive-VQE constructed ansätze

HO



DW



AHO

