VOL. 71 JUL 2026

NTU Q

UPCOMING EVENT

SCALING FOR QUANTUM ADVANTAGE AND BEYOND

- Time: 10th – 11th August
- Registration deadline: 25th July, 16:00 P.M.
- Venue: NCTS Physics Lecture Hall, 4F, Chee-Chun Leung Cosmology Hall, National Taiwan University

LEARN MORE

NTU-IBM QUANTUM SYSTEM USERS MEETING 2026

- Time: 12th August, 10:30 A.M. – 12:30 P.M.
- Registration deadline: 12th July, 16:00 P.M.
- Venue: 1F, Center for Condensed Matter Sciences and New Physics Building, National Taiwan University
- Please do not register if you are already accepted by the Qiskit Hackathon Taiwan 2026.

LEARN MORE

HIGHLIGHTING NEWS

STATUS OF QUANTUM PROGRAMMING LANGUAGES & FRAMEWORKS

Unlike classical programming, which operates on deterministic binary bits, quantum programming is built under the phenomena such as superposition, entanglement, and quantum interference, requiring specialized languages, compilers, and development environments. A recent survey overview the current landscape of quantum computing programming languages and software frameworks, emphasizing that quantum software is becoming an increasingly important layer of the quantum-computing ecosystem as hardware continues to mature. Generally, the software stack can be categorized into three broad levels: instruction sets (e.g., OpenQASM, Quil), which translate algorithms into hardware-level operations for quantum processors; software development kits (SDKs) (e.g., Qiskit, PannyLane), which provide simulation and hardware-access tools in Python; and higher-level quantum programming languages (e.g., Bloqade), which enable developers to express specific problems more efficiently.

Several of influential quantum platforms using Python has become the dominant entry point for quantum programming due to its extensive ecosystem and integration with major quantum programming frameworks such as Qiskit (IBM), Q# (Microsoft), Cirq (Google), and CUDA-Q (Nvidia). These leading development environments support circuit design, simulation, optimization, and execution on real quantum hardware. Looking ahead, as quantum computers progress toward practical applications, software platforms are expected to evolve similarly to modern machine-learning frameworks. This trend could significantly expand the pool of quantum developers by enabling programmers without deep expertise in quantum physics to build useful applications.

READ MORE (News)

ffsim LIBRARY TO SPEED UP FERMIONIC QUANTUM CIRCUITS

IBM has introduced ffsim, an open-source Python library designed for the fast classical simulation of fermionic quantum circuits, which are widely used to model molecules, materials, and other many-electron quantum systems. The motivation behind ffsim is that quantum simulators generally require the full 2^n-dimensional Hilbert space to represent an n-qubit system, which wastes the computational resources immensely. To overcome this inefficiency, ffsim exploits two conserved quantities common to fermionic systems—the total particle number and the spin-z component—thereby only the symmetry-constrained subspace is useful to express the physical states, which dramatically reduces both memory requirements and computational cost. As an illustrative example, a 64-qubit Hubbard-model simulation on a 4×8 lattice that would require approximately unfeasible 256 EiB of memory using a conventional state-vector simulator can be performed with only 19.3 GiB using ffsim.

Beyond its computational efficiency, ffsim provides a comprehensive platform for prototyping and validating quantum algorithms before implementing on real quantum hardware. The library includes number-preserving fermionic gates, Hamiltonian time evolution, variational ansatz, and efficient sampling routines, while integrating seamlessly with Qiskit and PySCF. IBM reports that ffsim outperforms existing fermionic simulators such as FQE by up to an order of magnitude on representative quantum-chemistry workloads and can also simulate a broader class of Hamming weight-preserving

quantum circuits beyond strictly fermionic applications. By enabling rapid benchmarking and verification of quantum algorithms on classical hardware, ffsim is intended to bridge the gap between algorithm design and hardware execution, providing researchers with a practical tool for developing quantum applications in chemistry, materials science, and related fields as quantum processors continue to advance.

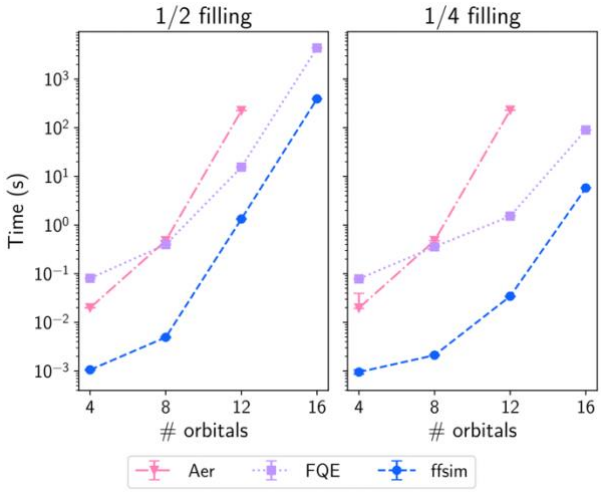

Fig. Comparison of Qiskit Aer, FQE, and ffsim runtimes for Trotterized time evolution of the molecular double factorized Hamiltonian.

READ MORE (Press Release & Research Article)

計畫補助單位：NSTC 國家科學及技術委員會 National Science and Technology Council

IBM Quantum Computer Hub at National Taiwan University

Rm.711, Dept. of Physics /Center for Condensed Building

No. 1, Sec.4 Roosevelt Rd., Da'an Dist. Taipei City 106319,

ntuq2018@gmail.com :+886 2-33669928 http://quantum.ntu.edu.tw/