NISQ Computers, a chemist point of view

Patrick CASSAM-CHENAÏ

Laboratoire J.A. Dieudonné, Nice, France

Nice 04/03/2022

Plan

Feynman's seminal ideas

International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

Simulating Physics with Computers

Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

Received May 7, 1981

Feynman's seminal ideas

- "(...) with a suitable class of quantum machines you could imitate any quantum system,(...)." $\rightsquigarrow$ idea of a quantum computer for quantum systems

- "What, (...), is the universal quantum simulator?" $\rightsquigarrow$ idea of the equivalent of a Turing machine for quantum computing

- "If you had discrete quantum systems, what other discrete quantum systems are exact imitators of it, and is there a class against which everything can be matched?" $\rightsquigarrow$ idea of quantum complexity classes

- "another system such that at each point in space-time this system has only two possible base states. Either that point is occupied, or unoccupied–those are the two states" $\rightsquigarrow$ idea of qubits as a two-level systems

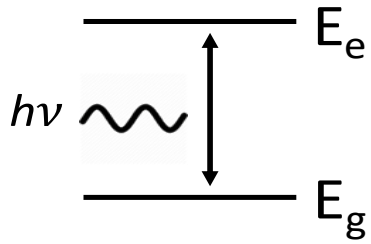
Qubits

0

1

classical bit

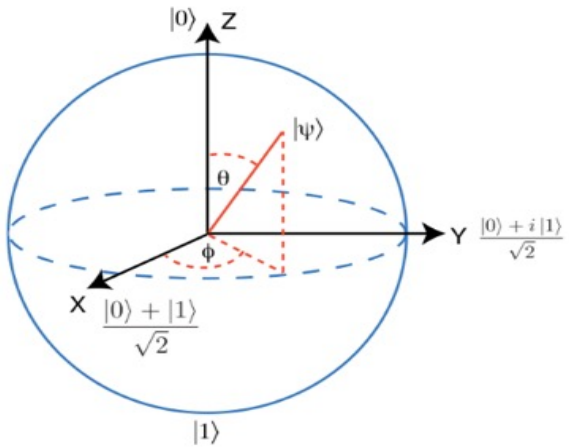
qubit : quantum bit



$|e\rangle \sim |1\rangle$

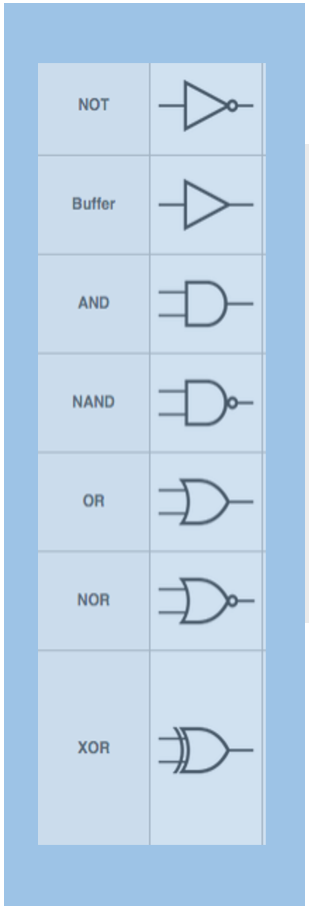
$|g\rangle \sim |0\rangle$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$



The Bloch sphere

gates = unitary operators



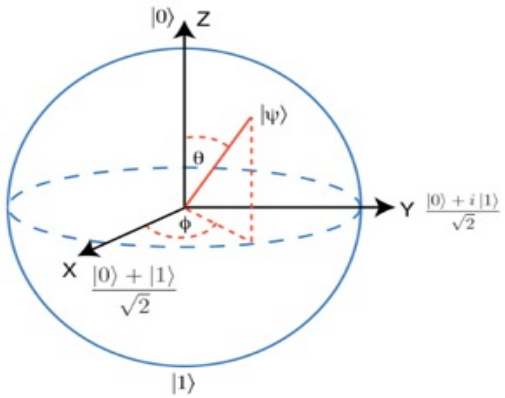
Controlling a qubit

« PAULI » Operators

rotation around x axis	⊕	$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$ qc.x(qr[n])	RX	$\begin{pmatrix}\cos\frac{\theta}{2} & -i\sin\frac{\theta}{2}\\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2}\end{pmatrix}$
rotation around y axis	Y	$\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$ qc.y(qr[n])	RY	$\begin{pmatrix}\cos\frac{\theta}{2} & -\sin\frac{\theta}{2}\\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2}\end{pmatrix}$
rotation around z axis	Z	$\begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}$ qc.z(qr[n])	RZ	$\begin{pmatrix}e^{-i\frac{\theta}{2}} & 0\\ 0 & e^{i\frac{\theta}{2}}\end{pmatrix}$
Identity	I	$\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$ qc.id(qr[n])		

Bloch Sphere

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}\,|1\rangle$$



superposition (X+Z) Hadamard gate

H $\quad \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$ qc.h(qr[n])

More operators are available from qiskit (S, T, swap, cswap, ccx, cz, ...)

CNOT : flips target qubit according to control qubit state.

$$\begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}$$

measurement measures quantum state in quantum register into classical register (0/1)

multi-qubit systems

- Register of n-qubits:

Hilbert space = tensor product of the Hibert spaces of each qubits

- Notation:

$$|i_1 \cdots i_n\rangle := |i_1\rangle \otimes \cdots \otimes |i_n\rangle \quad i_1, \ldots, i_n \in \{0, 1\}$$

$$|\Psi\rangle := \sum_{i_1 \ldots i_n} \alpha_{i_1 \ldots i_n} |i_1 \cdots i_n\rangle, \qquad \alpha_{i_1 \ldots i_n} \in \mathbb{C}$$

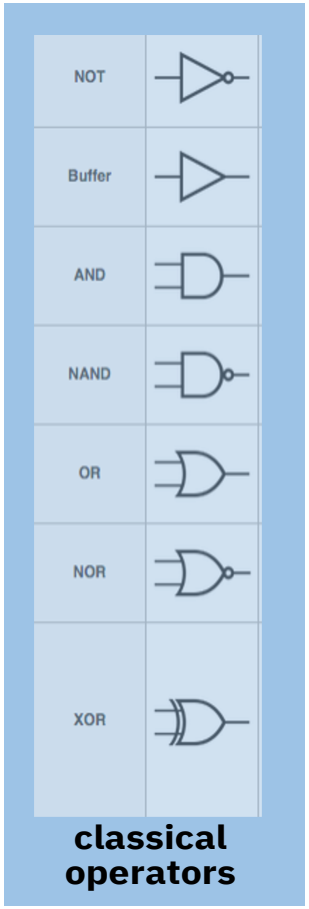
- Ex. 2-qubits:

Simple product states, $\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$

"Correlated" or "entangled" states such as Bell's

$$|\Psi^{00}\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) := \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$$

gates 2



classical operators

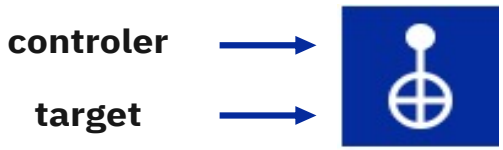
quantum operators :

H operator (Hadamard)

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

creates equal superposition of states $|0\rangle$ and $|1\rangle$

Control-Not operation



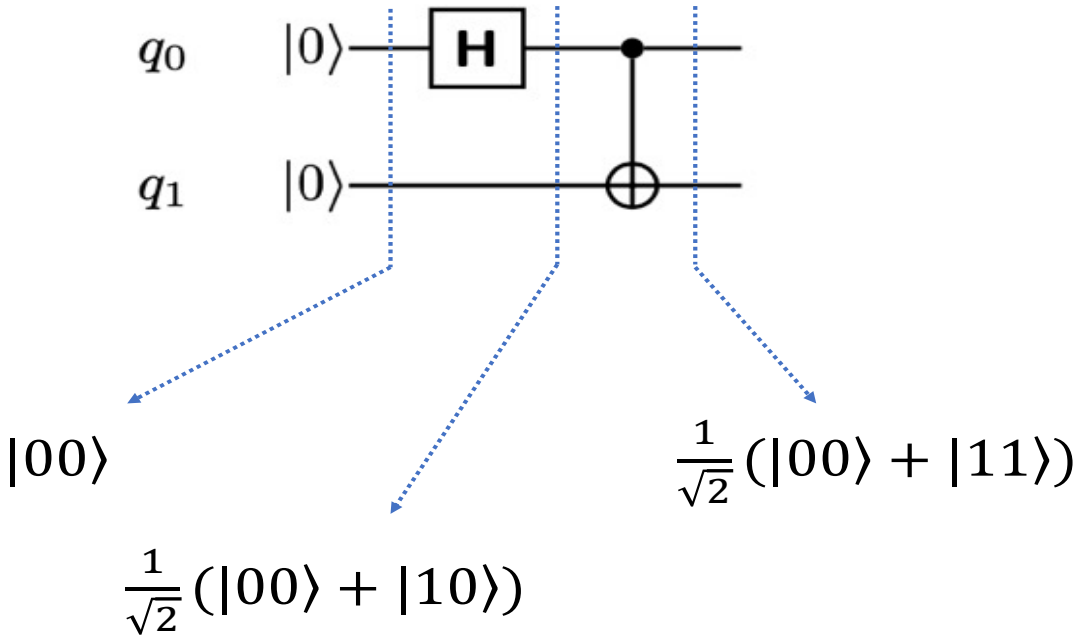
target qubit state is flipped if and only if the control qubit is in state $|1\rangle$

creates quantum entanglement of two qubits

Quantum circuit

```
print('Hello World!')
```

Hello World!



Complexity classes: hope is that $BQP \bigcap NP \neq \emptyset$

Table 1 Polynomial time complexity classes are separated by the resources the computer has access to while the non-deterministic polynomial time complexity classes are characterize by the resources of polynomial time computational verifiers

Class	Name	Computer type taking only poly. time
P	Polynomial time	Turing machine
BPP	Bounded error probabilistic polynomial time	Turing machine with access to random number generator
BQP	Bounded error quantum polynomial time	Turing machine with access to quantum resources
Class	**Name**	**Verifier's computer type taking only poly. time**
NP	Non-deterministic polynomial time	Turing machine
MA	Merlin–Arthur	Turing machine with access to random number generator
QMA	Quantum Merlin–Arthur	Turing machine with access to quantum resources

Phys. Chem. Chem. Phys., 2013, **15**, 397–411 | **401**

Quantum advantage

- Quantum supremacy (Preskill 2011)

Task not achievable classically but not necessarily useful (Google AI Quantum 2019)

- Quantum advantage:

Useful task where QC performs better than the best classical computer with the best classical algorithm in a broad sense, could be better approximate solution to an optimization pb under constraints, less power consumption, lower manufacturing cost, but in general:

- Quantum speedup:

does not necessarily implies a complexity class lowering, can be just a decrease in polynomial power within P, but the holy grail is going from exponential to polynomial cost $\rightsquigarrow$ Quantum Chemistry

Quantum computer classification

- Fault tolerant QC

circuit based + error correction method

- Adiabatic QC

based on the adiabatic theorem; polynomially equivalent to FTQC

- Noisy Intermediate Scale Quantum Computers:

circuit based; noise = errors depending on gate type; limited nb of qubits

- Quantum annealer:

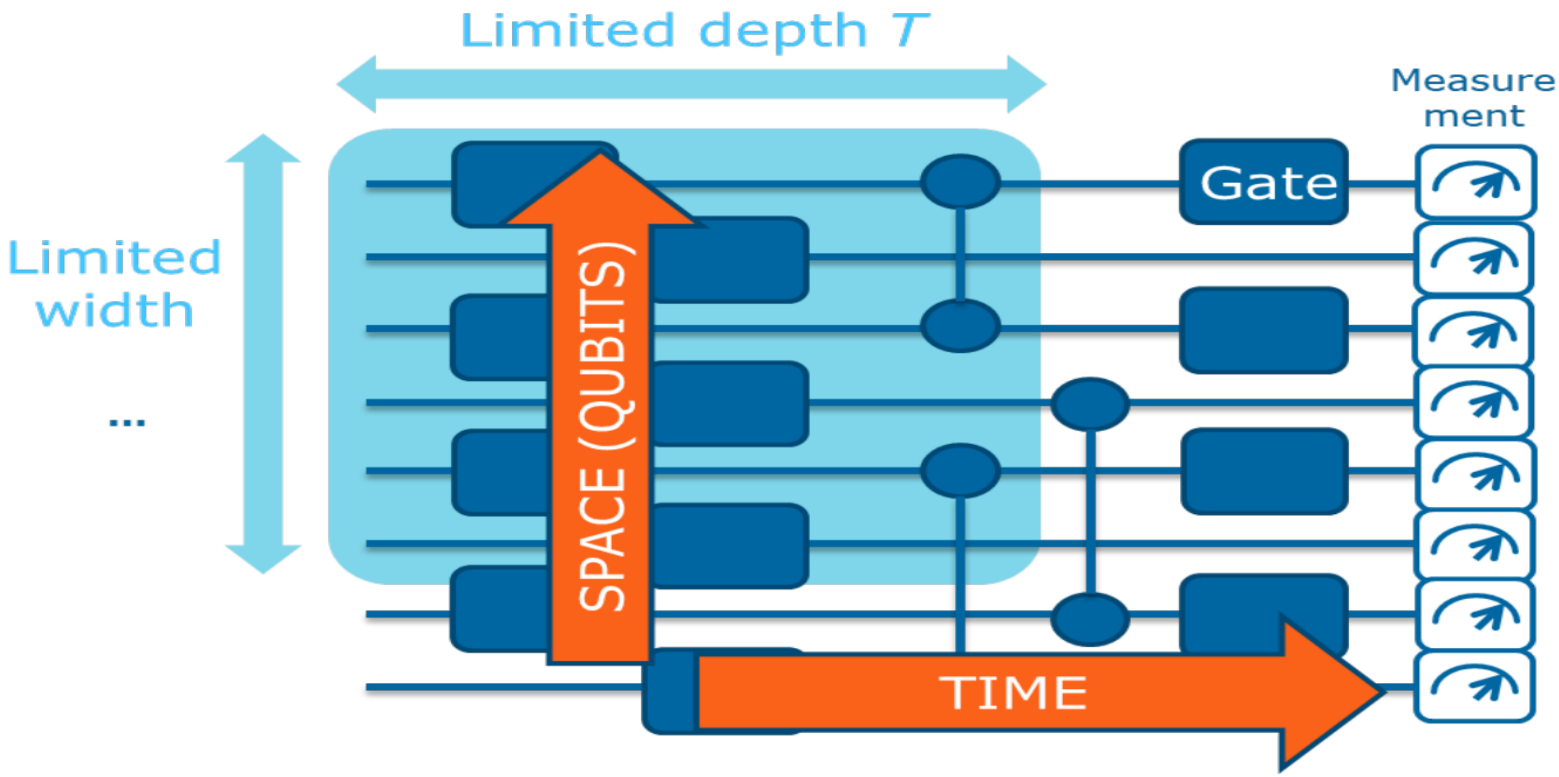
Adiabatic QC + noise; D-WAVE many qubits but low connectivity

(connectivity limits the NP pbs you can treat on a given hardware, Aida Todri-Sanial)

NISQ Limitations $\rightsquigarrow$ importance of the transpiler

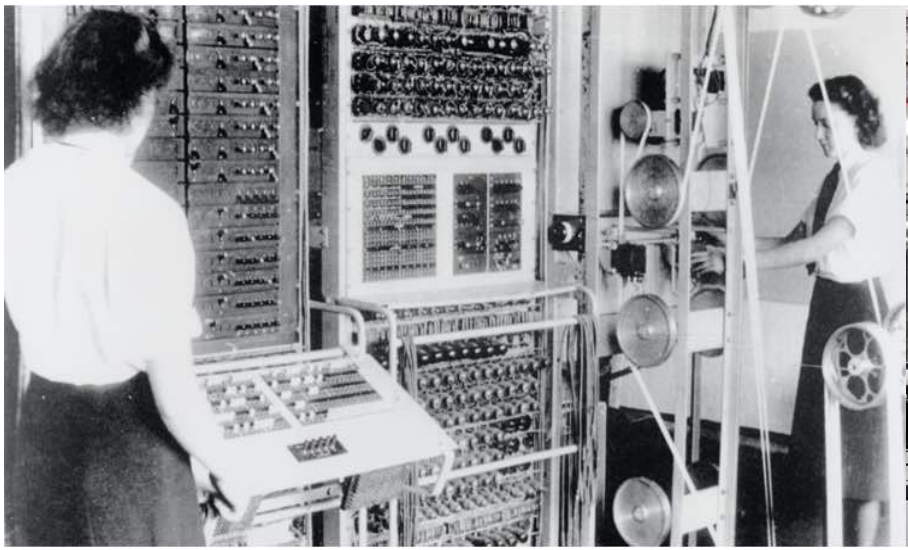
Limitations:

- **Few qubits + Short coherence times**



Historical perspective

We are in the early stages of a rapidly advancing new computing technology

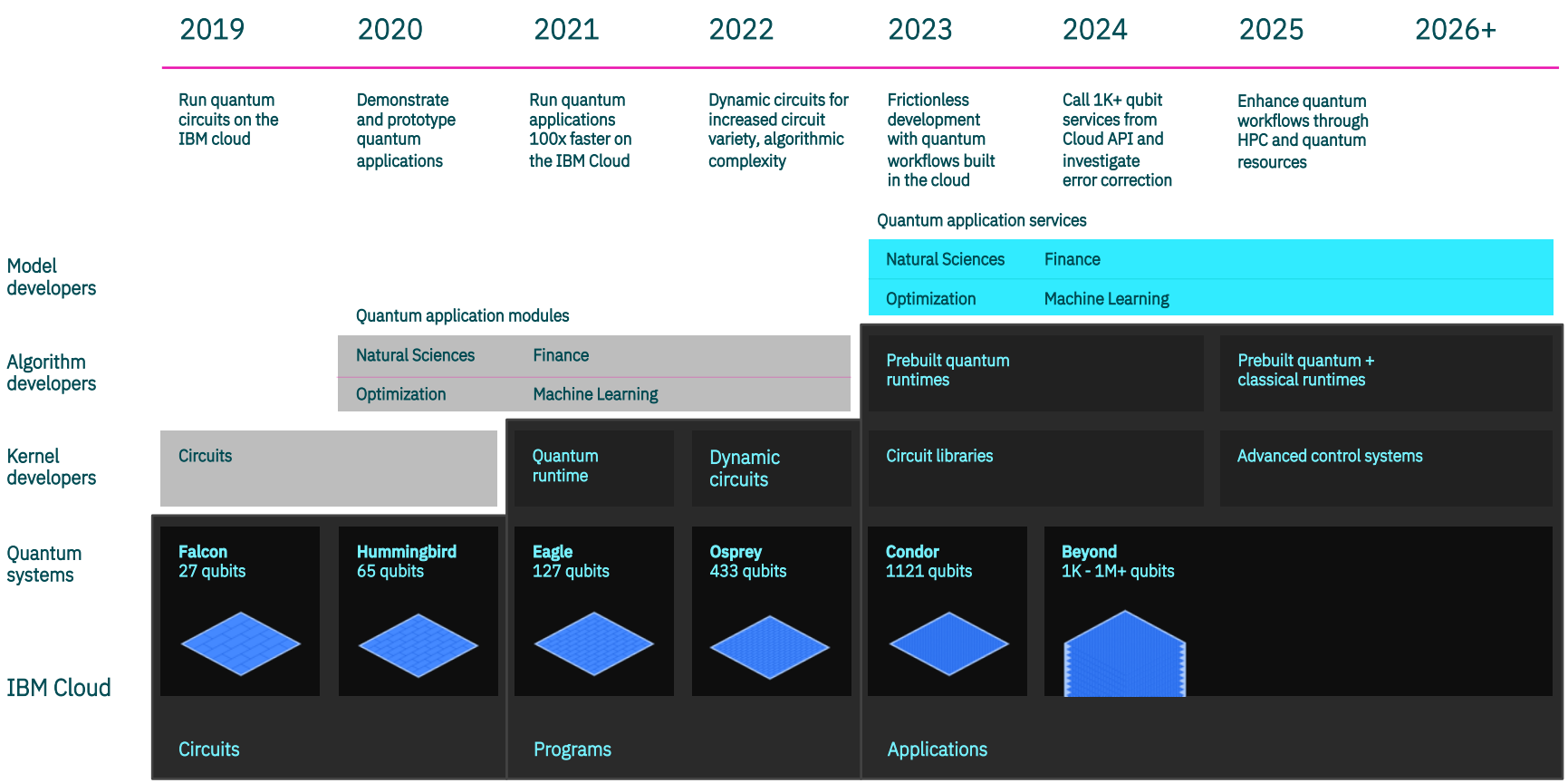


Classical Computer 1944

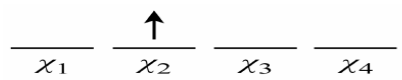
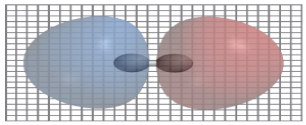
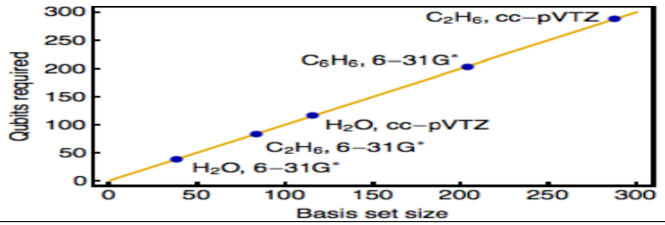
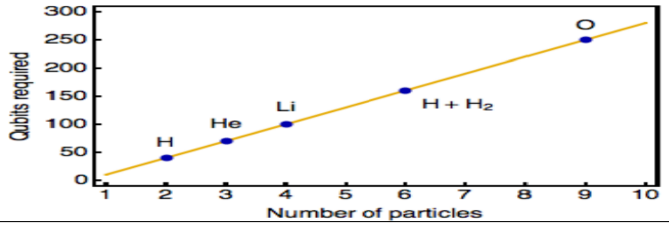
Quantum Computer 2019

Historical perspective

IBM Quantum Development Roadmap

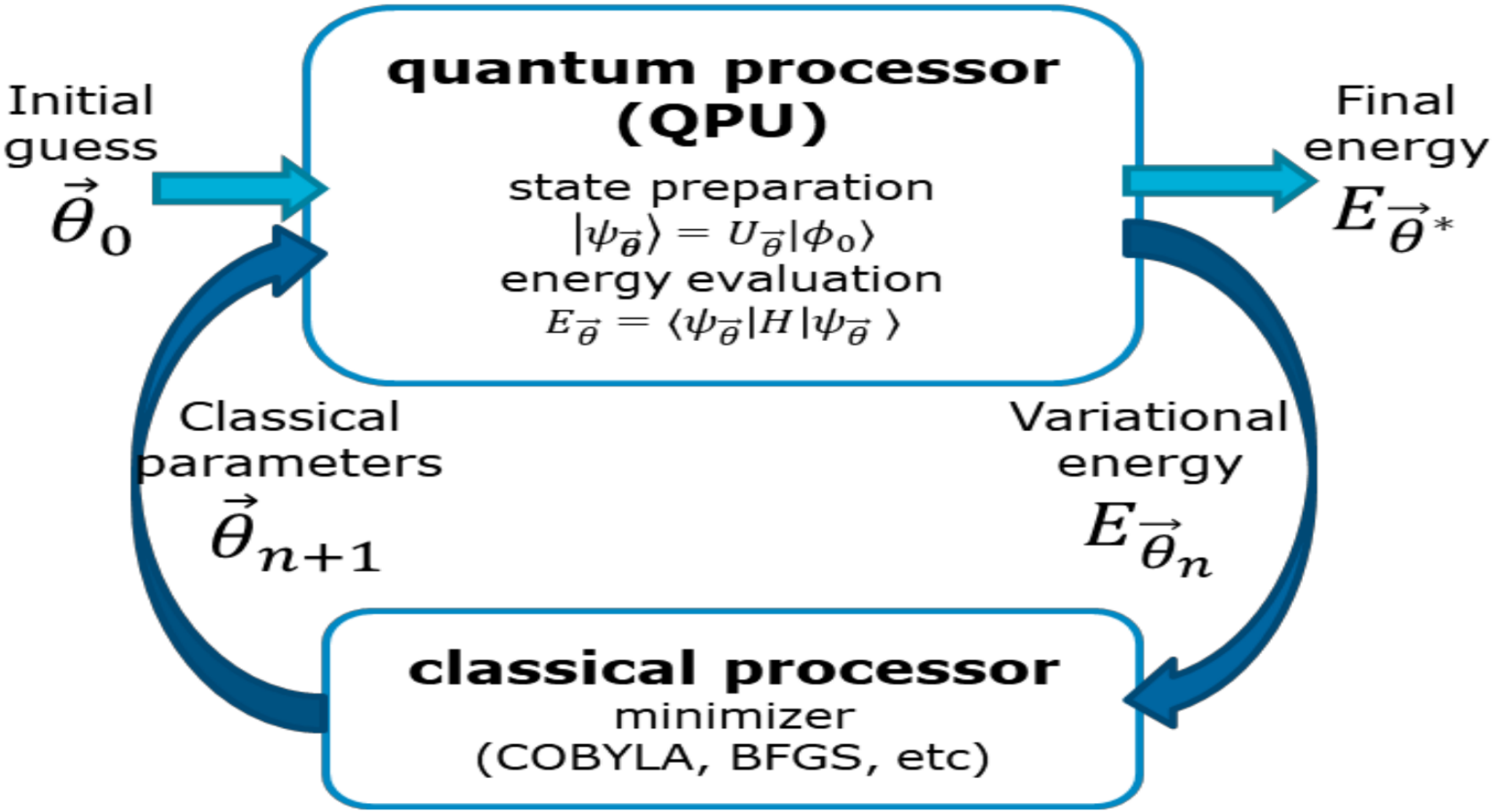


wave function encoding

	Second quantized	First quantized
wave function encoding	Fock state in a given basis:  $\lvert\psi\rangle = \lvert 0100\rangle$	On a grid of 2^n sites per dimension: $\lvert\psi\rangle = \sum_{\mathbf{x}} a_{\mathbf{x}} \lvert\mathbf{x}\rangle$ 
Qubits required to represent the wave function	One per basis state (spin-orbitals) 	$3n$ 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
Quantum gates required for simulation	$O(M^5)$ with number of basis states	$O(B^2)$ with number of particles
Advantages	• Compact wave function representation (requires fewer qubits) • Takes advantage of classical electronic-structure theory to improve performance • Already experimentally implemented	• Better asymptotic scaling (requires fewer gates) • Treats dynamics better • Can be used for computing reaction rates or state-to-state transition amplitudes

VQE based algo

Variational Quantum Eigensolving, Peruzzo `14



ATOS simulator

myQLM documentation

myQLM

myQLM-1.5.1-1

Search docs

BASIC USAGE

- Installing myQLM
- Getting started
- Writing quantum circuits
- Executing quantum circuits
- Running variational algorithms
- Combinatorial optimization
- Interoperability with myQLM

ADVANCED USAGE

- Main objects: Jobs, Observables, Circuits...
- Building custom execution stacks
- Advanced programming using pyAQASM
- Advanced combinatorial optimization
- The AQASM format
- Command-line tools

SOURCE CODE DOCUMENTATION

- Source code documentation

ADDITIONAL INFORMATION

- Support
- Notebooks

» Quantum Application Toolset

Welcome to the myQLM documentation

myQLM is a quantum software stack for writing, simulating, optimizing, and executing quantum programs. Through a Python interface, it provides

- **powerful semantics** for manipulating quantum circuits, with support for universal as well as custom gate sets, abstract parameters, advanced linking options, etc.;
- **a versatile execution stack** for running quantum jobs, including an easy handling of observables, special plugins for carrying out NISQ-oriented variational methods (such as VQE, QAOA), and easy API for writing customized plugins (e.g for compilation or error mitigation), as well as for connecting to any Quantum Processing Unit (QPU);
- **a seamless interface** to available quantum processors and major quantum programming frameworks.

Basic usage

- Installing myQLM
- Getting started
- Writing quantum circuits
- Executing quantum circuits
- Running variational algorithms
- Combinatorial optimization
- Interoperability with myQLM

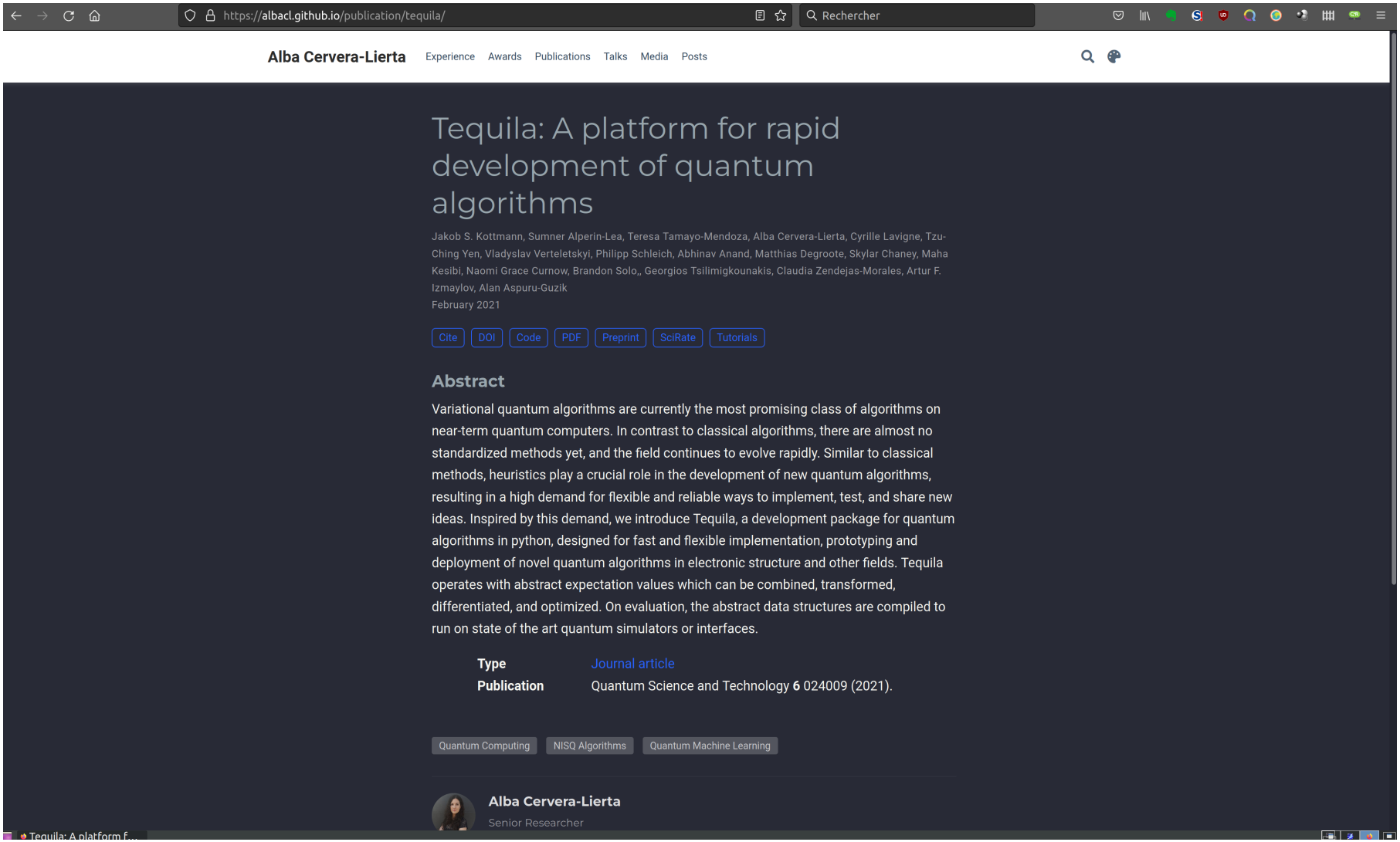
Advanced usage

- Main objects: Jobs, Observables, Circuits...
- Building custom execution stacks
- Advanced programming using pyAQASM
- Advanced combinatorial optimization
- The AQASM format
- Command-line tools

Source code documentation

- Source code documentation
  - Quantum Application ToolChain (QAT) Python libraries
  - The core library

Tequila


https://albacl.github.io/publication/tequila/
Rechercher
Alba Cervera-Lierta
Experience
Awards
Publications
Talks
Media
Posts
Tequila: A platform for rapid development of quantum algorithms
Jakob S. Kottmann, Sumner Alperin-Lea, Teresa Tamayo-Mendoza, Alba Cervera-Lierta, Cyrille Lavigne, Tzu-Ching Yen, Vladyslav Verteletskyi, Philipp Schleich, Abhinav Anand, Matthias Degroote, Skylar Chaney, Maha Kesibi, Naomi Grace Curnow, Brandon Solo,, Georgios Tsilimigkounakis, Claudia Zendejas-Morales, Artur F. Izmaylov, Alan Aspuru-Guzik
February 2021
Cite
DOI
Code
PDF
Preprint
SciRate
Tutorials
Abstract
Variational quantum algorithms are currently the most promising class of algorithms on near-term quantum computers. In contrast to classical algorithms, there are almost no standardized methods yet, and the field continues to evolve rapidly. Similar to classical methods, heuristics play a crucial role in the development of new quantum algorithms, resulting in a high demand for flexible and reliable ways to implement, test, and share new ideas. Inspired by this demand, we introduce Tequila, a development package for quantum algorithms in python, designed for fast and flexible implementation, prototyping and deployment of novel quantum algorithms in electronic structure and other fields. Tequila operates with abstract expectation values which can be combined, transformed, differentiated, and optimized. On evaluation, the abstract data structures are compiled to run on state of the art quantum simulators or interfaces.
Type
Journal article
Publication
Quantum Science and Technology 6 024009 (2021).
Quantum Computing
NISQ Algorithms
Quantum Machine Learning
Alba Cervera-Lierta
Senior Researcher

Xanadu

https://strawberryfields.readthedocs.io/en/stable/

STRAWBERRY FIELDS | Quantum Photonics | Install | Documentation | FAQ | Support | GitHub

Search

Using Strawberry Fields
- Introduction
- Hardware and cloud
- Circuits
- Operations
- States
- GBS datasets

Development
- Development guide
- Migration guides
- Research and contribution
- Release notes

API
- sf
- sf.apps
- sf.backends
- sf.compilers
- sf.circuitdrawer
- sf.decompositions
- sf.engine
- sf.io
- sf.ops
- sf.parameters
- sf.program
- sf.program_utils
- sf.plot
- sf.tdm
- sf.utils

Strawberry Fields Documentation

Release
0.21.0

Strawberry Fields is a full-stack Python library for designing, optimizing, and utilizing photonic quantum computers.

Contents
- Features
- How to cite
- Support
- License

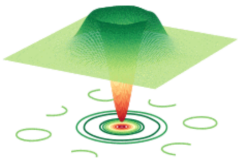
Using SF
Learn how to interact with a photonic quantum computer »

Developing
How you can contribute to Strawberry Fields »

API
Explore the Strawberry Fields API »

Features

- Execute photonic quantum algorithms directly on Xanadu's next-generation quantum hardware
- High-level functions for **solving practical problems** including **graph and network optimization**, **machine learning**, and **chemistry**
- Includes a suite of world-class simulators—based on cutting edge algorithms—to compile and simulate photonic algorithms
- Train and optimize your quantum programs with our end-to-end differentiable TensorFlow backend



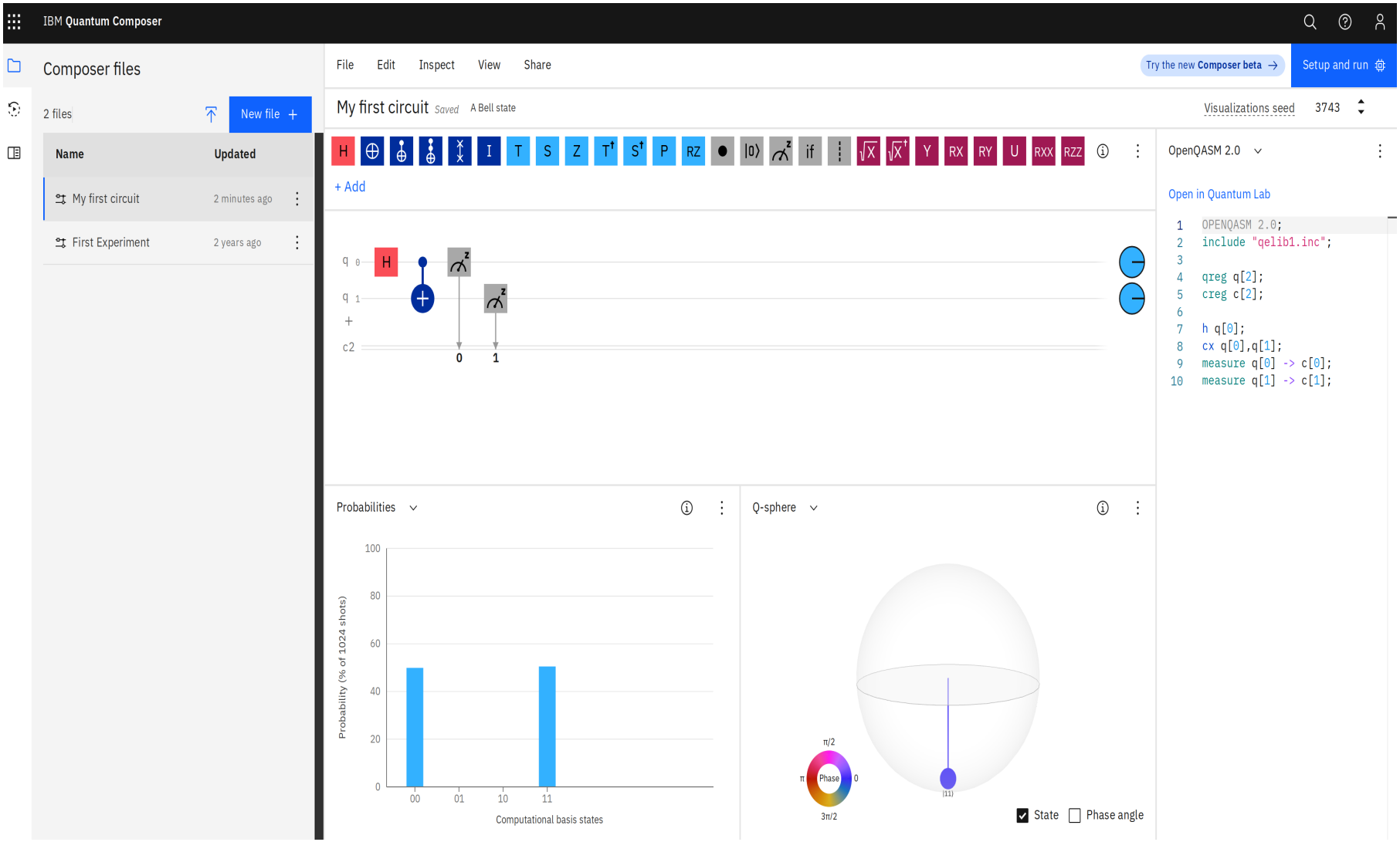
How to cite

If you are doing research using Strawberry Fields, please cite our papers:

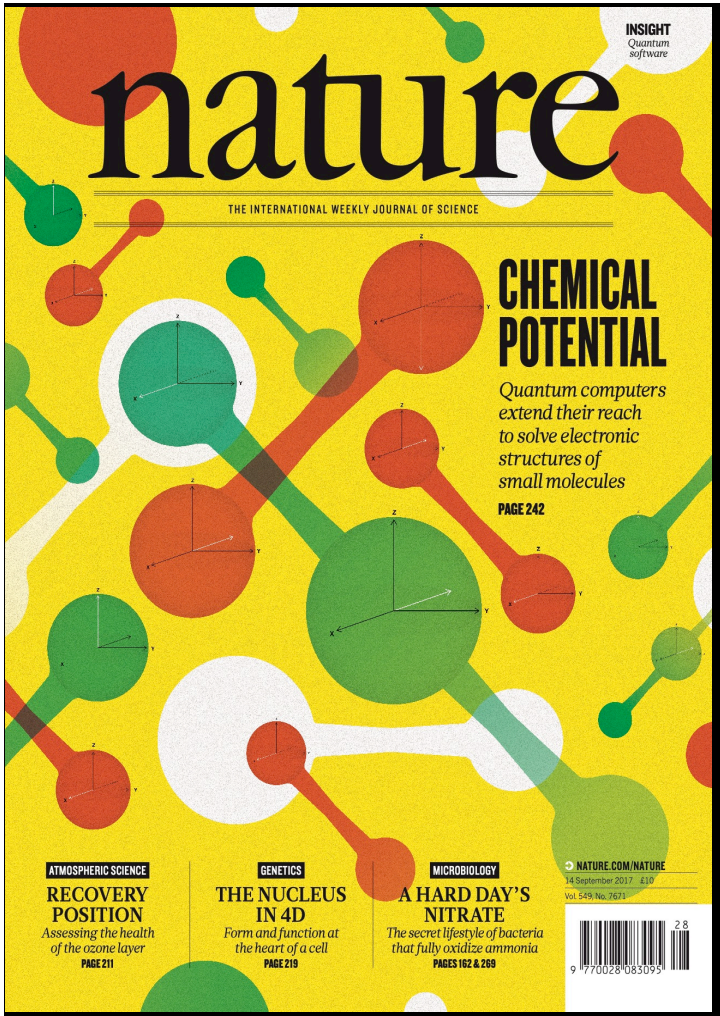
Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook. "Strawberry Fields: A Software Platform for Photonic Quantum Computing", Quantum, 3, 129 (2019).

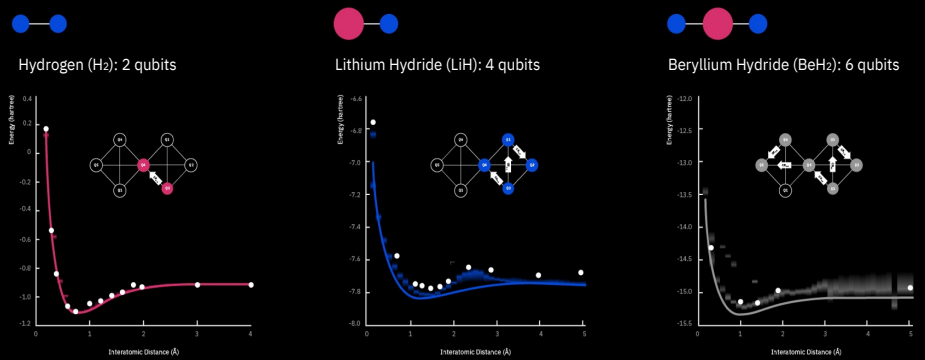
v: stable

QISKIT composer

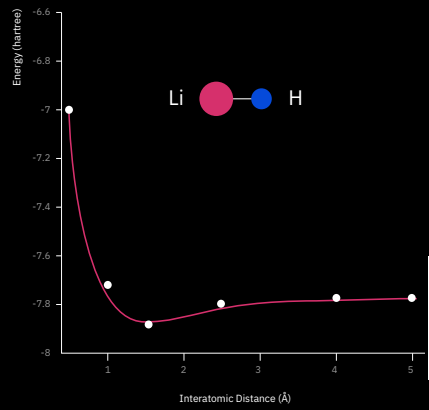


Chemical applications





IBM 2018



10X better
using software-based error mitigation

1

Theoretical chemists have proposed:

1. Chemical systems as candidates for making quantum devices.

Csaba Fábri, Sieghard Albert, Ziqiu Chen, Robert Prentner and Martin Quack, Phys. Chem. Chem. Phys. 20, 7387 (2018) A molecular quantum switch based on tunneling in meta-D-phenol C6H4 DOH

2. Chemical systems as candidates for making qubits

K. Shioya, K. Mishima and K. Yamashita, Mol. Phys. 105, 1283 (2007) Quantum computing using molecular vibrational and rotational modes
D. Weidinger and M. Gruebele, Mol. Phys. 105, 1999 (2007) Quantum computation with vibrationally excited polyatomic molecules: effects of rotation, level structure, and field gradients
D. Weidinger and M. Gruebele, Chem. Phys. 350, 139 (2008) Simulations of quantum computation with a molecular ion
Qi Wei, Sabre Kais, Bretislav Friedrich, and Dudley Herschbach, J Chem. Phys. 135, 154102 (2011) Entanglement of polar symmetric top molecules as candidate qubits

Theoretical chemists have designed quantum algorithms to:

1. determine molecular structures (by finding global minima with a modified Grover's algorithm)

Zhu Jing, Huang Zhen and Kais Sabre, Mol. Phys. 107, 2015 (2009) Simulated quantum computation of global minima

2. Calculate molecular energies (by solving Schrödinger equation for electrons, nuclei or both)

Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, Sukin Sim, Libor Veis, and Alán Aspuru-Guzik, Chem. Rev. 119, 10856 (2019), Quantum Chemistry in the Age of Quantum Computing
Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan, Rev. Mod. Phys. 92, 015003 (2020) Quantum computational chemistry
Alessandro Rossi, Paul G. Baity, Vera M. Schäfer, Martin Weides, Int J Quantum Chem. 121, e26688 (2021) Quantum computing hardware in the cloud: Should a computational chemist care?

Recent review

A. Rossi, P. G. Baity, V. M. Schäfer, M. Weides, Int J Quantum Chem. 121, e26688 (2021)
Quantum computing hardware in the cloud: Should a computational chemist care?

TABLE 1 Quantum computing hardware in the cloud (a non-exhaustive selection)

Manufacturer	Platform	Cloud access	Max # qubits	Gate fidelity (1-qubit, 2-qubit)	Max QV	Simulated molecules
IBM	Superconducting	IBM Quantum Experience (Open access)	15 (Melbourne)	99.97%, 99.16% (Santiago)	32 (Santiago)	H_2, LiH, BeH_2, NaH, KH, RbH
IonQ	Trapped Ions	Microsoft Azure or Amazon Bracket	11	99.50%, 97.50%	Not published	H_2O
QuTech	Silicon	Quantum Inspire	2 (Spin2-QPU)	$\approx$ 99%, $\approx$ 90%	Not published	none
Google	Superconducting	Google Quantum AI	53 (Sycamore)	99.85%, 99.35%	Not published	H_2N_2, H_6, H_8, H_{10}, H_{12}
Rigetti	Superconducting	Rigetti Quantum Cloud	31 (Aspen-8)	99.8%, 95.9%	8 (Aspen-4)	NaH, H_2
Honeywell	Trapped Ions	Microsoft Azure or Amazon Bracket	10 (H1)	99.97%, 99.5%	128	None

Note: Wherever more than one QPU is available, the relevant machine is indicated within brackets. Simulated molecules column denotes experiments run with any quantum machine from the relevant manufacturer, not necessarily one of those listed. IBM hardware considered is limited to open access services. Google cloud services are limited to emulators at present, although the reported chemical simulations have been performed with proprietary physical hardware.

Thank you for your attention.

We acknowledge copy-pasting elements from many other people presentations including: J.-M. Torres from IBM-Q, T. Ayral from ATOS Quantum lab., O. Hess (formerly IBM-Q, now ATOS), J. D. Whitefield from Univ. of Vienna,