Quantum computing in Quantum Chemistry and Physics: Part 1

Ivano Tavernelli

IBM Quantum, IBM Research - Zurich, Switzerland

EASISchool 3
Genoa, Italy
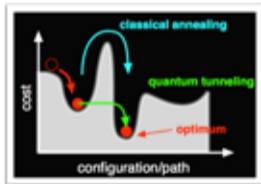
Sept 28th - Oct 9th , 2020

Types of Quantum Computing

Quantum Annealing

Optimization Problems

- Machine learning
- Fault analysis
- Resource optimization
- etc...

classical annealing

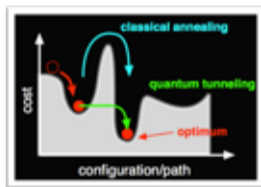quantum tunneling

optimum

cost

configuration/path

Many 'noisy' qubits can be built;
large problem class in optimization;
amount of quantum speedup unclear

Types of Quantum Computing

Quantum Annealing

Optimization Problems

- Machine learning
- Fault analysis
- Resource optimization
- etc...

classical annealing
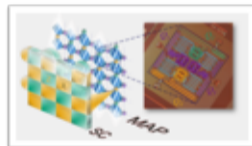
quantum tunneling

cost

optimum

configuration/path

Many 'noisy' qubits can be built; large problem class in optimization; amount of quantum speedup unclear

Fault-tolerant Universal Q-Comp.

Execution of Arbitrary Quantum Algorithms

- Algebraic algorithms (machine learning, cryptography,....)
- Combinatorial optimization
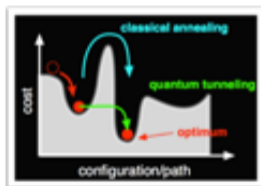- Digital simulation of quantum systems

Proven quantum speedup; error correction requires significant qubit overhead.

Types of Quantum Computing

Quantum Annealing

Optimization Problems

- Machine learning
- Fault analysis
- Resource optimization
- etc...

Many 'noisy' qubits can be built; large problem class in optimization; amount of quantum speedup unclear

Near-Term Quantum Comp.

Simulation of Quantum Systems, Optimization

- Quantum chemistry/physics
- Lattice field theory
- Material discovery
- Optimization (logistics, time scheduling,...)
- Machine Learning

Hybrid quantum-classical approach; already 50-100 "good" physical qubits could provide quantum speedup.

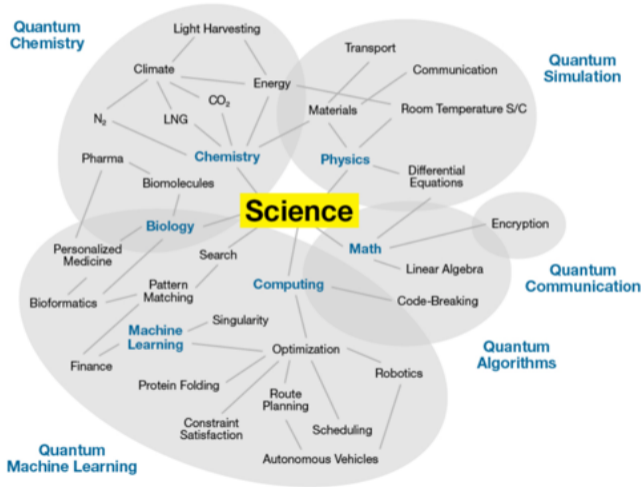## Fault-tolerant Universal Q-Comp.

Execution of Arbitrary Quantum Algorithms

- Algebraic algorithms (machine learning, cryptography,...)
- Combinatorial optimization
- Digital simulation of quantum systems
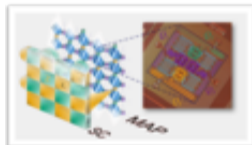
Proven quantum speedup; error correction requires significant qubit overhead.

Applications of quantum computing

Quantum computing may provide a new path to solve some of the hardest or most memory intensive problems in science and business.

Bloch Sphere representation

One qubit states: $|0\rangle$, $|1\rangle$

One qubit: linear superposition

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

with $\quad |\alpha_0|^2 + |\alpha_1|^2 = 1$

Shown here: $|\psi\rangle = 0.95|0\rangle + (0.18 + 0.25i)|1\rangle$

90.25% of the measurements give $|0\rangle$ and 9.75 % give $|1\rangle$.
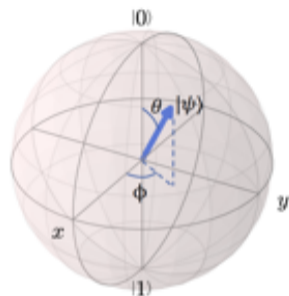
Two qubit states: $|00\rangle = |0\rangle\otimes|0\rangle$; $|01\rangle = |0\rangle\otimes|1\rangle$; $|10\rangle = |1\rangle\otimes|0\rangle$; $|11\rangle = |1\rangle\otimes|1\rangle$

One qubit gates

X gate (σ_x matrix)

bit-flip:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \begin{array}{l} |0\rangle \to |1\rangle \\ |1\rangle \to |0\rangle \end{array}$$

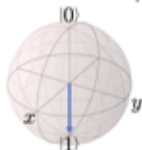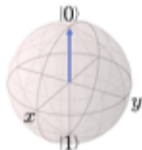Z gate (σ_z matrix)

phase-flip:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \begin{array}{l} |0\rangle \to |0\rangle \\ |1\rangle \to -|1\rangle \end{array}$$

Hadamard gate

(superposition):

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \begin{array}{l} |0\rangle \to \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle \to \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{array}$$
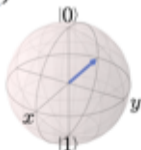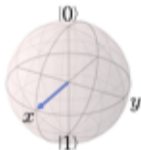
$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \ |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$
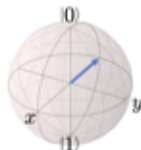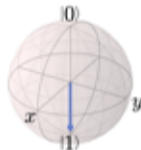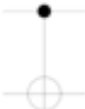
Two qubit gates

SWAP gate

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$|00\rangle \rightarrow |00\rangle; \ |01\rangle \rightarrow |10\rangle$
$|10\rangle \rightarrow |01\rangle; \ |11\rangle \rightarrow |11\rangle$

Preparation of the Bell's state

controlled NOT gate (CNOT)

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$|00\rangle \rightarrow |00\rangle; \ |01\rangle \rightarrow |01\rangle$
$|10\rangle \rightarrow |11\rangle; \ |11\rangle \rightarrow |10\rangle$

$$|\psi_{\text{Bell}}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Measurments

States and operators are expressed in a given basis. When not specified differently, the basis in each spin subsystem is the canonical (measurement) one made by the eigenstates of the σ_z operator: $\{|0\rangle_z, |1\rangle_z\} = \{|0\rangle, |1\rangle\}$.

An experiment like the one performed with Quantum Experience provides only probabilities for reading 0 and 1 (along the z-axes) for each qubit line:

$$P(0), \, P(1) \, .$$

For a multi-qubit experiment, e.g., a 2 qubit one, the measurement produces the probabilities:

$$P(00), \, P(01), \, P(10), \, P(11) \, .$$

All probabilities are evaluated by repeating the experiment a sufficient number of times.

Measurements

- Measure the expectation value $\langle\psi|\sigma_z|\psi\rangle$ for the state $|\psi\rangle = a|0\rangle + b|1\rangle$ (all in z-basis).

$$\langle\psi|\sigma_z|\psi\rangle = \begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$
$$= a^2 - b^2$$
$$= P(0) - P(1).$$

- For a two qubit state, like a product state $|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ we want to measure $\langle\Psi|\sigma_z \otimes \sigma_z|\Psi\rangle$

$$\langle\Psi|\sigma_z \otimes \sigma_z|\Psi\rangle = \langle\psi_1| \otimes \langle\psi_2|\sigma_z \otimes \sigma_z|\psi_1\rangle \otimes |\psi_2\rangle$$
$$= \langle\psi_1|\sigma_z|\psi_1\rangle\langle\psi_2|\sigma_z|\psi_2\rangle$$
$$= (a^2 - b^2)(c^2 - d^2) = a^2c^2 - a^2d^2 - b^2c^2 + b^2d^2$$
$$= P(00) - P(01) - P(10) + P(11)$$

where $|\psi_1\rangle = a|0\rangle + b|1\rangle$ and $|\psi_2\rangle = c|0\rangle + d|1\rangle$.

Measurements in different basis

Experiments can only measure along the z-axis. How do we compute expectation values like $\langle\psi|\sigma_x|\psi\rangle$?
We need to introduces rotations to map $x \to z$.

Assume we need to compute the expectation value of

$$\langle\psi|M|\psi\rangle$$

where M is the most general operator in SU(3), namely $M = \bar{\sigma} \cdot \bar{n}$ where $\bar{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ and
$\bar{n} = (n_x, n_y, n_z)$.
To measure the expectation value $\langle\psi|M|\psi\rangle$ we rotate $M = \bar{\sigma} \cdot \bar{n}$ so that it gets aligned to z.

$$\begin{aligned}
\langle\psi|\bar{\sigma} \cdot \bar{n}|\psi\rangle &= \langle\psi|R_n R_n^{-1} \, \bar{\sigma} \cdot \bar{n} \, R_n(\phi) R_n^{-1}(\phi)|\psi\rangle \\
&= \langle\psi|R_n(\phi)\sigma_z R_n^{-1}(\phi)|\psi\rangle \\
&= \langle\psi'|\sigma_z|\psi'\rangle
\end{aligned}$$

where $R_n(\phi)$ is the operator that rotates $\bar{n}$ into z and $|\psi'\rangle = R_n(\phi)|\psi\rangle$.

Measurements in different basis

Problem: Determine the matrix form for $R_n(\phi)$.

solution

$$R_n(\phi) = e^{-i\frac{\phi}{2}\,\vec{n}\cdot\vec{\sigma}} = \begin{pmatrix} \cos(\frac{\phi}{2}) - i n_z \sin(\frac{\phi}{2}) & (-i n_x - n_y)\sin(\frac{\phi}{2}) \\ (-i n_x + n_y)\sin(\frac{\phi}{2}) & \cos(\frac{\phi}{2}) + i n_z \sin(\frac{\phi}{2}) \end{pmatrix}$$

Decomposition of $R_n(\phi)$.
In order to implement $R_n(\phi)$ we need to decompose it in the fundamental gate set available on the device.
Standard universal (one qubit) gate set:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \; S = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}, \; T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

Measurements in different basis

Example: Evaluate $\langle \psi_{BS} | ZW | \psi_{BS} \rangle$ where $W = \frac{1}{\sqrt{2}}(\sigma_x + \sigma_z)$.

We have: $R = S H T^\dagger H$ with

$$S = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}, H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}, T^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{pmatrix}$$
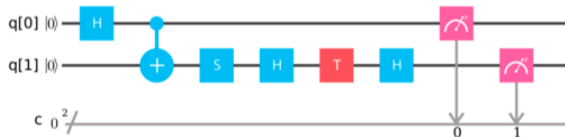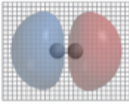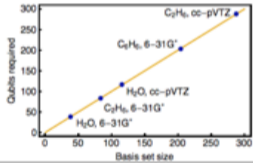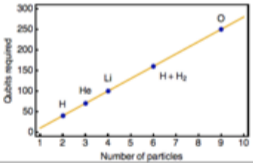
Corresponding circuit:

First vs. second quantization representations

	Second-quantized	First-quantized
Wavefunction encoding	Fock state in a given basis: $$\underset{\chi_1}{\underline{}}\ \overset{\uparrow}{\underset{\chi_2}{\underline{}}}\ \underset{\chi_3}{\underline{}}\ \underset{\chi_4}{\underline{}}$$ $\|\psi\rangle = |0100\rangle$	On a grid of 2^n sites per dimension: $\|\psi\rangle = \sum_{\mathbf{x}} a_{\mathbf{x}} \|\mathbf{x}\rangle$
Qubits required to represent the wavefunction	One per basis state (spin-orbital)	$3n$ per particle (nuclei & electrons)
Molecular Hamiltonian	$\sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2}\sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s$ Coefficients pre-computed classically	$\sum_i \frac{p_i^2}{2m_i} + \sum_{i<j} \frac{q_i q_j}{r_{ij}}$ Interaction calculated on the fly

I. Kassal et al., Ann. Rev. Phys. Chem. (2011), 62, 185.

Quantum chemistry - Formulation of the problem

Non-relativistic many-electron Schrödinger equation

$$H_{el}\ \psi(r_1, r_2, \ldots, r_{N_{el}}) = E_0\ \psi(r_1, r_2, \ldots, r_{N_{el}}), \quad H_{el} = -\frac{1}{2}\sum_{i=1}^{N}\nabla_i^2 - \sum_{i=1}^{N}\sum_{A=1}^{N_n}\frac{Z_A}{r_{iA}} + \sum_{i=1, j>i}^{N}\frac{1}{r_{ij}}$$

where N is the number of electrons and N_n the number of nuclei.

We are interested in the:

- Wavefunction (or simply the electronic density - Hohenberg-Kohn Theorem)

$$\psi(r_1, r_2, \ldots, r_N) = \sum_{i=1}^{n_{\text{conf}}} c_i |\phi_{i_1}(r_1)\phi_{i_2}(r_2), \ldots, \phi_{i_N}(r_N)\rangle$$

  Need to search in the 'exponentially' large Hilbert space of N electrons $\left(\frac{M!}{(M-N)!N!}\right)$.

- Energy (E_0) and all other observables *functional* of the system wavefunction, $O[\psi]$.

First vs. second quantization representations

Hamiltonian

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

$$h_{pq} = \int d\boldsymbol{r} \phi_p^*(\boldsymbol{r}) \left(-\frac{1}{2}\nabla^2 - \sum_I \frac{Z_I}{\boldsymbol{R}_I - \boldsymbol{r}} \right) \phi_q(\boldsymbol{r}) \quad \text{and} \quad h_{pqrs} = \int d\boldsymbol{r}_1 d\boldsymbol{r}_2 \frac{\phi_p^*(\boldsymbol{r}_1)\phi_q^*(\boldsymbol{r}_2)\phi_r(\boldsymbol{r}_2)\phi_s(\boldsymbol{r}_1)}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|}$$

The one-particle functions $\{\phi_i(\boldsymbol{r}_i)\}$ in the Hilbert space $\mathcal{H}$ define a basis in the Fock space

$$F_\nu(\mathcal{H}) = \bigoplus_{n=0}^\infty S_\nu \mathcal{H}^{\otimes n} = \mathbb{C} \oplus \mathcal{H} \oplus (S_\nu(\mathcal{H} \otimes \mathcal{H})) \oplus (S_\nu(\mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H})) \oplus \dots$$

A typical state vector is then given by

$$|\Psi\rangle_\nu = |\Psi_0\rangle_\nu \oplus |\Psi_1\rangle_\nu \oplus |\Psi_2\rangle_\nu \oplus \dots = a_0|0\rangle \oplus a_1|\psi_1\rangle \oplus \sum_{ij} a_{ij}|\psi_{2i}, \psi_{2j}\rangle_\nu \oplus \dots$$

where $\quad |\psi_{2i}, \psi_{2j}\rangle_\nu = \frac{1}{2}(|\psi_{2i}\rangle \otimes |\psi_{2j}\rangle + \nu|\psi_{2j}\rangle \otimes |\psi_{2i}\rangle) \in S_\nu(\mathcal{H} \otimes \mathcal{H})$ is a Slater determinant.

The Jordan-Wigner transformation (I)

The problem with the simulation of quantum chemistry with a quantum computer is that spins obey bosonic statistic while electrons are fermions.

Bosons

$[\sigma_i, \sigma_i] = 0, [\sigma_i^\dagger, \sigma_i^\dagger] = 0, [\sigma_i, \sigma_j^\dagger] = \delta_{i,j}$

Fermions

$\{a_i, a_i\} = 0, \{a_i^\dagger, a_i^\dagger\} = 0, \{a_i, a_i^\dagger\} = \delta_{i,j}$

We need therefore a 'bosonization' procedure.

Among the 'bosonization' procedures, the Jordan–Wigner is currently the most commonly used one in the context of electronic-structure Hamiltonians.

$$a_j = \overset{j-1}{\underset{i=1}{\otimes}} \sigma_i^z \otimes \left(\sigma_j^x + i\sigma_j^y\right) \text{ and } a_j^\dagger = \overset{j-1}{\underset{i=1}{\otimes}} \sigma_i^z \otimes \left(\sigma_j^x - i\sigma_j^y\right),$$

Note: the 'bosonization' procedures increase the many-body interactions from order 4 to K

$$\underbrace{\sigma_z \otimes \sigma_z \otimes \ldots \sigma^+}_{\text{N-local term}} \otimes \mathbb{1} \otimes \mathbb{1}$$

The Jordan-Wigner transformation (II)

After applying the JW transformation (or any other equivalent transformation) the Hamltonian becomes more 'complex' to read.

$$h_{pqrs}\, a_p^\dagger a_q^\dagger a_r a_s + h_{spqr}\, a_s^\dagger a_r^\dagger a_p a_q$$

$$\downarrow$$

$$\left(\bigotimes_{k=s+1}^{r-1} \sigma_k^z \right) \left(\bigotimes_{k=q+1}^{p-1} \sigma_k^z \right) \left(
\begin{array}{l}
\frac{\Re(h_{pqrs})}{8} \left(\begin{array}{l}
\sigma_s^x \sigma_r^x \sigma_q^x \sigma_p^x - \sigma_s^x \sigma_r^x \sigma_q^y \sigma_p^y + \sigma_s^x \sigma_r^y \sigma_q^x \sigma_p^y + \\
+ \sigma_s^x \sigma_r^x \sigma_q^x \sigma_p^y + \sigma_s^y \sigma_r^x \sigma_q^y \sigma_p^x - \sigma_s^y \sigma_r^y \sigma_q^x \sigma_p^x + \\
+ \sigma_s^x \sigma_r^y \sigma_q^y \sigma_p^x + \sigma_s^y \sigma_r^y \sigma_q^y \sigma_p^y
\end{array} \right) \\
+ \frac{\Im(h_{pqrs})}{8} \left(\begin{array}{l}
\sigma_s^y \sigma_r^x \sigma_q^x \sigma_p^x + \sigma_s^x \sigma_r^y \sigma_q^x \sigma_p^x - \sigma_s^x \sigma_r^x \sigma_q^y \sigma_p^x + \\
- \sigma_s^x \sigma_r^x \sigma_q^x \sigma_p^y - \sigma_s^y \sigma_r^x \sigma_q^y \sigma_p^y + \sigma_s^y \sigma_r^y \sigma_q^x \sigma_p^y + \\
+ \sigma_s^y \sigma_r^y \sigma_q^x \sigma_p^y + \sigma_s^y \sigma_r^y \sigma_q^y \sigma_p^x
\end{array} \right)
\end{array} \right)$$

However, the procedure can be automatised and the software (e.g., Qiskit) will take care of the transformation.

The reduction of the Fock space

The Hilbert space in second quantization is given by the Fock space

$$\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 \cdots \oplus \mathcal{H}_{N_\mathbf{m}} = \oplus_{i=1}^{N_\mathbf{m}} \mathcal{H}_i$$
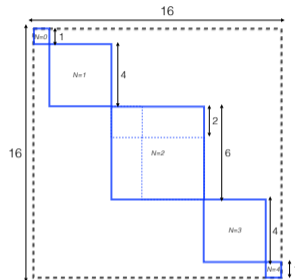
where the index indicates the number of particles in the system.

The original Hamiltonian is reduced to $\mathcal{H}_N$ in two steps:

- Projection

$$\hat{H}_N^{(K)} = \hat{P}_N^{(K)\dagger} \hat{H}^{(K)} \hat{P}_N^{(K)} \quad \text{with} \quad \hat{P}_N^{(K)} = \prod_{j \neq N}^{K} \frac{\hat{N} - j}{N - j} \quad \text{and} \quad \hat{N} = \sum_{j=1}^{K} \hat{c}_j^\dagger \hat{c}_j \ .$$

- Reduction by the 'scissors' operator (cut the desired $\mathcal{H}_N$ space).

N. Moll et al, J. Phys. A: Math. Theor, **49**, 295301 (2016).

Reduction of the degrees of freedom

Problem

- Size of the Fock space is unnecessarily large $\rightarrow$ confine to the sector with the desired number of electrons.
- The size of the problem (number of qubits) depends crucially on size of the basis set.

Strategies

- Modify the reference state. Design a Hamiltonian with a shifted ground state (e.g., the *particle/hole* picture)
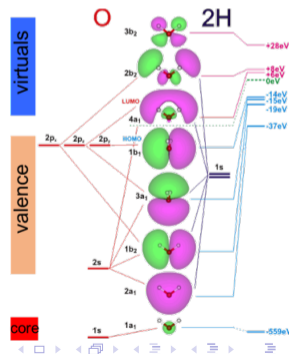
$$|\Phi_0\rangle = |0, 0, \ldots, 0\rangle \longrightarrow |\Phi_0\rangle = \prod_{i=1}^{N} \hat{a}_i^\dagger |\text{vac}\rangle$$

- Replace core electrons with effective core potentials (ECP)

$$h_{ij}^{\text{ECP}} = \int dr \, \phi_i^*(r) \left(-\frac{1}{2} \nabla_r^2 - \sum_{I=1}^{N_c} V_{\text{ECP}}(r_I) \right) \phi_j(r)$$

- Add a *penalty* potential term to the Hamiltonian

$$\hat{H} \rightarrow \hat{H} + \mu(\hat{N} - N)^2$$

Wavefunction: New strategies in quantum computing

Many-electron waverfunction

The Fock-space wavefunction
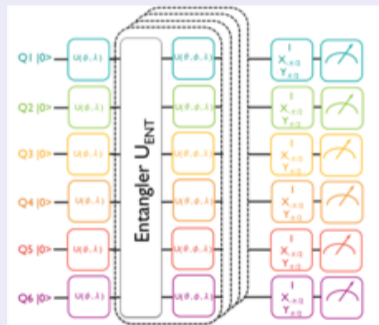
$$|\Psi\rangle_\nu = |\Psi_0\rangle_\nu \oplus |\Psi_1\rangle_\nu \oplus |\Psi_2\rangle_\nu \oplus \ldots$$

$$= a_0|0\rangle \oplus a_1|\psi_1\rangle \oplus \sum_{ij} a_{ij}|\psi_{2i}, \psi_{2j}\rangle_\nu \oplus \ldots$$

with

$$|\psi_{2i}, \psi_{2j}\rangle_\nu = \frac{1}{2}(|\psi_{2i}\rangle \otimes |\psi_{2j}\rangle + \nu\,|\psi_{2j}\rangle \otimes |\psi_{2i}\rangle)$$

$\Longrightarrow$

Quantum circuits

The mapping of the electronic wavefunction

There are a combinatoric large number of possible wavefunction configurations (Slater determinats).
Which kind of expansion can we use?

Generate orbitals (classical algorithms)	Hartree-Fock equation
Compute the system Hamiltonian	$F(\{\phi_i(r)\})\,\phi_i(r) = \epsilon_i\,\phi_i(r)$
Encode the wavefunction in the qubit register (parametrized in the qubit angles θ_i).	$\begin{aligned} |\psi(\theta)\rangle = {}& \theta_1|1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\rangle + \\ & \theta_2|1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\rangle + \\ & \qquad\qquad\qquad\quad \cdots \\ & \theta_{10}|1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\rangle + \\ & \qquad\qquad\qquad\quad \cdots \\ & \theta_{N_c}|0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1\rangle \end{aligned}$

Wavefunction: New strategies in quantum computing

Two main strategies for the expansion of the system wavefunction (Hilbert space sampling).

Classically inspired algorithms

- Hartree Fock (HF)
- Full Configuration interaction (CI, MRCI)
- Complete active space SCF (CASSCF)
- Coupled Cluster (CCSD, MR)
- Møller Plesset (MP2, MP3, MP4)
- ... and a zoo of other methods.

Genuine quantum algorithms

- Hardware efficient sampling of the Hilbert space:
  two-qubit gates distributed among the quantum register in order to guarantee a maximum entanglement between the qubits (maximal concurrence).
- Quantum machine learning
- Quantum networks

The Unitary Coupled Cluster wavefunction Ansatz

CCSD wavefunction:

$$|\Psi(\vec{\theta})\rangle = e^{\hat{T}(\vec{\theta}) - \hat{T}^\dagger(\vec{\theta})}|\Phi_0\rangle$$

where the 'cluster operator' is defined as:

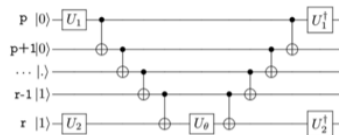$$\hat{T}(\vec{\theta}) = \hat{T}_1(\vec{\theta}) + \hat{T}_2(\vec{\theta})$$

with

$$\hat{T}_1(\vec{\theta}) = \sum_{i;m} \theta_i^m \, \hat{a}_m^\dagger \hat{a}_i$$

$$\hat{T}_2(\vec{\theta}) = \sum_{i,j;m,n} \theta_{i,j}^{m,n} \, \hat{a}_n^\dagger \hat{a}_m^\dagger \hat{a}_j \hat{a}_i$$
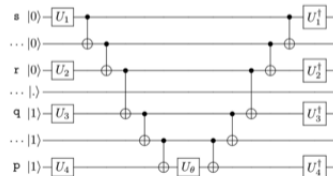
T_1 :

$$(U_1, U_2) = \{(Y, H), (H, Y)\} \quad \left(\text{where } Y = R_x(-\tfrac{\pi}{2})\right)$$

T_2 :

$$(U_1, U_2, U_3, U_4) = \{(H, H, H, Y, H), (Y, H, Y, Y), (H, Y, Y, Y), (H, H, H, H, Y),$$
$$(Y, H, H, H), (H, Y, H, H), (Y, Y, Y, H), (Y, Y, H, Y)\}$$

P. Barkoutsos et al, *Phys. Rev. A*, accepted (2018)
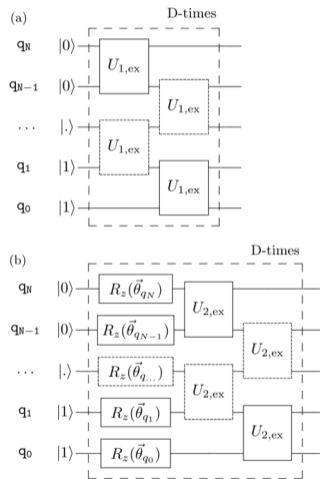
The Heuristic hardware efficient Ansatz

Heuristic wavefunction:

$$|\Psi(\vec{\theta})\rangle = \overbrace{\hat{U}^M(\vec{\theta})\hat{U}_{\text{ent}}\ldots\hat{U}^1(\vec{\theta})\hat{U}_{\text{ent}}}^{\text{D}-\text{times}}\,\hat{U}^0(\vec{\theta})|\Phi_0\rangle$$

where the two-qubit gates U_{ent} can have different forms:

$$U_{\text{SWAP}}(\theta_1, \theta_2) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_1 & e^{i\theta_2}\cos\theta_1 & 0 \\ 0 & e^{-i\theta_2}\sin\theta_1 & -\cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
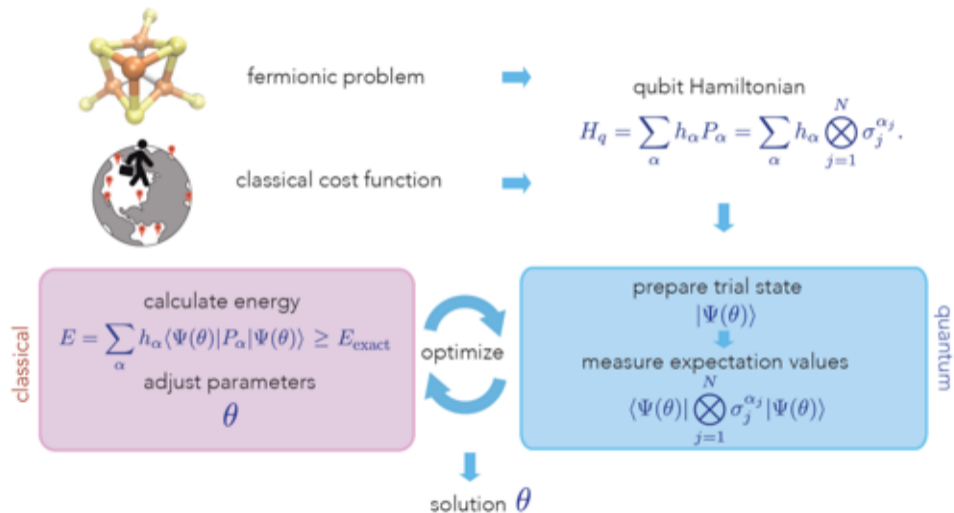
$$U_{\text{FLIP}}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 2\theta & -i\sin 2\theta & 0 \\ 0 & -i\sin 2\theta & -\cos 2\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
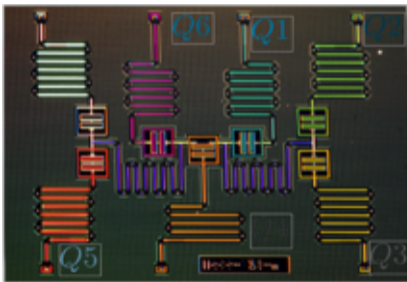

P. Barkoutsos et al, *Phys. Rev. A*, accepted (2018)

The Quantum algorithm: Variational Quantum Eigensolver (VQE)

The quantum algorithms: VQE

The quantum-classical hardware:

A small quantum processor is combined with a classical computer to jointly solve a computational task. At this stage, the quantum computer is seen as an *accelerator* of a classical processor.

Detailed VQE - part 1

(1) After setting the coordinates, charge and spin multiplicity of the molecule, perform a HF calculation in a given basis set using a classical driver.

(2) The matrix elements: $\langle r|\hat{h}|s \rangle$ and $\langle rs|\hat{g}|tu \rangle$ are then extracted and used to construct the molecular Hamiltonian using the parity or the JW fermion-to-qubit mapping. Exploiting the symmetries

$$[\hat{H}, \hat{N}_{\uparrow}] = [\hat{H}, \hat{N}_{\downarrow}] = 0 \,,$$

we can achieve two-qubit reduction (1) (one of each $\mathbb{Z}_2$ symmetry of the Hamiltonian) without modifying the lower part the energy spectrum (including the ground state). Finally, the frozen-core approximation is employed to reduce the number of possible single and double excitations and the qubit count.

---

(1): Bravyi, S.; Kitaev, A. Ann. Phys., **298**, 210-226. (2000)

Detailed VQE - part 2

(3) The qubits can be further tapered off (1) by finding the underlying symmetries of the Hamiltonian and using graph-based qubit encodings. The latter applies to the Hamiltonian, the q-UCC operator and the state vector.

(4) The parametrized trial wavefunction $|\Psi(\vec{\theta})\rangle$ is generated starting from the HF state $|\Phi_0\rangle$ by applying the q-UCC cluster operators or any other form of variational Ansatz.

---

(1): Bravyi, S.; Gambetta, J. M.; Mezzacapo, A.; Temme, K. Tapering off qubits to simulate fermionic Hamiltonians. arXiv:1701.08213

Detailed VQE - part 3

(5) The system energy $\langle \Psi(\vec{\theta})|\hat{H}|\Psi(\vec{\theta})\rangle$ is evaluated using

- the statevector (or matrix representation) simulator provided by Qiskit, which uses a matrix representation of the operators in the Hilbert space.
- the VQE optimization of the circuit parameters followed by a sequence of measurements of the observables. This can be done with or without gate noise, measurement noise and state decoherence.
- directly on hardware.

(6) Steps (4) and (5) are repeated until convergence using a classical optimizer. Examples are the Sequential Least Squares Quadratic Programming (SLSQP) and the L-BFGS-B optimization algorithms. Amplitudes $\vec{\theta}$ can be initialized to a fixed value or using a better guess such as MP2 amplitudes. The convergence criterion for the energy is set to 10^{-7}.

Issues with VQE

What are the challenges with VQE?

(1) Size of the Hilbert space vs. number of gate operations (circuit depth) required to achieve the desired accuracy for the expectation values (e.g., the system energy).

(2) Number of parameterized gates (number of parameter to be optimized) vs. magnitude of the gradients for the circuit variables.

(3) Size of the Hilbert space vs. number of measurements required to achieve convergence for the observables (e.g., the system energy).
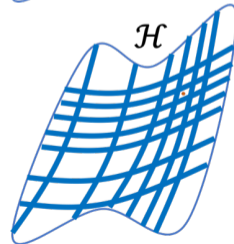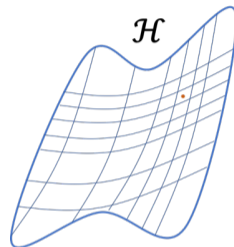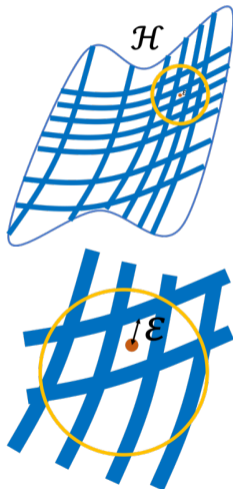
Issues with VQE

What are the challenges with VQE?

(1) Size of the Hilbert space vs. number of gate operations (circuit depth) required to achieve the desired accuracy for the expectation values (e.g., the system energy).
For a N qubits system the corresponding Hilbert space is 2^N.

The **Solovay-Kitaev theorem**:

for any target operation $U \in SU(2^N)$ and $\mathcal{G} = g_1, \ldots, g_m$ a dense set $SU(2^N)$ there is a sequence $S = g_1 g_2 \ldots g_m$ of length $l = \mathcal{O}(log^c(1/\varepsilon))$ in a dense subset of $SU(2^N)$ such that the error $d(U, S) < \varepsilon$, where $d(U, S) = \sup_{||\psi||=1} ||(U - S)\psi||$.

Ex: for any irrational α, $S_\alpha = \{R_x(\alpha), R_y(\alpha), R_z(\alpha)\}$ is dense in $SU(2)$.
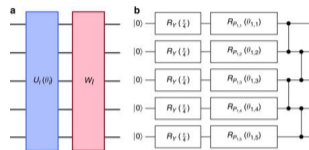
Issues with VQE

What are the challenges with VQE?

(2) Number of parameterized gates (number of parameter to be optimized) vs. magnitude of the gradients for the circuit variables.
This is a problem also known in classical optimization. In the quantum community it is known as the barren plateau.

The Barren plateau:
the observation that the size of the gradients on the gate parameters decrease exponentially with the number of qubits.

$$U(\theta) = \prod_{l=1}^{L} U_l(\theta_l)\, W_l$$

$$E(\theta) = \langle 0|U(\theta)^{\dagger} H U(\theta)|0\rangle$$

Are chemical variational VQE circuits (e.g., UCCSD) 'random enough' to share this property?
[J.R.M. McClean, et al, Nature Comm., 2018]

———————————

[Ledoux, M. *The Concentration of Measure Phenomenon* (American Mathematical Society, Providence, 2005)]

Issues with VQE

What are the challenges with VQE?

(3) Size of the Hilbert space vs. number of measurements required to achieve convergence for the observables. The number of measurements (shots) to sample an expectation value (e.g., the energy) determine the execution time of an algorithm. In noisy quantum computers each run must be completed in a few μs.

The number of measurements:
The larger the Hilbert space the larger is the number of shots needed to sample all possible configurations that contribute to the energy expectation value. For a system of about 100 electrons estimates point to $> 10^{10}$ shots.
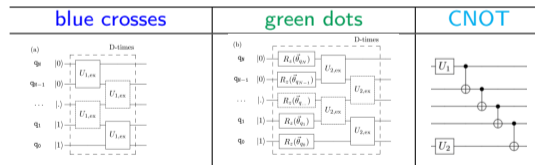

How many runs are necessary to measure an accurate expectation value, e.g., molecular energies within chemical accuracy?

Hardware-efficient results for the dissociation of water

P. Barkoutsos et al, *Phys. Rev. A*, 98 022322 (2018)
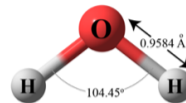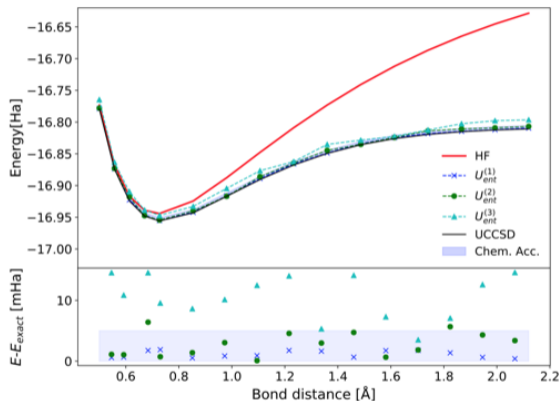
Upper: Dissociation profile of the H_2O molecule obtained for different circuit implementations:

blue crosses	green dots	CNOT

Lower: Errors along the dissociation profile.

Blue shaded area: range of chemical accuracy.

CCSD results on the dissociation of water

P. Barkoutsos et al, *Phys. Rev. A*, 98 022322 (2018)
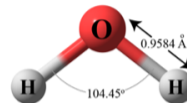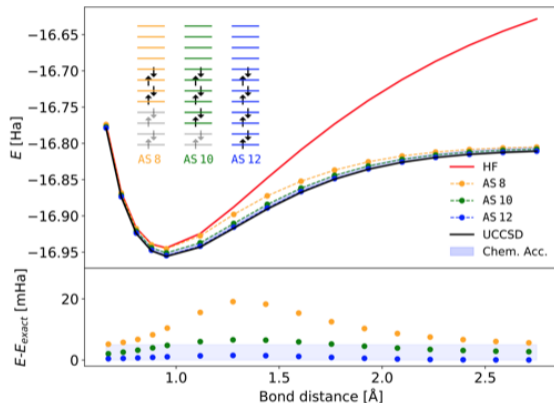
Upper: Dissociation profile of the H_2O molecule for different definitions of the active space (AS):

AS 8: 4 HF orbitals

AS 10: 6 occupied and all virtual orbitals;

AS 12: 8 occupied and all virtual orbitals.
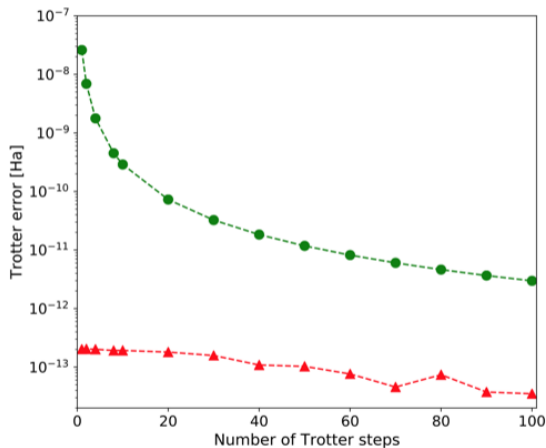
red curve: HF reference calculation

black curve: analytic solution.

Lower: Errors along the dissociation profile.
Blue shaded area: range of chemical accuracy.

Is circuit-UCC an exact map of classical UCC?

P. Barkoutsos et al, *Phys. Rev. A*, 98 022322 (2018)

Trotter decomposition of the UCC exponential:

$$e^{(\hat{T}_a + \hat{T}_b)} = lim_{n \to \infty} \left(e^{\frac{\hat{T}_a}{n}} e^{\frac{\hat{T}_b}{n}} \right)^n$$

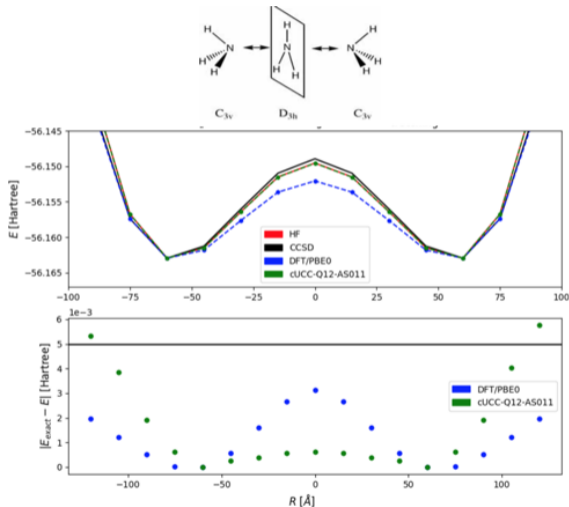Theoretical estimate: $n > 100$.

Evaluation of the Trotter error:

$$E_{\text{UCCSD}}^{opt/n}(n) = \langle \psi_{Tr}(\vec{\theta}_{opt}, n) | \hat{H}^{p/h} | \psi_{Tr}(\vec{\theta}_{opt}, n) \rangle$$

$$E_{\text{UCCSD}}^{\text{circ}/n}(n) = \min_{\vec{\theta}} \langle \psi_{Tr}(\vec{\theta}, n) | \hat{H}^{p/h} | \psi_{Tr}(\vec{\theta}, n) \rangle$$

... it is not: $\boxed{\text{UCC} \to \text{q-UCC}}$
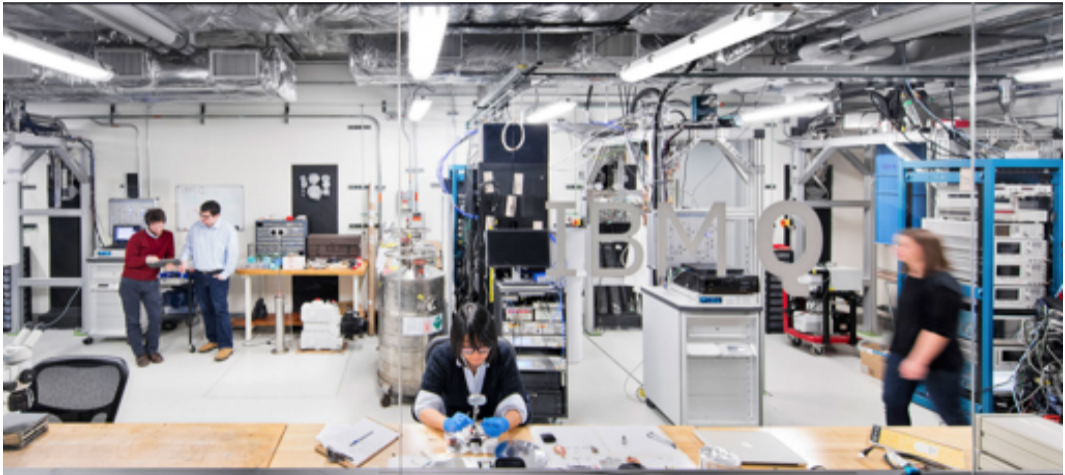
Inversion profile in ammonia, NH_3

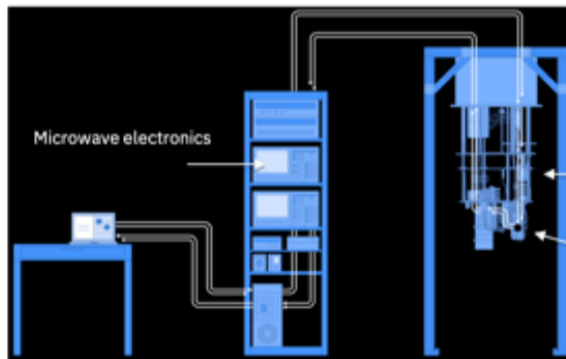The inversion of ammonia proceeds through an energy barrier.

It is well known that barriers are hard to capture with DFT (here DFT/PBE0).

q-UCC with a single Trotter step is able to capture the barrier hight within chemical accuracy (error < 2 kcal/mol).

. . . and in a real experiment . . .

... and in a real experiment ...

Microwave electronics

Refrigerator to cool qubits to 10 - 15 mK with a mixture of ^{3}He and ^{4}He

PCB with the qubit chip at 15 mK protected from the environment by multiple shields

Chip with superconducting qubits and resonators

40K
3K
0.9K
0.1K
0.015K

Ivano Tavernelli - ita@zurich.ibm.com

Calculations using IBM Q hardware . . .
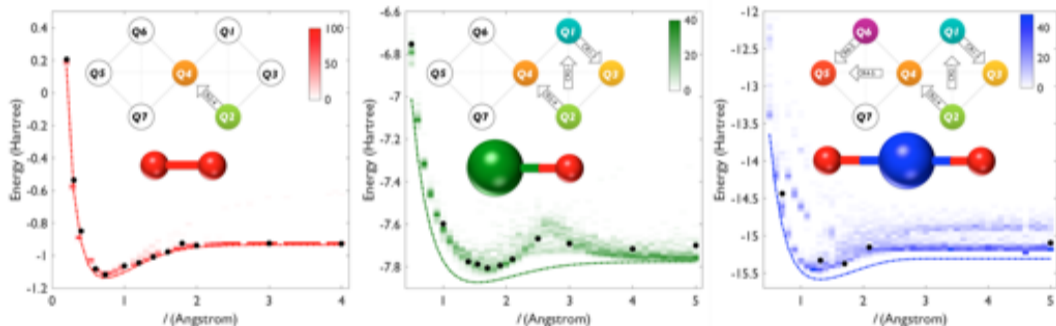
A. Kandala et al., Nature **549**, 242 (2017)

Error mitigation scheme
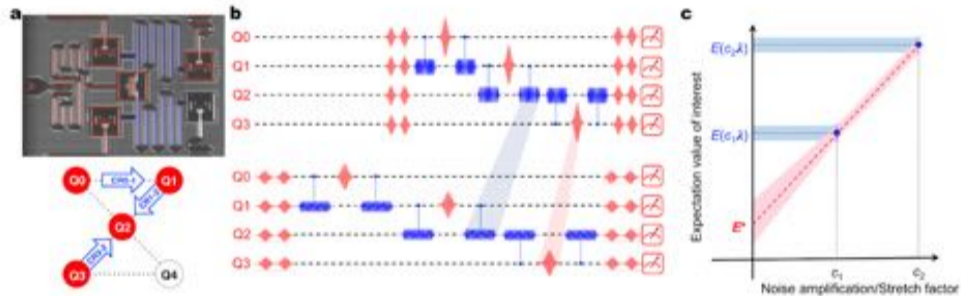
A. Kandala et al., Nature, **467**, 491-495 (2019)

Error mitigation scheme - Results

A. Kandala et al., Nature, **467**, 491-495 (2019)

The Equation of Motion Approach

Any excited state $|n\rangle$ can be created by applying an excitation operator $\hat{O}_n^\dagger$ on the ground state $|0\rangle$ of the system.
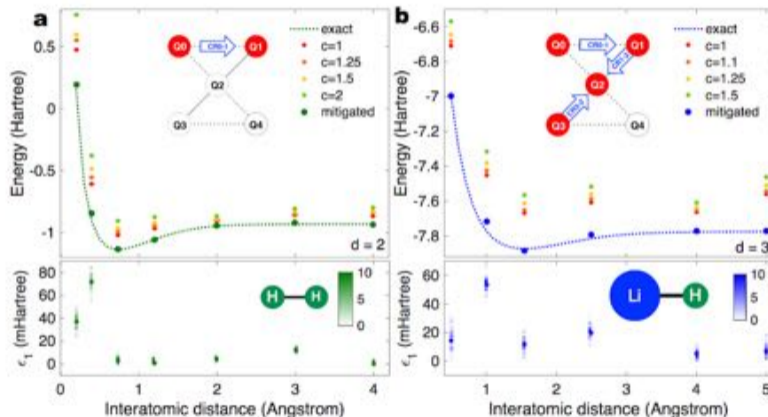
$$\hat{O}_n^\dagger = |n\rangle \langle 0|$$

with corresponding de-excitation

$$\hat{O}_n = |0\rangle \langle n| \ .$$

The EOM is obtained acting the superoperator $\hat{\hat{H}}$ on the excitation operator ($\hat{\hat{H}}\hat{O}_n^\dagger |0\rangle = [\hat{H}, \hat{O}_n^\dagger]$) giving

$$\hat{\hat{H}}\hat{O}_n^\dagger |0\rangle = \hat{H}\hat{O}_n^\dagger |0\rangle - \hat{O}_n^\dagger \hat{H} |0\rangle = (E_n - E_0)\hat{O}_n^\dagger |0\rangle \ ,$$

with after some arrangements leads to

$$\Delta E_{0n} = \frac{\langle 0| [\hat{O}_n, \hat{H}, \hat{O}_n^\dagger] |0\rangle}{\langle 0| [\hat{O}_n, \hat{O}_n^\dagger] |0\rangle}$$

with $[\hat{A}, \hat{B}, \hat{C}] = \frac{1}{2}\{[[\hat{A}, \hat{B}], \hat{C}] + [\hat{A}, [\hat{B}, \hat{C}]]\}$.

The Equation of Motion Approach

Introducing a basis set of HF or KS orbitals we have

$$\hat{O}_n^\dagger = \sum_\mu (X_\mu \hat{E}_\mu - Y_\mu \hat{E}_\mu^\dagger)$$

with the single and double excitation operators

$$\hat{E}_\mu = \{\{\hat{a}_m^\dagger \hat{a}_i\}, \{\hat{a}_m^\dagger \hat{a}_n^\dagger \hat{a}_i \hat{a}_j\}\}$$

$$\hat{E}_\mu^\dagger = \{\{\hat{a}_i^\dagger \hat{a}_m\}, \{\hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_m \hat{a}_n\}\}.$$

The stationary condition $\partial(\Delta E_{0n}) = 0$ leads to the derivation a set of the following secular equations:

$$\begin{pmatrix} \mathbf{M} & \mathbf{Q} \\ \mathbf{Q^*} & \mathbf{M^*} \end{pmatrix} \begin{pmatrix} \mathbf{X}_n \\ \mathbf{Y}_n \end{pmatrix} = \omega_n \begin{pmatrix} \mathbf{V} & \mathbf{W} \\ -\mathbf{W^*} & -\mathbf{V^*} \end{pmatrix} \begin{pmatrix} \mathbf{X}_n \\ \mathbf{Y}_n \end{pmatrix}$$

The Equation of Motion Approach

The stationary condition $\partial(\Delta E_{0n}) = 0$ leads to the derivation a set of the following secular equations:

$$\begin{pmatrix} \mathbf{M} & \mathbf{Q} \\ \mathbf{Q*} & \mathbf{M*} \end{pmatrix} \begin{pmatrix} \mathbf{X}_n \\ \mathbf{Y}_n \end{pmatrix} = \omega_n \begin{pmatrix} \mathbf{V} & \mathbf{W} \\ -\mathbf{W*} & -\mathbf{V*} \end{pmatrix} \begin{pmatrix} \mathbf{X}_n \\ \mathbf{Y}_n \end{pmatrix}$$

where

$$M_{\mu\nu} = \langle 0| \, [\hat{E}_\mu^\dagger, \hat{H}, \hat{E}_\nu] \, |0\rangle \,,$$
$$Q_{\mu\nu} = - \langle 0| \, [\hat{E}_\mu^\dagger, \hat{H}, \hat{E}_\nu^\dagger] \, |0\rangle \,,$$
$$V_{\mu\nu} = \langle 0| \, [\hat{E}_\mu^\dagger, \hat{E}_\nu] \, |0\rangle \,,$$
$$W_{\mu\nu} = - \langle 0| \, [\hat{E}_\mu^\dagger, \hat{E}_\nu^\dagger] \, |0\rangle$$

and ω_n are the first n excitation energies.

EOM implementation

1. Find the ground state wavefunction $|\psi_0\rangle$ parametrized in the angles $\{\theta_i\}$.

2. Generate $\hat{E}_\mu$ and $\hat{E}_\mu^\dagger$ operators for all possible n excitations.

3. For each matrix element (μ, ν):

   - Compute the commutators in matrix form, e.g. for $V_{\mu\nu}$ compute $\hat{E}_\mu^\dagger \hat{E}_\nu - \hat{E}_\nu \hat{E}_\mu^\dagger$.
   - Map the fermionic $\mathbf{M}$, $\mathbf{Q}$, $\mathbf{V}$ and $\mathbf{W}$ matrices to qubit space using the Jordan-Wigner transformation.
   - Evaluate in the circuit parametrized with the fix angles $\{\theta_i\}$ the matrices elements $M_{\mu\nu}, Q_{\mu\nu}, V_{\mu\nu}, W_{\mu\nu}$

4. Evaluate classically the $2n$ eigenvalues of the secular equation to get the first n excitation (and de-excitation) energies.

Results: numerical simulations
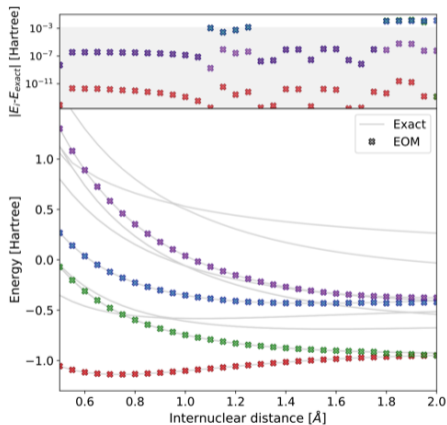
(a) H_2
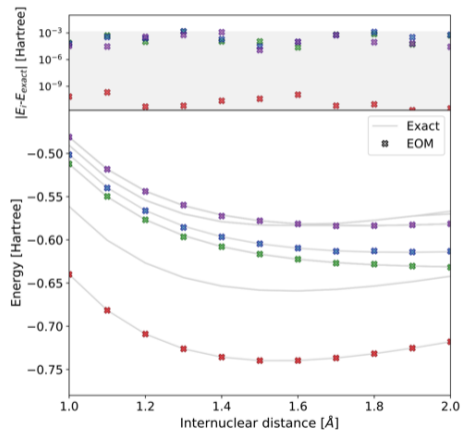
(b) LiH

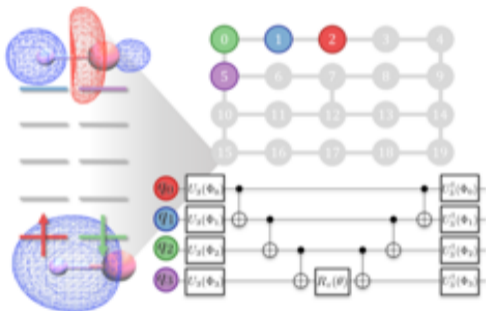# Results: numerical simulations with realistic noise (H$_2$)
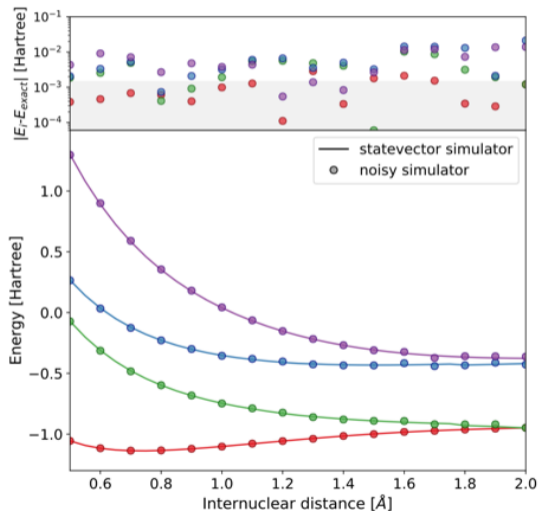
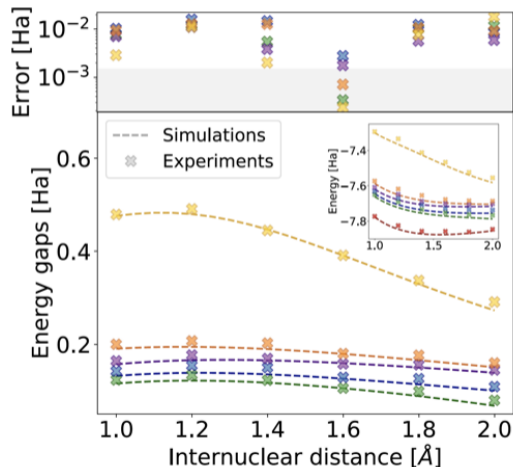# Results: on IBM quantum hardware - no error corrections
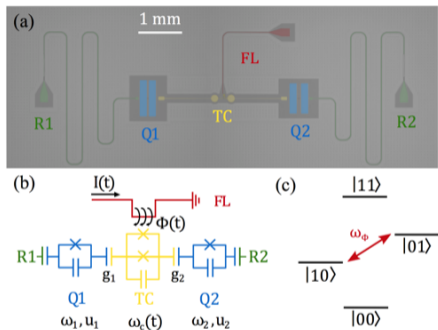
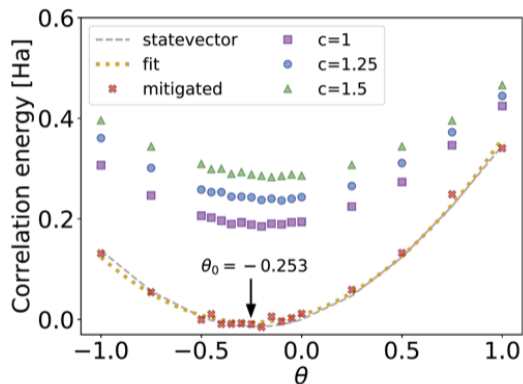Marc Ganzhorn, et al., Phys. Rev. App., **11**, 044092 (2019)
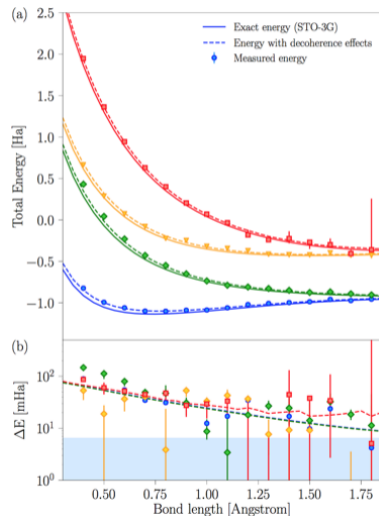
Excited States of LiH on 4 qubits - implementation and results

P. Ollitrault et al., preprint arXiv:1910.12890 (2019)

Excited States of LiH on 4 qubits - error mitigation

P. Ollitrault et al., preprint arXiv:1910.12890 (2019)
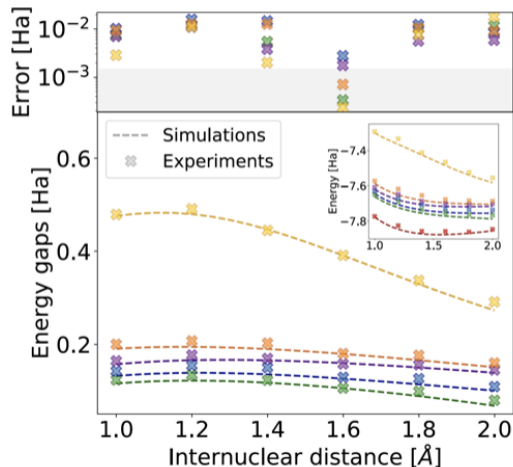
Literature on excited states

(1) Gate-Efficient Simulation of Molecular Eigenstates on a Quantum Computer

M. Ganzhorn, D.J. Egger, P. Barkoutsos, P. Ollitrault, G. Salis, N. Moll, M. Roth, A. Fuhrer, P. Mueller, S. Woerner, I. Tavernelli, and S. Filipp

(2) Quantum equation of motion for computing molecular excitation energies on a noisy quantum processor

Pauline J. Ollitrault,[1,2] Abhinav Kandala,[3] Chun-Fu Chen,[3] Panagiotis Kl. Barkoutsos,[1] Antonio Mezzacapo,[3] Marco Pistoia,[3] Sarah Sheldon,[3] Stefan Woerner,[1] Jay M. Gambetta,[3] and Ivano Tavernelli[1,*]

[1] IBM Research GmbH, Zurich Research Laboratory, Säumerstrasse 4, 8803 Rüschlikon, Switzerland
[2] Laboratory of Physical Chemistry, ETH Zürich, 8093 Zürich, Switzerland
[3] IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
(Dated: August 18, 2020)

arXiv:1910.12890 - To appear in Physical Review Research

Time propagation algorithm

Given an Hamiltonian H, time propagations deals with the calculation of

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

where

- we use atomic units $\hbar = 1$.
- $|\psi(0)\rangle$ in encoded in a qubit register.
- the propagation operators e^{-iHt} is encoded as a series of gate operations
- $|\psi(t)\rangle$ is read qubit-by-qubit at the circuit end.

The Hamiltonian is assumed to be expressed as a Pauli string, meaning a tensor product of a sequence of Pauli matrices.

$$H = \sum_i g_i\, \sigma_{i_1} \otimes \sigma_{i_2} \otimes \ldots \sigma_{i_N}$$

with $i = \{i_1, i_2, \ldots, i_N\}$.

Time propagation algorithm

Theorem 1:
For all $H = \sum_i^M H_k$ then

$$e^{-iHt} = e^{-iH_1 t} e^{-iH_2 t} \ldots e^{-iH_M t}$$

for all t, iff $[H_i, H_j] = 0$, $\forall i, j$.

Theorem 2 (The Trotter formula):
For two Hermitian operators A and B, for all t

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

which is true even when A and B do not commute.

Error:

$$e^{i(A+B)dt} = e^{iAdt} e^{iBdt} + \mathcal{O}(dt^2).$$

Time propagation algorithm

Extension (Baker-Campbell-Hausdorf formula):

$$e^{(A+B)dt} = e^{Adt} \, e^{Bdt} e^{-\frac{1}{2}[A,B]dt^2} + \mathcal{O}(dt^3).$$

Exercise: Prove this equation.

Algorithm:

1. Prepare initial state $|\psi(0)\rangle \rightarrow |\psi_0\rangle$.
2. Propagate for a small[1] time interval dt: $|\psi_{j+1}\rangle = U(t_j, t_j + dt)|\psi_j\rangle$.
3. $j := j + 1$ repeat from 2 until $j\,dt \geq T_f$, else
4. $|\psi(T_f)\rangle = |\psi_j\rangle$.

where $U(t_j, t_j + dt)$ is either the Trotter or the BCH operator.

Time propagation algorithm

Since the Hamiltonian is a tensor product of Pauli matrices (which are the generators of rotations in SU(2)), the propagator $U = e^{-iHt}$ describes a rotation of the state vector in the tensor space.
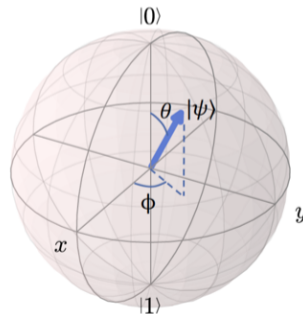
One-qubit space

Pauli matrices $\{\sigma_x, \sigma_y, \sigma_z\}$:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} .$$

State on the Bloch sphere:

$$|\Psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle .$$

Time propagation algorithm

One-qubit rotations

Case 1: $e^{i\Delta_t \Gamma \sigma_x}$:

$$-\boxed{R_x(-2\Delta_t\Gamma)}-$$

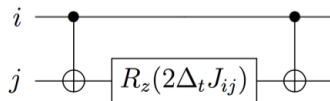Case 2: transform it to a $e^{i\Delta_t \Gamma \sigma_z}$ rotation by applying a change of basis.
In this case we have first to make a basis transformation and then apply the rotation in z:

$$-\boxed{H}-\boxed{R_z(-2\Delta_t\Gamma)}-\boxed{H}-$$

Two-qubit rotations

Case 3: $e^{i\delta\sigma_z\sigma_z} = e^{i\delta\sigma_z\otimes\sigma_z}$ $(\delta = \Delta_t\Gamma)$:
We will show that this can be computed using:

Time propagation algorithm

We first derive the matrix for $e^{i\delta\sigma_z\otimes\sigma_z}$

$$\sigma_z \otimes \sigma_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \rightarrow \quad e^{i\delta\sigma_z\otimes\sigma_z} = \begin{pmatrix} e^{i\delta} & 0 & 0 & 0 \\ 0 & e^{-i\delta} & 0 & 0 \\ 0 & 0 & e^{-i\delta} & 0 \\ 0 & 0 & 0 & e^{i\delta} \end{pmatrix}$$

and

$$e^{i\delta\sigma_z\otimes\sigma_z} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} e^{-i\delta} & 0 & 0 & 0 \\ 0 & e^{i\delta} & 0 & 0 \\ 0 & 0 & e^{-i\delta} & 0 \\ 0 & 0 & 0 & e^{i\delta} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$= \quad (\text{CNOT}) \quad . \quad (\mathbb{I} \otimes R_z(\delta)) \quad . \quad (\text{CNOT}).$$

Time propagation algorithm

Exercise:

Prove that $e^{i\delta\sigma_z\otimes\sigma_z}$ performs a rotation of $-\delta$ if the two qubits are in the same state, $|00\rangle, |11\rangle$ and of $+\delta$ when the spins are opposite, $|10\rangle, |01\rangle$.

This result can be generalized to any number of qubits (see lecture on quantum chemistry)
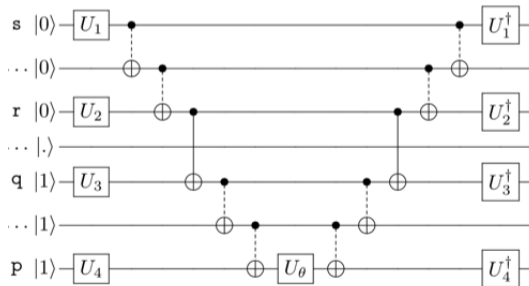
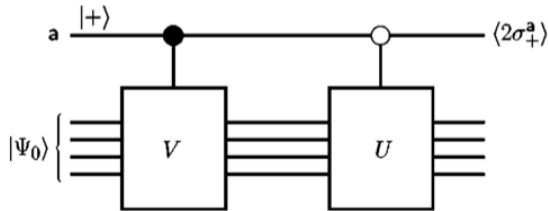$(U_1, U_2, U_3, U_4) = \{(H, H, Y, H), (Y, H, Y, Y), (H, Y, Y, Y), (H, H, H, Y),$

Time propagation algorithm - Properties

Theorem (Laflamme et al. Phys. Rev. A **65** 042323):
The correlation function $\langle \Psi_0 | U^\dagger V | \Psi_0 \rangle$ can be evaluate with the help of an unique ancilla qubit using

$$\langle \sigma_+^a \rangle_+ = \langle \Psi_0 | U^\dagger V | \Psi_0 \rangle$$

where $\sigma_+^a = \sigma_x^a + i\sigma_y^a$ ($|+\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle)$) and V and U are unitary operations. The corresponding circuit is

Laflamme et al. Phys. Rev. A **65** 042323.

Time propagation algorithm - Properties

We know want to compute an observable like the time correlation function between the initial and the final states

$$C(t) = \langle \psi_0 | \psi(t) \rangle \,.$$

We can use the previous theorem with the mapping:

$$V = U^\dagger = e^{-iHt/2}$$

and therefore

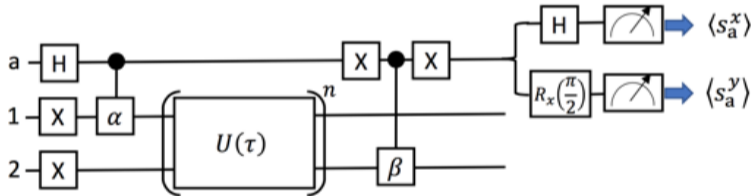$$\langle \sigma_x^a + i \sigma_y^a \rangle_a = \langle \psi_0 | e^{-iHt} | \psi_0 \rangle = C(t) \,.$$

A. Chiesa, et al. *Quantum hardware simulating four-dimensional inelastic neutron scattering Nat. Phys..* **15**, 455-459 (2019).

Time propagation algorithm - Properties

We know want to compute an observable like the time correlation function between the initial and the final state

$$C(t) = \langle \psi(0) | \psi(t) \rangle.$$

For a 2-qubit system, the corresponding circuit will look like

where $U(\tau)$ is the time-evolution unitary block (Trotter step) repeated n times.

A. Chiesa, et al. *Quantum hardware simulating four-dimensional inelastic neutron scattering* Nat. Phys.. **15**, 455-459 (2019).

Application: Four-dimensional inelastic neutron scattering

The new generation of spectrometers, equipped with arrays of position-sensitive detectors, enable to efficiently measure **the neutron cross-sections** as a function of **energy and of the three components of the momentum transfer vector Q**, in vast portions of the reciprocal space.

Exploiting these capabilities, it is now possible to obtain an unprecedented insight into the **coherent spin dynamics** of molecular clusters.
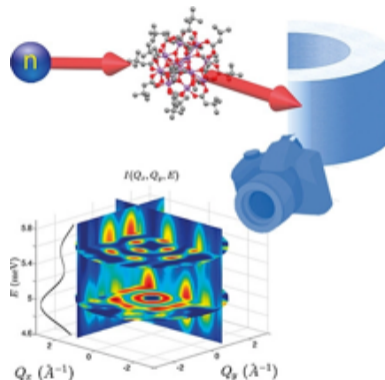
E, Garlatti *et al.*, Neutron Scattering in Coordination Chemistry,

2019, 1106 (2019).

Four-dimensional inelastic neutron scattering

Scattering function

$$I(\mathbf{Q}, \omega) \propto \sum_{\alpha\beta} \left(1 - \frac{Q_\alpha^2}{Q_\beta^2}\right) \sum_{p=1}^{2^N} \sum_{i \geq j=1}^{N} F_i(Q) \cdot F_j(Q) \cos(\mathbf{Q} \cdot \mathbf{R}_{ij}) \langle 0|s_\alpha(i)|p\rangle \langle p|s_\beta(j)|0\rangle \, \delta(\omega - E_p/\hbar)$$

- $\alpha, \beta = x, y, z$
- $\mathbf{Q}$ scattering vector
- $F_i(Q)$ form factor of atom i.
- N number of spins with positions $\mathbf{R}_i$ and distance vectors $\mathbf{R}_{ij}$
- $p : 1 \ldots 2^N$ labels the elements of the Hilbert space
- E_p eigenenergy associated to $|p\rangle$.

Four-dimensional inelastic neutron scattering

Scattering function

$$I(\mathbf{Q}, \omega) \propto \sum_{\alpha\beta} \left(1 - \frac{Q_\alpha^2}{Q_\beta^2}\right) \sum_{p=1}^{2^N} \sum_{i \geq j=1}^{N} F_i(Q) \cdot F_j(Q) \cos(\mathbf{Q} \cdot \mathbf{R}_{ij}) \langle 0|s_\alpha(i)|p\rangle \langle p|s_\beta(j)|0\rangle \, \delta(\omega - E_p/\hbar)$$

- $\alpha, \beta = x, y, z$
- $\mathbf{Q}$ scattering vector
- $F_i(Q)$ form factor of atom i.
- N number of spins with positions $\mathbf{R}_i$ and distance vectors $\mathbf{R}_{ij}$
- $p : 1 \ldots 2^N$ labels the elements of the Hilbert space
- E_p eigenenergy associated to $|p\rangle$.

Calculation of $I(\mathbf{Q}, \omega)$

The limiting step in the calculation of $I(\mathbf{Q}, \omega)$ is the correlation function

$$c_{ij}^{\alpha\beta}(t) = \langle s_i^\alpha(t) s_j^\beta \rangle = \sum_p \langle 0 | s_\alpha(i) | p \rangle \langle p | s_\beta(j) | 0 \rangle e^{-iE_p t}$$

- compute the eigen-spectrum of the system Hamiltonian (2^N dimensional) $\rightarrow \{E_p, |p\rangle\}$
  - evaluate $\sum_p \langle 0 | s_\alpha(i) | p \rangle \langle p | s_\beta(j) | 0 \rangle e^{-iE_p t}$
  - compute $I(\mathbf{Q}, \omega)$

- time evolution using the propagator, $\exp(-iHt/\hbar)$
  - compute $c_{ij}^{\alpha\beta}(t) = \langle s_i^\alpha(t) s_j^\beta \rangle$
  - extract $\{E_p, |p\rangle\}$ using a fitting procedure or a Fourier transform
  - compute $I(\mathbf{Q}, \omega)$

A. Chiesa, et al. *Quantum hardware simulating four-dimensional inelastic neutron scattering Nat. Phys.*. **15**, 455-459 (2019).
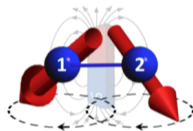
Model systems

Due to the limitations of the NISQ quantum hardware (about 10 qubits) we study the scattering properties of the following model systems:

$$\mathcal{H} = \sum_{i=1}^{N-1} \left[J_p \left(s_i^x s_{i+1}^x + s_i^y s_{i+1}^y \right) + J_z s_i^z s_{i+1}^z \right] + B \sum_{i=1}^{N} g_i s_i^z$$
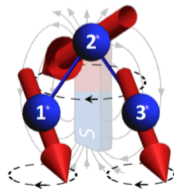

i. Spin dimer:
   a. Heisenberg model: $J_p = J_z = J$, $g_1 = g_2 = g$.
   b. Heisenberg model with nonequivalent ions: $J_p = J_z = J$, $g_1 \neq g_2$.
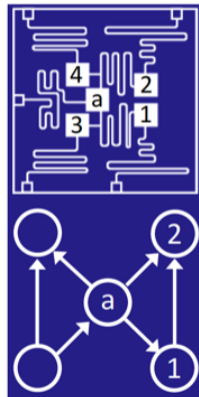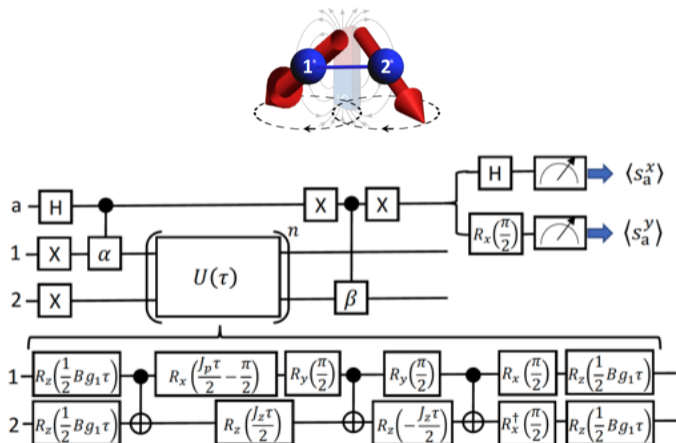   c. Ising model with nonequivalent ions $J_p = 0$; $J_z = J$, $g_1 \neq g_2$.

ii. Spin trimer:
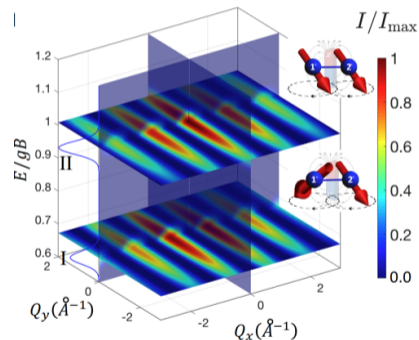   Heisenberg nearest-neighbors interactions with
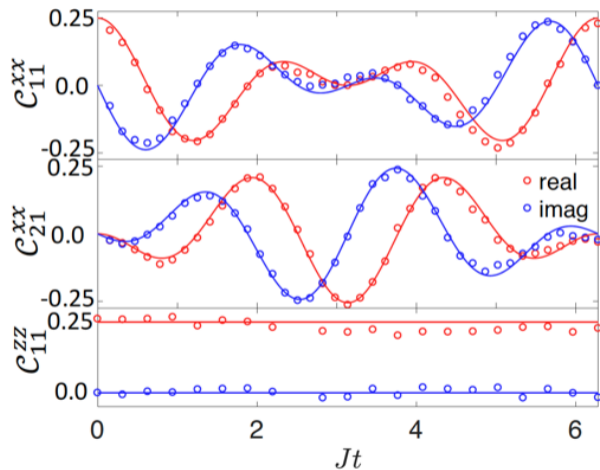 $J_p = J_z = J$, $g_1 = g_2 = g_3 = g$.
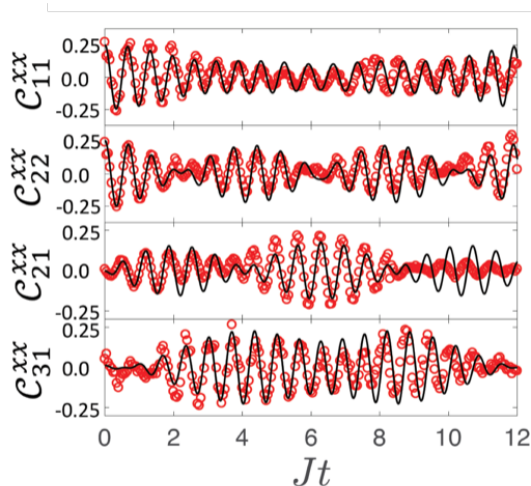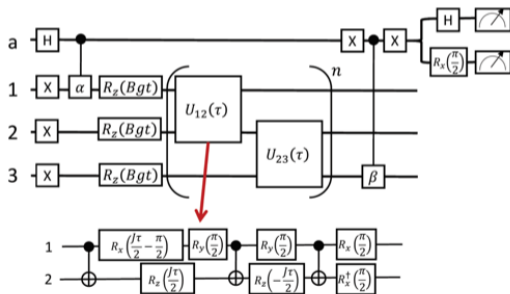
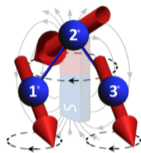# Circuit implementation: 2-spin system

A. Chiesa, et al. *Nat. Phys.*, **15**, 455 (2019).

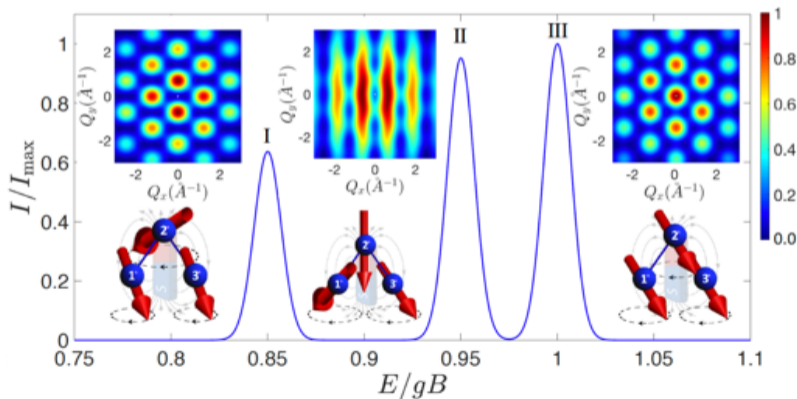# Results: 2-spin system

Time correlation functions

Results: 3-spin system

A. Chiesa, et al. *Nat. Phys.*, **15**, 455 (2019).

Results: 3-spin system

$$I(\mathbf{Q}, \omega) \propto \sum_{\alpha\beta} \left(1 - \frac{Q_\alpha^2}{Q_\beta^2}\right) \sum_{p=1}^{2^N} \sum_{i \geq j=1}^{N} F_i(Q) \cdot F_j(Q) \cos(\mathbf{Q} \cdot \mathbf{R}_{ij}) \langle 0|s_\alpha(i)|p\rangle \langle p|s_\beta(j)|0\rangle \, \delta(\omega - E_p/\hbar)$$

ooVQE Algorithm

Rotation matrix:

$$U = e^{-\kappa}$$

induces the orbital rotation

$$C = C_{\mathrm{RHF}} e^{-\kappa}$$

The one- and two-body integrals used to generate the Hamiltonian matrix are modified according to

$$\langle r|\hat{\tilde{h}}|s\rangle = \sum_{\alpha\beta} C_{\alpha r}^* C_{\beta s} \langle \alpha|\hat{h}|\beta\rangle \ (1)$$

$$\langle pq|\hat{\tilde{g}}|rs\rangle = \sum_{\alpha\beta\gamma\theta} C_{\alpha p}^* C_{\beta q}^* C_{\gamma r} C_{\theta s} \langle \alpha\beta|\hat{g}|\gamma\theta\rangle \ (2)$$

Two additional steps in the conventional VQE:

$(1)^*$ Extracted RHF integrals $\langle r|\hat{h}|s\rangle$ and $\langle rs|\hat{g}|tu\rangle$ undergo orbital rotation using Eqs. (1) and (2)

$(2)^*$ Matrix elements $\vec{\kappa}$ are also optimized in addition to the VQE variational parameters.
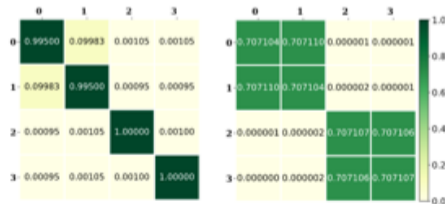
Can Quantum Algorithms Outperform their Classical Equivalents?
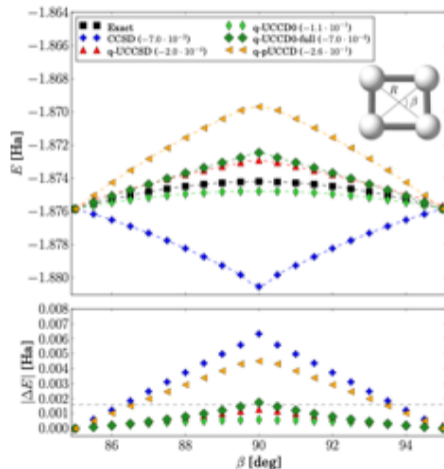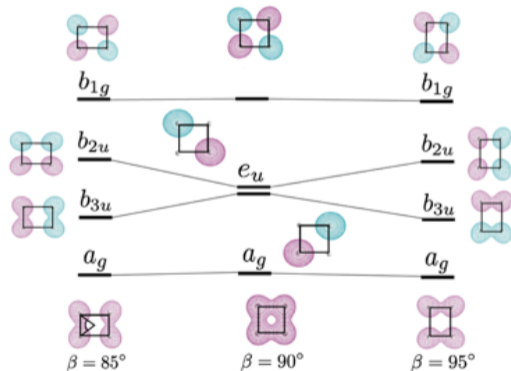
Can Quantum Algorithms Outperform their Classical Equivalents?
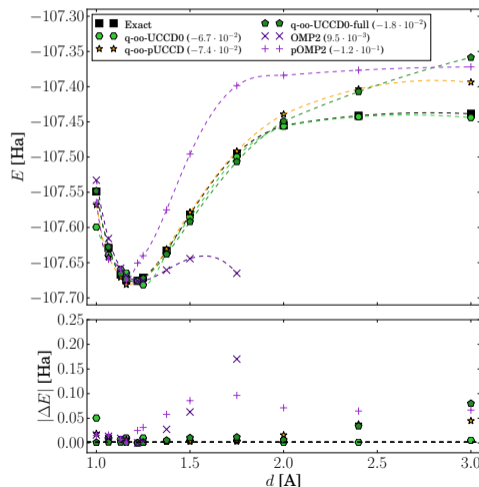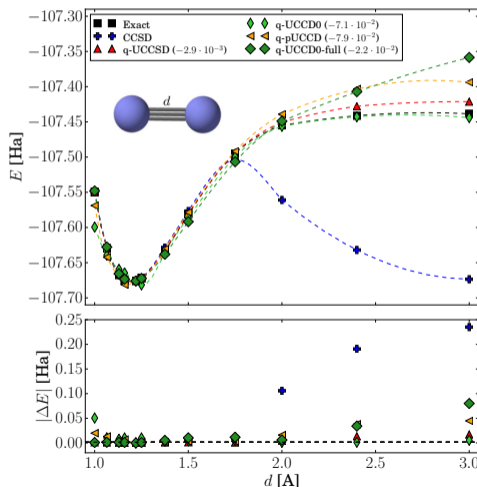
Literature on ooVQE

Quantum orbital-optimized unitary coupled cluster methods in the strongly correlated regime: Can quantum algorithms outperform their classical equivalents?

Igor O. Sokolov[1,a], Panagiotis Kl. Barkoutsos[1], Pauline J. Ollitrault[1,2], Donny Greenberg[3], Julia Rice[4], Marco Pistoia[3,5], and Ivano Tavernelli[1,b]

arXiv:1911.10864

Can Quantum Algorithms Outperform their Classical Equivalents?

The forces $\mathbf{F}_I = (F_{Ix}, F_{Iy}, F_{Iz})$ are defined as

$$F_{I\alpha}(\mathbf{R}) = \left.\frac{dE}{dR_{I\alpha}}\right|_{\mathbf{R}},$$

with $\alpha \in \{x, y, z\}$.
The total derivative is given explicitly by

$$\begin{aligned}
F_{I\alpha}(\mathbf{R}) = {} & \langle\Psi(\theta)|\partial_{I\alpha}\hat{H}(\mathbf{R})|\Psi(\theta)\rangle \\
& + \langle\partial_{I\alpha}\Psi(\theta)|\hat{H}(\mathbf{R})|\Psi(\theta)\rangle \\
& + \langle\Psi(\theta)|\hat{H}(\mathbf{R})|\partial_{I\alpha}\Psi(\theta)\rangle,
\end{aligned}$$

The first term corresponds to the Hellmann-Feynman force.

Two possible solutions:

$$F_{I\alpha}^{\text{FD}}(\mathbf{R}) = \frac{\langle\Psi_+|\hat{H}_+|\Psi_+\rangle - \langle\Psi_-|\hat{H}_-|\Psi_-\rangle}{2\Delta R},$$

$$F_{I\alpha}^{\text{H-F}}(\mathbf{R}) = \left\langle\Psi_0\left|\frac{\hat{H}_+ - \hat{H}_-}{2\Delta R}\right|\Psi_0\right\rangle,$$

where Ψ_+ is the wavefunction optimized for the displaced Hamiltonian $\hat{H}_+$ where all $\mathbf{R}$ are displaced

$$\mathbf{R} \to \mathbf{R} + d\mathbf{R}_{x,y,z}$$

along each cartesian coordinate (for a total of $3N_A$ displacements)

Algorithmic error mitigation: the Lanczos approach

With the VQE-Lanczos algorithm, we evaluate expectation values as

$$\langle\Psi_0|\hat{O}|\Psi_0\rangle \rightarrow \frac{\langle\Psi_0|(\hat{H}(\mathbf{R}) - d)\hat{O}(\hat{H}(\mathbf{R}) - d)|\Psi_0\rangle}{\langle\Psi_0|(\hat{H}(\mathbf{R}) - d)^2|\Psi_0\rangle} = L_{d,\mathbf{R}}(\hat{O})$$

where $d \in \mathbb{R}$ is a tunable parameter that need to be optimized a priori.

Algorithmic Error Mitigation Scheme for Current Quantum Processors

Philippe Suchsland,[1,2,3] Francesco Tacchino,[2] Mark H. Fischer,[3]
Titus Neupert,[3] Panagiotis Kl. Barkoutsos,[2] and Ivano Tavernelli[2]

[1] Institute for Theoretical Physics, ETH Zurich, 8093 Zurich, Switzerland
[2] IBM Quantum, IBM Research – Zurich, 8803 Rueschlikon, Switzerland
[3] Department of Physics, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland
(Dated: August 26, 2020)

arXiv:2008.10914

Validation: geometry optimization

Table: Geometry optimization results using the MR (reference), the VQE and the VQE with Lanczos (VQE-L) algorithms. In these two last cases we use 8'192 measurements for the evaluation of the energy and force components. The equilibrium bond distance of H_2 is given by R_{eq}. The structure of H_3^+ (see Fig. **??**) is characterized by three parameters: (i) the distance between atoms 1 and 2 (R_{12}), (ii) the distance between atoms 1 and 3 (R_{13}), and (iii) the angle α_{213} formed between the bonds H_1-H_2, and H_1-H_3 (α_{213}). We use Å for distances and degrees for the angles.

		MR	VQE	VQE-L
H_2	R_{eq}	0.735	0.742	0.733
H_3^+	R_{12}	0.985	1.006	0.990
	R_{13}	0.985	0.999	0.990
	α_{213}	60.0	59.8	59.9

Micro-canonical versus canonical MD

Micro-canonical dynamics at constant E

Newton's equations of motion

$$
\begin{aligned}
\dot{\mathbf{v}} &= \mathbf{F}(\mathbf{R})/\mathbf{m}, \\
\dot{\mathbf{R}} &= \mathbf{v},
\end{aligned}
$$

Verlet algorithm

$$
\mathbf{R}(t+\Delta t) = 2\mathbf{R}(t) - \mathbf{R}(t-\Delta t) + \frac{1}{2}\mathbf{F}(t)/\mathbf{m}\,\Delta t^2,
$$

Canonical dynamics at constant T

Langevin Dynamics:

$$
\begin{aligned}
\dot{\mathbf{v}} &= -\gamma(\mathbf{R})\cdot\mathbf{v} + \mathbf{F}(\mathbf{R})/\mathbf{m} + \eta(t), \\
\dot{\mathbf{R}} &= \mathbf{v}, \\
\langle\eta(t)\rangle &= 0, \\
\langle\eta_i(t)\eta_j(t')\rangle &= \alpha_{ij}(\mathbf{R})\,\delta(t-t'),
\end{aligned}
$$

MD algorithms
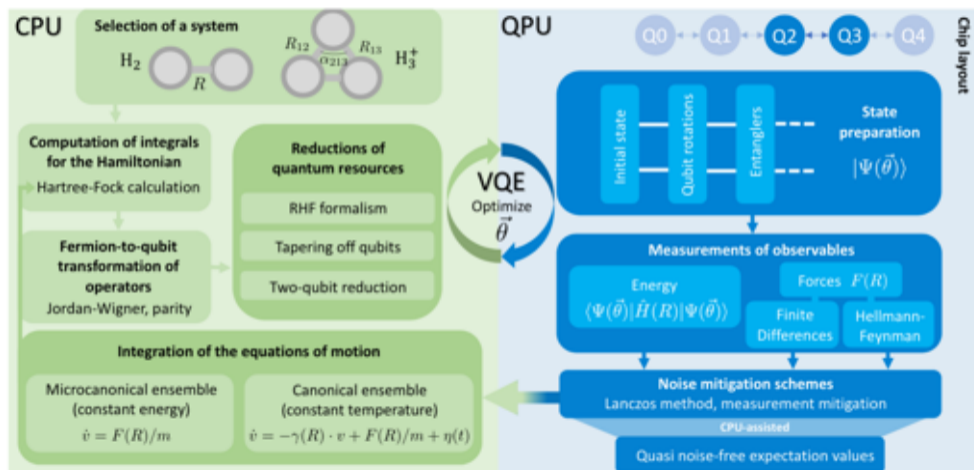
MD time series - statistical noise and hardware noise
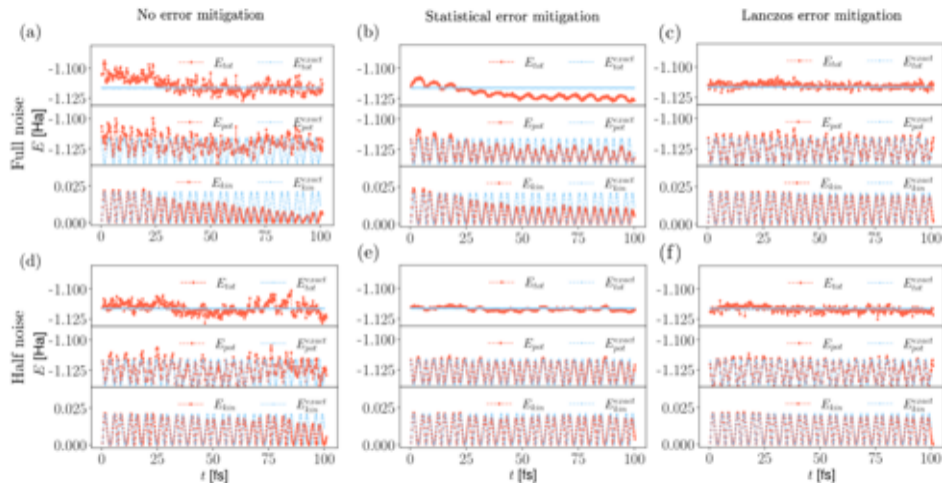
MD of H₂ on a quantum computer (ibmq_athens)
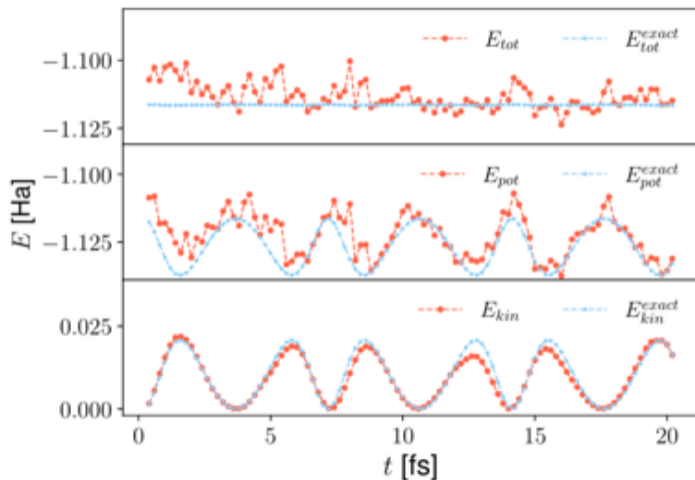
Phase space trajectories for H_2: microcanonic *vs.* canonic MD
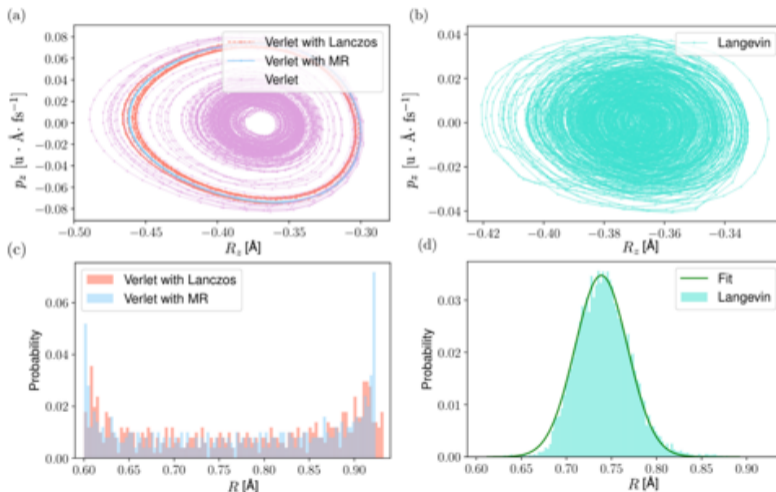
Literature on MD

Microcanonical and finite temperature *ab initio* molecular dynamics simulations on quantum computers

Igor O. Sokolov,[1,2] Panagiotis Kl. Barkoutsos,[1] Lukas Moeller,[1]
Philippe Suchsland,[1,3] Guglielmo Mazzola,[1] and Ivano Tavernelli[1,*]

[1] *IBM Quantum, IBM Research – Zurich, Switzerland*
[2] *Department of Chemistry, University of Zurich,*
Winterthurerstrasse 190, 8057 Zurich, Switzerland
[3] *Department of Physics, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland*
(Dated: August 20, 2020)

arXiv:2008.08144

Quantum Information Software Kit Qiskit

Qiskit is an open-source framework for working with noisy quantum computers at the level of pulses, circuits, and algorithms.

Terra and is the foundation on which the rest of Qiskit is built.

Aqua (Algorithms for QUantum computing Applications) providing a library of cross-domain algorithms upon which domain-specific applications can be built.

Ignis provides tools for quantum hardware verification, noise characterization, and error correction.

Aer provides high-performance quantum computing simulators with realistic noise models.

Quantum Information Software Kit Qiskit

| Qiskit Optimization, ML, Chemistry, Finance |
| Qiskit Aqua |
| Qiskit Terra |

Qiskit Ignis

Qiskit Aer

| Open Pulse | Open QASM |

| IBM QX |

| Hardware/Simulator |

Hardware/Simulator

IBM Quantum Hardware

Hardware/Simulator

Using Quantum Computers IBM Quantum Experience

Interacting with Quantum Computers Open Pulse / QASM

Code editor   Open in Quantum Lab

```
QASM  ⌄                              ⋮

1   OPENQASM 2.0;
2   include "qelib1.inc";
3
4   qreg q[3];
5   creg c[3];
6
7   z q[0];
8   h q[2];
9   cx q[0],q[1];
10  x q[2];
11  swap q[1],q[2];
```


sched9

Open Pulse	Open QASM

IBM QX
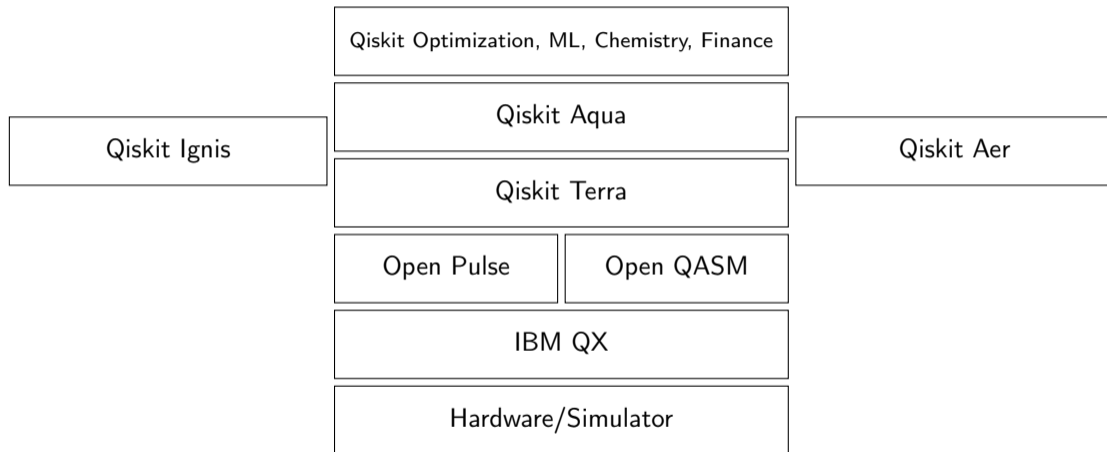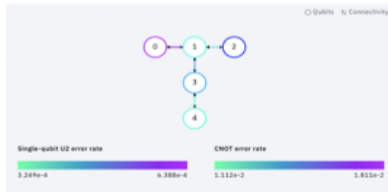
Hardware/Simulator

Interacting with Quantum Computers Qiskit Terra

```python
>>> from qiskit import *
>>> qc = QuantumCircuit(2, 2)
>>> qc.h(0)
>>> qc.cx(0, 1)
>>> qc.measure([0,1], [0,1])
>>> backend_sim = BasicAer.get_backend('qasm_simulator')
>>> transpiled_qc = transpile(qc, backend_sim)
>>> result = backend_sim.run(assemble(transpiled_qc)).result()
>>> print(result.get_counts(qc))
```

In this case, the output will be:

```
{'00': 513, '11': 511}
```

Qiskit Terra	
Open Pulse	Open QASM
IBM QX	
Hardware/Simulator	

Programming Applications on Quantum Computers Qiskit Chemistry


```
[6]: # from qiskit import IBMQ
     # provider = IBMQ.load_account()

[7]: backend = Aer.get_backend('statevector_simulator')

[8]: # setup COBYLA optimizer
     max_eval = 200
     cobyla = COBYLA(maxiter=max_eval)

     # setup HartreeFock state
     HF_state = HartreeFock(num_spin_orbitals, num_particles, map_type,
                            qubit_reduction)

     # setup UCCSD variational form
     var_form = UCCSD(num_orbitals=num_spin_orbitals, num_particles=num_particles,
                      active_occupied=[0], active_unoccupied=[0, 1],
                      initial_state=HF_state, qubit_mapping=map_type,
                      two_qubit_reduction=qubit_reduction, num_time_slices=1)

     # setup VQE
     vqe = VQE(qubitOp, var_form, cobyla)
     quantum_instance = QuantumInstance(backend=backend)
```

Step 4: Run algorithm and retrieve the results

```
[9]: results = vqe.run(quantum_instance)
     print('The computed ground state energy is: {:.12f}'.format(results.eigenvalue.real))
     print('The total ground state energy is: {:.12f}'.format(results.eigenvalue.real + energy_shift + nuclear_repulsion_energy))
     print("Parameters: {}".format(results.optimal_point))

     The computed ground state energy is: -1.057852461436
     The total ground state energy is: -7.861864759732
     Parameters: [3.03358125e-06 7.13391799e-05 3.03361004e-06 6.99450928e-05]
```

Qiskit Chemistry
Qiskit Aqua
Qiskit Terra

Open Pulse	Open QASM

IBM QX
Hardware/Simulator

Dedicated Software The User

Users without any prior knowledge to quantum computing and algorithms

Users developing quantum algorithms and qc software

Users developing hardware primitives

Users developing hardware

Qiskit Chemistry

Qiskit Aqua

Qiskit Terra

Open Pulse	Open QASM

IBM QX

Hardware/Simulator