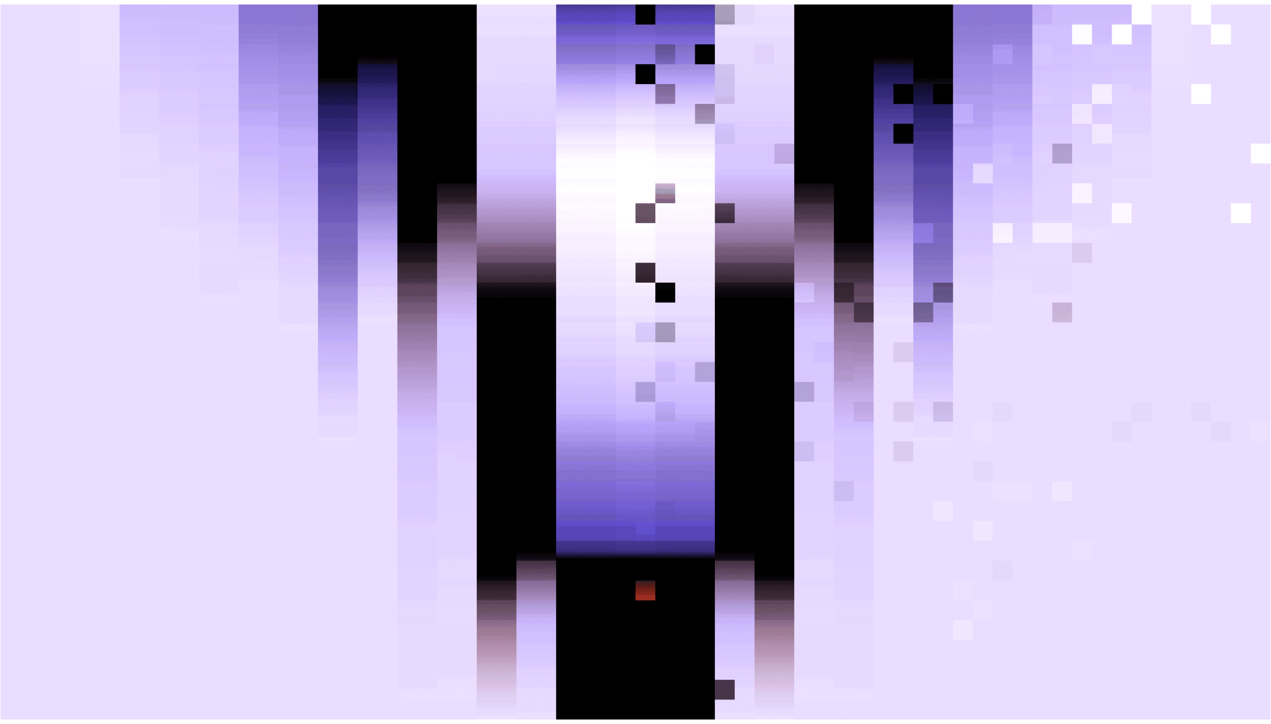
Simulating the universe’s most extreme environments with utility-scale quantum computation

Recent research out of the University of Washington and Lawrence Berkeley National Laboratory demonstrates scalable techniques that could one day enable simulations of fundamental physics experiments at the highest energies.



Marc Illa
Anthony N. Ciavarella
Martin J. Savage
Robert Davis

Topics

Error Correction & Mitigation

Research

Use Cases

Qiskit

Share this blog

we know about the tiny quantum-scale particles that make up our everyday world. It is a remarkable achievement, but it’s also incomplete — rife with unanswered questions. To fill the gaps in our knowledge, and discover new laws of physics beyond the Standard Model, we must study the exotic phenomena and states of matter that *don’t* exist in our everyday world. These include the high-energy collisions of particles and nuclei that take place in the fiery heart of stars, in cosmic ray events occurring all across earth’s upper atmosphere, and in particle accelerators like the Large Hadron Collider (LHC) at CERN or the Relativistic Heavy Ion Collider at Brookhaven National Laboratory.

Computer simulations of fundamental physics processes play an essential role in this research, but many important questions require simulations that are much too complex for even the most powerful classical supercomputers. Now that utility-scale quantum computers have demonstrated the ability to simulate quantum systems at a scale beyond exact or “brute force” classical methods, researchers are exploring how these devices might help us run simulations and answer scientific questions that are inaccessible to classical computation. In two recent papers published in PRX Quantum (PRX)[1] and Physical Review D (PRD)[2], our research group did just that, developing scalable techniques for simulating the real-time dynamics of quantum-scale particles using the IBM® fleet of utility-scale, superconducting quantum computers.

The techniques we’ve developed could very well serve as the building blocks for future quantum computer simulations that are completely inaccessible to both exact and even approximate classical methods — simulations that would demonstrate what we call “quantum advantage” over all known classical techniques. Our results provide clear evidence that such simulations are potentially within reach of the quantum hardware we have today.

Utility-scale simulations for

nuclear and high-energy physics

We are a team of researchers from the University of Washington and Lawrence Berkeley National Laboratory who have spent years investigating the use of quantum hardware for simulations of quantum chromodynamics (QCD).

This work was supported, in part, by the U.S. Department of Energy grant DE-FG02-97ER-41014 (Farrell), by U.S. Department of Energy, Office of Science, Office of Nuclear Physics, InQubator for Quantum Simulation (IQuS) under Award Number DOE (NP) Award DE-SC0020970 via the program on Quantum Horizons: QIS Research and Innovation for Nuclear Science (Anthony Ciavarella, Roland Farrell, Martin Savage), the Quantum Science Center (QSC) which is a National Quantum Information Science Research Center of the U.S. Department of Energy (DOE) (Marc Illa), and by the U.S. Department of Energy (DOE), Office of Science under contract DE-AC02-05CH11231, through Quantum Information Science Enabled Discovery (QuantISED) for High Energy Physics (KA2401032) (Anthony Ciavarella).

This work is also supported, in part, through the Department of Physics and the College of Arts and Sciences at the University of Washington.

This research used resources of the Oak Ridge Leadership Computing Facility (OLCF), which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

We acknowledge the use of IBM Quantum services for this work.

This work was enabled, in part, by the use of advanced computational, storage and networking infrastructure provided by the Hyak supercomputer system at the University of Washington.

This research was done using services provided by the OSG Consortium, which is supported by the National Science Foundation awards #2030508 and #1836650.”

QCD is the theory of the strong force that binds the nuclei of atoms together, and it presents enormous challenges when it comes to observing, modeling, and predicting the behavior of fundamental particles.

One prominent example of these challenges comes from the field of collider physics. Physicists use colliders like the LHC to smash beams of particles and atomic nuclei into each other at extraordinarily high energies, recreating the kinds of collisions that take place in stars and cosmic ray events. Collider experiments give physicists the ability to observe how matter behaves in the universe’s most extreme discover new physics beyond the Standard Model.

Let’s say we want to use the data from collider experiments to identify new physics theories. To do this, we must be able to accurately predict the way *known* physics theories like QCD contribute to the exotic physics processes that occur in collider runs, and we must be able to quantify the uncertainties of the corresponding theoretical calculations. Performing these tasks requires detailed simulations of systems of fundamental particles. These simulations are impossible to achieve with classical computation alone, but should be well within reach for a sufficiently capable quantum computer.

Quantum computing hardware is making rapid progress toward the day when it will be capable of simulating complex systems of fundamental particles, but we can’t just sit back and wait for quantum technology to reach maturity. When that day comes, we’ll need to be ready with scalable techniques for executing each step of the simulation process.

The research community is already beginning to make significant progress in this field, with most efforts today focused on simulations of simplified, low-dimensional models of QCD and other fundamental physics theories. This is exactly what our research group has been working on, with our experiments primarily centering on simulations of the widely used Schwinger model, a one-dimensional model of QCD that describes how electrons and positrons behave and interact through the exchange of photons.

In a paper submitted to arXiv in 2023, and published in PRX Quantum this past April, we used the Schwinger model to demonstrate the first essential step in building future simulations of high-energy collisions of matter: preparing a simulation of the quantum vacuum state in which particle collisions would occur. Our follow-up to that paper, published in PRD in June, shows techniques for performing the next step in this process — preparing a beam of particles in the quantum

More specifically, that follow-up paper shows how to prepare hadron wavepackets in a 1-dimensional quantum simulation and evolve them forward in time. In this context, you can think of a hadron as a composite particle made up of a positron and electron and bound together by something analogous to the strong force that binds neutrons and protons together in nuclei.

Due to the uncertainty principle, it is impossible to precisely know both the position and momentum of a particle. The best you can do is to create a wavepacket, a region of space over which a particle will appear with some probability and with a range of different momenta. The uncertainty in momentum causes the wavepacket to spread out or "propagate" across some area of space.

By evolving our hadron wavepacket forward in time, we effectively create a simulation of pulses or beams of hadrons moving in this 1-dimensional system, just like the beams of particles we smash into each other in particle colliders. The wavepacket we create has an equal probability of propagating in any direction. However, since we're working in 1-dimensional space, essentially a straight line, it's more accurate to say the particle is equally likely to propagate to the left or to the right.

Our simulation method

We've established that our primary goal is to simulate the dynamics of a composite hadron particle moving through the quantum vacuum in one-dimensional space. To achieve this, we'll need to prepare an initial state with the hadron situated on a simplified model of space made up of discrete points — also known as a "lattice." Then, we'll have to perform what we call "time evolution" so we can see the hadron move around and study its dynamics.

Our first step is to determine the quantum circuits we'll need to run on developed a new state preparation algorithm, Scalable Circuits ADAPT-VQE. This algorithm uses the popular ADAPT-VQE algorithm as a subroutine, and is able to find circuits for preparing the state with the lowest energy — i.e., the “ground state” — as well as a hadron wavepacket state. A key feature of this technique is the use of classical computers to determine circuit blocks for preparing a desired state on a small lattice that can be systematically scaled up to prepare the desired state on a much larger lattice. These scaled circuits cannot be executed exactly on a classical computer and are instead executed on a quantum computer.

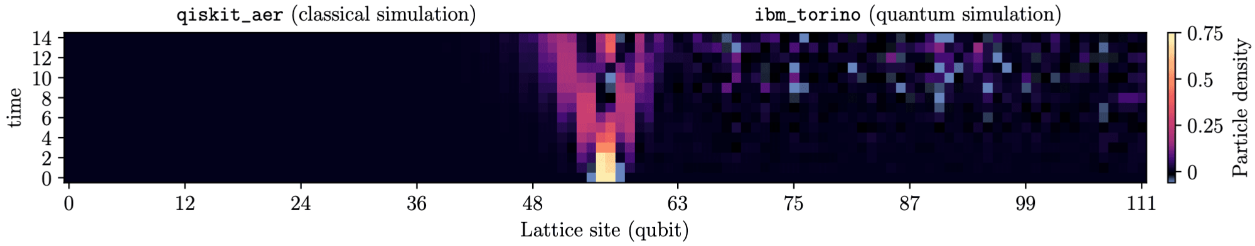
Once we have the initial state, our next step is to apply the time evolution operator. This is a mathematical tool that allows us to take a quantum state as it exists at one point in time and evolve it into the state that corresponds to some future point in time. In our experiment, we use the conventional Trotterized time evolution, where you split up the different mathematical terms representing the Hamiltonian energy equation that describes the quantum system and convert each term into quantum gates in your circuit.

This, however, is where we run into a problem. Even the simplified Schwinger model states that interactions between individual matter particles in our system are “all-to-all.” In other words, every matter particle in the system must interact with every other particle in the system, meaning every qubit in our circuit needs to interact with every other qubit.

This poses a few challenges. For one thing, an all-to-all interaction causes the number of quantum gates required for time evolution to scale quadratically with the simulation volume, making these circuits much too large to run on current quantum hardware. Another key challenge is that, as of today, even the most advanced IBM Quantum™ processor allows only for native interactions between neighboring qubits — so, for example, the fifth qubit in an IBM Quantum Heron processor can technically interact only with qubits 4 and 6. While there are special techniques that let us get around this linear connectivity setting would make it so the required two-qubit gate depth also scales quadratically in the simulation volume.

To get around this problem, we used the emergent phenomenon of confinement — one of the features that the Schwinger model also shares with QCD. Confinement tells us that interactions are significant only over distances around the size of the hadron. This motivated our use of approximate interactions, where the qubits need to interact only with at most next-to-next-to-nearest neighbor qubits, e.g., qubit 5 needs to interact only with qubits 2, 3, 4, 6, and 7. We established a formalism for constructing a systematically improvable interaction and turned that interaction into a sequence of gates that allowed us to perform the time evolution.

Once the time evolution is complete, all we need to do is measure some observable in our final state. In particular, we wanted to see the way our simulated hadron particle propagates on the lattice, so we measured the particle density. At the beginning of the simulation (t=0), the hadron is localized in a specific area. As it evolves forward in time it propagates with a spread that is bounded by the speed of light (a 45º angle).



This figure depicts the results of our simulation of hadron dynamics. The time direction is charted on the lefthand-side Y-axis, and the points on the lattice –qubits 0 to 111— are charted on the X-axis. The colors correspond to the particle density, with higher values (lighter colors) corresponding to having a higher probability of finding a particle at that location. The left-half of this figure shows the results of error-free approximate classical simulation methods, while the right half shows the results obtained from performing simulations on real Quantum hardware (specifically, the `ibm_torino` system). In an error free simulation, the left and right halves would be mirror images of each other. Deviations from this are due to device errors.

Keeping in mind that this is a simplified simulation in one spatial dimension, we can say this behavior mimics what we would expect to see from a hadron propagating through the vacuum, such as the hadrons produced by a device like the Large Hadron Collider.

How Qiskit made our experiments possible

Utility-scale IBM quantum hardware played an essential role in enabling our research. Our experiment used 112 qubits on the IBM Quantum Heron processor `ibm_torino` to run circuits that are impossible to simulate with brute force classical methods. However, equally important was the Qiskit software stack, which provided a number of convenient and powerful tools that were absolutely critical in our simulation experiments.

Quantum hardware is extremely susceptible to errors caused by noise in the surrounding environment. In the future, IBM hopes to develop quantum error correction, a capability that allows quantum computers to correct errors as they appear during quantum computations. For now, however, that capability remains out of reach.

Instead, we rely on quantum error suppression methods to anticipate and avoid the effects of noise, and we use quantum error mitigation post-processing techniques to analyze the quantum computer's noisy outputs and deduce estimates of the noise-free results.

In the past, leveraging these techniques for quantum computation could be enormously difficult, often requiring researchers to hand-code error suppression and error mitigation solutions specifically tailored to both the experiments they wanted to run, and the device they wanted to use. Fortunately, the recent advent of software tools meaningful results out of quantum hardware while taking advantage of built-in error handling capabilities.

In particular, we relied heavily on the Qiskit Runtime `Sampler` primitive, which calculates the probabilities or quasi-probabilities of bitstrings being output by quantum circuits, and makes it easy to compute physical observables like the particle density.

`Sampler` not only simplified the process of collecting these outputs, but also improved their fidelity by automatically inserting an error suppression technique known as dynamical decoupling into our circuits and by automatically applying quantum readout error mitigation to our results.

Obtaining accurate, error-mitigated results required running many variants of our circuits. In total, our experiment involved roughly 154 million "shots" on quantum hardware, and we couldn't have achieved this by running our circuits one by one. Instead, we used Qiskit execution modes, particularly Session mode, to submit circuits to quantum hardware in efficient multi-job workloads. The sequential execution of many circuits meant that the calibration and noise on the device was correlated between runs facilitating our error mitigation methods.

Sending circuits to IBM Quantum hardware while taking advantage of the Sampler primitive and Session mode required just a few lines of code, truly as simple as:

```
from qiskit_ibm_runtime import Sampler, Session, QiskitRuntimeService, Options

# Insert your token
Service = QiskitRuntimeService(token=’”', channel='ibm_quantum')
# Change to the backend of your choice
backend = service.get_backend('ibm_torino')

options = Options()
# Enable dynamical decoupling
options.optimization_level = 2
# Enable readout error mitigation using Matrix-free Measurement Mitigation (M3)
options.resilience_level = 1
```

```
        sampler = Sampler(session=session, options=options)
            sampler.run([is a circuit])
    session.close()
```

Our team did several runs both with and without Qiskit Runtime’s built-in error mitigation, and found that methods offered natively via the `Sampler` primitive significantly improved the quality and accuracy of our results. In addition, the flexibility of Session and Sampler allowed us to add additional, custom layers of error mitigation like Pauli twirling and operator decoherence renormalization. The combination of all these error mitigation techniques enabled us to successfully perform a quantum simulation with 13,858 CNOTs and a CNOT depth of 370!

What is CNOT depth?
CNOT depth is an important measure of the complexity of quantum circuits. A CNOT gate, or “controlled NOT” gate, is a quantum logic gate that takes two qubits as input, and performs a NOT operation that flips the value of the second (“target”) qubit depending on the value of the first (“control”) qubit. CNOT gates are an important building block in many quantum algorithms and are the noisiest gate on current quantum computers. “CNOT depth” of a quantum simulation refers to the number of layers of CNOT gates across the whole device that have to be executed (each layer can have multiple CNOT gates acting on different qubits, but they can be applied at the same time, i.e., in parallel). Without the use of quantum error handling techniques like those offered by the Qiskit software stack, reaching a CNOT depth of 370 would be impossible.

Looking ahead to the future of quantum simulation technology

Over the course of two research papers, we have demonstrated techniques for using utility-scale quantum hardware to simulate the quantum vacuum, and to simulate the dynamics of a beam of particles on top of that vacuum. Our research group is already hard at work on the logical next step in this progression — simulating collisions

If we can simulate these collisions at high enough energy, we believe we can demonstrate the long-sought goal of quantum computational advantage. Today, no classical computing method is capable of accurately simulating the collision of two particles at the energies we've set our sights on, even using simplified physics theories like the Schwinger model. However, our research so far indicates that this task could be within reach for near-term utility-scale quantum hardware. This means that, even without achieving full quantum error correction, we may soon be able to use quantum hardware to build simulations of systems of fundamental particles that were previously impossible, and use those simulations to seek answers to some of the most enduring mysteries in all of physics.

At the same time, IBM hasn't given up hope for quantum error correction, and neither have we. Indeed, we've poured tremendous effort into ensuring that the techniques we've developed in our research are scalable, such that we can transition them from the noisy, utility-scale processors we have today to the hypothetical error-corrected processors of the future. If achieved, the ability to perform error correction in quantum computations will make quantum computers considerably more powerful, and open the door to rich, three-dimensional simulations of incredibly complex physics processes. With those capabilities at our fingertips, who knows what we'll discover?

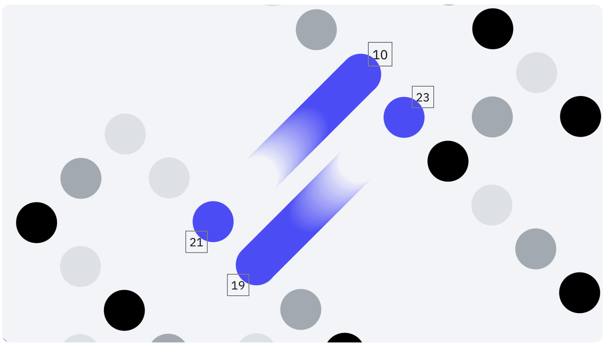
References

1. Farrell, R. C., Illa, M., Ciavarella, A. N., & Savage, M. J. (2024, April 18). *Scalable Circuits for Preparing Ground States on Digital Quantum Computers: The Schwinger Model Vacuum on 100 Qubits.* PRX Quantum, https://journals.aps.org/prxquantum/abstract/10.1103/PRXQuantum.5.020315 | ↩

2. Farrell, R. C., Illa, M., Ciavarella, A. N., & Savage, M. J. (2024, June 10). *Quantum simulations of hadron dynamics in the Schwinger model using 112 qubits.* Physical Review D, https://journals.aps.org/prd/abstract/10.1103/PhysRevD.109.114510 | ↩

Keep exploring

View all blogs →



Dynamic circuits enable essential circuit cutting methods for quantum-centric...

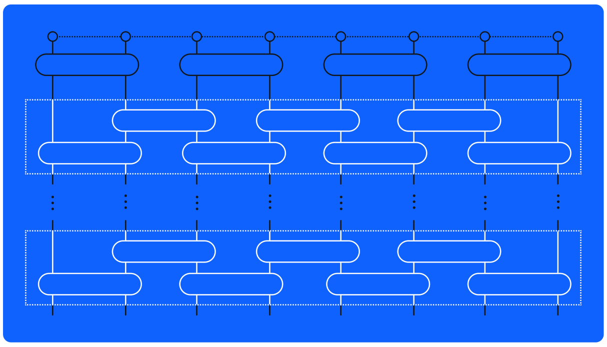
11 Dec 2024 • Daniel J. Egger, Maika Takita, Robert Davis

Error Correction & Mitigation | Research | Systems

IBM Quantum delivers on performance challenge made two years ago

13 Nov 2024 • Jay Gambetta, Ryan Mandelbaum

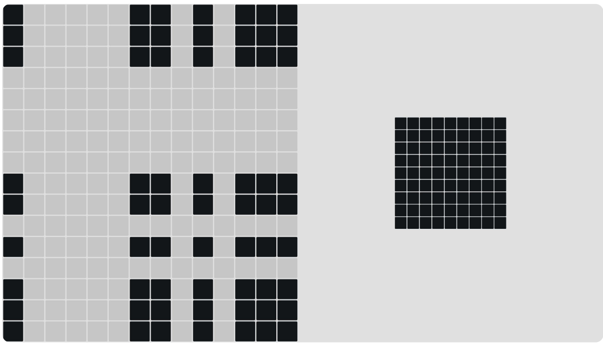
Community | Error Correction & Mitigation | Network | Software | Systems | Qiskit



Advancing quantum algorithms for large-scale simulations of quantum spin chains

30 Oct 2024 • Simone Cantori, Marcel Pfaffhauser, Fabio Scafirimuto, Robert Davis

Error Correction & Mitigation | Research



Exploring measurement error mitigation with the Mthree Qiskit extension

3 Apr 2024 • Hwajung Kang