Kick-off
&
Introduction to Quantum Computing

Quantum Computing for Software Engineers

12.06. - 22.06.2023

Plan for today

- Some organizational remarks
- Talk: Introduction to quantum computing
- Please ask questions!

Outline of the the live sessions

- 12.06., 10:00 - 11:30:
  - Organization; Introduction to Quantum Computing
- 15.06., 16:00 - 17:00
  - Discussion of week 1, tutorial and questions
- 19.06., 10:00 - 11:30
  - Deep Dive: Grover's Algorithm
- 22.06., 16:00 - 17:00
  - Discussion of week 2, tutorial, questions, and closing

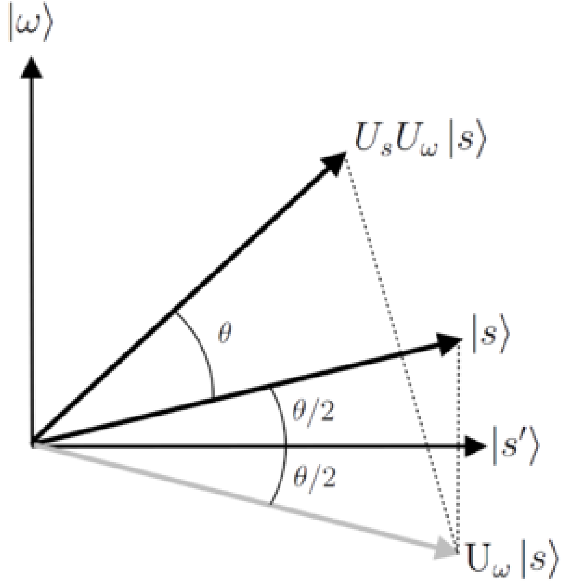
Tutorials

- We will use pennylane as the quantum computing software framework for the exercises in this workshop.
- For more information, please visit the pennylane documentation:
  - https://docs.pennylane.ai/en/stable/
- The tutorials are provided in Jupyter Notebooks.
- Access is provided via a git repository:
  - https://github.com/dfki-ric-quantum/q3up_workshop_june2023

PENNYLANE

Content of tutorials

- Week 1:
  - Introduction to Quantum Circuits and Algorithms
  - Goal: Solving Simon's Problem
- Week 2:
  - Grover's Algorithm
  - Goal: Using Grover's Algorithm to Solve Sudoku



Your team for this workshop

Hans Hohenfeld
Tutorial 1. Week &
Introduction 2. Week

Dirk Heimann
Tutorial 2. Week

Felix Wiebe
Tutorial 2. Week

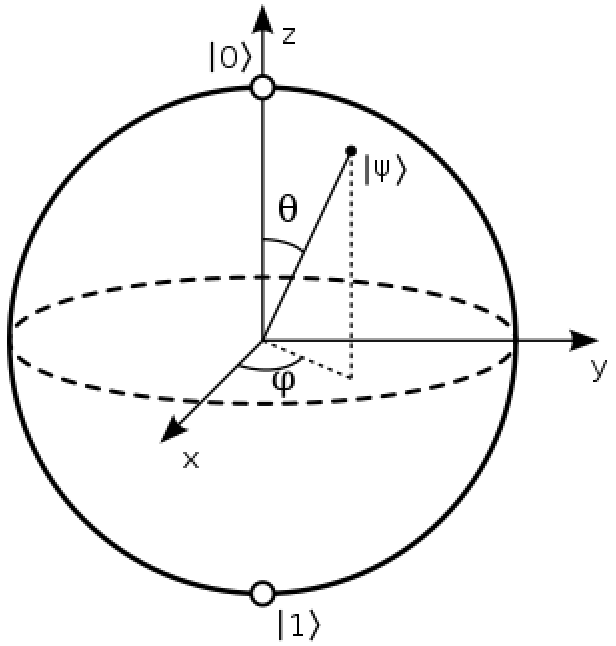
Gunnar Schönhoff
Introduction 1. Week &
Organization

Some further ressources

- Quantum Computation and Quantum Information (10th Anniversary edition) M. Nielsen, I. Chuang
- Introduction to Quantum Information Science. Lecture Notes. Scott Aaronson. https://www.scottaaronson.com/qclec.pdf
- pennylane.ai/qml/
- qiskit.org/textbook

Introduction to Quantum Computing

- Comparison to classical computing
- Qubits & a tiny bit of quantum physics
- Quantum Hardware
- Quantum Gates
- Quantum Software & Algorithms
- Applications
- Outlook



Verizon
1:10 AM
Tuesday
11
Calendar
Photos
Camera
Videos
280
Maps
Weather
Passbook
Notes
Reminders
Clock
Stocks
Newsstand
iTunes Store
App Store
Game C
FaceTime
Music
Phone
Mail
Safari
Messages

What is classical computing?

Processing of electrical signals in the hardware

Programming of this signal processing on different software levels

Fundamental unit: bit

What is quantum computing?

What is quantum computing?

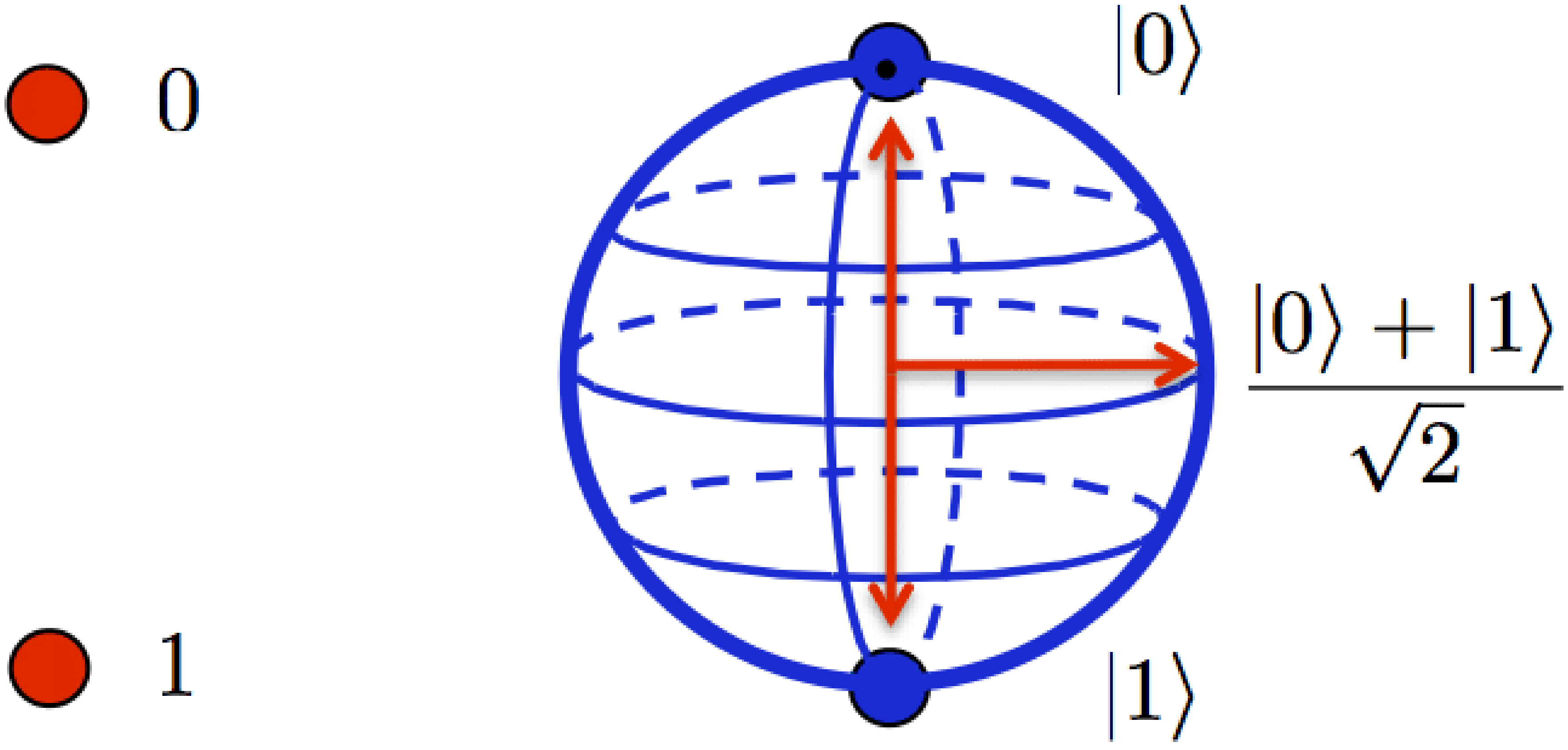
"I like to call it the study of what we can't do even with computers we don't have."

Scott Aaronson, UT Austin

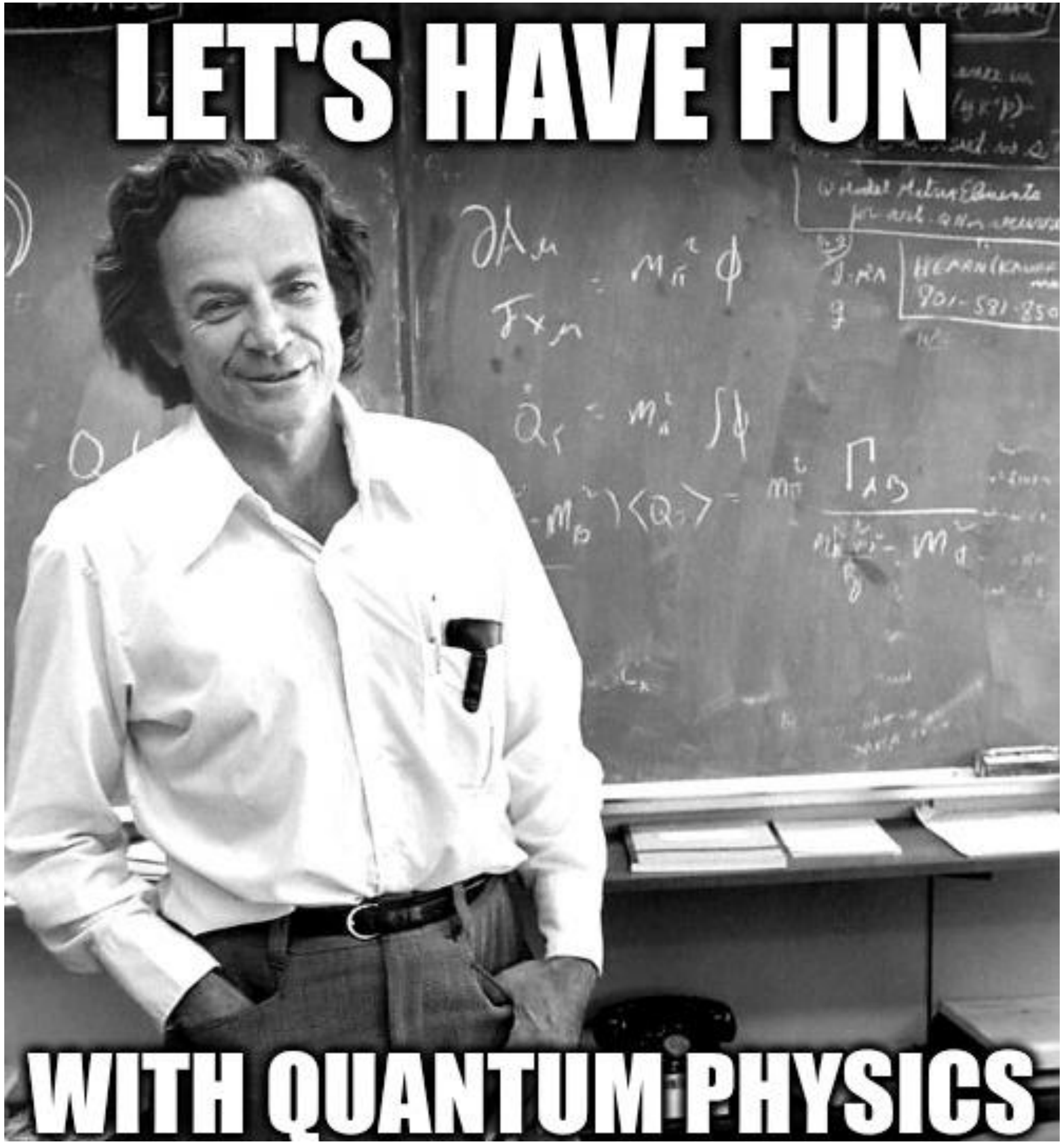
IBMs Q System One (Archivbild) Foto: --- / dpa

What is a quantum computer?

- A quantum computer uses quantum physical properties to **perform calculations** through the **manipulation of quantum states**.
- Quantum computers work with **qubits**.
- Qubits are manipulated through **operators** (theory) or **pulse sequences** (experiment).



Classical Bit **Qubit**


LET'S HAVE FUN
WITH QUANTUM PHYSICS

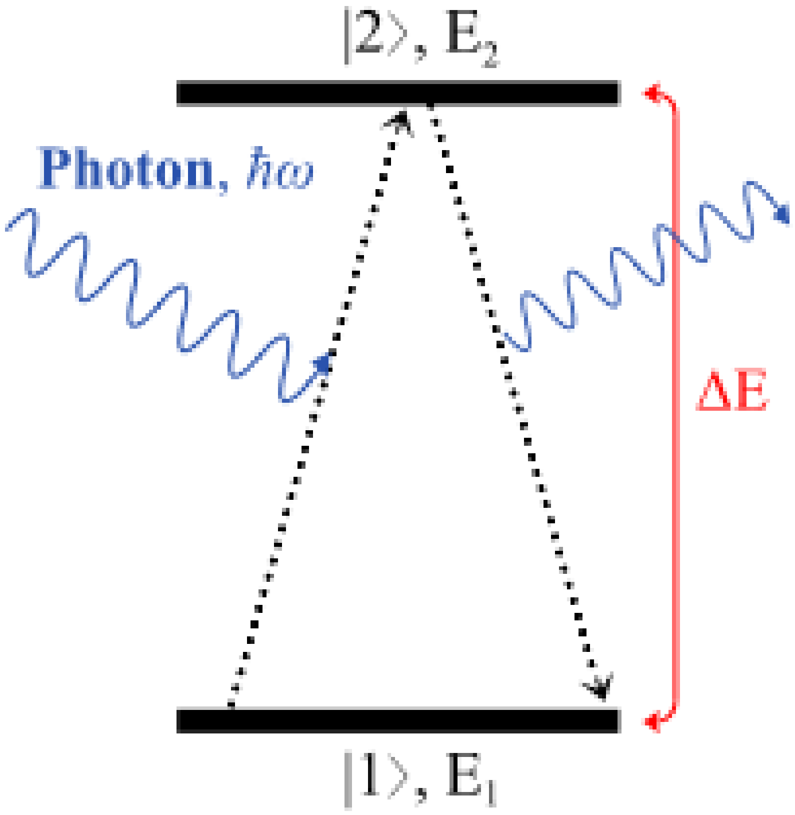
Quantum Physics: Unitaries acting on vectors in Hilbert space

$$\boldsymbol{H}|\psi\rangle = \boldsymbol{E}|\psi\rangle$$

$$\begin{pmatrix} H_{11} & H_{12} \\ H_{12}^{*} & H_{22} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = E \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

Single Qubit: Two-level system

$$|\psi\rangle = c_1|1\rangle + c_2|2\rangle$$



Properties of the wave function

$$|\psi\rangle = \sum_i \boldsymbol{c_i} \left|\boldsymbol{k_i}\right\rangle$$

$$\boldsymbol{\langle\psi|\psi\rangle = 1,} \quad \sum_i \left|c_i\right|^2 = 1.$$

Quantum Computers make use of

Superposition

Entanglement

Collapse of the wave function

Superposition

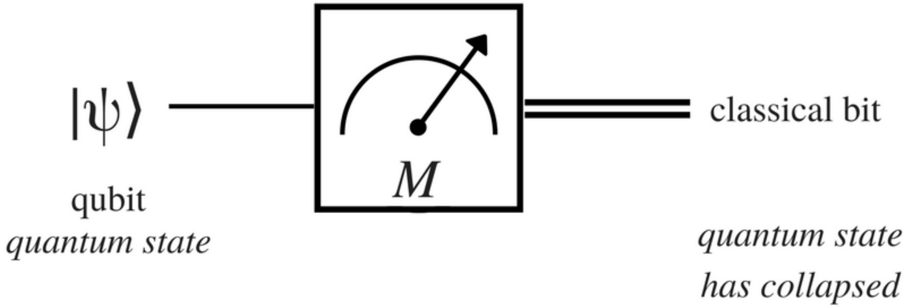
- Quantum mechanical systems can be in a superposition of states.
- This applies for single-particle/single-qubit systems as well as for systems of multiple particles/qubits.
- When measured, the wave function collapses and one only sees a single state.

$$|\psi\rangle = c_1|1\rangle + c_2|2\rangle$$

Classical bits			Qubits	
2xn			2^n	
(2x2)	*4*	n = 2	(2^2)	*4*
(2x3)	*6*	n = 3	(2^3)	*8*
(2x4)	*8*	n = 4	(2^4)	*16*
(2x5)	*10*	n = 5	(2^5)	*32*
...	...		...	...
(2x100)	*200*	n = 100	(2^{100})	*1267650600228229401496 703205376*

Measurement & Sampling

- To get an output in the form of classical bits, the state of a qubit has to be measured.
- This results in the collapse of the wave function.
  - **No access to the full wave function**!
- In many cases, one has to sample several (a lot of!) times from a quantum circuit to get a meaningful expectation value.



Entanglement

- Individual parts of a system cannot be treated separately, i.e., they are not separable and cannot be described as product states.
- Entangled subsystems are **non-classically correlated with each other**.
- Reason: quantum-mechanical interaction.

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle_A \otimes |1\rangle_B - |1\rangle_A \otimes |0\rangle_B\right)$$

In case you want to know more...

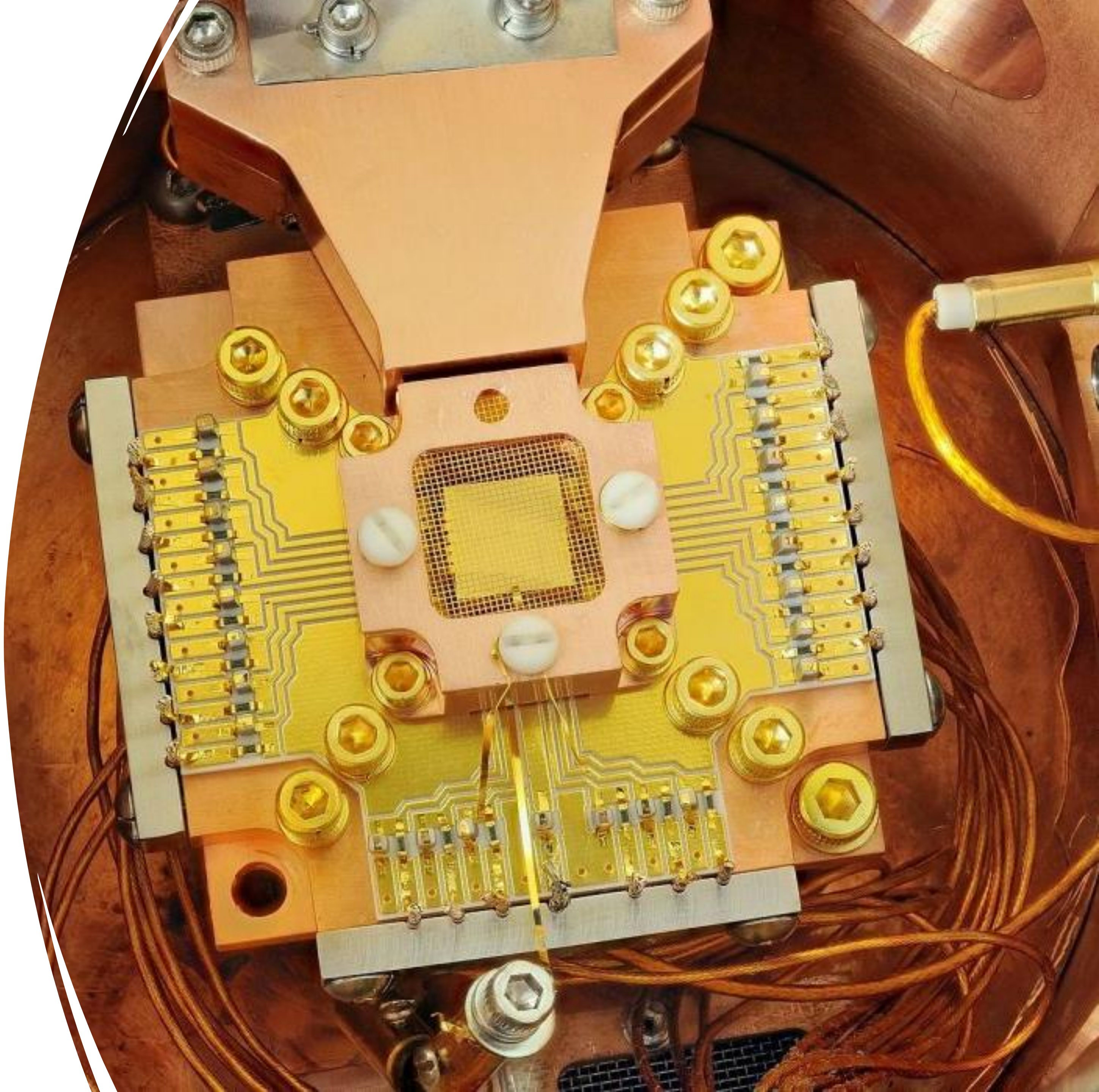
"Quantum mechanics essentials: Everything you need for quantum computation", Tobias Osborne (https://www.youtube.com/watch?v=28ABEInFxBQ)

(and a lot of other sources)

--> **Quantum physics as a probabilistic theory**

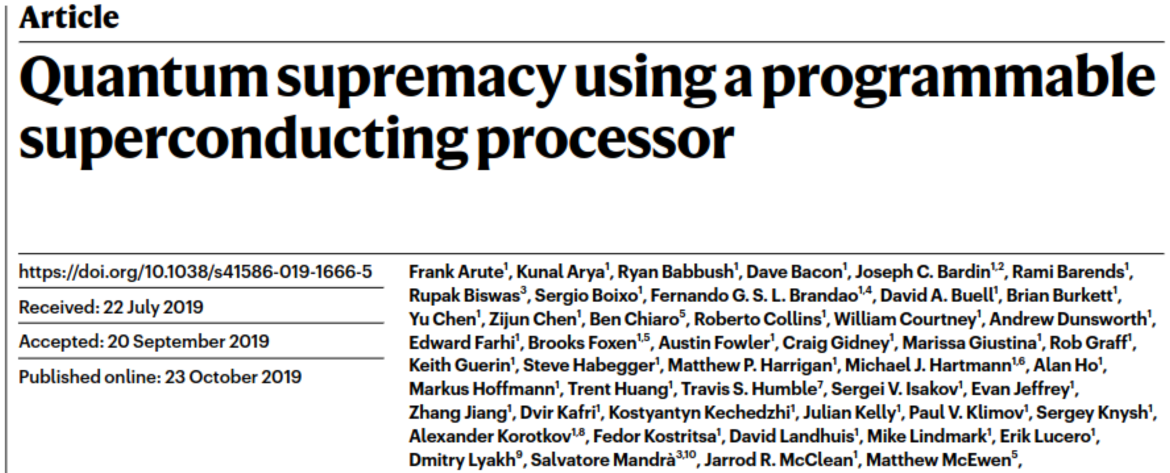
Overview of QC Hardware implementations

- Some major ones, especially:
  - Superconducting
  - Ion traps
  - Photonic
- More experimental ones (NV-centers, electron spins, topological, …)



Superconducting Qubits

- Developed by Google, IBM among others
- Use of Cooper pairs/Josephson junctions
- Cooled down to near absolute zero
- Manipulation of qubits through microwave pulses



Article

Quantum supremacy using a programmable superconducting processor

https://doi.org/10.1038/s41586-019-1666-5

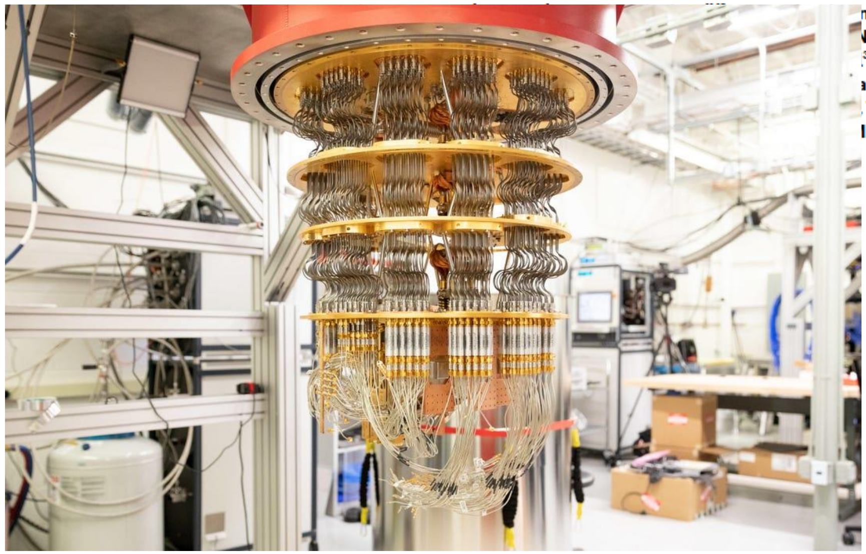
Received: 22 July 2019

Accepted: 20 September 2019

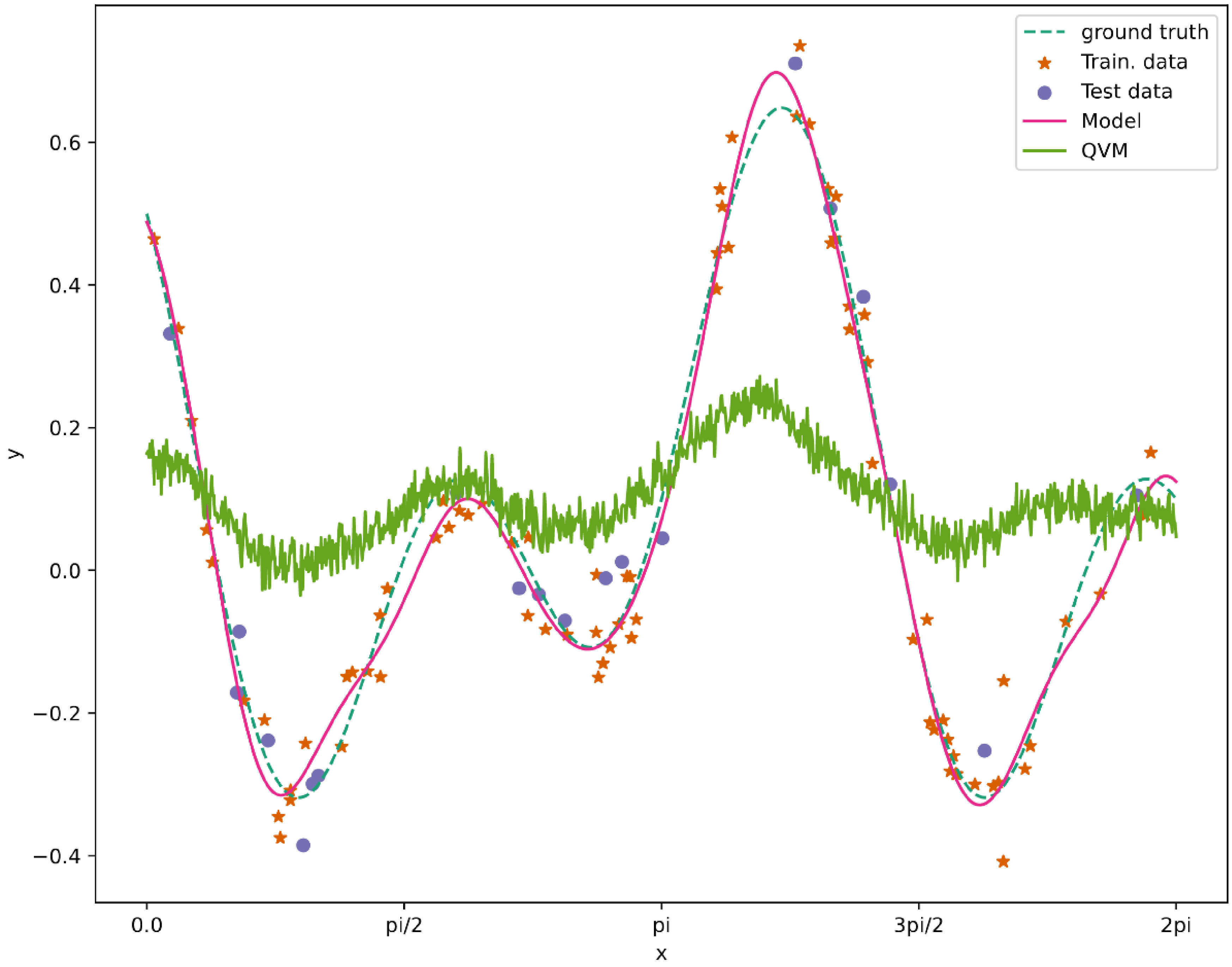
Published online: 23 October 2019

Frank Arute[1], Kunal Arya[1], Ryan Babbush[1], Dave Bacon[1], Joseph C. Bardin[1,2], Rami Barends[1], Rupak Biswas[3], Sergio Boixo[1], Fernando G. S. L. Brandao[1,4], David A. Buell[1], Brian Burkett[1], Yu Chen[1], Zijun Chen[1], Ben Chiaro[5], Roberto Collins[1], William Courtney[1], Andrew Dunsworth[1], Edward Farhi[1], Brooks Foxen[1,5], Austin Fowler[1], Craig Gidney[1], Marissa Giustina[1], Rob Graff[1], Keith Guerin[1], Steve Habegger[1], Matthew P. Harrigan[1], Michael J. Hartmann[1,6], Alan Ho[1], Markus Hoffmann[1], Trent Huang[1], Travis S. Humble[7], Sergei V. Isakov[1], Evan Jeffrey[1], Zhang Jiang[1], Dvir Kafri[1], Kostyantyn Kechedzhi[1], Julian Kelly[1], Paul V. Klimov[1], Sergey Knysh[1], Alexander Korotkov[1,8], Fedor Kostritsa[1], David Landhuis[1], Mike Lindmark[1], Erik Lucero[1], Dmitry Lyakh[9], Salvatore Mandrà[3,10], Jarrod R. McClean[1], Matthew McEwen[5],

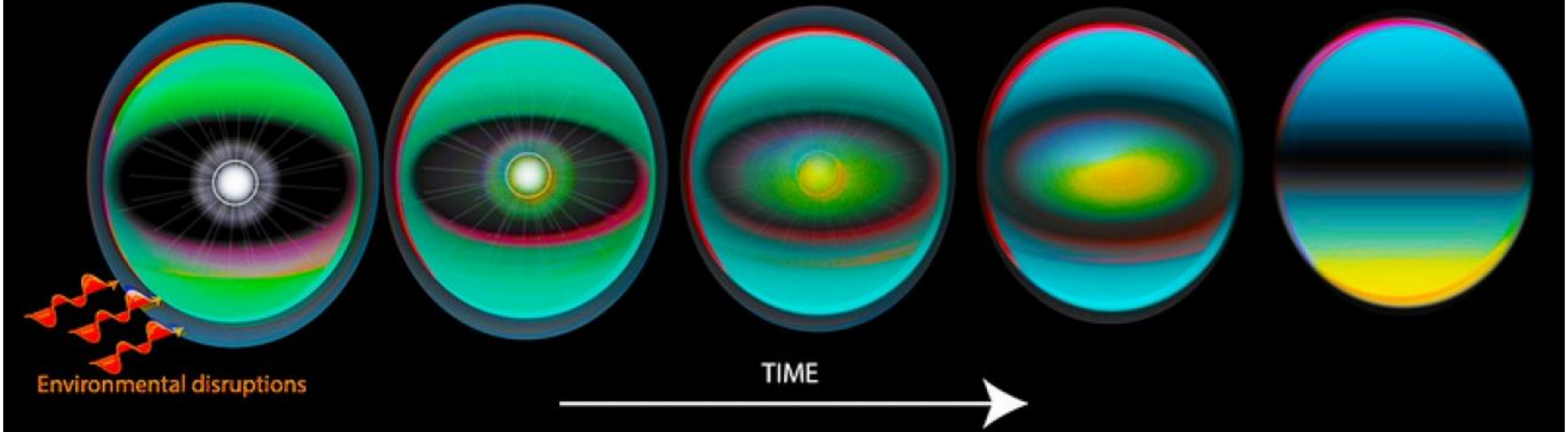
ni[1], Josh Mutus[1],
Niu[1], Eric Ostby[1],
[3], Pedram Roushan[1],
anskiy[1], Kevin J. Sung[1,13],
Theodore White[1],
I. Martinis[1,5*]

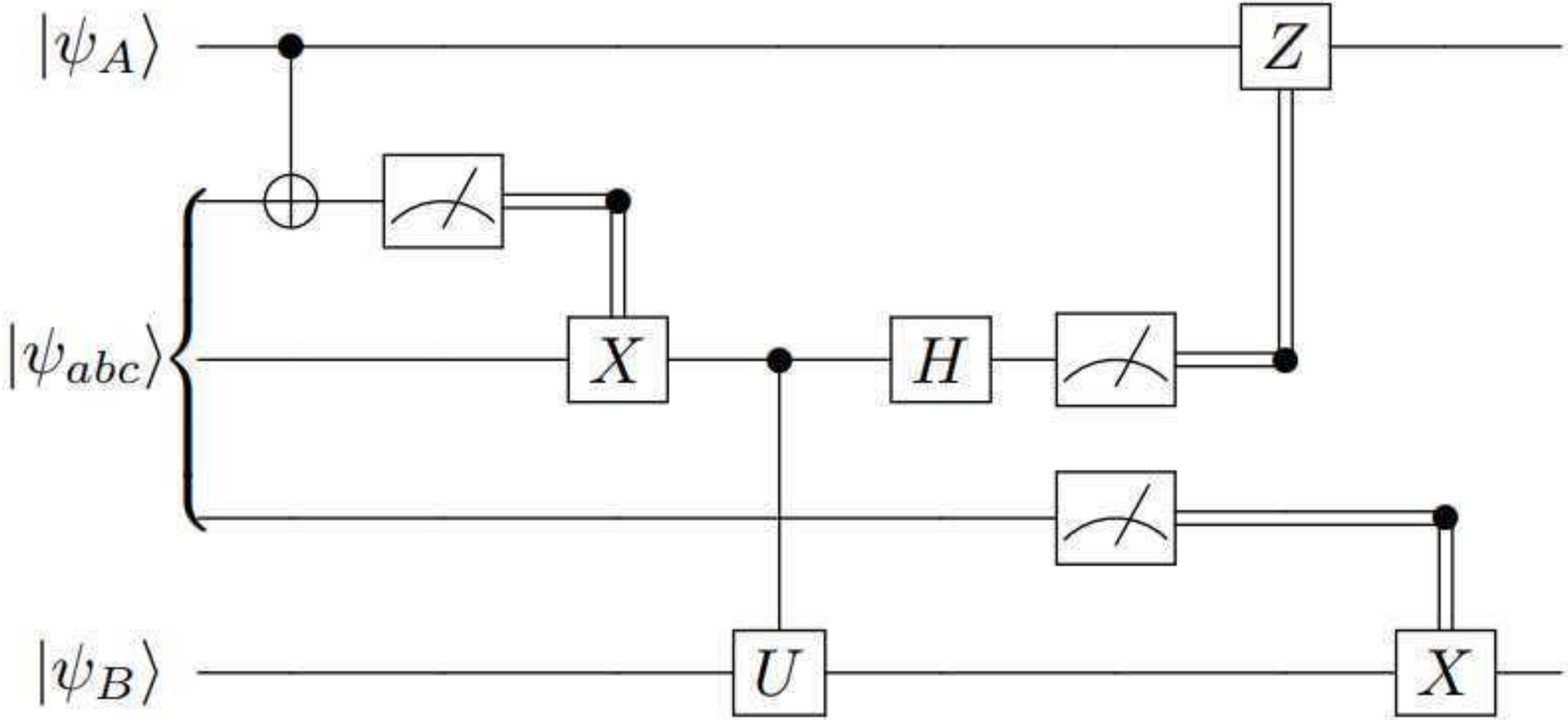


Why do I need to know about different hardware implementations?


ground truth
Train. data
Test data
Model
QVM
y
0.6
0.4
0.2
0.0
−0.2
−0.4
x
0.0
pi/2
pi
3pi/2
2pi

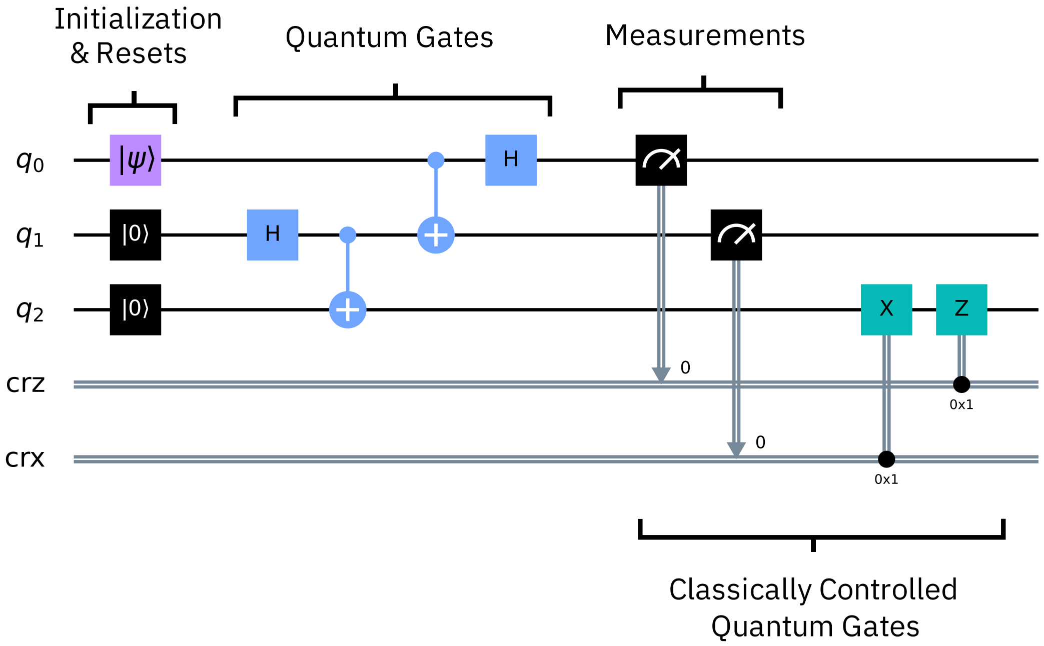
Coherence time: How long can a quantum state live?




$|\psi_A\rangle$
$|\psi_{abc}\rangle$
$|\psi_B\rangle$
X
U
H
Z
X

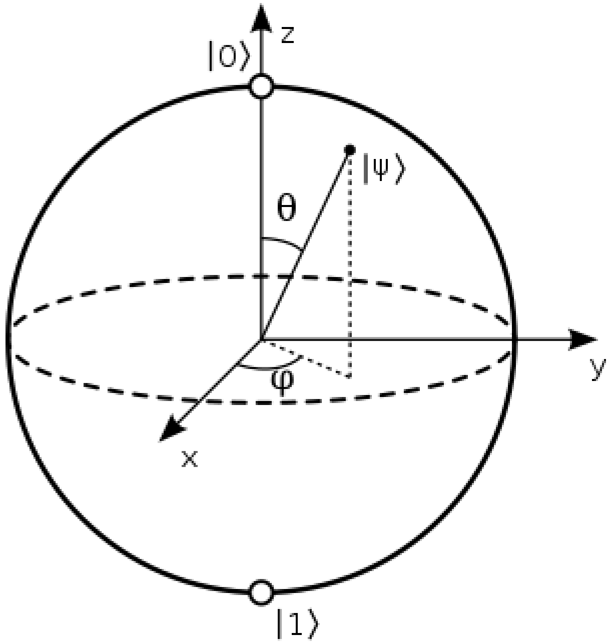
Gate/Circuit model of Quantum Computing

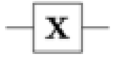
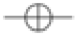
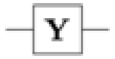
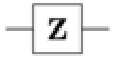
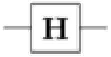
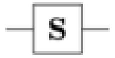
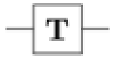
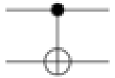
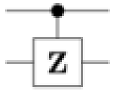
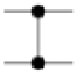
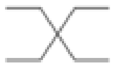
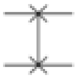
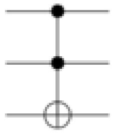
- **De-facto standard** to describe quantum computing operations and algorithms
- The gate model is universal, i.e., any computation possible with a quantum computer can be described by it
- **Initialization** of qubits followed by **manipulations** and **measurements**
- Any quantum program can be represented by a sequence of quantum circuits and non-concurrent classical computation
- Other models: measurement-based QC (universal)



Quantum gates

- **Reversible unitary transformations** on one or more qubits
- I.e., every gate corresponds to a unitary matrix ("unitaries")
- Different hardware implementations have **different native gate sets**
- A gate set is **universal** if any unitary transformation can be efficiently approximated arbitrarily well as a sequence of gates in the set



Operator	Gate(s)		Matrix
Pauli-X (X)			$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)			$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)			$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)			$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)			$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
π/8 (T)			$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)			$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Example: the Hadamard Gate $-\boxed{H}-$

- The Hadamard gate is defined as follows:

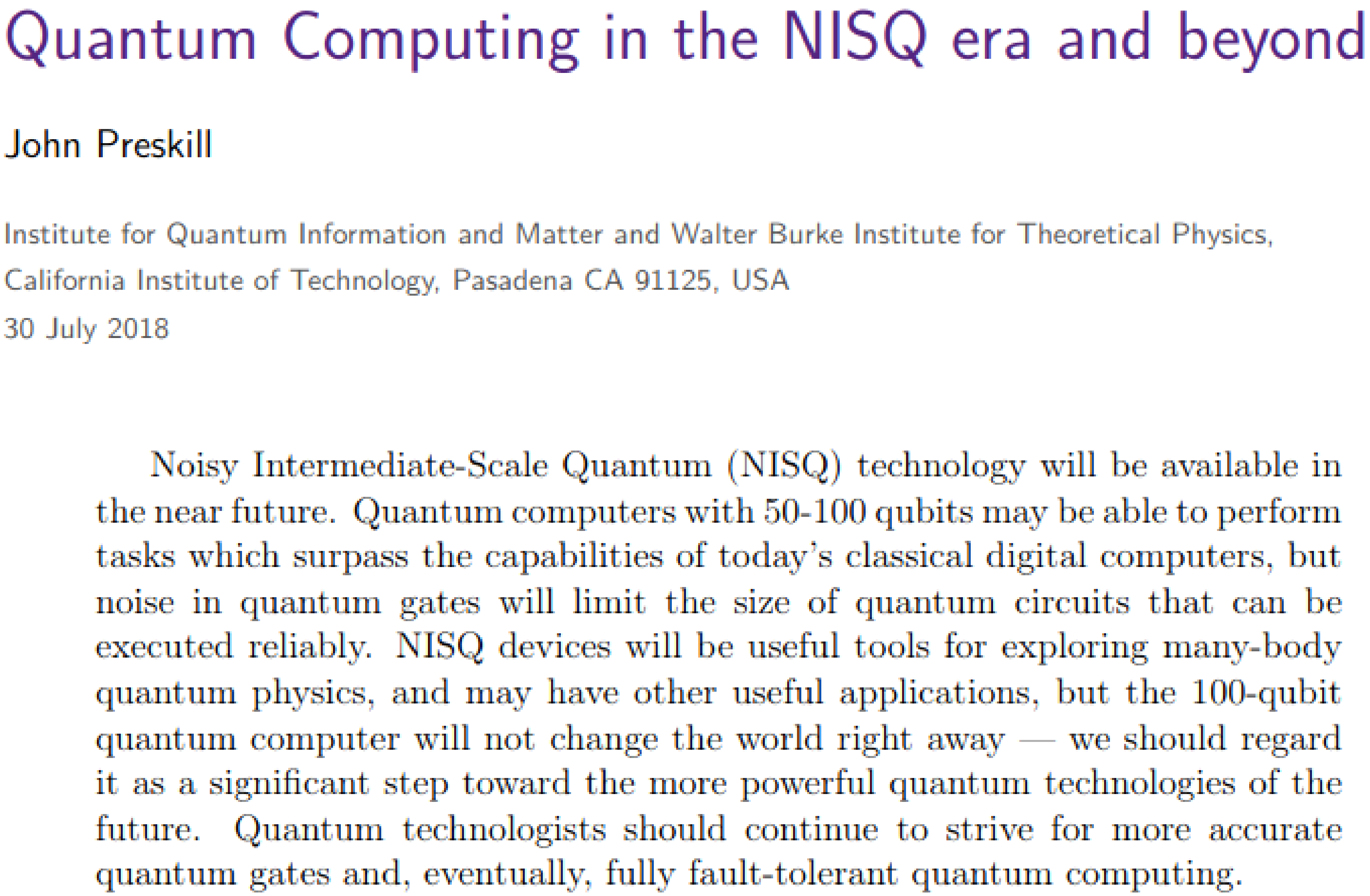
$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- It puts basis states into superposition

$$H(|0\rangle) = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle =: |+\rangle$$
$$H(|1\rangle) = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle =: |-\rangle$$

- The inverse of the Hadamard gate is again the Hadamard gate:

$$H(|+\rangle) = H\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{2}\Big(|0\rangle + |1\rangle\Big) + \frac{1}{2}\Big(|0\rangle - |1\rangle\Big) = |0\rangle$$
$$H(|-\rangle) = H\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{2}\Big(|0\rangle + |1\rangle\Big) - \frac{1}{2}\Big(|0\rangle - |1\rangle\Big) = |1\rangle$$



Quantum Computing in the NISQ era and beyond

John Preskill

Institute for Quantum Information and Matter and Walter Burke Institute for Theoretical Physics,
California Institute of Technology, Pasadena CA 91125, USA

30 July 2018

Noisy Intermediate-Scale Quantum (NISQ) technology will be available in the near future. Quantum computers with 50-100 qubits may be able to perform tasks which surpass the capabilities of today's classical digital computers, but noise in quantum gates will limit the size of quantum circuits that can be executed reliably. NISQ devices will be useful tools for exploring many-body quantum physics, and may have other useful applications, but the 100-qubit quantum computer will not change the world right away — we should regard it as a significant step toward the more powerful quantum technologies of the future. Quantum technologists should continue to strive for more accurate quantum gates and, eventually, fully fault-tolerant quantum computing.

Quantum algorithms

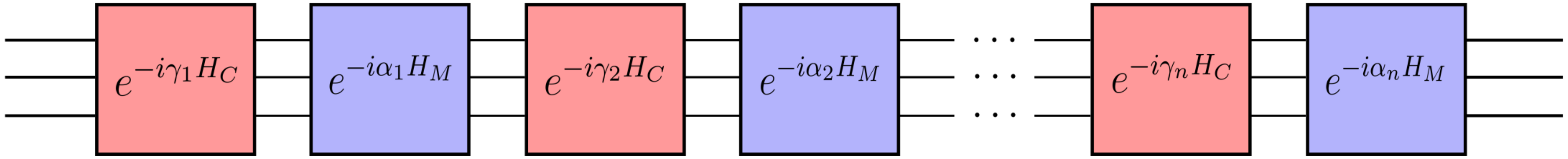
- Similar to classical algorithms: step-by-step procedure to be executed on a quantum computer
- Two flavours
    - Long established fault-tolerant algorithms
        - Some have theoretically proven speedups
        - Only really work once we have full-scale, fault-tolerant quantum computers
    - "NISQY" algorithms
        - Work on current quantum hardware
        - Speedups are unclear

Some famous quantum algorithms

- **Deutsch-Jozsa** algorithm (1992)
    - Determines whether an oracle function is balanced or constant
    - First algorithm to show "quantum advantage"
- **Shor**'s algorithm (1994)
    - for prime factorization
    - Based on quantum Fourier transform
    - Exponential speedup over best known classical algorithm
- **Grover**'s algorithm (1996)
    - for searching an unstructured database
    - Based on amplitude amplification
    - Quadratic speedup over best known classical algorithm; proven to be optimal
- **HHL** algorithm (2008)
    - for inversion of sparse quadratic matrices
    - Exponential speedup proven under certain assumptions

"NISQY" algorithms

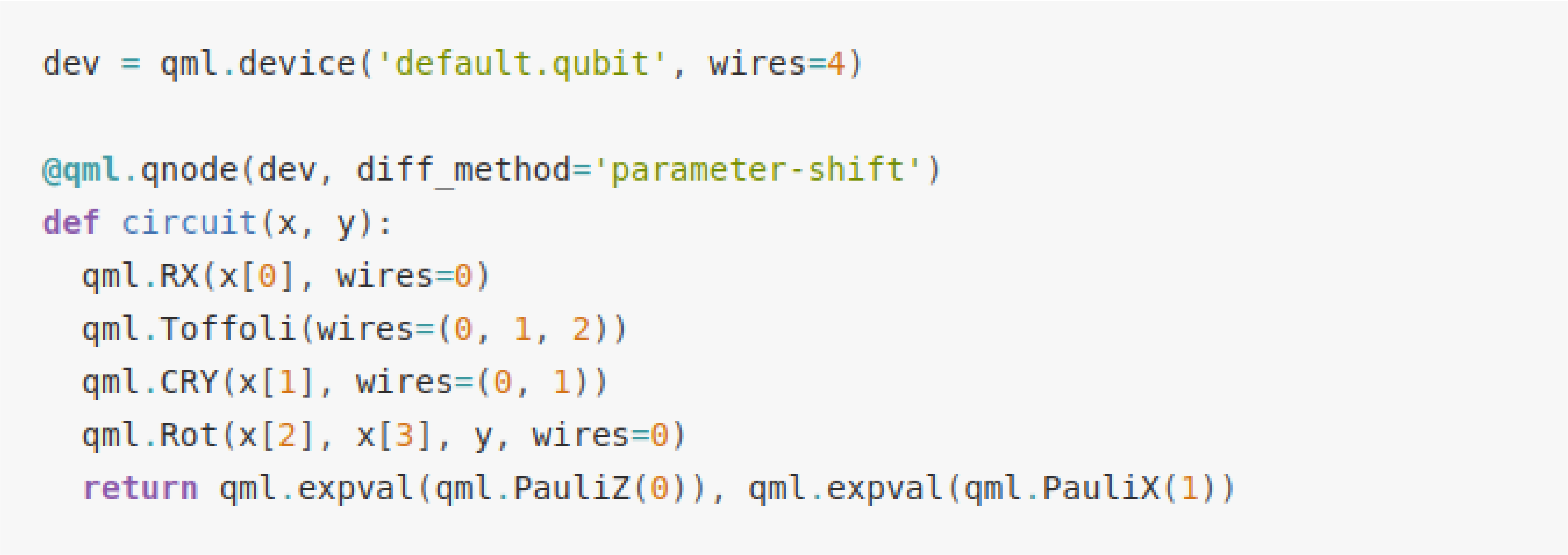
- Variational quantum algorithms (VQAs)
  - Hybrid quantum-classical algorithms
  - Classical part: parameter optimization
  - Quantum part: execution of parametrized quantum circuits (PQCs)
  - Example: Quantum Approximate Optimization Algorithm (QAOA)
- Other approaches, including quantum annealing



Current quantum software

- Different software frameworks to program quantum algorithms exist
- Mostly: programming on gate level
- Integrated with existing classical software (e.g., TensorFlow, Pytorch)
- Integration with simulated or real quantum backends

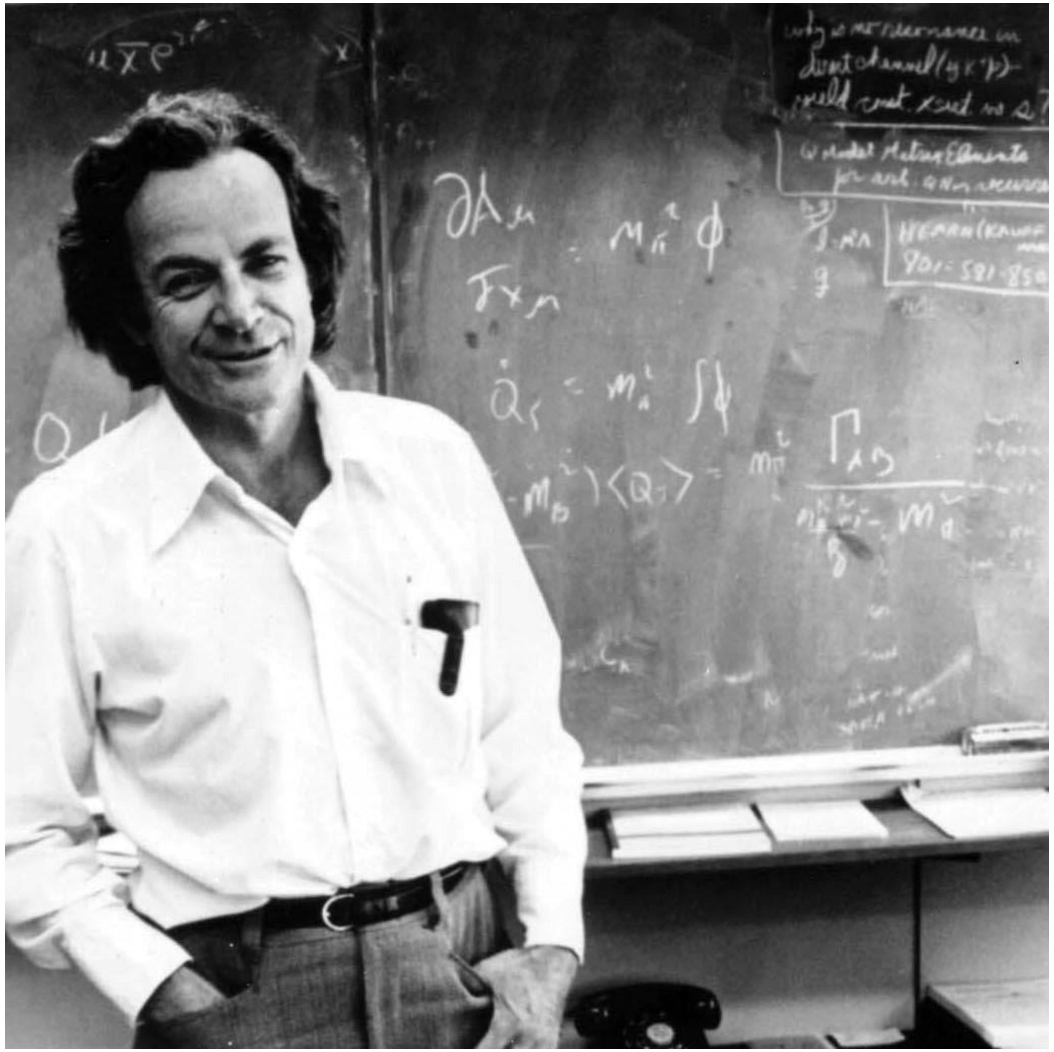
Qiskit PENNYLANE TensorFlow Quantum



```python
dev = qml.device('default.qubit', wires=4)

@qml.qnode(dev, diff_method='parameter-shift')
def circuit(x, y):
  qml.RX(x[0], wires=0)
  qml.Toffoli(wires=(0, 1, 2))
  qml.CRY(x[1], wires=(0, 1))
  qml.Rot(x[2], x[3], y, wires=0)
  return qml.expval(qml.PauliZ(0)), qml.expval(qml.PauliX(1))
```

Is this any useful?



"Now I explicitly go to the question of how we can simulate with a computer … the quantum mechanical effects... But the full description of quantum mechanics for a large system with R particles is given by a function which we call the amplitude to find the particles at x1, x2, . . . xR, and therefore because it has too many variables, it cannot be simulated with a normal computer."

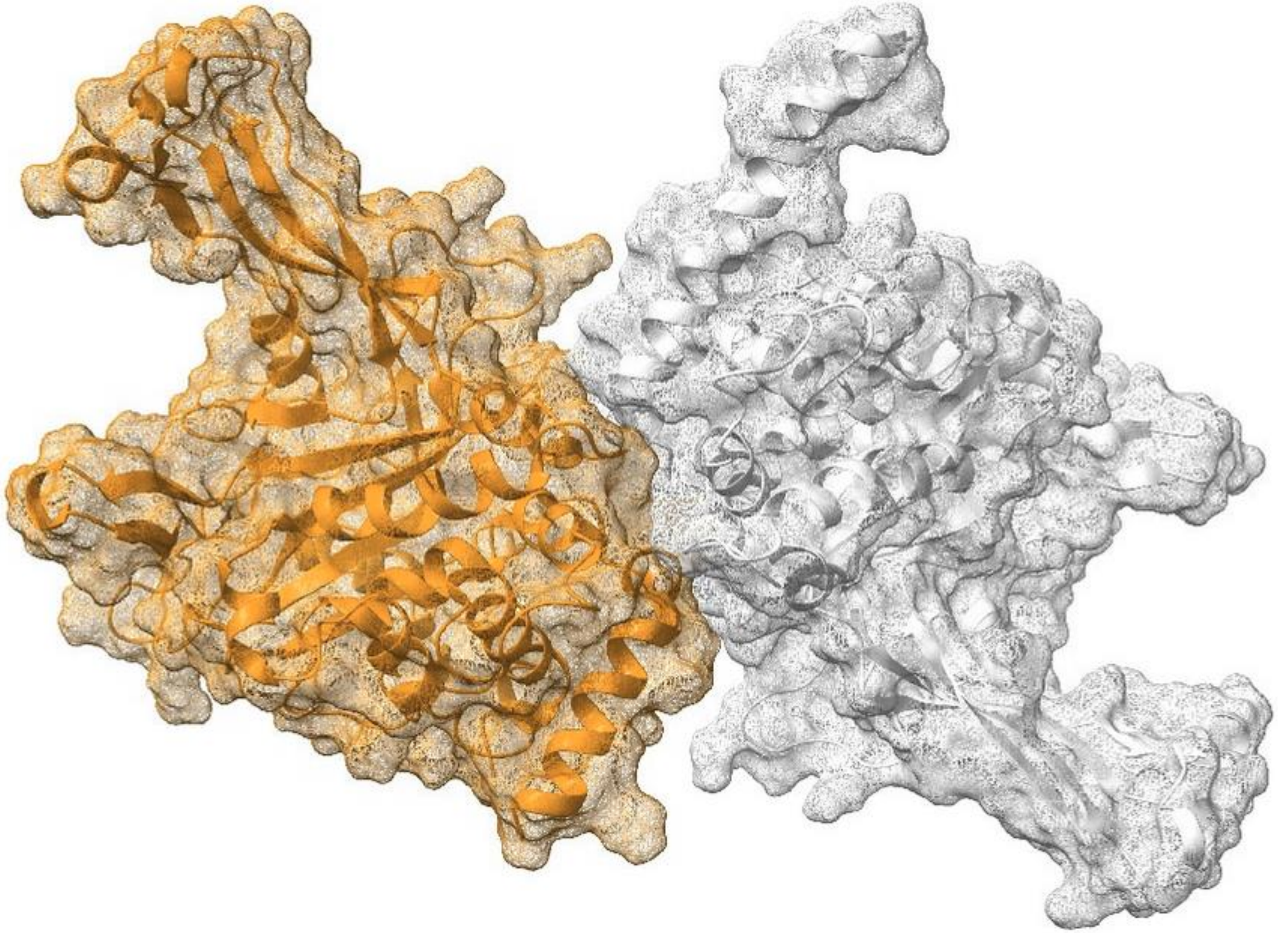
"Can you do it with a new kind of computer — a quantum computer? Now it turns out, as far as I can tell, that you can simulate this with a quantum system, with quantum computer elements."

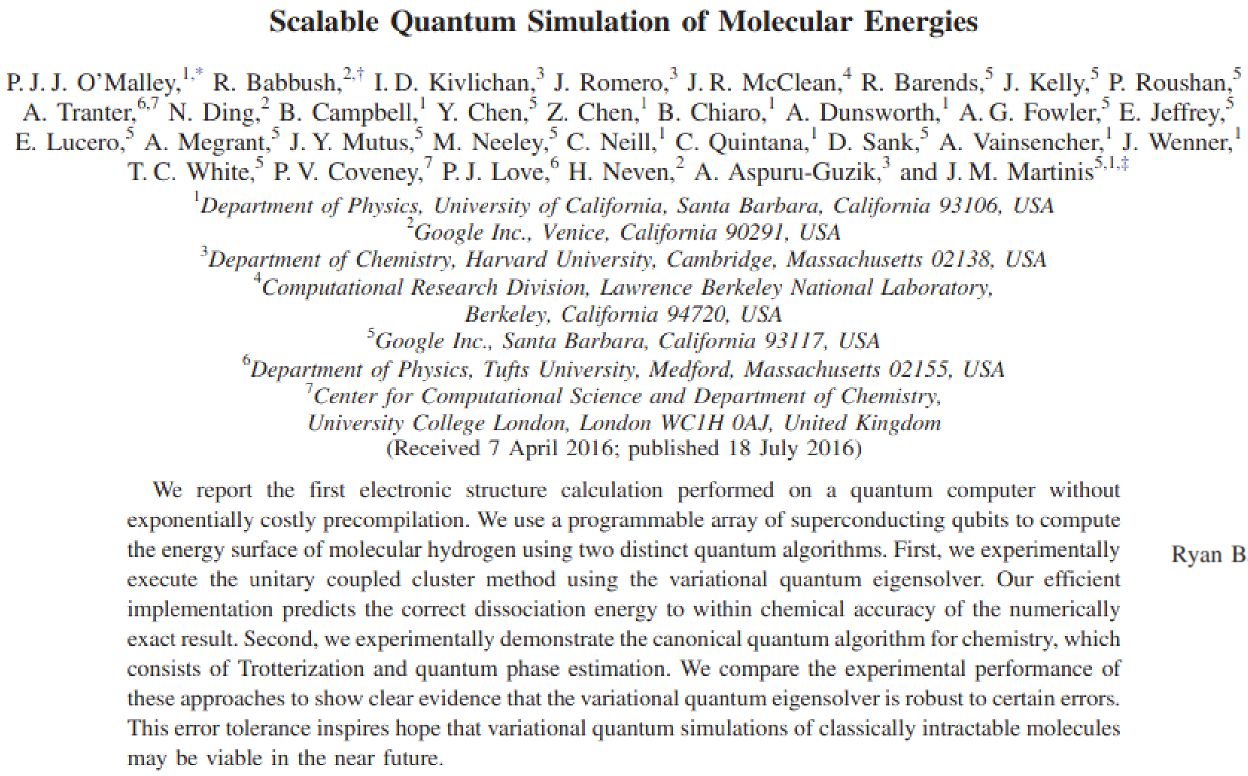
(Richard Feynman, 1981)

Qubit Pharmaceuticals Accelerates Drug Discovery With Hybrid Quantum Computing

Startup adopts NVIDIA QODA to reduce the time and investment needed to identify promising treatments for incurable diseases.

November 30, 2022 by CRAIG RHODES





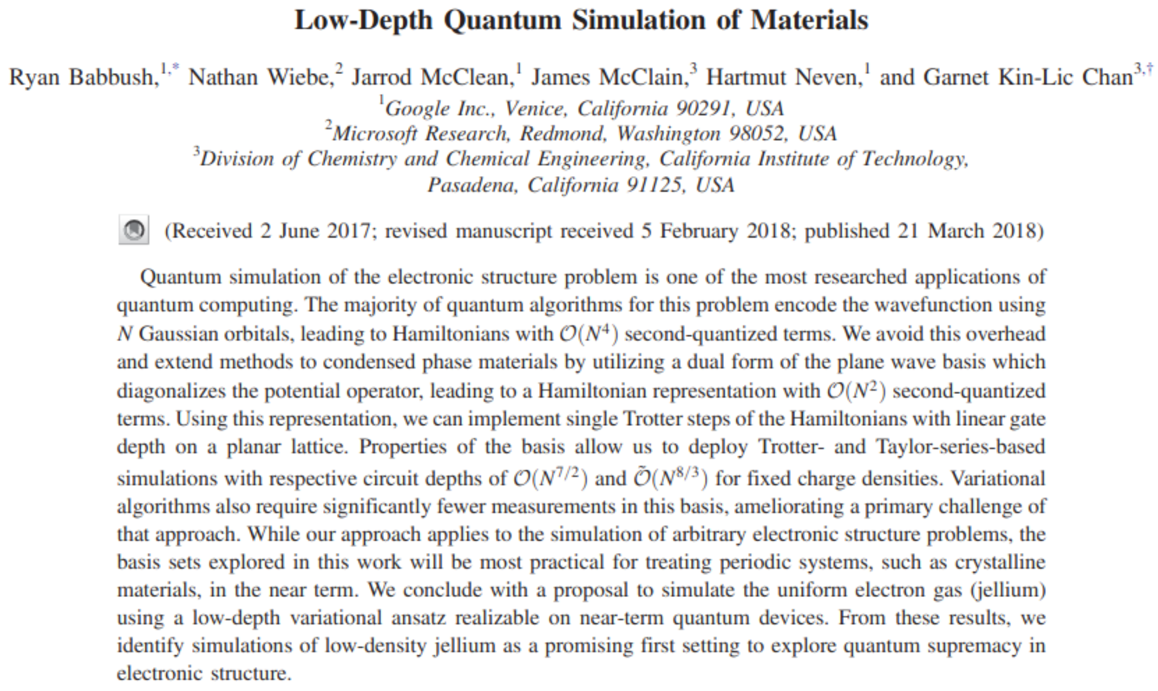
Scalable Quantum Simulation of Molecular Energies

P. J. J. O'Malley,[1,*] R. Babbush,[2,†] I. D. Kivlichan,[3] J. Romero,[3] J. R. McClean,[4] R. Barends,[5] J. Kelly,[5] P. Roushan,[5]
A. Tranter,[6,7] N. Ding,[2] B. Campbell,[1] Y. Chen,[5] Z. Chen,[1] B. Chiaro,[1] A. Dunsworth,[1] A. G. Fowler,[5] E. Jeffrey,[5]
E. Lucero,[5] A. Megrant,[5] J. Y. Mutus,[5] M. Neeley,[5] C. Neill,[1] C. Quintana,[1] D. Sank,[5] A. Vainsencher,[1] J. Wenner,[1]
T. C. White,[5] P. V. Coveney,[7] P. J. Love,[6] H. Neven,[2] A. Aspuru-Guzik,[3] and J. M. Martinis[5,1,‡]

[1]*Department of Physics, University of California, Santa Barbara, California 93106, USA*
[2]*Google Inc., Venice, California 90291, USA*
[3]*Department of Chemistry, Harvard University, Cambridge, Massachusetts 02138, USA*
[4]*Computational Research Division, Lawrence Berkeley National Laboratory,*
Berkeley, California 94720, USA
[5]*Google Inc., Santa Barbara, California 93117, USA*
[6]*Department of Physics, Tufts University, Medford, Massachusetts 02155, USA*
[7]*Center for Computational Science and Department of Chemistry,*
University College London, London WC1H 0AJ, United Kingdom

(Received 7 April 2016; published 18 July 2016)

We report the first electronic structure calculation performed on a quantum computer without exponentially costly precompilation. We use a programmable array of superconducting qubits to compute the energy surface of molecular hydrogen using two distinct quantum algorithms. First, we experimentally execute the unitary coupled cluster method using the variational quantum eigensolver. Our efficient implementation predicts the correct dissociation energy to within chemical accuracy of the numerically exact result. Second, we experimentally demonstrate the canonical quantum algorithm for chemistry, which consists of Trotterization and quantum phase estimation. We compare the experimental performance of these approaches to show clear evidence that the variational quantum eigensolver is robust to certain errors. This error tolerance inspires hope that variational quantum simulations of classically intractable molecules may be viable in the near future.



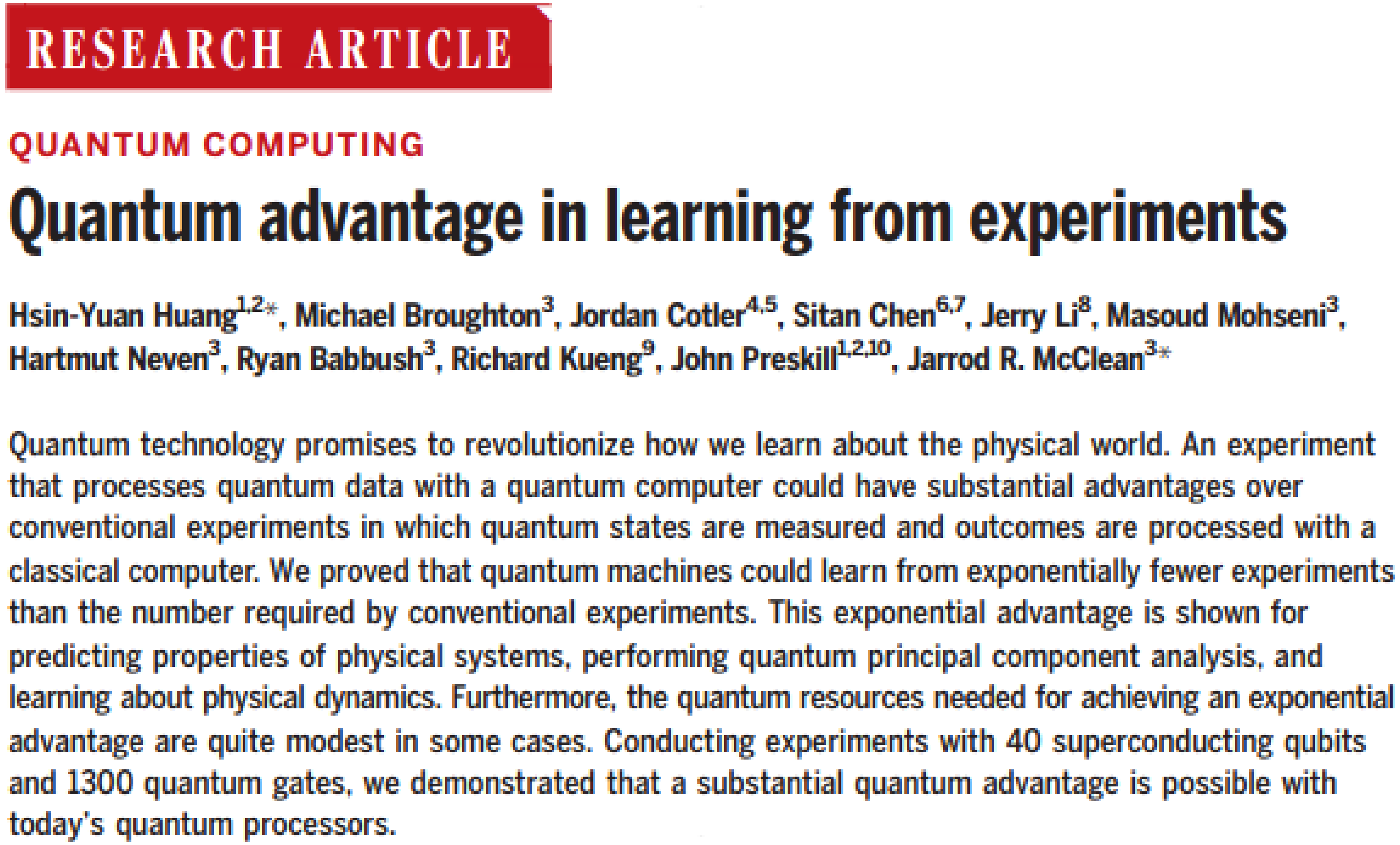
Low-Depth Quantum Simulation of Materials

Ryan Babbush,[1,*] Nathan Wiebe,[2] Jarrod McClean,[1] James McClain,[3] Hartmut Neven,[1] and Garnet Kin-Lic Chan[3,†]

[1]*Google Inc., Venice, California 90291, USA*
[2]*Microsoft Research, Redmond, Washington 98052, USA*
[3]*Division of Chemistry and Chemical Engineering, California Institute of Technology,*
Pasadena, California 91125, USA

(Received 2 June 2017; revised manuscript received 5 February 2018; published 21 March 2018)

Quantum simulation of the electronic structure problem is one of the most researched applications of quantum computing. The majority of quantum algorithms for this problem encode the wavefunction using N Gaussian orbitals, leading to Hamiltonians with $\mathcal{O}(N^4)$ second-quantized terms. We avoid this overhead and extend methods to condensed phase materials by utilizing a dual form of the plane wave basis which diagonalizes the potential operator, leading to a Hamiltonian representation with $\mathcal{O}(N^2)$ second-quantized terms. Using this representation, we can implement single Trotter steps of the Hamiltonians with linear gate depth on a planar lattice. Properties of the basis allow us to deploy Trotter- and Taylor-series-based simulations with respective circuit depths of $\mathcal{O}(N^{7/2})$ and $\tilde{\mathcal{O}}(N^{8/3})$ for fixed charge densities. Variational algorithms also require significantly fewer measurements in this basis, ameliorating a primary challenge of that approach. While our approach applies to the simulation of arbitrary electronic structure problems, the basis sets explored in this work will be most practical for treating periodic systems, such as crystalline materials, in the near term. We conclude with a proposal to simulate the uniform electron gas (jellium) using a low-depth variational ansatz realizable on near-term quantum devices. From these results, we identify simulations of low-density jellium as a promising first setting to explore quantum supremacy in electronic structure.



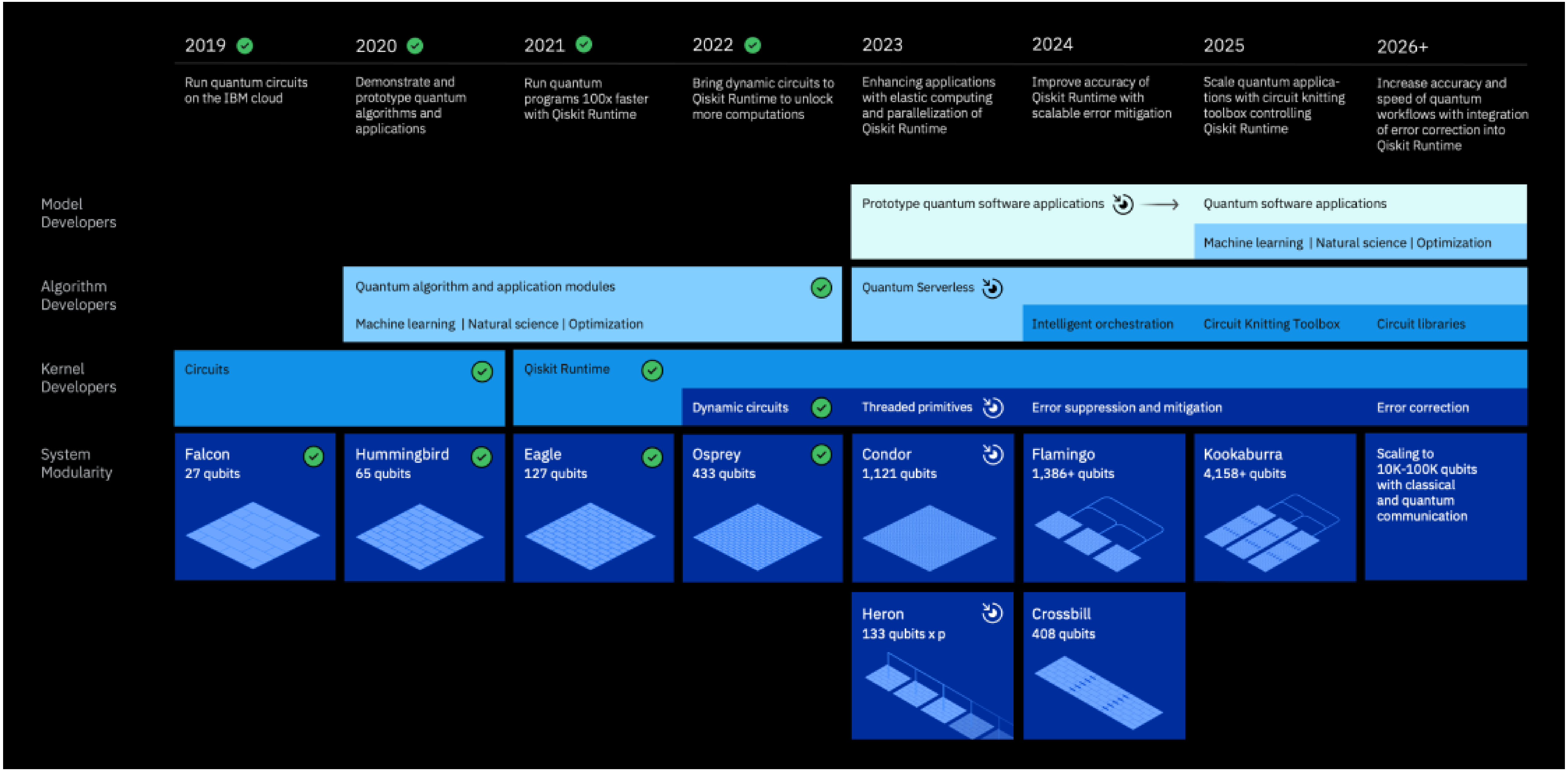
RESEARCH ARTICLE

QUANTUM COMPUTING

Quantum advantage in learning from experiments

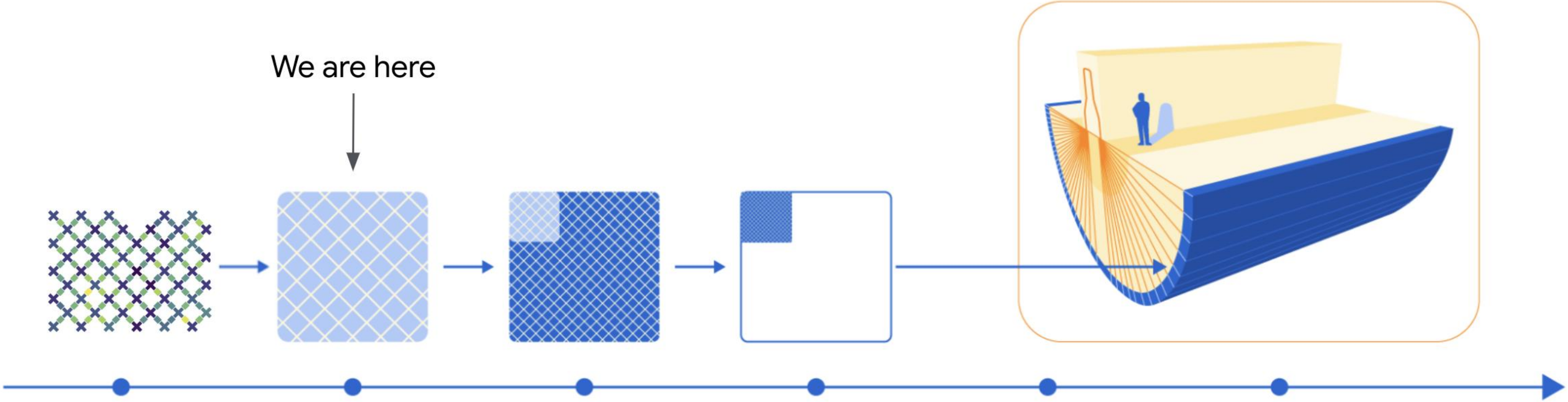
Hsin-Yuan Huang[1,2*], Michael Broughton[3], Jordan Cotler[4,5], Sitan Chen[6,7], Jerry Li[8], Masoud Mohseni[3], Hartmut Neven[3], Ryan Babbush[3], Richard Kueng[9], John Preskill[1,2,10], Jarrod R. McClean[3*]

Quantum technology promises to revolutionize how we learn about the physical world. An experiment that processes quantum data with a quantum computer could have substantial advantages over conventional experiments in which quantum states are measured and outcomes are processed with a classical computer. We proved that quantum machines could learn from exponentially fewer experiments than the number required by conventional experiments. This exponential advantage is shown for predicting properties of physical systems, performing quantum principal component analysis, and learning about physical dynamics. Furthermore, the quantum resources needed for achieving an exponential advantage are quite modest in some cases. Conducting experiments with 40 superconducting qubits and 1300 quantum gates, we demonstrated that a substantial quantum advantage is possible with today's quantum processors.



	2019 ✓	2020 ✓	2021 ✓	2022 ✓	2023	2024	2025	2026+
	Run quantum circuits on the IBM cloud	Demonstrate and prototype quantum algorithms and applications	Run quantum programs 100x faster with Qiskit Runtime	Bring dynamic circuits to Qiskit Runtime to unlock more computations	Enhancing applications with elastic computing and parallelization of Qiskit Runtime	Improve accuracy of Qiskit Runtime with scalable error mitigation	Scale quantum applications with circuit knitting toolbox controlling Qiskit Runtime	Increase accuracy and speed of quantum workflows with integration of error correction into Qiskit Runtime
Model Developers					Prototype quantum software applications ↻ →		Quantum software applications Machine learning | Natural science | Optimization	
Algorithm Developers		Quantum algorithm and application modules ✓ Machine learning | Natural science | Optimization			Quantum Serverless ↻	Intelligent orchestration	Circuit Knitting Toolbox	Circuit libraries
Kernel Developers	Circuits ✓		Qiskit Runtime ✓	Dynamic circuits ✓	Threaded primitives ↻	Error suppression and mitigation		Error correction
System Modularity	Falcon ✓ 27 qubits	Hummingbird ✓ 65 qubits	Eagle ✓ 127 qubits	Osprey ✓ 433 qubits	Condor ↻ 1,121 qubits	Flamingo 1,386+ qubits	Kookaburra 4,158+ qubits	Scaling to 10K-100K qubits with classical and quantum communication
					Heron ↻ 133 qubits x p	Crossbill 408 qubits		

We are here



physical qubits

54
Beyond classical
✔
M1 (2019)

10²
Logical qubit prototype
✔
M2 (2023)

10³
1 long-lived logical qubit
M3 (2025+)

10⁴
Tileable module (logical gate)
M4

10⁵
Engineering scale up
M5

10⁶
Error-corrected quantum computer
M6

?