Quantum Computing at DESY

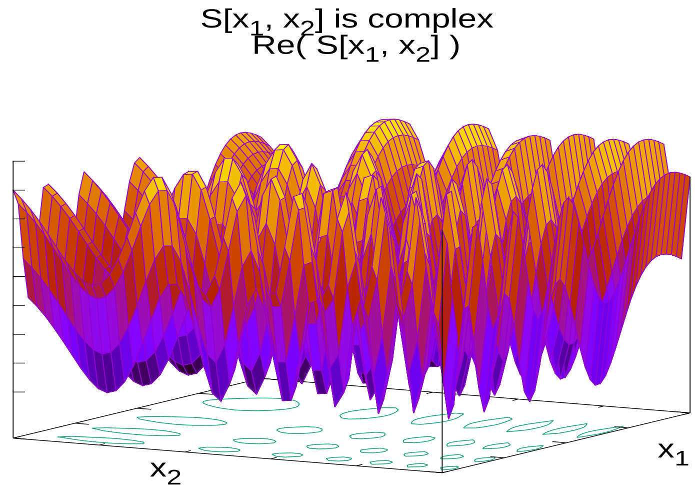
Karl Jansen

Markov Chain Monte Carlo (MCMC) Method

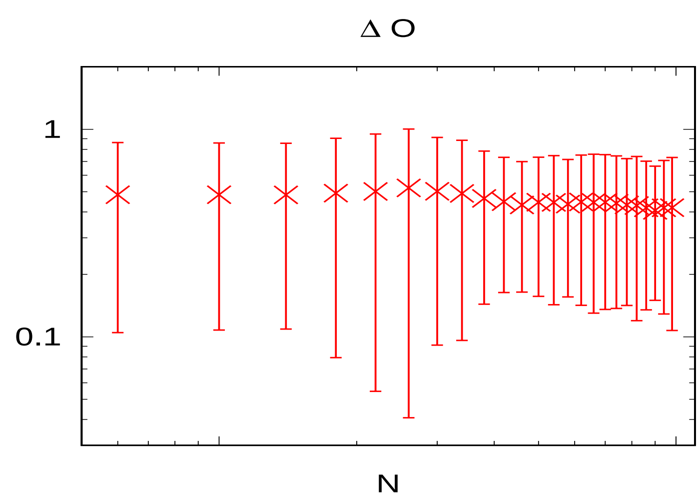
$\langle \mathcal{O} \rangle = \int \mathcal{D}_{\text{Fields}} \mathcal{O} e^{-S} / \int \mathcal{D}_{\text{Fields}} e^{-S}$

- very successful but needs real and positive probability density measure $\mathcal{D}_{\text{Fields}} e^{-S}$
- complex action not accessible to standard MCMC
  - no notion of probability distribution
  - importance sampling not possible
- MC: constant error $O(1)$ as function of sample size N

$\Rightarrow$ Sign problem

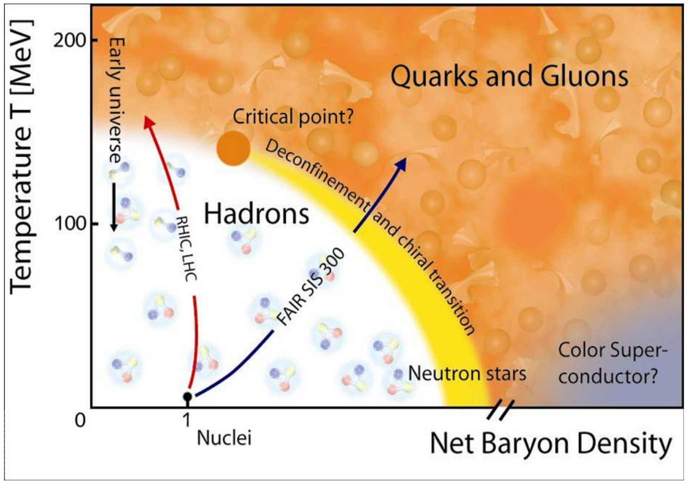


$\downarrow$

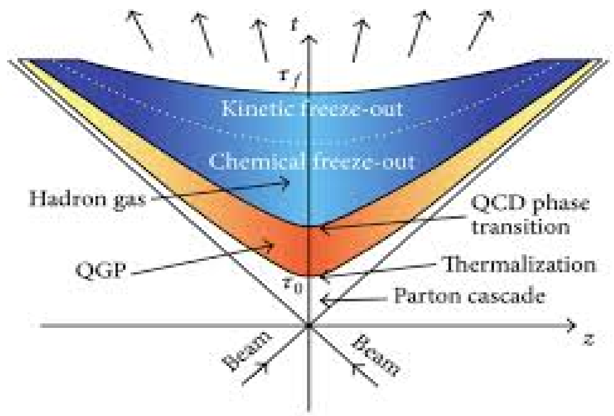


Sign problems in particle physics

- understanding the very early universe



- real time processes (particle collisions)



- strong CP-violation $\rightarrow$ topological term

Quantum Computing: applications at DESY

- Theoretical particle physics
  → variational quantum simulations of models in high energy physics
  (e.g.: T. Hartung, K.J., quantum computing ζ-regularized vacuum expectation values, arXiv:1808.06784 to appear in J.Math.Phys.)

- Experimental particle physics[1]
  → quantum annealing
  (e.g.: F. Babst et.al., A pattern recognition algorithm for quantum annealers, arXiv:1902.08324)

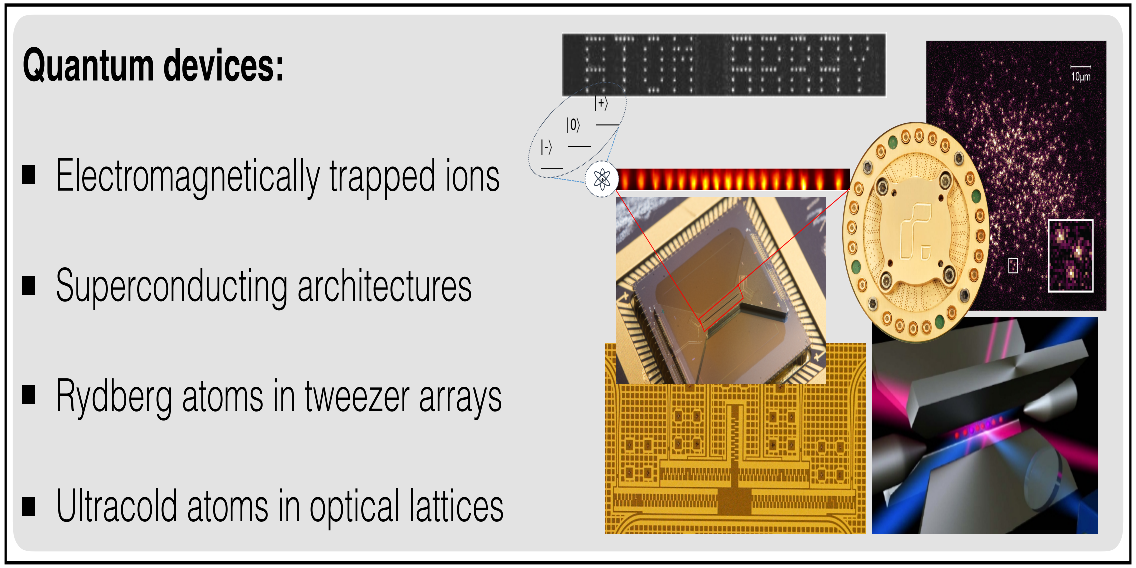
- Astroparticle physics
  → quantum networks
  (e.g.: E. T. Khabiboulline et.al., Quantum-Assisted Telescope Arrays arXiv:1809.03396, Phys.Rev.A100, 022316 (2019))

- Photon science
  → integrated x-ray optics for quantum optical applications
  (R. Röhlsberger, T. Salditt et.al.)

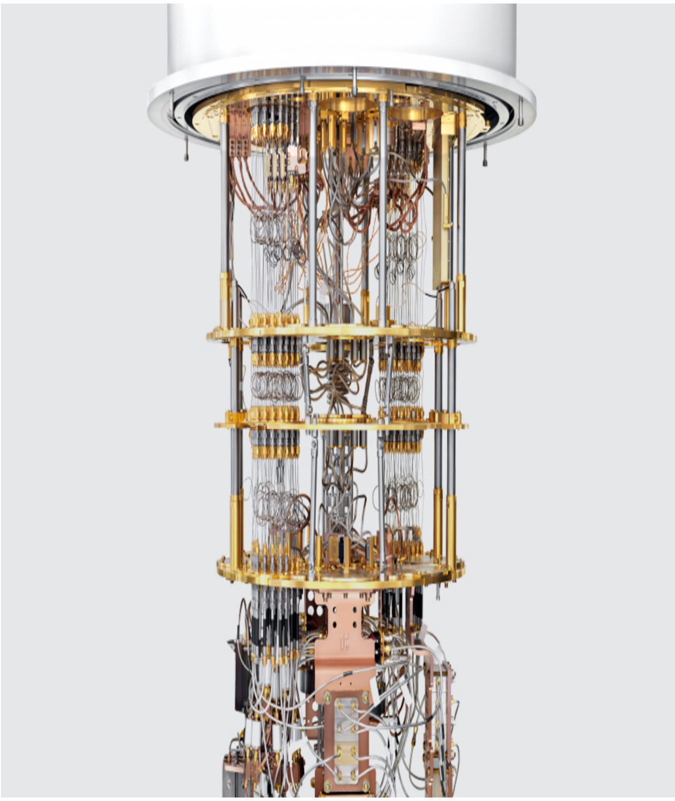
[1] CERN Workshop: https://indico.cern.ch/event/719844/timetable/

Quantum computing platforms

- Examples of quantum devices



Quantum Computing: when?



(Rigetti Aspen)

- Shielded to 50,000 times less than Earths magnetic field
- In a high vacuum: pressure is 10 billion times lower than atmospheric pressure
- Cooled 180 times colder than interstellar space (0.015 Kelvin)

→ prevent decoherence

- qubits based based on Josephson junction⋆
- application of unitary gate operations

→ generate entanglement

- reaching 128 qubits this year
- IBMQ: Experimental 50 qubits, Premium 20 qubits, Free 14 qubits

⋆ *A Josephson junction is formed by two superconducting regions that are separated by a very thin insulating barrier*

Quantum Computing: how?

- python programming language

  → company provides quantum libraries

- very convenient setup

  → simulator runs on your local machine

  → hardware usable through quantum cloud service

  → build on reservation system

- documentation, tutorials and examples availabe on website

  → you can start now

Example: hydrogen atom

(T. Hartung, K.J., arXiv:1808.06784, JMP)

- Hamiltonian

$$H = -\frac{\partial^2}{2m} + U(x)\ , \quad U = \begin{cases} x & ; x \in (0,\pi) \\ 0 & ; x \in (-\pi, 0] \end{cases}$$

- Hilbert space: $L_2(-\pi,\pi)$ with discretization

$$\varphi_k(x) := \frac{1}{\sqrt{2\pi}} e^{ikx}$$

- orthogonal projection

$$P_n[\mathcal{H}] = \mathrm{lin}\,\{\varphi_k; -n \leq k \leq n-1\}$$

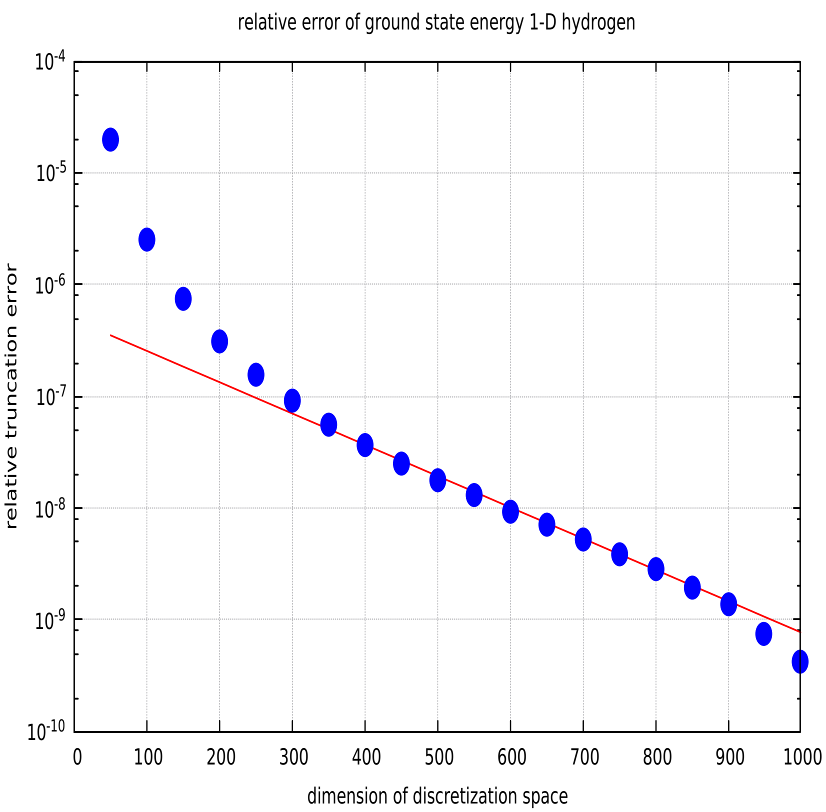
- matrix elements (for $m = 1$)

$$\langle \varphi_l, H\varphi_k \rangle = \frac{\big((-1)^{k-l}(1-i\pi(k-l))-1\big)}{2\pi(k-l)^2}$$

Truncation error

- exact diagonalization of matrix with elements

$$\langle \varphi_l, H\varphi_k \rangle = \frac{\left((-1)^{k-l}(1-i\pi(k-l))-1\right)}{2\pi(k-l)^2}$$



- relative error $\frac{E(j)-E(1024)}{E(1024)}$
- blue points: exact diagonalization
- red line: exponential fit
- find exponentially fast convergence

Hamiltonian for quantum computation

- the qubit Hamiltonian

$$H_q := \langle e_j, He_k \rangle_{j,k \in 2^Q}$$

- Pauli basis $[(1), \sigma_x, \sigma_y, \sigma_z]$

$$\left\{ S^q = \sigma^{q_{Q-1}} \otimes \sigma^{q_{Q-2}} \otimes \ldots \otimes \sigma^{q_0}; q \in 4^Q \right\}$$

- projecting H_q onto Pauli basis

$$H_Q = \sum_{q \in 4^Q} \frac{\mathrm{tr}\left(H_q S^q\right)}{2^Q} S^q$$

Variational quantum simulation

- start with some initial state $|\Psi_{\rm init}\rangle$
- apply succesive gate operations $\equiv$ unitary operations $e^{-iS\theta}$
- examples for S: σ_x, σ_y, σ_z, parametric CNOT

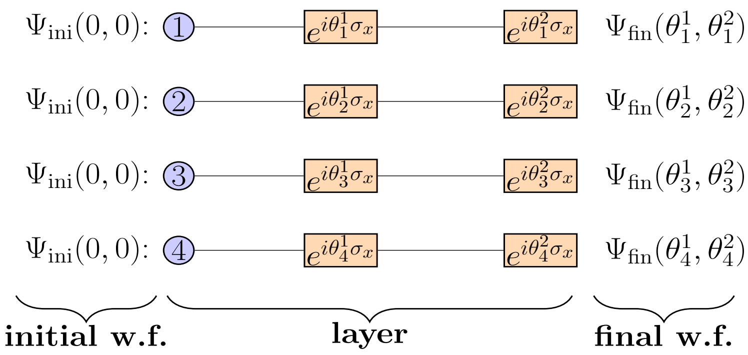
$$|\Psi(\vec{\theta})\rangle = e^{-iS_{(n)}\theta_n} \ldots e^{-iS_{(1)}\theta_1}|\psi_{\rm init}\rangle$$

- with $R_j := e^{-iS_{(j)}\theta_j}$ we obtain cost function

$$C := \left\langle \psi_{\rm init} \left| \left(\prod_{j=1}^n R_j\right)^\dagger H \prod_{j=1}^n R_j \right| \psi_{\rm init} \right\rangle$$

- goal: minimize C over the angles $\vec{\theta}$
  $\rightarrow$ obtain minimal energy, i.e. ground state

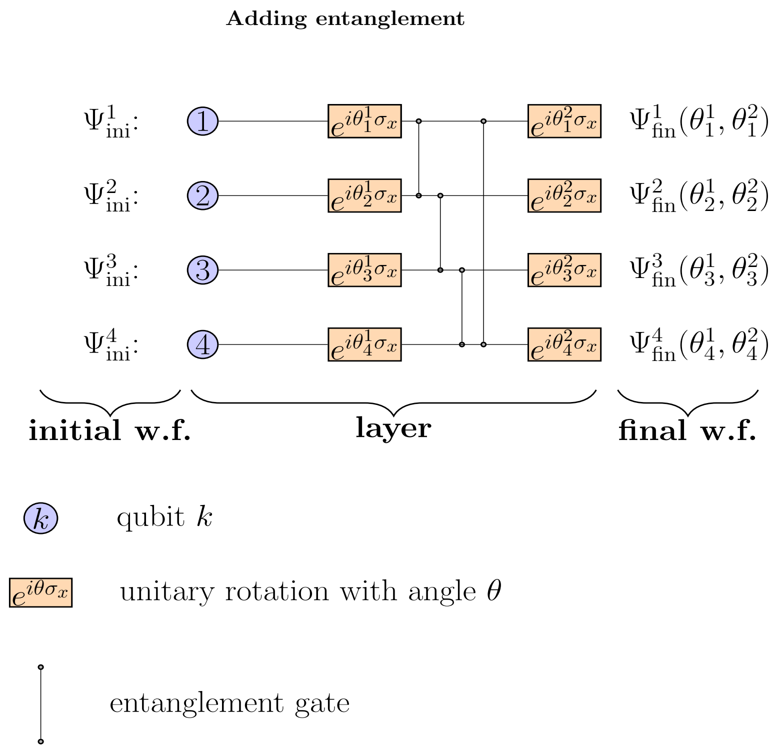
Example: only qubit rotations



k: qubit k

$e^{i\theta\sigma_x}$ unitary rotation with angle θ

Adding entanglement



Variational quantum simulation: brute force method

$\Leftarrow$ no library package provided worked

- method used in T. Hartung, KJ, J. Math.Phys.
  - quantum computer: evaluation of cost function
  - minimization on classical computer
  - minimize over one angle at a time

    $\rightarrow$ divide interval $\theta \in [0, 2\pi]$ in N steps

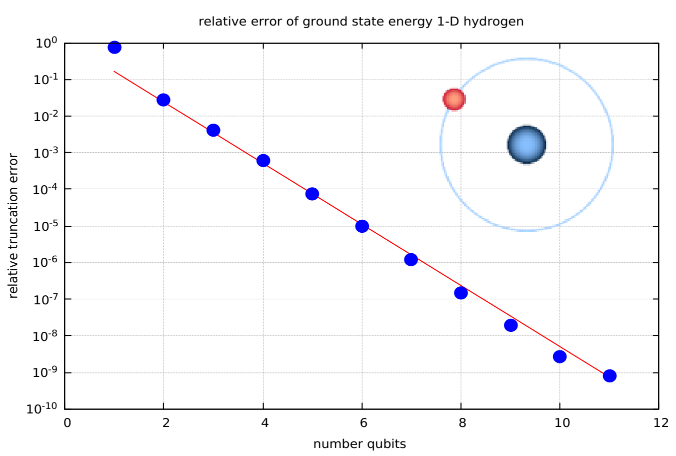
    $$\theta_j = 2\pi j/N \pm \epsilon \ , \ j = 1, ..., N$$

    ϵ random noise
  - loop several times over the complete set of angle

Simulation

- noise free simulation on Rigetti's QVM

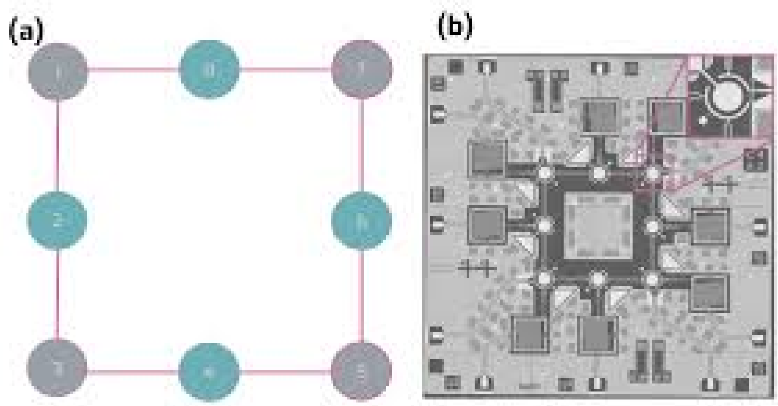
Q qubits	min. eig. H_Q	$\langle \psi_{2Q}, H_Q \psi_{2Q} \rangle$
1	.392108816647	.392108816647
2	.229395425745	.229395425968
3	.224258841712	.224258841747
4	.223452200306	.223452200445
5	.223336689755	.223336690423



- relatice error $\Delta E = \frac{E(q) - E(q{=}12)}{E(q{=}12)}$
- blue points: quantum simulator results
- find exponentially fast convergence
- line: fit $\Delta E(q) \approx 1.14 \cdot e^{-1.92q}$

Running on the Rigetti hardware

- Practical example: compute ground state energy of 1-dimensional hydrogen atom on Rigetti's 8-qubit Agave chip
  - performed variational quantum simulation



Agave chip

- hardware performance
  - select the qubit of the day
  - found gate fidelity $F_{1Q} = 0.982$, readout fidelity $F_{RO} = 0.94$
  - ground state energy with 4.9% error
  - more qubits: no significant result

Implement better algorithm

(T. Hartung, P. Stornati, KJ)

- new algorithm: quantum gradient descent
  - example for 2 unitaries, cost function:

$$C := \left\langle \psi_{\text{init}} \middle| \left(e^{i\sigma_x\theta_1}e^{i\sigma_y\theta_2}\right)^\dagger H e^{i\sigma_x\theta_1}e^{i\sigma_y\theta_2} \middle| \psi_{\text{init}} \right\rangle$$

  - derivative D of $R = e^{i\sigma_x\theta_1}e^{i\sigma_y\theta_2}$

    $D = (\frac{\partial R}{\partial\theta_1}, \frac{\partial R}{\partial\theta_2})$

  - obtain gradient of cost function

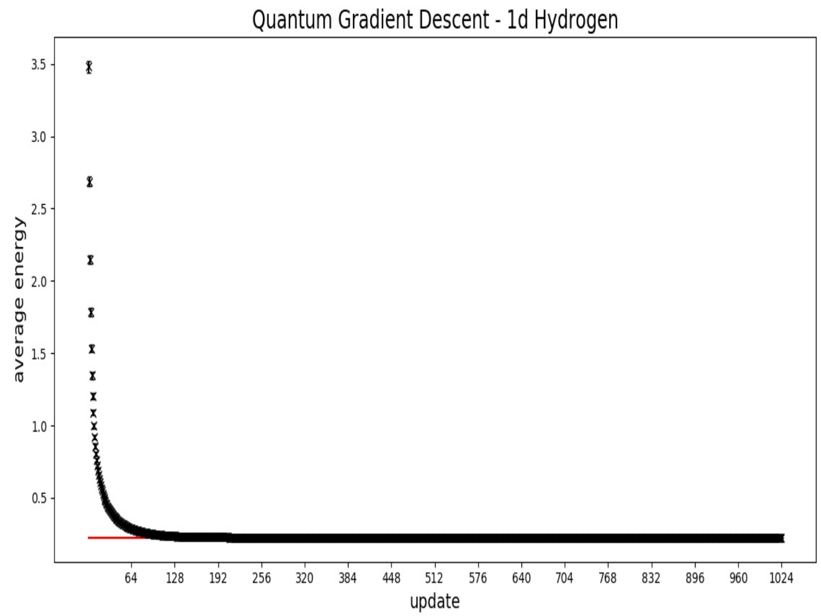
$$\partial/\partial\theta_1 C = \left\langle \psi_{\text{init}} \middle| \left(e^{i\sigma_x\theta_1}e^{i\sigma_y\theta_2}\right)^\dagger \left[Hi\sigma_x^\dagger + i\sigma_x H\right] e^{i\sigma_x\theta_1}e^{i\sigma_y\theta_2} \middle| \psi_{\text{init}} \right\rangle$$

- can re-use generated state vector, measure $Hi\sigma_x^\dagger + i\sigma_x H$
- obtain new vector of angles: $\vec{\theta}^{new} := \vec{\theta}^{old} - \eta\nabla C\left(\vec{\theta}^{old}\right)$
- tune "learning rate" η

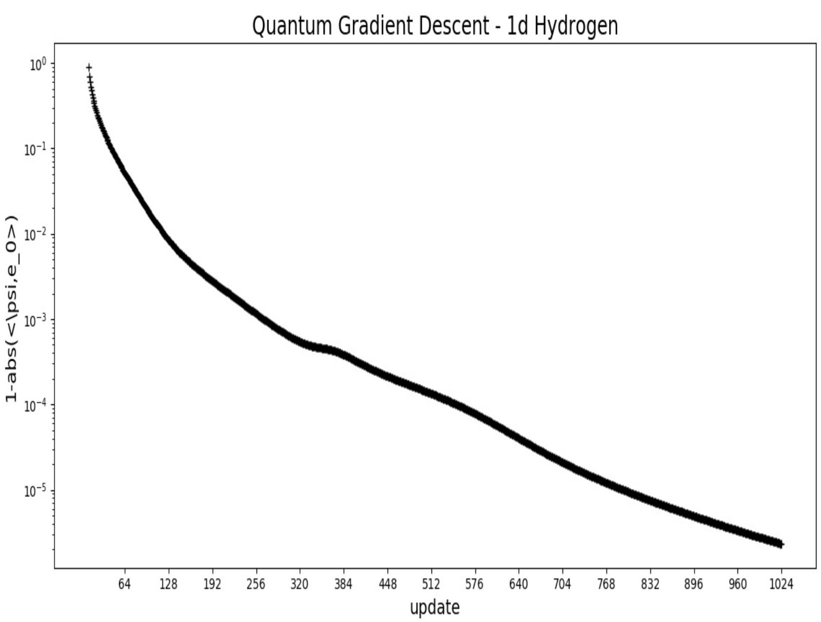
Generalization

- general form of the quantum gradient descent algorithm
  - cost function, $R_j = e^{iS_{(j)}\theta_j}$
$$C := \left\langle \psi_{\text{init}} \left| \left(\prod_{j=1}^{n} R_j \right)^{\dagger} H \prod_{j=1}^{n} R_j \right| \psi_{\text{init}} \right\rangle$$
  - define derivative operator
$$D_k := \left(\prod_{j=k+1}^{n} R_j \right) R'_k \left(\prod_{j=k+1}^{n} R_j \right)^{\dagger}$$
  - express CNOT gate in Pauli matrices
$$CNOT_{1\to 2} = \frac{\mathbb{1}_1 + \sigma_1^z}{2} \otimes \mathbb{1}_2 + \frac{\mathbb{1}_1 - \sigma_1^z}{2} \otimes \sigma_2^x$$
  - obtain kth-component of gradient
$$\partial_k C = \left\langle \prod_{j=1}^{n} R_j \psi_{\text{init}} \left| H D_k + (H D_k)^{\dagger} \right| \prod_{j=1}^{n} R_j \psi_{\text{init}} \right\rangle$$
- can re-use generated state vector, measure $HD_k + (HD_k)^{\dagger}$
- obtained new vector of angles: $\vec{\theta}^{new} := \vec{\theta}^{\text{old}} - \eta \nabla C \left(\vec{\theta}^{old} \right)$

Testing the algorithm



- average energy
  - simulator
  - no noise
  - 3 qubits
  - red line: exact result

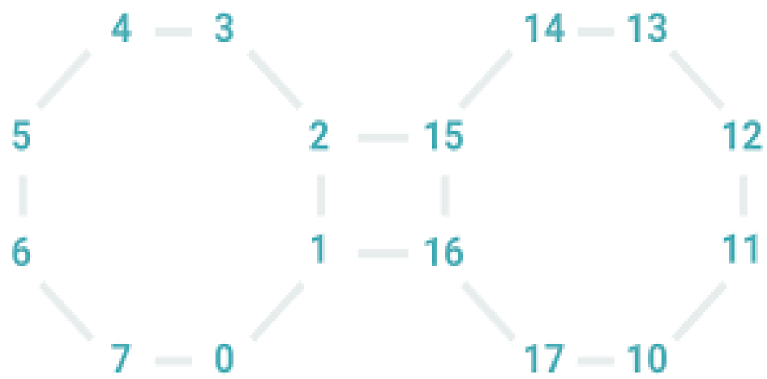


- distance to groundstate
- $|1 - \langle \Psi(j) | \Psi_0 \rangle|$

  Ψ_0 ground state wave function

New hardware

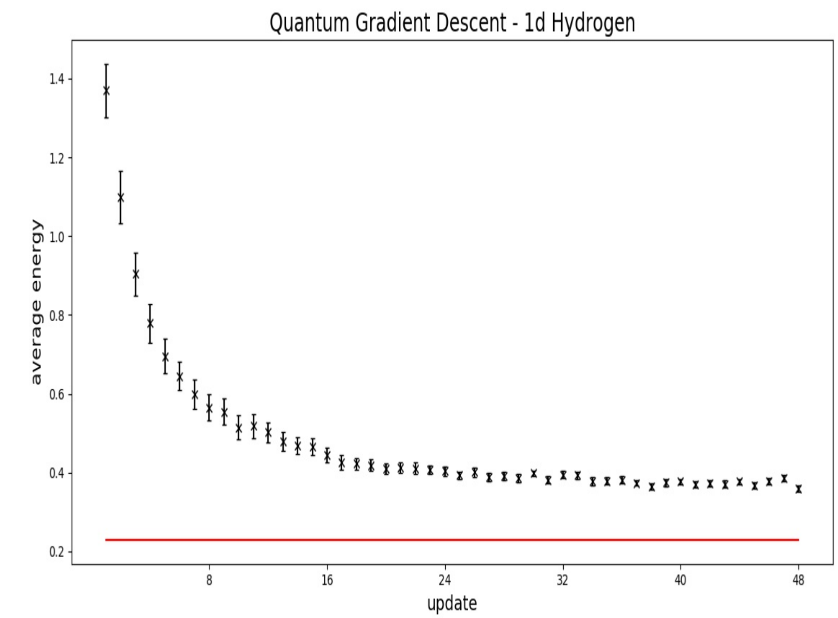
- new hardware: Aspen line (up to 128 qubits)



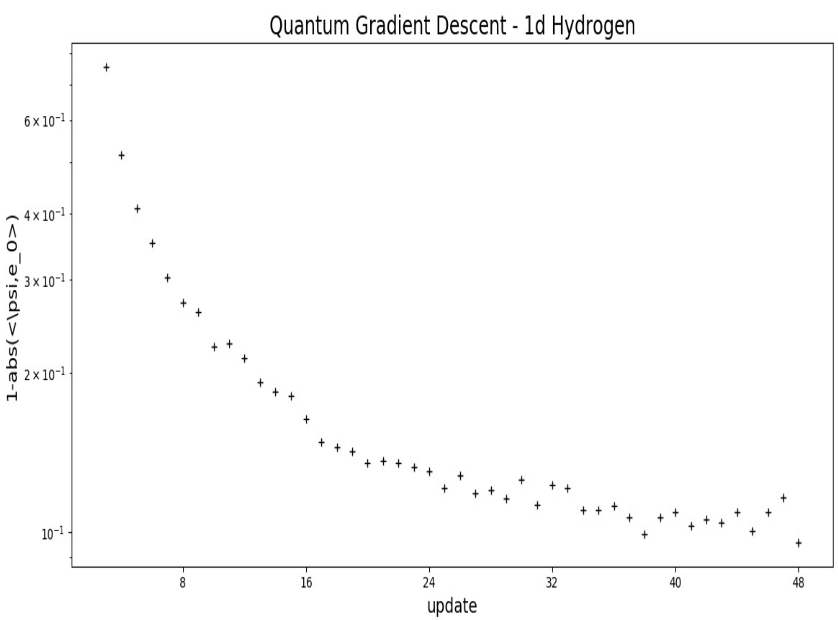
Aspen layout

- Rigetti's setup
  - Quantum Cloud Service
  - Pyquil (Python based) program enviroment
  - mean single qubit fidelity 95.5%

Running on Rigetti's Aspen hardware



- average energy
  - 2 qubits
  - red line: exact result
  - 60% fidelity



- distance to groundstate
- 90% fidelity
- reaching machine precision

Remarks

- with new hardware and better algorithm
  from impossible (2 qubits) $\rightarrow$ 90% fidelity for ground state

- still, 3 qubits: no significant result

- quantum gradient descent very expensive
  - justified for 1-d hydrogen atom (expensive itself)
  - for local Hamiltonians (e.g. Heisenberg)
    $\rightarrow$ need better algorithm (... and we are working on this)
    $\rightarrow$ can reach O(10) qubits

- quantum simulations of Schwinger model on real hardware
  - E.A. Martinez et.al., Nature 534 (2016) 516 (trapped ions)
  - N. Klco et.al., Phys.Rev. A98 (2018) no.3, 032331 (IBMQ)
  - C. Kokail et.al., Nature 569 (2019) no.7756, 355 (trapped ions, > 10 qubits)

Quantum Computing at DESY

- Helmholtz strategy paper: Quantentechnologies in the Helmholtz-Society
  - Quanten computing
  - Quanten communication
  - Quanten sensoring
  - Quanten materials
  - Simulation und numerical methods

- European project Quantentechnologies for Lattice Gauge Theorie (QTFLAG) (within European Quantum Flagship)

- Variational quantum simulations for lattice gauge theories

- one-way computing: collaboration with photon science (R. Röhlsberger, J. von Zanthier, K.J.)

- hope to find applications in experimental particle and astroparticle physics