Quantum emulators and scientific research

Thomas Ayral

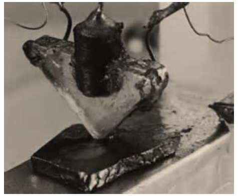
Atos Quantum Lab (France)

Thursday, June 13th 2019

Will These Get Quantum 1.0?

Fredrik Bull’s
“sorting-recording-adding” machine
1919

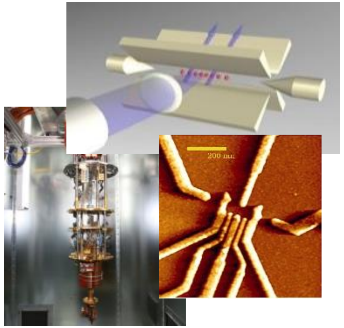
First quantum
revolution



Bull sequana
supercomputer

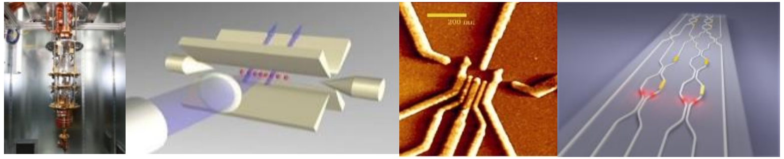
Will These Get Quantum 2.0?

Second quantum revolution



?

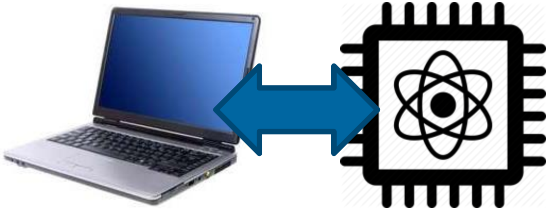
Why quantum hardware is not enough



- “Good qubits”: necessary… but not sufficient
- Beyond hardware:

Quantum algorithmics: very few known algorithms…
 - how to quickly program+validate new ideas?

NISQ era: before fault-tolerance
 - how to **compile** a program for a targeted hardware?
 - can we design **noise-resilient** algorithms?
 - **analog vs digital** approaches? …

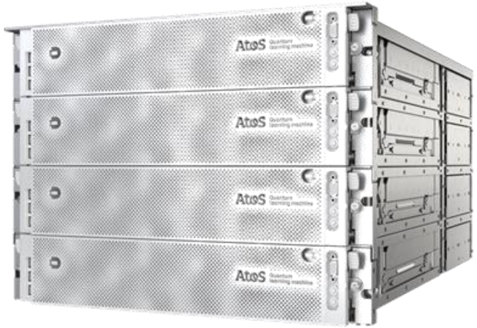


Quantum applications: “killer app(s)” yet to be found!
 - quantum-classical interface? (variational methods, error mitigation, pre- and post-treatment, parallelism…)

Need for (classical) programming/simulation/optimization tools

Outline

- **Modeling** quantum processors
- **Simulating** noisy quantum circuits with classical computers
- **Optimizing** noisy quantum circuits
- **Using** quantum processors to solve hard problems? Schwinger model example



Our collaborations

Quantum programming, classical simulation

BPI project Quantex (with Paris Saclay, CNRS-LORIA, CEA-LETI). *Quantum programming, hardware-acceleration for classical simulation of quantum circuits*

ANR SoftQPro (with Paris Saclay, CNRS-LORIA, CEA-LIST). *Numerical simulation of high-level quantum programming languages*.

CIFRE PhD thesis with Supélec/Saclay: *numerical techniques for quantum circuit generation*

Quantum algorithmics

QUANTERA QuantAlgo (with CNRS-IRIF, CWI-QuSoft, Cambridge, Univ. of Latvia, Univ. Libre de Bruxelles). *Machine Learning, exploration of use cases.*

ANR QuData (with CNRS-IRIF, Paris-Sorbonne, CNRS-LABRI). *Assessment of industrial use cases*.

CIFRE PhD thesis with IRIF: *algorithms for Quantum Machine Learning.*

Quantum hardware

EU-Flagship project AQTION (with Univ. Innsbruck (Blatt group), Oxford, ETHZ, Mainz, Fraunhofer, Swansea, Toptica). *Programming frontend, compilation, industrial use cases*

Chaire industrielle NASNIQ (with CEA-DRF Quantronics lab). *Computational architecture, noise models*

Merlion project Siliquon qubits (with CNRS-Néel, Singapore Institute of Quantum Computing). *Noise simulation of Si Qubits*

Analog quantum simulation

EU-Flagship PASQUANS Flagship project (with Institut d’Optique, Univ. Innsbruck (Zoller group), ETHZ, Univ Munich, LKB, Univ Strathclyde, Univ Ulm, Univ Padova, Univ Heidelberg).

WP leader of applications

1

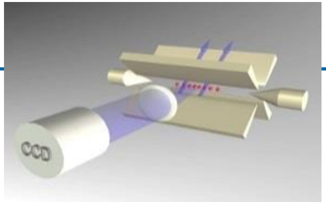
Modeling quantum processors

All qubits are not created equal

Superconducting qubits

- Coupled anharmonic oscillators (Josephson junctions)

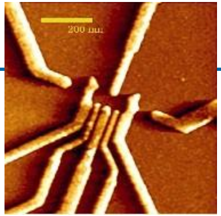
- Gates ~ 200 ns, coherence ~ 50 μs: 250 gates, 1-f=0.002
- ~10 entangled qubits, ~50 qbit chips announced
- Limited connectivity
- Limited scalability (need very precise calibration)?



Trapped ions

- Entangle internal degrees of freedom via motional mode

- Gates ~ 50 us, coherence ~ 200 ms: 4k gates, 1-f = 0.05
- All-to-all connectivity
- ~ 15 entangled qubits
- Limited scalability (large crystal: vibration modes…)?
- Alternatives: shuttling…



(etc)

Quantum dots

- Entangle electron spins via exchange J

- 1 and 2-qbit gates demonstrated… not more!
- Leverage CMOS technology: Scalable?

Yet, we want to be able to program them in a hardware-agnostic way.

How to describe noisy quantum computers?

- Quantum processors: fragile quantum systems in classical environment
  - Noise: -> mixed quantum state: **density matrix $\boldsymbol{\rho}$** (instead of $|\psi\rangle$)

    $2^n \times 2^n$ matrix instead of 2^n vector! (n: number of qbits)

- **Several approaches to describe noise**
  - Continuous-time approach: master equation

    e.g Lindblad master equation (Markovian noise: memory-less env.)

$$i\frac{d\rho}{dt} = [H,\rho] + i\sum_n \gamma_n \left(L_n \rho L_n^+ - \frac{1}{2}\{L_n^+ L_n, \rho\} \right)$$

    - requires knowledge of γ_n, L_n
    - can reduce cost to 2^n: quantum trajectories
    - continuous time: not very suitable for optimized compilation:

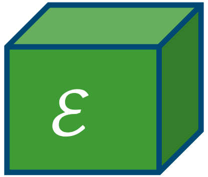
we would like to adopt **discrete** description

Building blocks: Noisy gates

- Quantum state evolution via ‘quantum channels’ or ‘CPTP maps’

$$\rho \to \mathcal{E}(\rho)$$

  - **complete positivity** (CP): “unitary total evolution”
  - **trace preservation** (TP): “no leakage”



- Several representations of quantum channels:
  - **Kraus representation**:

$$\mathcal{E}(\rho) = \sum_k E_k \rho E_k^{+}$$

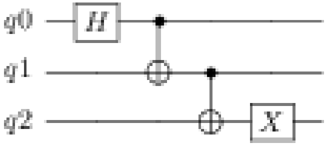
  - **Pauli transfer matrix** (PTM): (Chow et al ‘12)

$$\overrightarrow{\mathcal{E}(\rho)} = \boldsymbol{R} \cdot \vec{\rho}$$

how to specify noisy gates for a given hardware?

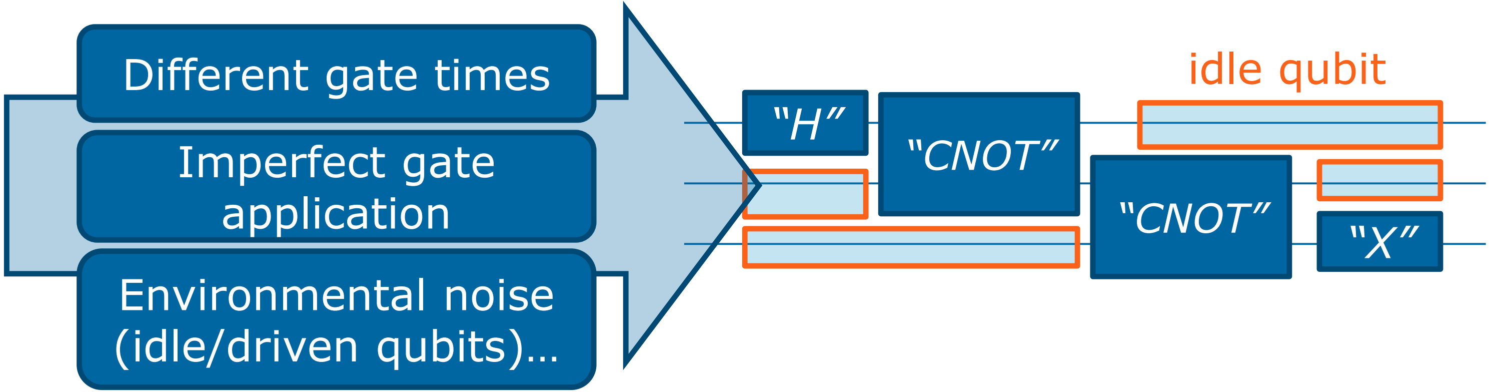
Discrete modeling of noise

Ideal circuit



Hardware model

Noisy circuit


Different gate times

Imperfect gate application

Environmental noise (idle/driven qubits)…

- Modular, hence **scalable** approach
- Captures **large variety of noises** (incl. spatially correlated, crosstalk, leakage…)

- **Caveat**: does not capture all memory effects

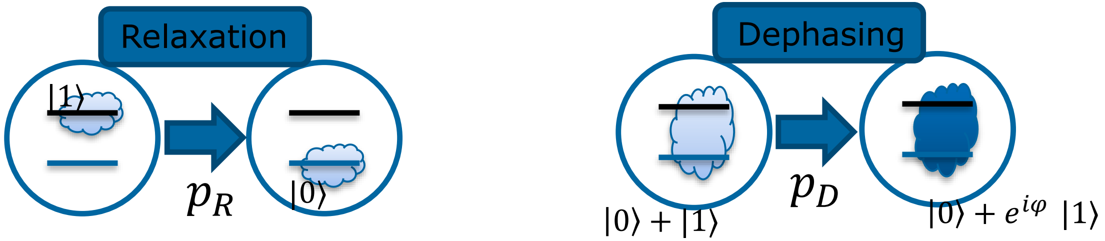
What do the boxes look like?

Textbook noise models…

- **Bit-flip noise**: $\mathcal{E}(\rho) = (1 - p_F)\rho + p_F X\rho X$
  "flip bit with probability p_F"
  Two Kraus operators: $E_1 = \sqrt{1 - p_F}\,\mathrm{Id}$ and $E_2 = \sqrt{p_F}\,X$

see e.g. Nielsen & Chuang

- **Relaxation** (aka amplitude damping) **and dephasing:**

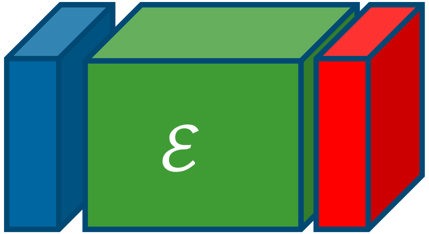


Challenge: what is p_F, p_R, p_D … for a given hardware?
Are there other important types of noise?

… and more ‘ab initio’ approaches: tomography

- Characterize $\mathcal{E}(\rho)$: **“process tomography”**
- Two complementary strategies:

prepare
($\geq 4^{nqbits}$)
basis states



project on
basis
states

Experiment/phenomenology

- **Quantum process tomography**: 16 circuits for 1-qbit gate, 256 for 2-qbit… (x lots of shots)
- **Gateset tomography**: consistent treatment of “SPAM” error

Merkel et al 2013

Advantage: phenomenological (“black box”) approach

Numerics

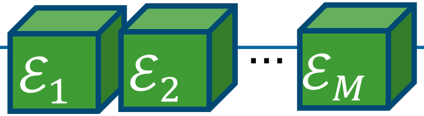
- Write **Hamiltonian model** for given operation
- Solve **Schrödinger/master equation** for all inputs

Advantage: inputs are usually experimentally accessible (microscopic) parameters.

2

Classical simulation (emulation) of ideal and noisy circuits

What we want to compute



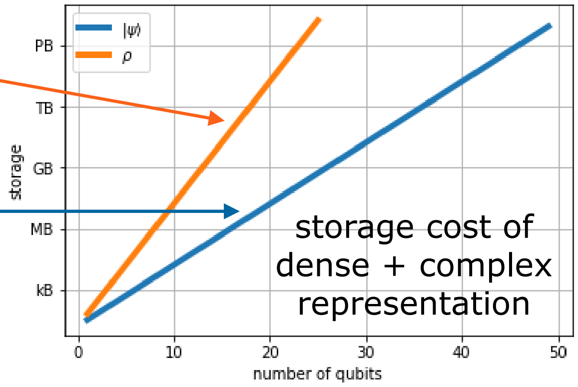
- Final density matrix after M operations:

$$\mathcal{E}(\boldsymbol{\rho}) = \sum_{k_M}\left(E_{k_M}^{(M)} \dots \left(\sum_{k_1} E_{k_1}^{(1)}\boldsymbol{\rho}E_{k_1}^{(1)\dagger}\right)\dots E_{k_M}^{(M)\dagger}\right)$$

  - Large memory requirement ($2^n \times 2^n$ matrices)

- Compare to **"ideal" quantum computation**:
  - Pure state $\boldsymbol{\rho} = |\psi\rangle\langle\psi|$…: "only" 2^n complex numbers
  - Only one Kraus op.: $\mathcal{E}(\boldsymbol{\rho}) = U\boldsymbol{\rho}U^\dagger$:

    unitary evolution $U_M \dots U_1|\psi\rangle$



storage cost of dense + complex representation

- Both require advanced numerical techniques **to reach few tens of qbits**

Ideal circuit simulation

Unitary evolution $|\psi_f\rangle = U_M \dots U_1 |\psi_i\rangle$

- **"Brute-force" simulation:**
  - store 2^n amplitude vector:

$$|\psi\rangle = \sum_{b_1..b_N} a_{b_1..b_N} |b_1..b_N\rangle$$

$a_{b_1 b_2 b_3 b_4} =$

  - up to 40-41 qubits

Ideal circuit simulation

Unitary evolution $|\psi_f\rangle = U_M \ldots U_1|\psi_i\rangle$

- **“Brute-force” simulation:**
  - store 2^n amplitude vector:

$$|\psi\rangle = \sum_{b_1..b_N} a_{b_1..b_N}|b_1..b_N\rangle$$

$a_{b_1b_2b_3b_4} =$ 

  - up to 40-41 qubits

- **Stabilizer simulation**
  - Only “Clifford” gates: can represent state with $O(n^2)$ cost
  - can simulate >1000 qubits
  - Extensions for Clifford+T…

Gottesman & Knill

Ideal circuit simulation

Unitary evolution $|\psi_f\rangle = U_M \ldots U_1 |\psi_i\rangle$

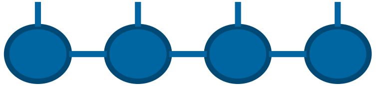
- **“Brute-force” simulation:**
  - store 2^n amplitude vector:

$$|\psi\rangle = \sum_{b_1..b_N} a_{b_1..b_N} |b_1..b_N\rangle$$

$$a_{b_1 b_2 b_3 b_4} =$$

  - up to 40-41 qubits

- **Using entanglement structure: Matrix product state (and tensor networks)**
  - MPS representation (4 qbits):



see e.g
Schollwöck ‘10,
Orus ‘14

  - physics insights from “bond dimension”
  - can simulate $\gg$ 40 qubits

- **Stabilizer simulation**
  - Only “Clifford” gates: can represent state with $O(n^2)$ cost
  - can simulate >1000 qubits
  - Extensions for Clifford+T…

Gottesman & Knill

Ideal circuit simulation

Unitary evolution $|\psi_f\rangle = U_M \ldots U_1 |\psi_i\rangle$

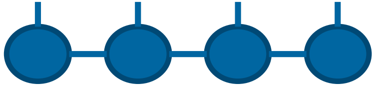
- **"Brute-force" simulation:**
  - store 2^n amplitude vector:

$$|\psi\rangle = \sum_{b_1..b_N} a_{b_1..b_N} |b_1 .. b_N\rangle$$

$a_{b_1 b_2 b_3 b_4} =$

  - up to 40-41 qubits

- **Using entanglement structure: Matrix product state (and tensor networks)**
  - MPS representation (4 qbits):



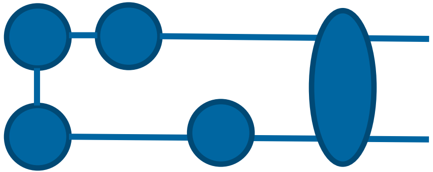
see e.g
Schollwöck '10
Orus '14

  - physics insights from "bond dimension"
  - can simulate $\gg 40$ qubits

- **Stabilizer simulation**
  - Only "Clifford" gates: can represent state with $O(n^2)$ cost
  - can simulate >1000 qubits
  - Extensions for Clifford+T…

Gottesman & Knill

- **Tensor network contraction**
  - space+time network (2 qbits + 4 gates)



  - challenge: find fast contraction order

Noisy simulation

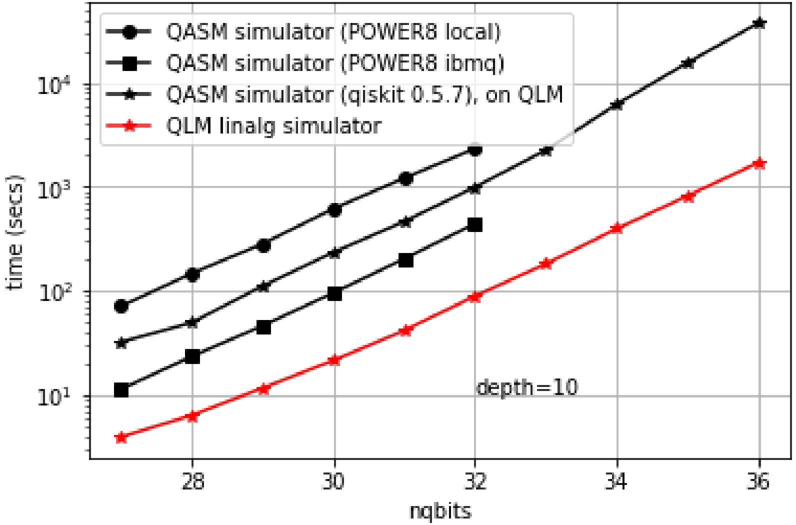
- Density-matrix evolution: $\mathcal{E}(\boldsymbol{\rho}) = \sum_{k_M} \left(E_{k_M}^{(M)} \dots \left(\sum_{k_1} E_{k_1}^{(1)} \boldsymbol{\rho} E_{k_1}^{(1)\dagger} \right) \dots E_{k_M}^{(M)\dagger} \right)$
  - ~OK for <20 qubits
- More qubits: rewrite as sum over “**trajectories**”

$$\mathcal{E}(\boldsymbol{\rho} = |\boldsymbol{\psi_0}\rangle\langle\boldsymbol{\psi_0}|) = \sum_{k_1..k_M} E_{k_M}^{(M)} \dots E_{k_1}^{(1)} |\boldsymbol{\psi_0}\rangle\langle\boldsymbol{\psi_0}| E_{k_1}^{(1)\dagger} \dots E_{k_M}^{(M)\dagger}$$

- One trajectory $k_1 .. k_M$: apply “gates” $E_{k_1}^{(1)} \dots E_{k_M}^{(M)}$ to $|\psi_0\rangle$
  - Reduction of storage cost… but exponential #trajectories
- **Stochastic sampling** over all possible trajectories

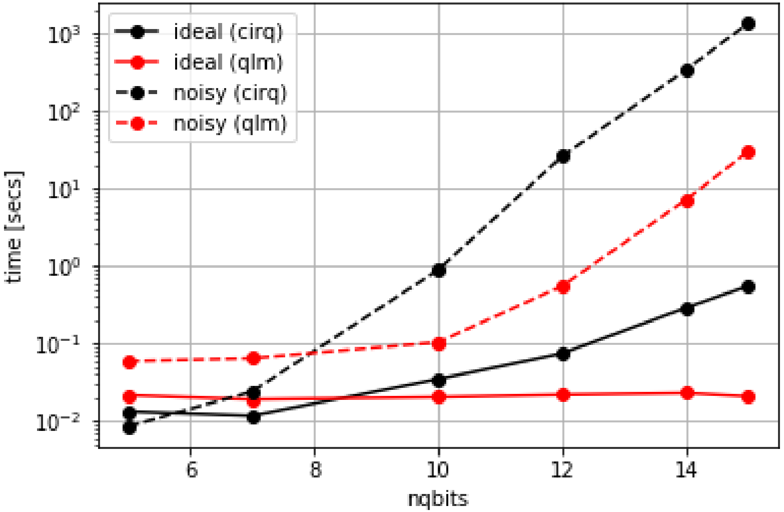
Speed comparison

Atos QLM vs IBM Qiskit simulator
Ideal simulation
'Quantum volume' benchmark circuit



QLM: **qat.linalg** simulator
IBM POWER8 benchmark data from
www.ibm.com/blogs/research/2018/05/quantum-circuits

Atos QLM vs Google cirq simulator
Ideal + noisy simulation
H+CNOT circuit



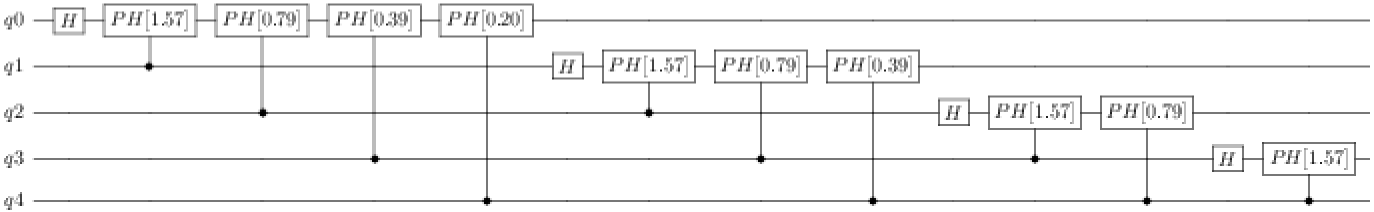
QLM: **qat.linalg and qat.noisy** simulator

2

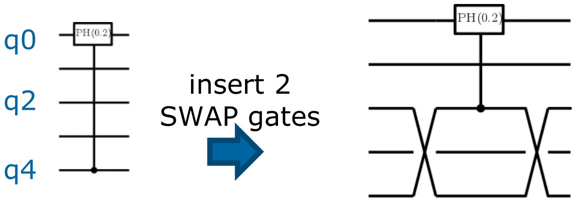
A word on compilation

Compilation under connectivity constraints

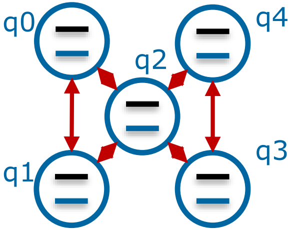
- Concrete example: Quantum Fourier Transform



- Compilation for connectivity:



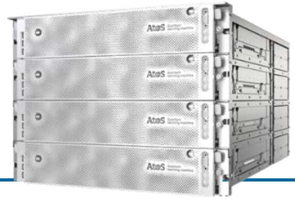
insert 2
SWAP gates



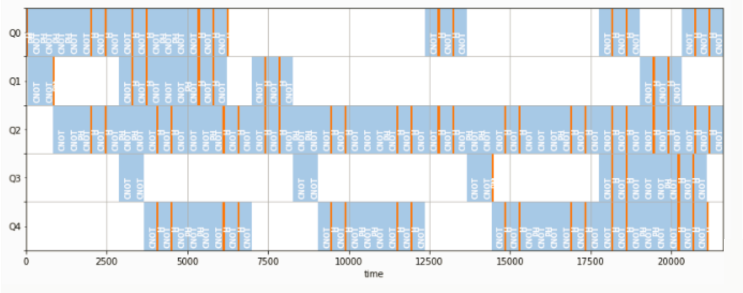
(here: undirected graph)

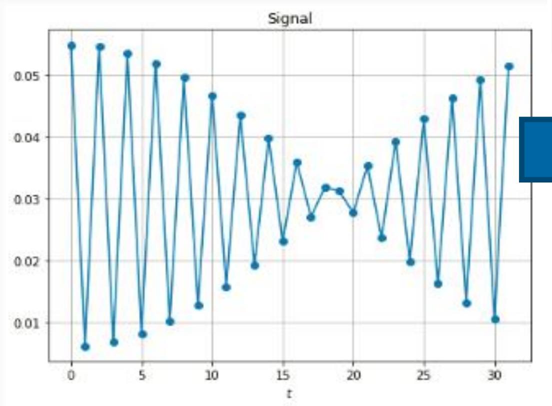
Large gate count overhead!

Example: Quantum Fourier Transform on 5 noisy qubits

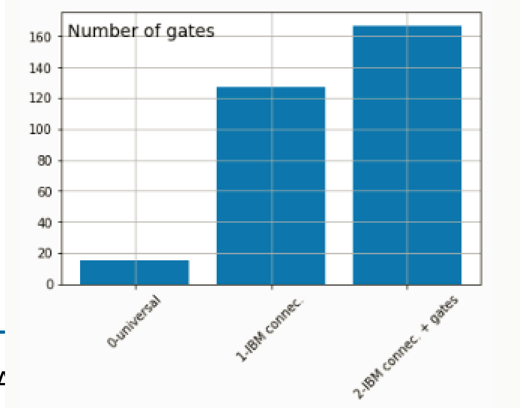


Decoherence noise only
on idle qubits,
perfect gates

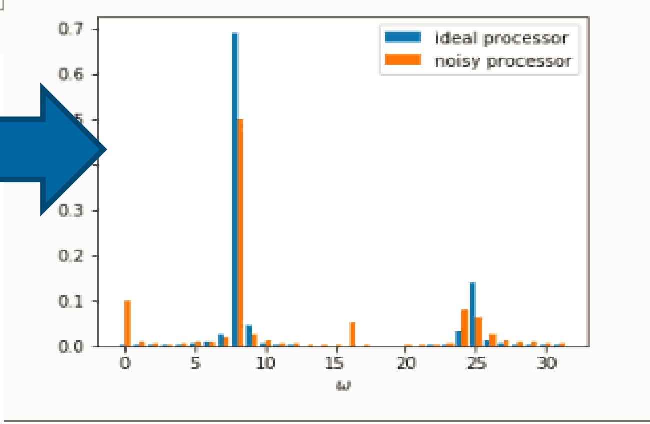




compilation, simulation



Typical noise
parameters
for IBM
device

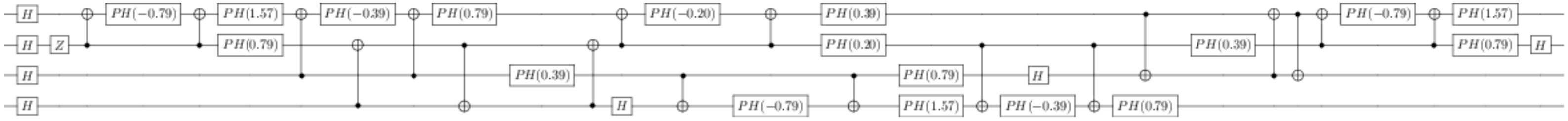


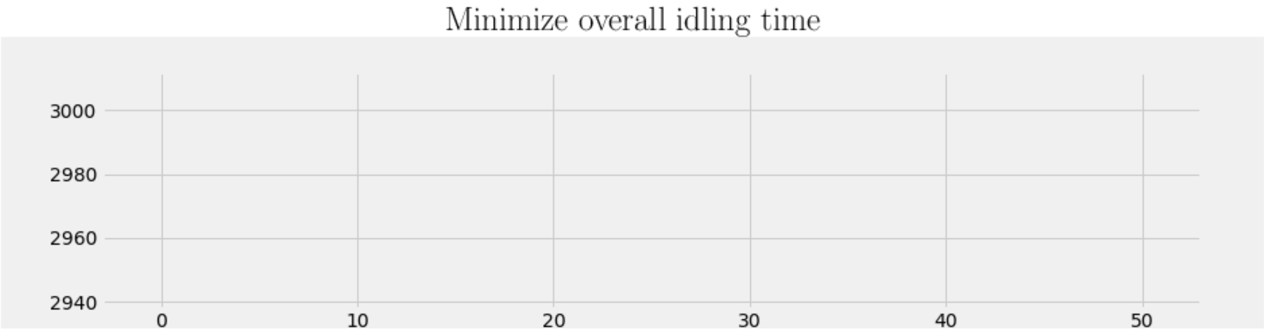
3

Optimizing quantum circuits

Example 1: Minimization of the total idling time

- Start from QFT compiled for connectivity & gateset
- Use **commutation patterns** to reduce **total idling time**
- Minimization via (classical) **simulated annealing**





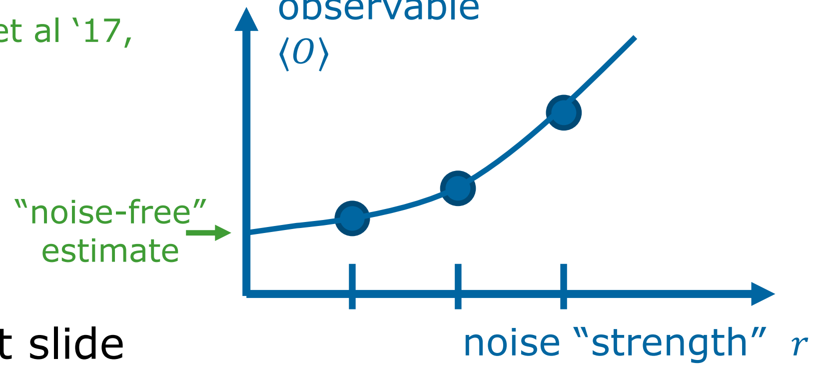
total idling time

annealing steps

Example 2: Error mitigation

- Error mitigation: before full-fledged quantum error correction

- Several methods, e.g   Li & Benjamin ‘17, Temme et al ‘17, Endo et al ‘17…
  - **Zero-noise extrapolation**: artificially enhance noise and extrapolate
  - **Quasiprobability error mitigation**: next slide



Software-based methods requiring
good control and/or knowledge of hardware

Quasiprobability error mitigation

(Temme et al `17, Endo et at `17)

- Observable: $\langle O\rangle = \ll O|\mathcal{E}_M \dots \mathcal{E}_1|\rho \gg \neq \langle O^{(0)}\rangle = \ll O^{(0)} \left|\mathcal{E}_M^{(0)} \dots \mathcal{E}_1^{(0)}\right| \rho^{(0)} \gg$

  with $\mathcal{E}_i$ ($\mathcal{E}_i^{(0)}$): imperfect (perfect) hardware operations

- Two steps:

1. **Linear decomposition** of perfect ops. on hardware ops.:

$$\mathcal{E}_i^{(0)} = \sum_{j_i} q_{j_i}^{(i)} \mathcal{E}_{j_i}$$

   Obtain sum over (exponentially many) "trajectories":

$$\langle O^{(0)}\rangle = \sum_{j_1 \dots j_M} q_{j_1}^{(1)} \dots q_{j_M}^{(M)} \ll O|\mathcal{E}_{j_M} \dots \mathcal{E}_{j_1}|\rho \gg$$

2. **Stochastic sampling** over trajectories

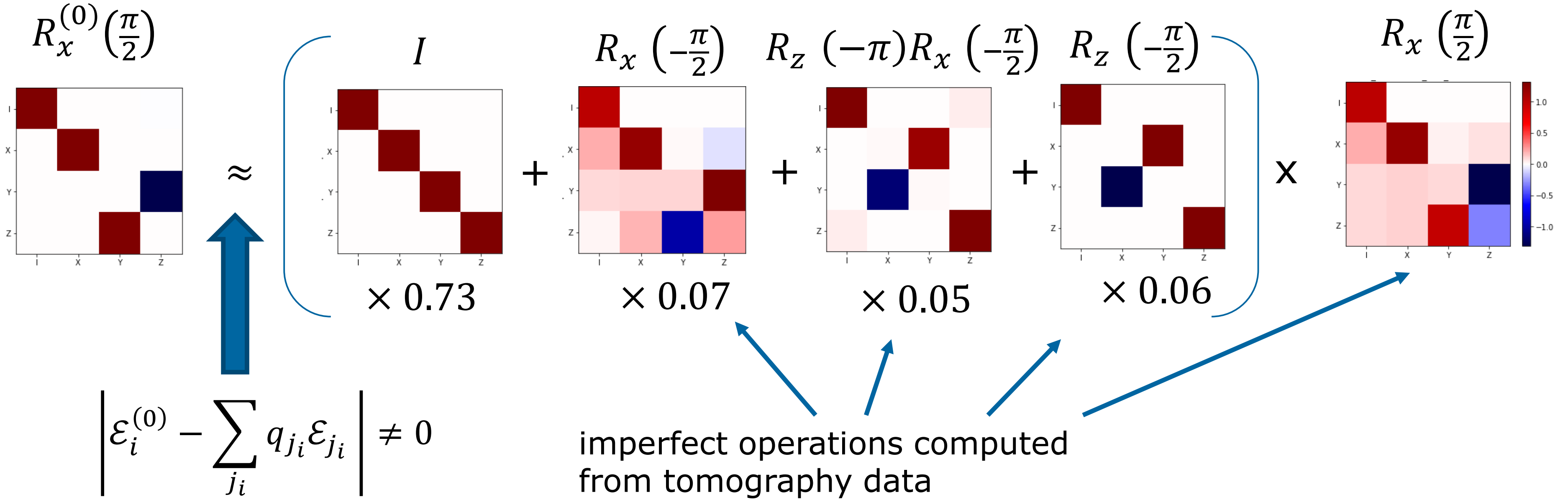
Caveats:

1) $\mathcal{E}_{j_i}$ should form a basis! (incl. non-unitary ops)

2) q_j not necessarily positive: "quasiprobabilities" (-> increased variance: 'sign problem')

Example of decomposition

- Using gateset tomography data on IBM QX4:



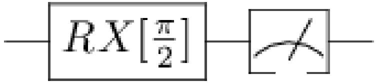
$$R_x^{(0)}\left(\frac{\pi}{2}\right) \approx \left(I \times 0.73 + R_x\left(-\frac{\pi}{2}\right) \times 0.07 + R_z(-\pi)R_x\left(-\frac{\pi}{2}\right) \times 0.05 + R_z\left(-\frac{\pi}{2}\right) \times 0.06 \right) \text{x}\; R_x\left(\frac{\pi}{2}\right)$$

$$\left| \mathcal{E}_i^{(0)} - \sum_{j_i} q_{j_i}\mathcal{E}_{j_i} \right| \neq 0$$

imperfect operations computed from tomography data

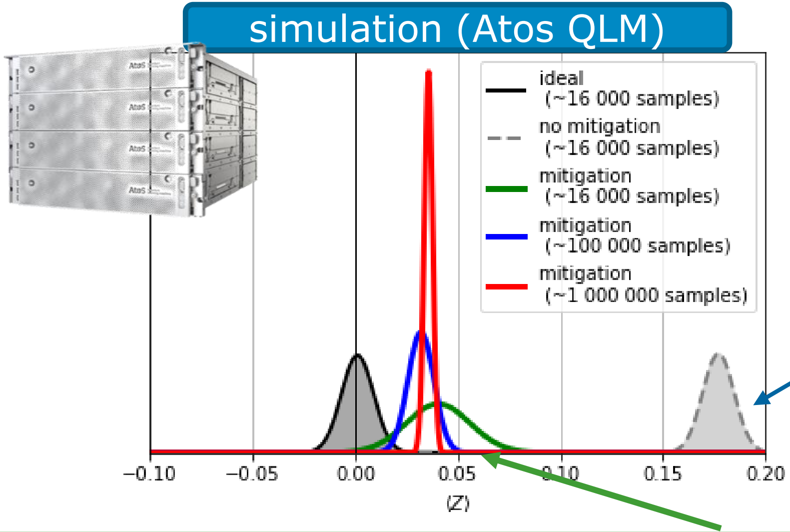
Comparison: noisy simulation vs. experiment

- Example circuit:

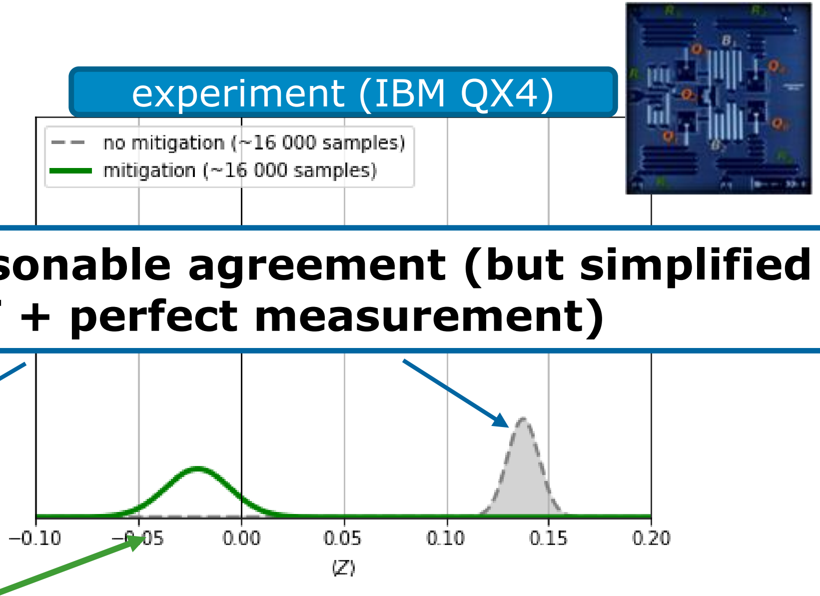


- Output state is (supposedly) $\propto (|0\rangle - i|1\rangle)$. Observable: $\langle Z \rangle = 0$
- Preliminary results:

simulation (Atos QLM)



experiment (IBM QX4)



reasonable agreement (but simplified GST + perfect measurement)

much more accurate, but still biased (limited operation set $\left|\mathcal{E}_i^{(0)} - \sum_{j_i} q_{j_i}\mathcal{E}_{j_i}\right| \neq 0$**)**

4

Potential applications of NISQ computers

Applications

- **Natural target: Quantum simulation** (equilibrium and dynamics)
  - quantum chemistry
  - condensed-matter physics

$$H|\psi\rangle = E|\psi\rangle \qquad i\frac{d|\psi\rangle}{dt} = H(t)|\psi\rangle$$

exponential cost with system size (or Monte-Carlo sign problem)

- **(Classical) Combinatorial optimization problems**
  - e.g clustering

  - Minimize cost function… which can be discretized -> Boolean function $C(z_1, \ldots, z_N)$ (classical Ising Hamiltonian <-> spin glass problems)

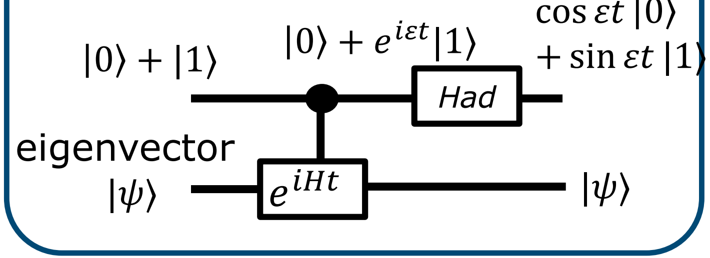
- **And also** (later): factoring (Shor), quantum machine learning, database search (Grover)…

Two approaches to quantum simulation

- **Analog approach**
  - Ultra-cold atoms, trapped ions, Rydberg atoms…
    - Good (global) control of a given Hamiltonian
  - Superconducting circuits: quantum annealing (d-wave)
    - Find ground state of (classical) Ising model

- **Digital approach:** gate-based (universal)
  - Superconducting circuits, trapped ions…
  - **Long term:**
    - **Quantum phase estimation:** find eigenvalue of an eigenvector: $\mathrm{e}^{iHt}|\psi\rangle = e^{i\varepsilon t}|\psi\rangle$
    - Requires very deep (long) quantum circuits: state preparation, trotterization…

"Phase kick-back"



Hybrid variational algorithms

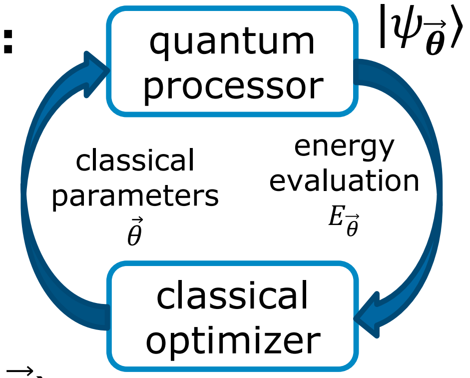
- **NISQ (Noisy intermediate scale quantum) computers:**
  - small number of qubits + short coherence times

- **Variational Quantum Simulation** (VQE, Peruzzo ’14, VQS, Kokail ’19…)

Goal: find ground state energy of Hamiltonian H

1. Choose (smart) variational ansatz $|\psi_{\vec{\boldsymbol{\theta}}}\rangle$ (with polynomial $\vec{\boldsymbol{\theta}}$)

   e.g (unitary) coupled cluster $\left|\psi_{\vec{\boldsymbol{\theta}}}\right\rangle \propto e^{i\sum \theta c^{+}c+h.c}|\phi_{HF}\rangle$
2. Find (short) quantum circuit / hamiltonian that creates $\left|\psi_{\vec{\boldsymbol{\theta}}}\right\rangle$
3. Measure variational energy $E_{\vec{\theta}} = \left\langle\psi_{\vec{\boldsymbol{\theta}}}|H_T|\psi_{\vec{\boldsymbol{\theta}}}\right\rangle$ with $H_T = \sum_\alpha \lambda_\alpha P_\alpha$, P_α product of Pauli operators
4. Use classical optimizer to find optimal $\vec{\theta}^*$



quantum processor

$|\psi_{\vec{\boldsymbol{\theta}}}\rangle$

classical parameters $\vec{\theta}$

energy evaluation $E_{\vec{\theta}}$

classical optimizer

Applications of variational quantum algorithms

- Quantum chemistry
  - Many small molecules have been studied (H2, LiH, …)
  - Well-known variational states: unitary coupled cluster (UCC), etc.

- Combinatorial optimization
  - Quantum approximate optimization algorithm (QAOA): special ansatz inspired from quantum annealing

- **Focus:** Variational quantum algorithms for **quantum field theory**?
  - Lattice QCD: a gauge theory plagued by Monte-Carlo sign problem in interesting regimes (hot quark-gluon plasma, neutron stars…)
  - Here, take Schwinger model (1+1-dim QED) as proxy for lattice QCD physics

Challenge I: translate problem to quantum computer language

- (Kogut-Susskind) **fermions**… in **spin**/qbit-based quantum computer?
  - Jordan-Wigner transformation

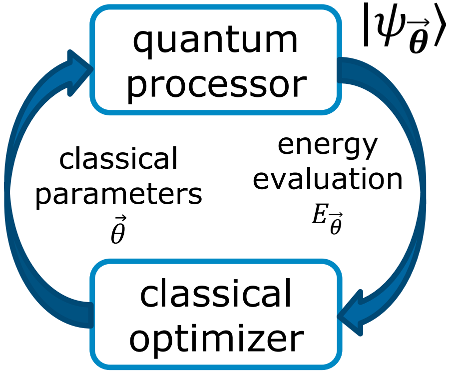
- infinite gauge degrees of freedom… in finite-dim quantum computer?
  - Use Gauss law to eliminate gauge d.o.f -> traded for exotic **long-range spin-spin interactions**

- **Final Hamiltonian:**

$$H_T = \sum_j \sigma_j^+ \sigma_{j+1}^- + h.c + \frac{m}{2} \sum_j (-)^j \sigma_j^z + \frac{g}{4} \sum_j \left(\sum_{l \leq j} \sigma_l^z + (-)^l \right)^2$$

with mass m, coupling g

Challenge II: Experimental realization



quantum processor

$|\psi_{\vec{\theta}}\rangle$

energy evaluation $E_{\vec{\theta}}$

classical optimizer

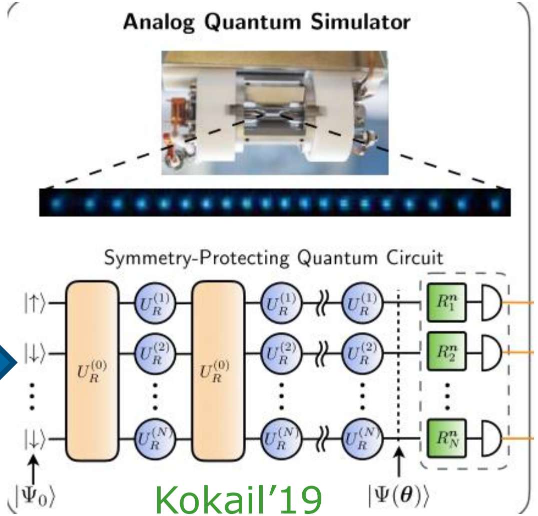
classical parameters $\vec{\theta}$

- Implementation of

$$H_T = \sum_j \sigma_j^+ \sigma_{j+1}^- + h.c + \frac{m}{2}\sum_j (-)^j \sigma_j^z + \frac{g}{4}\sum_j \left(\sum_{l\leq j} \sigma_l^z + (-)^l\right)^2$$

- Many recent publications:
  - digital computation with 4 ions (Martinez et al Nature ’16), 5 SC qubits (Klco et al PRA ’18), …
  - analog computation with 20 ions (Kokail et al Nature ‘19)

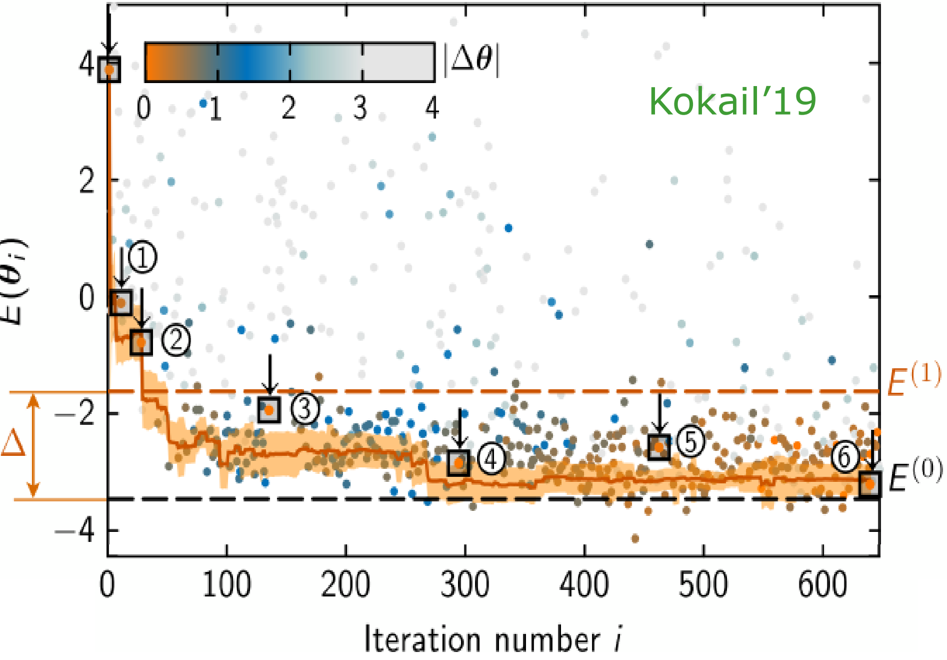
- **Here**: focus on analog computation
  - Preparation of $|\psi_{\vec{\theta}}\rangle$ with “resource” Hamiltonians H_R
  - Bonus: H_T and H_R share symmetries



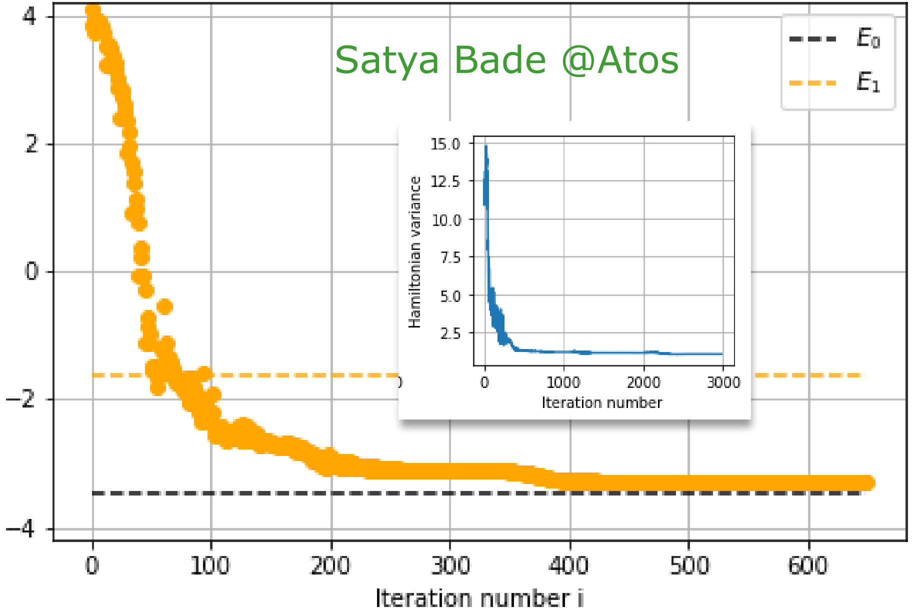
Kokail’19

Experimental results and simulation

- Experimental result (Nions=8):



- Our simulation (Nions=8):

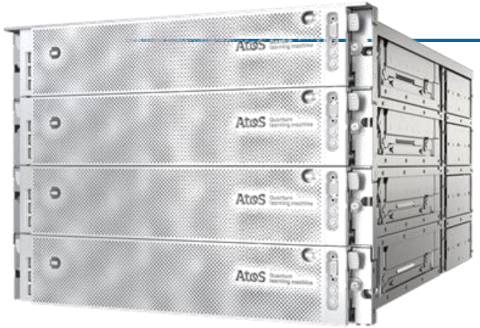


5

Conclusion: a dedicated platform for quantum simulation and computation

A platform to research and experiment quantum software

The Atos Quantum Learning Machine



QLM: a special-purpose classical simulator

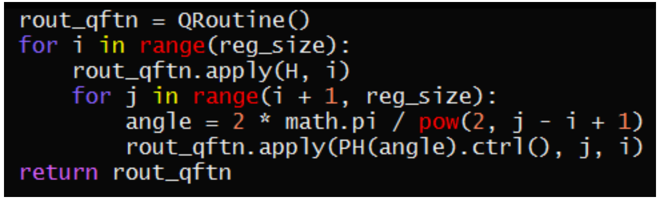
quantum programming libraries

quantum routines (QFT…), oracles, custom & abstract gates…

circuit optimization

connectivity constraints, graph-based optimization…

c++, python, jupyter, …



```
rout_qftn = QRoutine()
for i in range(reg_size):
    rout_qftn.apply(H, i)
    for j in range(i + 1, reg_size):
        angle = 2 * math.pi / pow(2, j - i + 1)
        rout_qftn.apply(PH(angle).ctrl(), j, i)
return rout_qftn
```

hybrid quantum-classical programming

perfect circuit simulation

at least 40 “logical” qubits

Various methods: full vector state, Matrix Product State, stabilizer, sum-over-histories…

noisy simulation

noise and hardware models

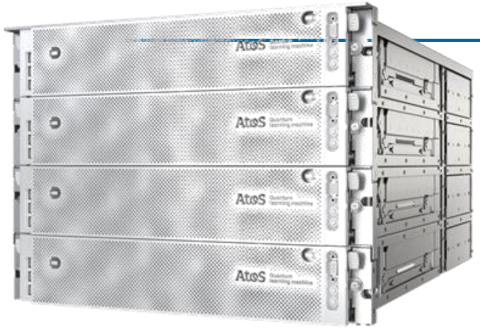
Various methods: full density matrix, Matrix Product Operator, quantum trajectories

application libraries

Spin/Fermionic models and translation to circuits
VQE + QAOA (optimizers…)

A platform to research and experiment quantum software

The Atos Quantum Learning Machine



QLM: a special-purpose
classical simulator

quantum programming libraries

quantum routines (QFT…), oracles, custom
& abstract gates…

circuit optimization

connectivity constraints, graph-based optimization…

hybrid quantum-classical programming

perfect circuit simulation

at least 40 “logical” qubits

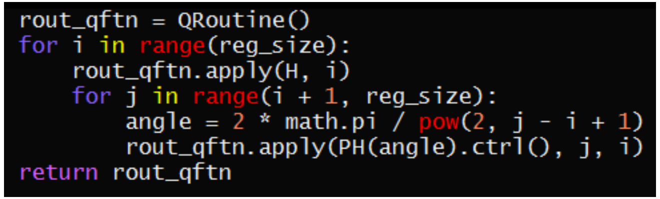
Various methods: full vector state, Matrix Product State, stabilizer, sum-over-histories…

noisy simulation

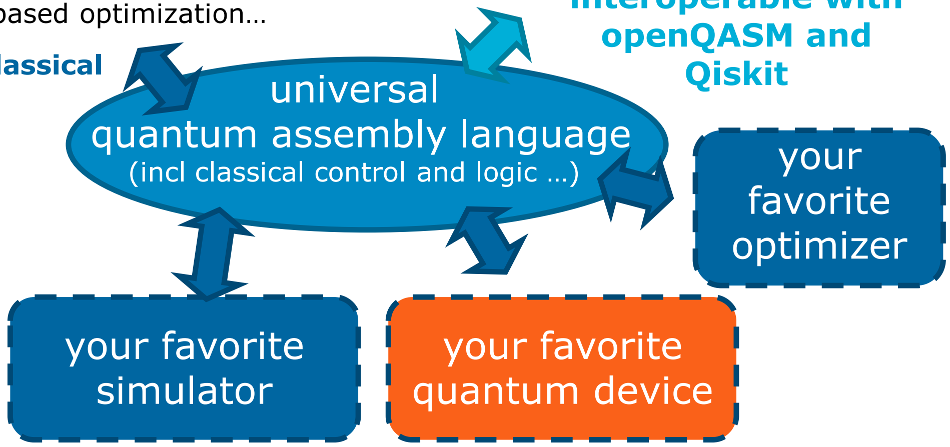
noise and hardware models

Various methods: full density matrix, Matrix Product Operator, quantum trajectories

c++, python, jupyter, …



```
rout_qftn = QRoutine()
for i in range(reg_size):
    rout_qftn.apply(H, i)
    for j in range(i + 1, reg_size):
        angle = 2 * math.pi / pow(2, j - i + 1)
        rout_qftn.apply(PH(angle).ctrl(), j, i)
return rout_qftn
```



interoperable with openQASM and Qiskit

universal
quantum assembly language
(incl classical control and logic …)

your favorite optimizer

your favorite simulator

your favorite quantum device

Atos QLM Customers

OAK RIDGE
National Laboratory

DTU
Technical University
of Denmark

JÜLICH
FORSCHUNGSZENTRUM

SURF SARA

ccrt
Centre de calcul recherche et technologie

UNIVERSITÉ
DE REIMS
CHAMPAGNE-ARDENNE

TOTAL

Argonne
NATIONAL LABORATORY

Hartree Centre
Science & Technology Facilities Council

fh
OBERÖSTERREICH
UNIVERSITY
OF APPLIED SCIENCES
UPPER AUSTRIA

Thanks

thomas.ayral@atos.net