Real time evolution of a SU(2) pure gauge lattice theory on a IBM quantum hardware. †

Emanuele Mendicelli
Department of Mathematical Sciences, Theoretical Physics, University of Liverpool

Workshop: Towards quantum simulation of gauge/gravity duality and lattice gauge theory

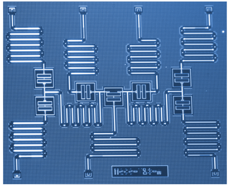
6th March 2024, Queen Mary University of London

† Based on works done at York University, Toronto, Canada:

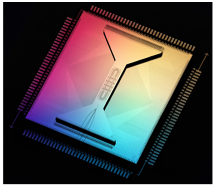
[S. A. Rahman, R. Lewis, E. Mendicelli, and S. Powell, "Self-mitigating Trotter circuits for SU(2) lattice gauge theory on a quantum computer", Phys. Rev. D 106, 074502, 2022, arXiv:2205.09247]

Quantum computer and qubit


IBM

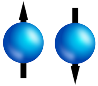

IBM 7 qubits

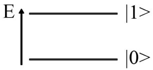

D-Wave


Quantinuum

- What is a quantum computer?
  It is a hardware that makes it possible to calculate or simulate by exploiting quantum mechanical properties like superposition and entanglement which are not present in classical systems.

- What is a qubit?
  A qubit is the fundamental unit of quantum information, in a certain way it is the quantum version of the bit. It can be realized by a spin-1/2 or a generic two level quantum system:

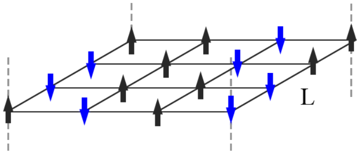




Different than a bit, a qubit can be in any superposition of its two states, and can be entangled with another qubit, meaning that a measure of the state of one qubit affects the state of the other qubit.

Why do we like qubits? Classical Vs quantum resources

Let's consider a simple Ising model system with spin-1/2



Let's estimate the computational resources needed to study a system with N spins:

$$Z = \int dx\, dy\, dz\, e^{-S} \longrightarrow \approx 2^N = 2^{L^3} = 2^V$$

Classically we need 2^N resources, memory slots, instead if we use qubits, it is just N qubits

---

Let's double the system: N spins $\longrightarrow$ $2N$ spins

Classically

- $2^N \longrightarrow \left(2^{2N}\right) = \left(2^N\right)^2$

Quantum

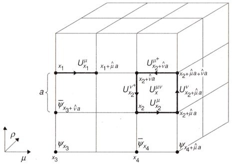
- N qubits $\longrightarrow$ $2N$ qubits

(escape importance sampling Monte Carlo)

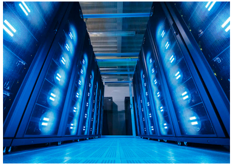
Technologically it is common to increase the hardware resources linearly, but not quadratically!

Lattice Quantum Chromodynamics (LQCD) in a nutshell

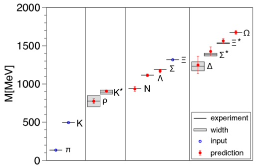
- The interaction between quarks and gluons has many phenomena that cannot be studied analytically.
- Lattice gauge theory is a theoretical approach in which theoretical models are directly formulated to be studied through simulations on supercomputers using the importance sampling Monte Carlo method:



Discrete spacetime



Classical Supercomputer



Predict particle masses

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

- 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!!!

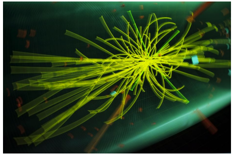
(Sign problem $\Longrightarrow$ The numerical calculation requires an exponentially large amount of resources!)

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

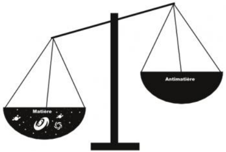
Since the late '70s many phenomena are inaccessible to LQCD simulations due to the infamous sign problem:

Real-time evolution

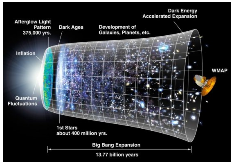
nonzero chemical potential



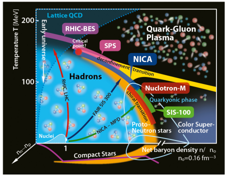
propagations/collisions



Imbalance matter-antimatter



Early Universe



QCD phase diagram

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

Challenges in the use of the available quantum hardware:

- Few qubits with low connectivity
- Errors in reading the qubits
- Noisy gates due to imperfect realization
- Low computation resources
- No error correction techniques available on current hardware $\Rightarrow$ **Only error mitigation techniques**!!!

Presentation Plan

- Section I: SU(2) pure gauge lattice theory
- Section II: IBM superconducting gate-based universal quantum computer
- Conclusions

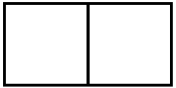
Hamiltonian for SU(2) pure gauge lattice theory

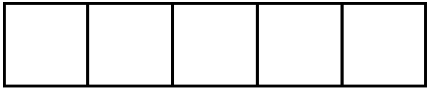
$$\mathcal{L} = -\frac{1}{4} F_{\mu\nu}(x) F^{\mu\nu}(x)$$

$$\Downarrow$$

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

We used a single-row lattice with few plaquettes:





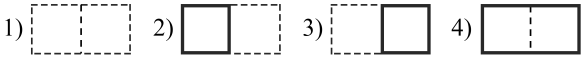
2 and 5 plaquettes with closed boundary conditions.

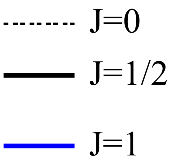
- g is the gauge coupling and $x \equiv 2/g^4$.
- $\hat{E}_i^2$ is the chromoelectric field for the ith lattice link. (Returns the chromoelectric energy stored on the lattice)
- $\hat{\Box}_i$ is the plaquette operator trace of the product of four gauge link operators of the ith plaquette. (Adds or subtracts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

States of the theory on 2 plaquettes

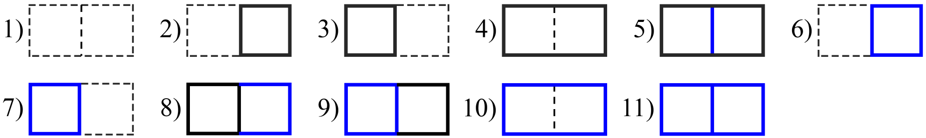
- The physical states can obtained by applying the plaquette operators on the vacuum, state 1), and by respecting the SU(2) Gauss's law at each node:

- Fixing the maximum energy flux to $j_{max} = 1/2$, there are 4 states:





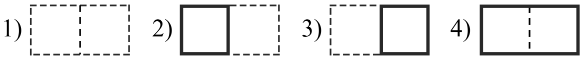
- While for $j_{max} = 1$, there are 11 states:



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

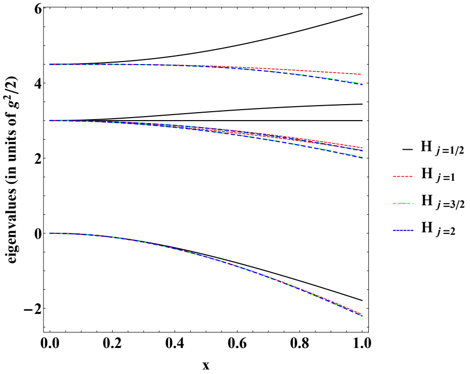
Representing the Hamiltonian for 2 plaquettes

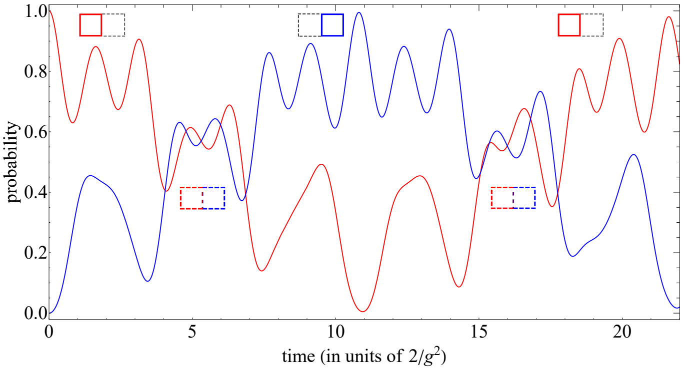
- Let's use the states to represent the Hamiltonian:



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

- 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?**

Section II: IBM superconducting gate-based universal quantum computer

IBM gate-based quantum computer

There are many gates available like the Pauli matrices X, Y, Z, rotations RX, RY, RZ, the $CNOT$ and few more. Some of the most common are:

Gate	Symbol	Matrix Representation
Z-Gate		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard-Gate		$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
RY-Gate		$\begin{pmatrix} cos(\theta/2) & -sin(\theta/2) \\ sin(\theta/2) & cos(\theta/2) \end{pmatrix}$
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 is: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- The Hadamard gate changes basis from z to x : $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
- The CNOT gate acts on two qubits, applies a X on the second qubit when the first is $|1\rangle$:

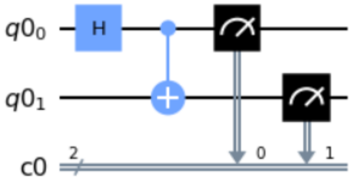
$$\boldsymbol{CNOT|0\rangle|0\rangle = |0\rangle|0\rangle} \text{ and } \boldsymbol{CNOT|1\rangle|0\rangle = |1\rangle|1\rangle}$$

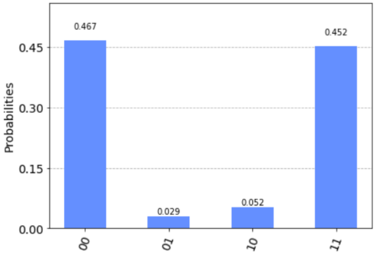
An example of using a IBM hardware: Create and measure a Bell state

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

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

```
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_o
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]

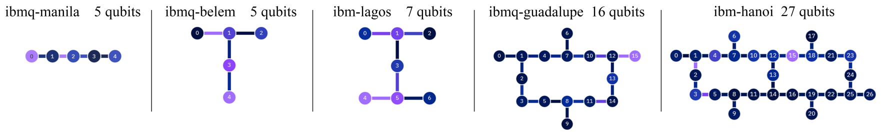
IBM superconducting gate-based universal quantum computer

- Each hardware has is 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	$\sim (0.5 - 2.5)\%$

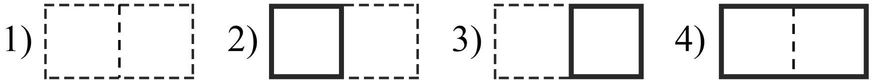
  - Hardware and qubit connectivity geometry:



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

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:

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

- The Hamiltonian representation can be rewritten in gates as:

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

$$= \frac{3}{8}(7I_1 \otimes I_0 - 3I_1 \otimes Z_0 - Z_1 \otimes Z_0 - 3Z_1 \otimes I_0) - \frac{x}{2}(3I_1 + Z_1) \otimes X_0 - \frac{x}{2}X_1 \otimes (3I_0 + Z_0)$$

Writing the time evolution operator for the 2-plaquette case in gates

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

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

For the 2 plaquettes lattice Hamiltonian we obtain:

$$\begin{aligned} e^{-iHt} \approx\; & e^{i(xt/4)Z_1Y_0} e^{i(3t/16)Z_0Z_1} e^{i(3xt/4)Y_0} e^{i(9t/16)Z_1} \times \\ & e^{i(9t/16)Z_0} e^{i(3xt/4)Y_1} e^{i(xt/4)Z_0Y_1} e^{i(xt/4)Z_0Y_1} \times \\ & e^{i(3xt/4)Y_1} e^{i(9t/16)Z_0} e^{i(9t/16)Z_1} e^{i(3xt/4)Y_0} \times \\ & e^{i(3t/16)Z_0Z_1} e^{i(xt/4)Z_1Y_0} \end{aligned}$$

Fundamental gate identities:

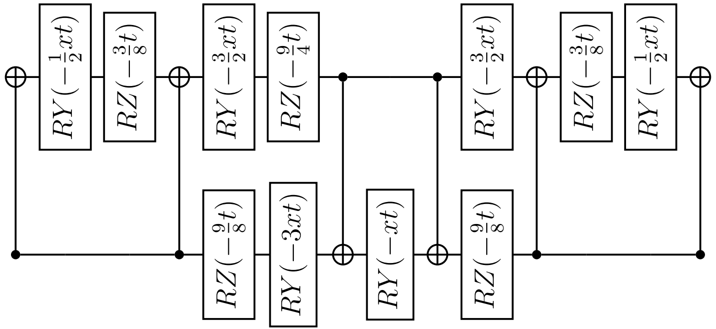
- $e^{-i\theta Z_j} = RZ_j(2\theta)$
- $e^{-i\theta Y_j} = RY_j(2\theta)$
- $e^{-i\theta Z_jZ_k} = CX_{jk}RZ_k(2\theta)CX_{jk}$
- $e^{-i\theta Z_jY_k} = CX_{jk}RY_k(2\theta)CX_{jk}$

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

*[Naomichi Hatano and Masuo Suzuki, Lect. Notes Phys. **679**, 37 (2005), 2005, pp. 37–68., doi:10.1007/11526216-2]

Time evolution circuit 2-plaquette case

- The corresponding circuit of a single second-order Suzuki-Trotter step for the 2-plaquettes lattice is:



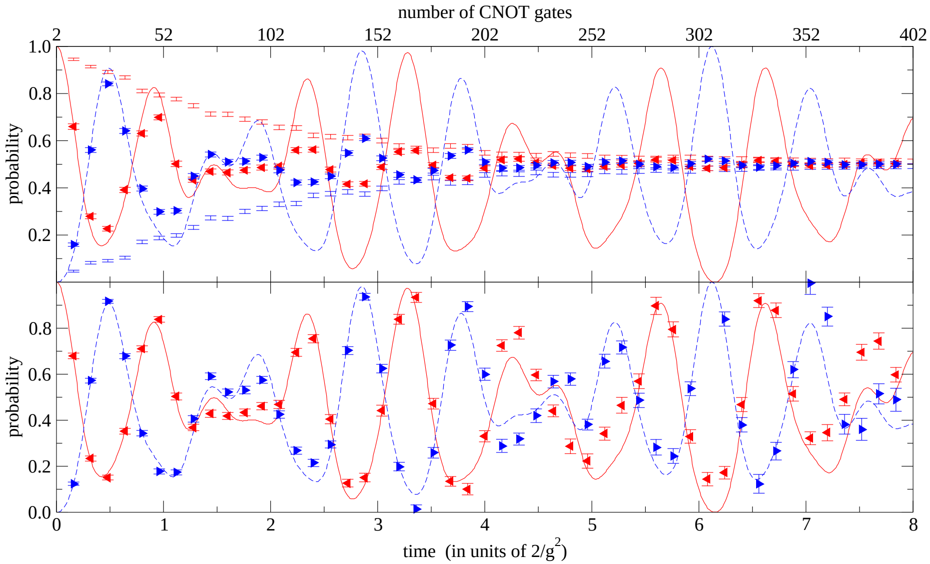
Our code makes some optimizations:

- The edge CNOTs cancel out, leaving 4 CNOTs per Trotter step.
- The edge rotation gates are combined in one gate.

To obtain the time evolution of the 2-plaquette system we need to:

- Prepare an initial state
- Apply N times the time evolution circuit to reach time Ndt

Results time evolution 2 plaquettes case

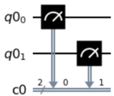


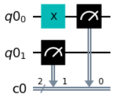
- Initial state: left plaquette "on" $j = 1/2$.
- Upper panel "raw data" after measurement error mitigation and randomized compiling (Pauli-twirling)
- Lower panel: final data, using our approach called "Self-mitigation"

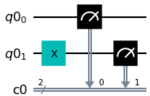
- $x = 2.0$; dt=0.08; $P\,[2) \to 2)] + P\,[2) \to 4)]$, $P\,[2) \to 3)] + P\,[2) \to 4)]$
- red solid (blue dashed) curve exact probability left (right) plaquette has $j = 1/2$
- red and blue triangles (error bars) are physics data (mitigation data) from the ibm_lagos

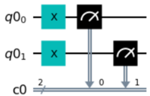
Error mitigation technique: Mitigation of measurement error

- 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:

1) 

2) 

3) 

4) 

A calibration matrix looks like :

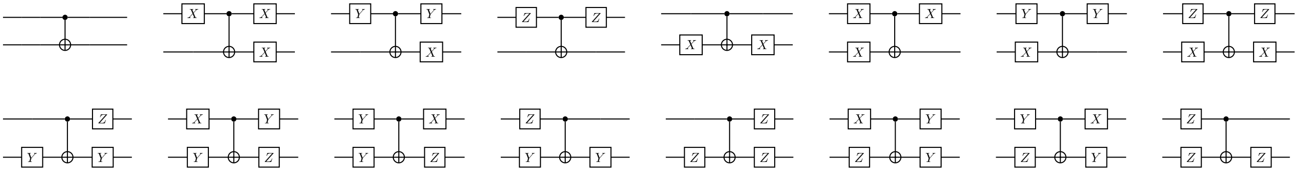
$$\begin{pmatrix} 0.9865 & 0.0131 & 0.0064 & 0.0003 \\ 0.0084 & 0.9817 & 0.0002 & 0.008 \\ 0.0049 & 0.0001 & 0.9832 & 0.0125 \\ 0.0002 & 0.0051 & 0.0102 & 0.9792 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

- 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:



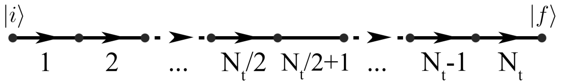
- The circuit should be run a sufficient number of times to randomly access a large part of identity combinations for each CNOT gate present in the physics circuit.

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

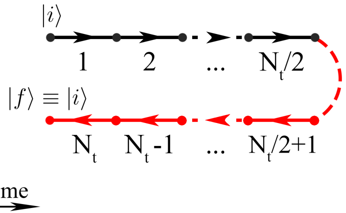
Error mitigation technique: Self-mitigation [1/2]

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



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

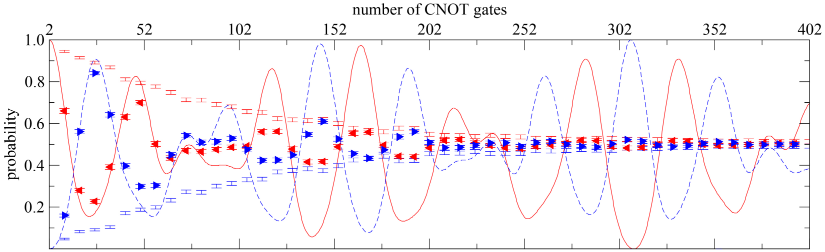


- Our error estimation circuit has the same gates, in the same order and the identical or opposite variables inside the rotations gates, and therefore it closely reproduces the physics circuit noise.

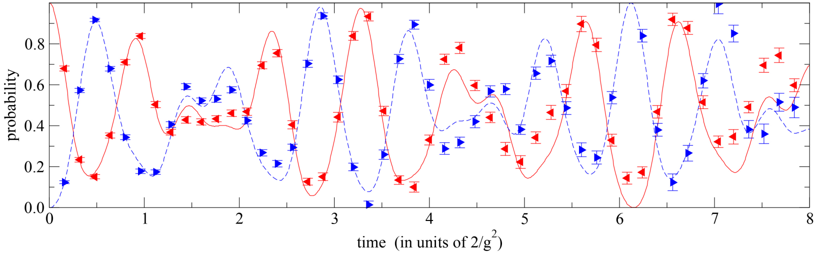
- The hardware error can be estimated in how far the final state of the error estimation circuit is measured from the initial state.

*Inspire by [Miroslav Urbanek, Benjamin Nachman, Vincent R. Pascuzzi, Andre He, Christian W. Bauer, and Wibe A. de Jong, Phys. Rev. Lett. 127, 270502]

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

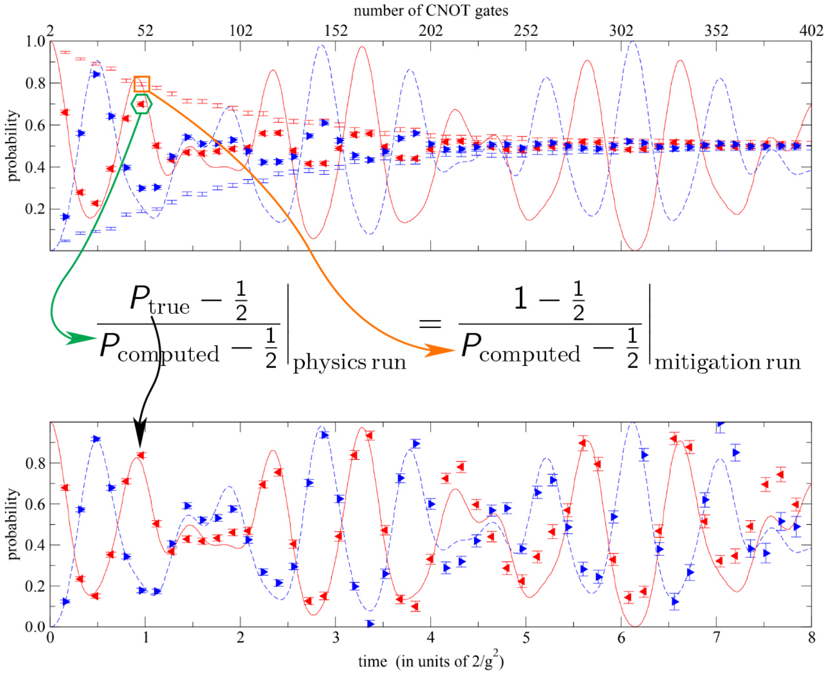


$$\left.\frac{P_{\text{true}} - \frac{1}{2}}{P_{\text{computed}} - \frac{1}{2}}\right|_{\text{physics run}} = \left.\frac{P_{\text{true}} - \frac{1}{2}}{P_{\text{computed}} - \frac{1}{2}}\right|_{\text{mitigation run}}$$



- 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`.

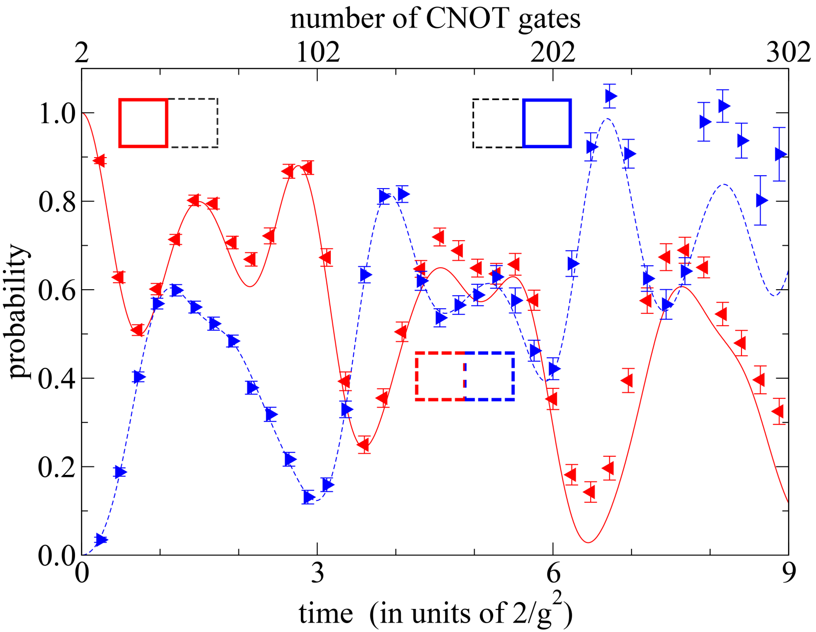
Measurement protocol

- 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.

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 $j = 1/2$, $P\,[2) \to 2)] + P\,[2) \to 4)]$
- blue dashed curve exact probability right plaquette has $j = 1/2$, $P\,[2) \to 3)] + P\,[2) \to 4)]$
- red and blue triangles are calculations on the `ibm_lagos`.

- For $x < 1$ the dominant states are the low energy ones and these are the single-plaquette states, therefore traveling excitations across the lattice are visible.

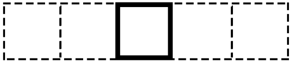
Going toward a larger lattice: The 5 plaquettes case with $j_{max} = 1/2$

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

$$
\begin{aligned}
H &= \frac{g^2}{2}(h_E + h_B)\,, \\
h_E &= \frac{3}{8}(3N+1) - \frac{9}{8}(Z_0 + Z_{N-1}) - \frac{3}{4}\sum_{n=1}^{N-2} Z_n \\
&\quad - \frac{3}{8}\sum_{n=0}^{N-2} Z_n Z_{n+1}\,, \\
h_B &= -\frac{x}{2}(3 + Z_1)X_0 - \frac{x}{2}(3 + Z_{N-2})X_{N-1} \\
&\quad - \frac{x}{8}\sum_{n=1}^{N-2}(9 + 3Z_{n-1} + 3Z_{n+1} + Z_{n-1}Z_{n+1})X_n\,.
\end{aligned}
$$

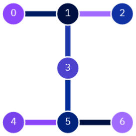
5 plaquettes lattice:

Initial state used:



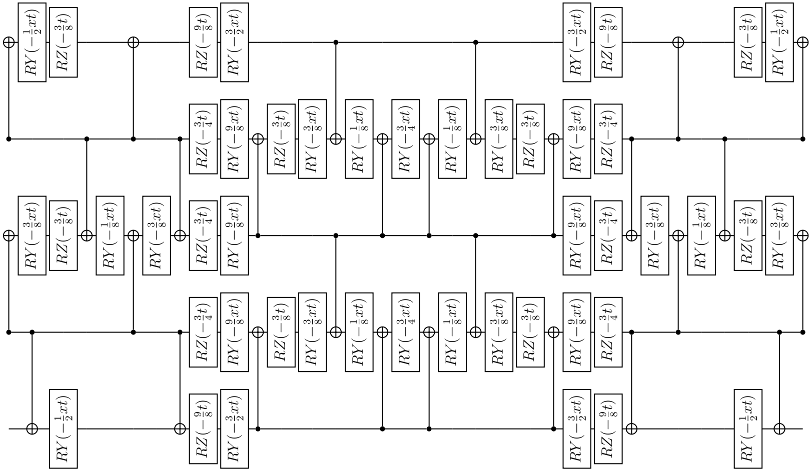
Dashed line $j = 0$ solid tick line $j = 1/2$

- 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

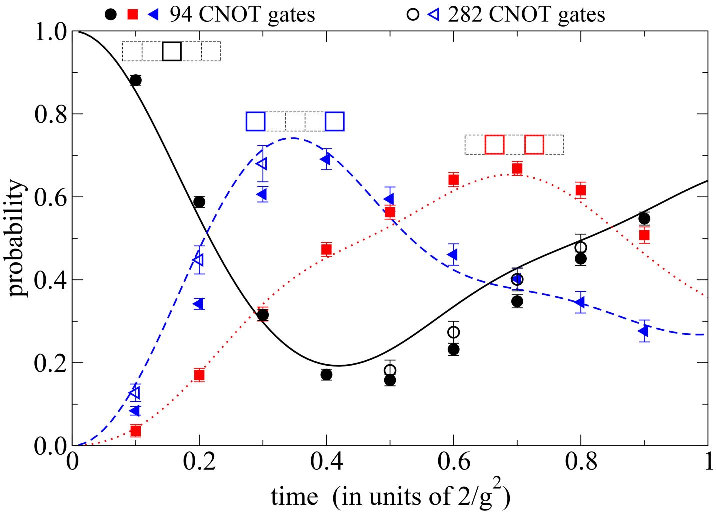
- The corresponding time evolution circuit of a single second order Suzuki-Trotter step for the 5-plaquettes lattice is:



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.

Going toward a longer lattice: 5 plaquettes



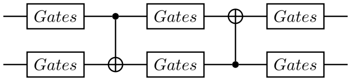
- Initial state: center plaquette "on" $j = 1/2$
- $x = 2.0$, 4 Trotter steps, various time step sizes;
- full symbols are calculations on the `ibm_lagos` after self-mitigation.
- open symbols are obtained using zero-noise extrapolation with 3 CNOTs every CNOT.

- 5 plaquettes is the largest lattice that can be implemented on a 7-qubits hardware without swap gates due to qubits connectivity.
- Zero-noise extrapolation consists of creating new circuits with extra CNOT gates and then fitting the result to extract the zero noise limit.

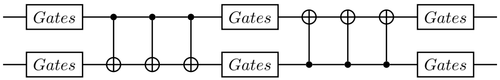
Error mitigation technique: Zero-noise extrapolation *

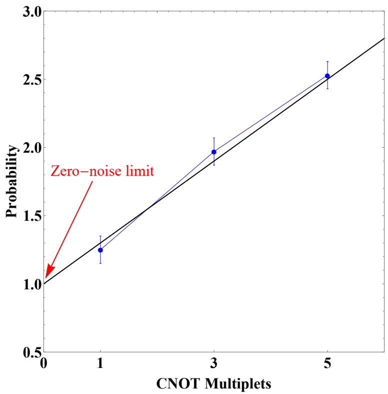
The method consists of studying the circuit noise by creating copies of the original circuits where the noise is artificially increased by replacing each CNOT gate by odd multiplets (triplet, quintet etc.) of CNOTS.

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.

*[Y. Li and S.C. Benjamin, "Efficient Variational Quantum Simulator Incorporating Active Error Minimization", Phys. Rev X **7** 021050 (2017), doi:10.1103/PhysRevX.7.021050]

Take-home message from our works

- The IBM gate based quantum computer can be successfully used to study the real-time evolution of a non-Abelian lattice gauge theory on a small lattice size.
- The real-time evolution study was extended to a time range much larger than previous study on non-Abelian gauge theories.
- Real-time dynamical process like local excitations moving across the lattice were observed.
- Simple error mitigation techniques like measurement mitigation, randomized compiling, zero-noise extrapolation and self-mitigation can be used to extend the usability of NISQ quantum hardware.
- Future studies should address how quantum hardware can be better used to investigate lattice gauge theories on larger lattice size and with larger gauge truncation.

Current challenges in encoding LGT on NISQ quantum hardware:

- How can we encode a lattice gauge theory on a quantum hardware?

$$\Downarrow$$

- What basis should be used to represent the Hamiltonian?
- How can we efficiently prepare an initial state?
- How can we efficiently find the ground state?
- How can we efficiently perform the time evolution?
- Which error mitigation techniques can scale well with the quantum resources?

Thank you all so much for your time!

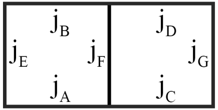
Time for Questions!

Backup Slides

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

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

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



$$= \left| E_A^B F_C^D G \right\rangle$$

$$|\psi\rangle = |j_E\rangle\,|j_A\rangle\,|j_B\rangle\,|j_F\rangle\,|j_C\rangle\,|j_D\rangle\,|j_G\rangle$$

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

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

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

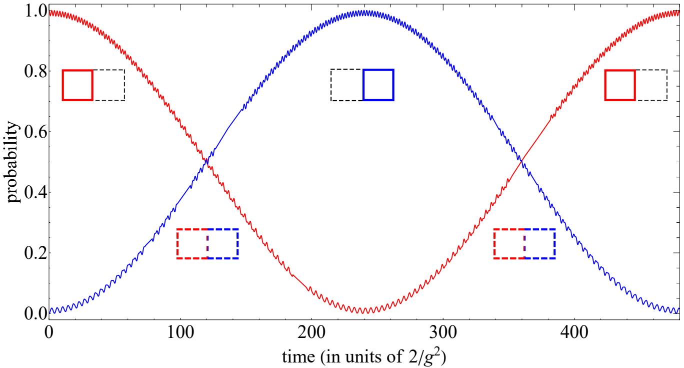
$\hat{\Box}_i$ is the plaquette operator trace of the product of four gauge link operators of the ith plaquette.

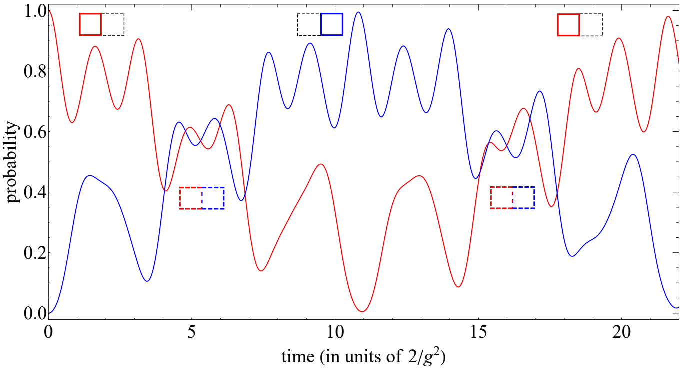
$$\begin{aligned}\langle\psi_{\text{final}}|\,\Box_1\,|\psi_{\text{initial}}\rangle \;=\;& (-1)^{j_A+j_B+j_C+j_D+2J_E+2J_F+2j_I+2j_J} \\ & \sqrt{2j_E+1}\sqrt{2J_E+1}\sqrt{2j_J+1}\sqrt{2J_J+1}\sqrt{2j_F+1}\sqrt{2J_F+1}\sqrt{2j_I+1}\sqrt{2J_I+1} \\ & \begin{Bmatrix} j_A & j_E & j_I \\ \frac{1}{2} & J_I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

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.

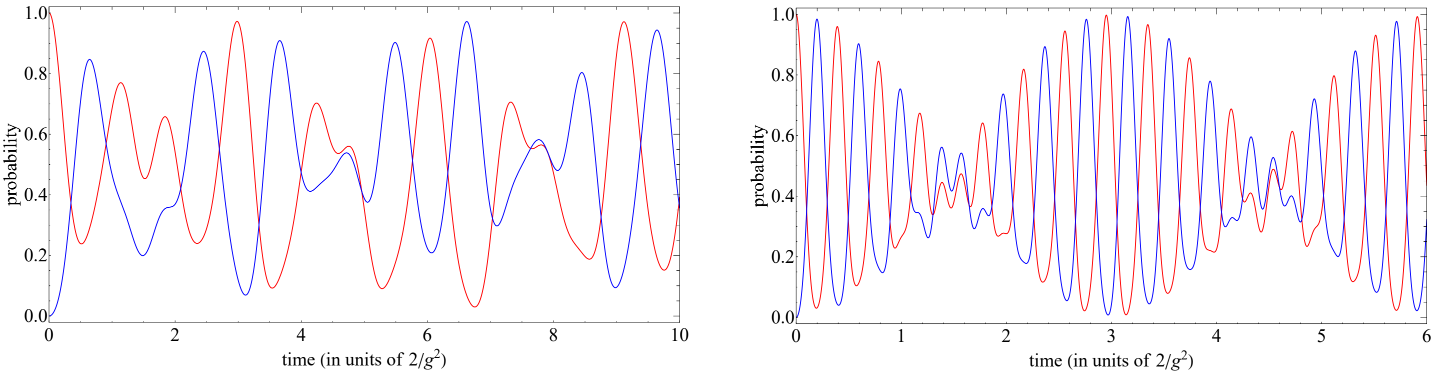




- The single-plaquette propagation time is larger for small x, diverging for $x = 0$, where the Hamiltonian is diagonal, containing only the chromoelectric term. The single-plaquettes are eigenstates, therefore they are constant in time.
- In the right figure, increasing x lets higher frequencies appear 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 $x = 5.0$