IBM **Quantum**

The Coherence Times

A Quarterly Readout

Cooking up algorithms

Our quarterly leadership update

A note from Jay

Hi all, and welcome to the Q3 2025 edition of the Coherence Times: Cooking up algorithms.

This year, we demonstrated what it will take to build a fault-tolerant quantum computer. We also developed a framework for realizing quantum advantage. The ingredients are prepared—now, you need the equipment required to cook up quantum advantages themselves using IBM Quantum systems.

How? First, our team is hard at work designing and benchmarking the circuits that will deliver the first quantum advantages, relying on techniques like peaked circuits, quantum diagonalization, and error mitigation. We're also developing the algorithms that will make use of these circuits. Earlier this year, we announced a new algorithms and applications mission at IBM Research focused on finding algorithms for areas that hold promise for quantum—differential equations, optimization, Hamiltonian simulation, machine learning, and more.

Yet, equally important is the work of you, our community members, who must be able to leverage the outputs of these teams. Herein lies the continued importance of Qiskit, our open-source software development kit—Qiskit is how you will use and build on these algorithms, and ultimately incorporate quantum into their workflows.

So, as we continue to push forward, I want you to remember these three things. First, Qiskit is open for quantum information science, and we will continue to support our community working at the bleeding edge of quantum theory. Second, Qiskit is open for high-performance computing, and we must ensure that both classical and quantum developers can employ workload management tools to realize quantum-centric supercomputing. Third, Qiskit is open for algorithm discovery, and we must ensure our domain experts have the tools available to continue pushing the field forward. As we approach our annual Quantum Developer Conference, we must ensure that our users in these areas feel supported and empowered to develop, innovate, and ultimately to bring useful quantum computing to the world.

To that end, I hope you all continue contributing to this vibrant quantum ecosystem. We will continue celebrating your work, and helping to grow this community so we can all realize useful quantum computing together.

Thank you, as always, for your hard work.

Qiskit is open for quantum information science

Evolving the primitives to support customizable error mitigation

Since introducing Qiskit Runtime primitives in 2022, we've aimed to provide users with the tools they need to exploit quantum algorithms at speed and scale. But as algorithms and applications have evolved, so has the functionality of the primitives. In 2023, we brought parameterized compilation to Qiskit Runtime, and this year we have enabled boxes and annotations with Qiskit 2.0 and 2.1, respectively. These releases have given users more control over the circuit compilation process and noise model definition while also paving the way for an exciting new interface—a primitive that enables customizable error mitigation.

Designed for quantum information scientists looking to build new capabilities while still leveraging the efficiencies afforded by Qiskit Runtime, this next-generation "executor" primitive provides extended flexibility in circuit layering and twirling for enhanced error suppression and mitigation. With this primitive, users can build atop, or mix and match among, existing error mitigation techniques to advance quantum computing capabilities.

The key element that makes this possible is the **samplex**, a graph describing how a user intends the transformation and randomization for sampling to be handled. You can think of the samplex like a recipe that a user can create, either by modifying an existing recipe or creating one from scratch. Annotations are the cooking steps listed in the recipe, and the new primitive is like a professional kitchen that can efficiently "cook the meal" by following the recipe. So why do we need a "recipe" to begin with? Take, for example, Pauli twirling—a common technique for regulating noise that is used in

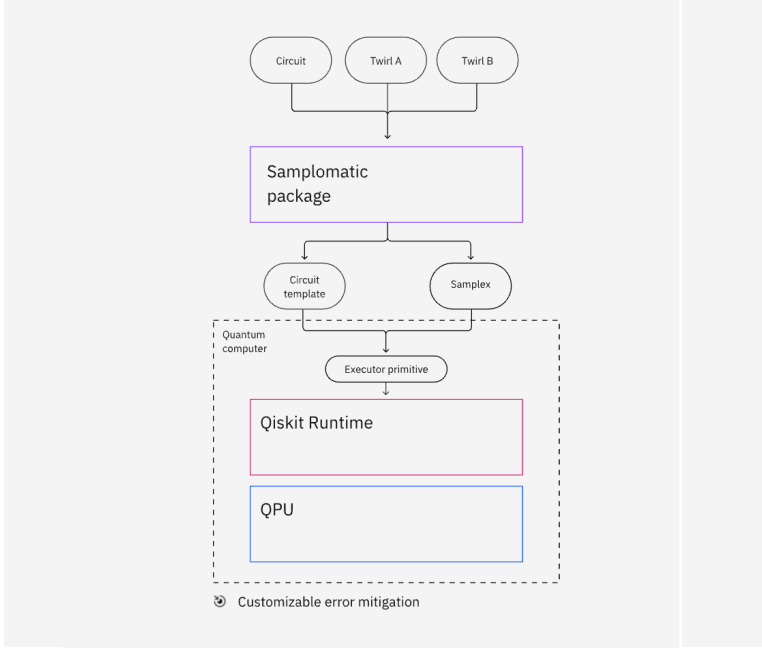
Qiskit is open for quantum information science

many error mitigation protocols. Today, users can implement Pauli twirling on the client side; however, it is highly inefficient as it requires sending a large number of parameters to the server. With samplex, users can provide the directives without generating and sending parameters, giving them the control, transparency, and efficiency required to do foundational work in quantum computing.

For now, users must first install the **Samplomatic** package from GitHub, though it will soon be included in Qiskit. This open-source Python package contains the build method that generates the samplex along with a circuit template, both of which become inputs for the "executor" primitive. Samplomatic currently supports Pauli twirling; upcoming milestones include measurement twirling, dynamic circuit twirling, noise injection, and more.

Essentially, we are restructuring the primitives to make the runtime leaner. Sampler and Estimator will still offer the same user experience as they did before, but they will internally create a samplex to be consumed by the runtime without any need for user input. These primitives are basically out-of-the-box instances of a new highly customizable primitive framework based around the samplex. We are evolving the level of control available to users while ensuring user experience and workflows don't change.

The Qiskit Runtime primitives will continue to feature prepackaged error mitigation techniques for users who want to take advantage of cutting-edge capabilities at an abstracted level. But for users seeking fine-grained control of modular error mitigation tools, Samplomatic and the new samplex-accepting primitive facilitate the design of advanced error mitigation methods that may bring quantum advantage closer.



Qiskit is open for high-performance computing

With Qiskit 2.2, the C API enables full workflows outside of Python and unlocks new possibilities for integrations

You've heard it before: The future of advanced computing is quantum-centric supercomputing, a heterogeneous compute architecture that brings together QPUs, CPUs, and GPUs to solve problems better than quantum and classical alone. Ushering in this future will bring quantum advantage closer. But it also requires a software stack that supports hybrid compute models, plus the programming languages best suited for them.

That's why we've focused our Qiskit 2.x release series around building interfaces for other programming languages via a C API. With last year's Qiskit 1.x series, we rewrote most of Qiskit's core data model in Rust, improving runtime over 60x and enhancing the memory overhead. Not only did the transition create performance gains, but it also enabled the development of a C API for Qiskit. That's because Rust has native support for building C foreign function interfaces (FFIs), which expose an interface for consumption from compiled languages like C.

Over the course of the Qiskit 2.x release series, we have steadily increased the functionality of the C API. Qiskit 2.0 could create an observable with C, Qiskit 2.1 could run a circuit with standard gates and generate an internal QPU model, and now, Qiskit 2.2 can compile circuits for the target hardware and even customize certain transpiler passes. C users have progressively gained the ability to build and optimize Qiskit circuits, speeding up their circuit construction and execution by a factor of at least 10. Ultimately, this means lower development overhead, a consistent development ecosystem, and the ability to avoid context switching.
In short, HPC users who don't typically program in Python can finally call to the Qiskit library from their programming language of choice, without writing any Python code.

Not only does Qiskit 2.2 mark a major milestone in the effort to make quantum native to HPC environments, but it also unlocks the potential to expand the software ecosystem built on top of Qiskit. If there is existing software that might need to engage quantum in some way, the C API facilitates its integration with Qiskit—regardless of the language in which it was written. We look forward to seeing what our users build with Qiskit in the programming language of their choice.

Benchmarking for advantage

Pushing the performance of our quantum computers requires benchmarks that let us efficiently monitor improvements and compare hardware. Recently, our team has published important results with implications for how we, and the broader industry, validate the quality of the circuits that quantum hardware can run.

A frequent question asked of IBM is why we don't use random circuit sampling (RCS) to benchmark our systems—in essence, why we don't run Google's quantum supremacy experiment. This experiment generates circuits with a random selection of gates and samples from the outputs to create a representative distribution.

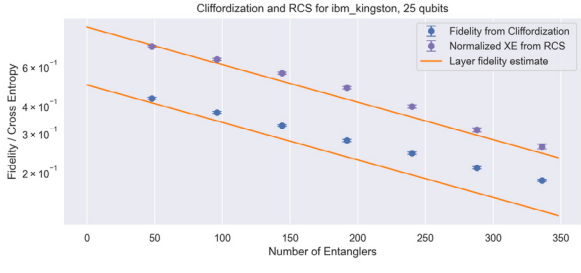
The first answer is that random circuit sampling is not verifiable. We have no way to compare the output distribution against the ideal distribution, because the circuits are beyond the scale that classical computers can efficiently simulate. However, more recent work from IBM in collaboration with researchers at Sandia National Laboratories **has demonstrated** that we can glean the same benefits from benchmarking with RCS using a set of simpler circuits that can be efficiently simulated classically, called Clifford circuits.

This work shows that the fidelity of these Clifford circuits accurately predicts the performance of RCS circuits provided that they have the same pattern of entangling gates. In these cases, the only difference between Clifford and non-Clifford circuits are in one of the gates, called the virtual Z gates—where we can simply tweak the angle of the rotation in the software. In short, we can "Cliffordize" RCS circuits to calculate the fidelities of large circuits without the immense classical overhead of RCS.

Cliffordized circuits let us extract the actual information we need, our system's ability to run quantum circuits. We measured fidelities from Cliffordized RCS circuits, as well as the error per layered gate (EPLG) and the RCS cross-entropy, on up to 65 qubits of our Heron device ibm_kingston. The three metrics are consistent in the range where cross-entropy can still be calculated classically (~25 qubits). More importantly, Cliffordization and EPLG can be pushed to scales needed to benchmark for quantum advantage—while still being easily verifiable with HPC.

We also compared our results with Google's inferred RCS values for 67 qubits (arXiv:2304.11119) as a function of entangling gate counts (figure 2). Fitting Google's data predicts an EPLG of 5.6e-3, which is similar to our largest devices. Our latest ibm_pittsburgh is about twice as good on this metric.

What implications does this have for our benchmarking efforts? Well, Cliffordization (and, in turn, RCS) are well-predicted by a much simpler experiment that also returns finer-grained detail: the error per layered gate (EPLG) experiment that we've long used to benchmark our systems and reported on the platform, which also gives us the discrete error rates of each gate. Therefore, while we can't say that our benchmarking circuits allow us to calculate things a septillion times faster than classical, our benchmarking efforts are perfect for our needs—and together will help us achieve the scale, quality, and speed required to hit quantum advantage.



This graph compares fidelities of three different benchmarking techniques: Cliffordization, Random Circuit Sampling, and layer fidelity. The y-axis means different things for the different techniques, so what's most important is the slope. Having the same slope means that the techniques serve as equivalent benchmarks.

Algorithms and applications

Improving hardware is only part of the story when it comes to improving our quantum capabilities. We must also develop new uses for this hardware. While classical computing is approaching fundamental scaling limits (e.g., Moore's law), quantum computing is still in its emergent era. We feel confident that there are undiscovered algorithms that can lead to valuable, disruptive, and hopefully life-bettering applications. The classical analogues are algorithms like the now-ubiquitous fast Fourier transform and hashing, which enabled real-time signal processing and search.

And unlike the emergent classical era, we now have powerful computing technology in the form of classical HPC and graphics processing units (GPUs) to support quantum algorithm development.

IBM Research now runs a mathematics of computation division tasked with uncovering new algorithms and maturing them into applications—especially those making use of classical, quantum, and AI together. This division will focus on spaces where quantum-centric supercomputing demonstrates promise:

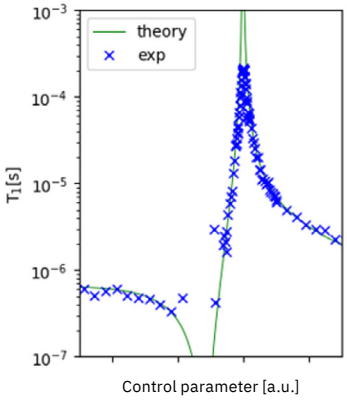
- **Differential equations.** Differential equation appear across the physical sciences, but traditional solvers can require the world's largest supercomputers. New research is beginning to demonstrate high potential for quantum speedups in this space.
- **Optimization**. Many of the most pressing business problems today rely on finding optimal solutions to some problem. AI can have an important role in exploiting problem structure, while quantum circuits could provide further gains by exploiting the mathematics of quantum. A working group of IBM and non-IBM experts are actively benchmarking quantum against leading classical optimization algorithms seeking speedups.
- **Linear algebra.** Linear algebra is the common language of science; AI-driven algorithm discovery can reduce computations, while faster factorization techniques and quantum-enhanced eigensolvers lift the ceiling on model size and precision. This could help with some of the most promising quantum applications, those that aid in the simulation of Hamiltonians—the equations that govern the quantum state of a system.
- **Quantum machine learning.** New techniques use quantum mathematics to seek scalable speedups over existing classical-only ML algorithms or schemes for stochastic processes. This includes quantum kernels, which find structures in data only visible to quantum algorithms, and encoding data into the ground states of Hamiltonians.

Though the division itself is new, its mission is not—IBM Research has an 80-year history of developing new algorithms. Expect a drumbeat of blog and social posts highlighting this work over the next few months.

From Representation to Revolution: Celebrating the FFT and the Future of Computing

Opening a new era in algorithmic development

Controlling leakage



This graph charts T1—how long it takes for a qubit to reset to its ground state—against the parameter used to control the gadget. You can see that when the parameter is at 0 (turned off), T1 is at the microsecond level, as required for quantum computation. To the left, the gadget is turned on and the T1 quickly drops to just a few hundred nanoseconds.

We compute in binary code—strings of 0s and 1s—and the same is true for quantum computing. Quantum computers accept inputs as strings of binary digits, perform quantum operations, and then output strings of binary digits. But unlike classical computers that rely on innately two-state systems to represent 0 and 1, qubits use just the lowest two energy levels of a quantum system to represent 0 and 1. Excess energy, such as from noise or from readout, can sometimes send a qubit into the 2, 3, or even higher states.

Qubits entering states beyond the two used for computation is called "leakage," an established challenge we must overcome. Once a qubit has entered these higher states, it's no longer available for computation until it "cools down" into the lower states. This can take milliseconds—but we don't have time to wait so long, since we need to run computations as quickly as possible. This is especially true as we introduce dynamic circuits and error correction—error correction requires many readouts and qubit resets, which becomes untenable if we must frequently wait for the qubit to cool down, since other qubits can decay during this cool-down period.

Now, the team has created a gadget that can reset qubits in leaked states back down to zero in just a few hundred nanoseconds or even less. Essentially, a switching element attached to each qubit on the chip allows us to selectively protect the qubit or couple it to a dissipative material. When the switch is off, the qubit is protected from the environment and has a long T1/coherence time. When turned on, the element acts like a refrigerator, sucking the extra energy out of the qubit and sending it back to its ground state on demand.

"This will be a big improvement in terms of both quality and speed of the reset operation," said Andreas Fuhrer-Janett, senior research staff member based at of IBM Research, Zurich, "And it will be very important for error correction circuits."

The team has already run experimental demonstrations of the gadget on a handful of qubits. Their work shows that when the gadget is turned off, the connected qubits maintain coherence times at hundreds of microseconds—comparable to what we would expect of these devices in isolation. When the gadget is turned on, the qubits return to the 0 state in tens of nanoseconds.

Now that we have a functioning proof-of-concept, the team hopes to implement the gadget at scale. The device will be added to both IBM Quantum Loon and the second generation of IBM Quantum Nighthawk. We expect to see a significant decrease in the amount of leakage, and therefore an improvement in the speed of our chips and our ability to run dynamic circuits.

Quantum advantage white paper

What it is: IBM and quantum startup Pasqal wrote a white paper laying out a clear definition of quantum advantage and outlining strategies for achieving it in the near term and for validating advantage claims. They define advantage as the ability to execute a task on a quantum computer in a way that can be scientifically validated, and that can be performed with a "quantum separation" demonstrating improved efficiency, cost-effectiveness, or accuracy over classical methods. The paper highlights sampling problems, variational problems, and calculations of expectation values of observables as promising candidates for near-term advantage demonstrations.

Why it matters: As more research organizations begin exploring interesting scientific questions with quantum computers, we're beginning to see some of the first credible claims of quantum advantage emerging. However, "claiming" quantum advantage and "proving" quantum advantage are two different tasks. As we approach advantage, the research community will need to verify these claims, as outlined in the IBM-Pasqal paper.

BasQ explores fundamental physics at utility scale

What it is: Basque Quantum (BasQ) and IBM have teamed up on a new paper running simulations to help study the strong force, a fundamental force in the Standard Model of particle physics, with utility-scale quantum computation. Their experiments represent some of the largest quantum computations ever performed, with simulations running on up to 144 qubits at quantum circuit depths of up to 7872 two-qubit gates. They validate their experiments on classical computers using sophisticated tensor networks to show that the quantum simulations were producing good results.

Why it matters: The strong force binds matter together at the very smallest scales, shaping quarks into the protons and neutrons that make up atomic nuclei. However, there is much that we still don't understand about the complexities of the strong interaction, and the way it behaved in the earliest moments of the universe following the Big Bang. The most powerful classical supercomputers in the world are unable to model that complexity, and real-world experiments conducted in particle colliders can cost billions. This research shows that quantum computers could enable explorations of fundamental physics beyond what is achievable with classical methods.

RIKEN gets first IBM Quantum System Two outside U.S.

What it is: IBM and RIKEN, the national research laboratory based in Japan, have unveiled the first IBM Quantum System Two that is neither in the U.S. nor in an IBM Quantum Data Center. The new device is powered by a 156-qubit IBM Quantum Heron, and is also co-located with Fugaku, one of the most powerful classical high-performance computing (HPC) systems in the world.

Why it matters: At the end of 2024, VP of IBM Quantum Jay Gambetta predicted that quantum advantage would occur within the next two years—but only if the quantum and classical HPC communities worked together. Now, with the most advanced IBM quantum computer co-located alongside one of the world's most famous classical supercomputers, researchers from RIKEN and IBM can explore practical quantum + HPC hybrid workflows that push the limits of the possible.

IBM and AMD tackle quantum-centric supercomputing

What it is: IBM and global technology company AMD have announced plans to develop cutting-edge computing architectures combining quantum computers with classical high-performance computing systems. Together, they aim to explore integrations of AMD CPUs, GPUs, and more with IBM quantum computers to help accelerate new hybrid quantum + HPC algorithms that are showing promise for near-term demonstrations of quantum advantage.

Why it matters: This new collaboration is another example of how IBM is teaming up with the classical HPC community to enable quantum advantage in the near future. AMD's extensive HPC expertise will prove invaluable in helping to refine and optimize the quantum-centric supercomputing architectures being explored by partners like RIKEN and Oak Ridge National Laboratory. The partnership could also play a role in enabling the longer-term vision of IBM for large-scale, fault-tolerant quantum computing.

HSBC shows progress in quantum finance applications

What it is: New research from global banking institution HSBC highlights the potential for quantum computers to deliver value for real-world problems in algorithmic bond trading. Working in collaboration with researchers from IBM, the HSBC team explored the use of quantum computers to optimize "Requests for Quote," a process bond traders use to estimate the likelihood of a trade being filled at a quoted price. Their experiments yielded an up to 34% improvement in predictions for real, production-scale trading data over common classical methods.

Why it matters: While much more work is needed to yield practical advantages for quantum finance applications, this work provides evidence of the potential value that today's quantum computers may bring to the financial services sector. It's important to note that this work has not been reproduced on quantum simulators yet, and we're still working to understand the mechanism behind the observed speedup.

Revamping the Qiskit advocate program

What it is: Founded in 2019, the Qiskit advocate program is a global community initiative that aims to recognize and support those making meaningful contributions to the Qiskit ecosystem. Now, we're introducing a revamped version of the program to deliver new resources and pathways for the next generation of quantum community leaders.

Why it matters: Version 2.0 of the program introduces a strategic, tiered structure that provides advocates with greater benefits and a clearer path of progression through the advocate journey. In addition to getting access to mentorship, education, and networking opportunities, advocates who progress to higher tiers will enjoy early access to new quantum tools and resources and bonus QPU time. Learn more on the IBM Quantum blog.

Qiskit developer certification v2.x

What it is: The Qiskit developer certification is an exam that tests your understanding of quantum computation and your ability to use Qiskit. Since the certification first launched in 2021, more than 1,300 Qiskitters across 71 countries have successfully passed the exam to become IBM-certified quantum computational scientists. Now, we're updating it for the era of Qiskit SDK v2.x and beyond.

Why it matters: The new v2.x certification is a valuable tool for anyone looking to boost their quantum credentials and demonstrate proficiency as a Qiskit developer. IBM certifications are widely respected, and Qiskit certification can make a real impact in advancing one's career in quantum computation. Qiskit certification also brings special benefits within the Qiskit community itself. For example, it serves as a requirement for progression to higher tiers of the new Qiskit advocate program.

Qiskit SDK v2.2

What it is: IBM has rolled out a new minor release in the Qiskit SDK v2.x series. The new release brings several new capabilities and performance improvements, the most notable being the transpiler is now available in Qiskit's C API.

Why it matters: Qiskit was originally built using the Python programming language, but with the release of Qiskit v2.0, we began building out a new C API, which makes it possible to bind Qiskit natively to programming languages that are more commonly used by the classical HPC community. The C transpiler allows us to complete step 2 of Qiskit patterns—optimize for target hardware. With this in place, it is now possible to build a full end-to-end quantum workflow in C that does not touch the Python programming language at all. This is an essential step in realizing the full potential of quantum + HPC workloads.

Qiskit Fall Fest 2025

What it is: Qiskit Fall Fest returns this October-November for its fifth annual installment. Visit the Fall Fest website to learn more about how you can participate in upcoming events in your area.

Why it matters: Each year, Qiskit Fall Fest brings the quantum community together to celebrate quantum technology, research, and collaboration through events held on university campuses all around the world. The primarily student-led events include quantum challenges, hackathons, coding competitions, workshops, and more—and offer a great way to connect with your local quantum community.

Coherence Times podcast

What it is: IBM has launched a brand-new quantum-themed podcast hosted by IBM Quantum editor-in-chief Ryan Mandelbaum. In each episode, Ryan takes listeners on a deep dive into the world of quantum computing through interviews with some of the industry's leading experts.

Why it matters: Whether you're an experienced quantum computational scientist or just a curious observer, the Coherence Times podcast is a great way to get an inside view of the latest developments in quantum hardware, software, algorithms research, and more. Subscribe wherever you listen to podcasts or check it out on IBM Research YouTube.

What's new at IBM Quantum

What it is: IBM Quantum has launched a new content series covering the latest developments in quantum capabilities and research and providing readers with direct links to code, documentation, and experiments to help get started putting these new innovations to use right away. This will live first as blog posts, then as a regular web update and newsletter.

Why it matters: It's been a remarkably productive year here at IBM Quantum, and with so many developments happening so quickly, it can be hard to keep up with all the latest. The "What's new at IBM Quantum" roundup gives us a new way to keep our users and partners updated.

IBM wins IEEE Synergy Award

What it is: IBM was named among the winners of the 2025 IEEE Quantum Technical Community (QTC) Distinguished Synergy Award, with the IEEE Quantum Technical Community citing our "decades of leadership in advancing quantum computing from theory to global practice, fostering a diverse and collaborative ecosystem that accelerates innovation, commercialization, and societal impact worldwide."

Why it matters: Each year, the QTC Distinguished Synergy Award is presented to organizations for their outstanding contributions to and effective partnerships with the international quantum computing community. This recognition is a timely reminder of the tremendous impact IBM is making in the quantum community through our collaborations with research organizations around the globe.

Quantum Networking

Interview with Yuki Hayashino

Senior Expert leading IBM Application Research Project at Tokyo Electron Limited

What is Tokyo Electron's strategy for quantum computing?

TEL is a semiconductor production equipment company based in Tokyo, Japan.

Quantum computing right now is a dynamically changing environment. We see a great deal of morphing and changing in the coming years. We are getting closer to actual social adoption of the technology, and the time scales are becoming more realistic.

This is our strategy: We will leverage next-generation computing technologies–including hybrid computing that combines quantum computing, AI, and HPC. This will enhance the performance and productivity of our business activities and models. Additionally, we will create a high-quality ecosystem to support the adoption of these advanced computing technologies, and enhance their value. We aren't limited in our approach to quantum computing. We are exploring applications across multiple fields, including chemistry, optimization, and simulation.

What is your role at TEL?

I lead the practical development of quantum computing applications. Specifically, I assess how quantum computing can address current and future business challenges. I also identify barriers in developing and implementing next-generation computing technologies and work to find optimal solutions.

We work with domain experts, people who have the frontline technical expertise. It's important for them to collaborate with the data scientists and quantum experts in a very cohesive environment to assess problems holistically in a hands-on way and find solutions using quantum computing.

How did you find and pursue this career?

To explain the background, I need to go back to my university days. When I was an undergraduate student, a quantum chemistry professor said "the age of quantum will come in the future." He encouraged students to join his research group. I considered

his invitation, but I decided to join another group. However, I couldn't forget what he said.

Later, when I took my doctoral qualification exam in graduate school, my supervisor mentioned that my research might be useful for future quantum technology and quantum computing. That was the first time I heard the term "quantum computing." At that time, in 2002, it was predicted that quantum computers would not become a reality for another 70 to 80 years. So, I thought it was uncertain whether they would appear in my lifetime.

After that, I kept thinking that the quantum era was not coming for many years, and that quantum computing was an extremely difficult technology to realize. That's why I clearly remember when IBM unveiled its quantum computer in 2016. I was amazed and inspired, and from then on, I wanted to learn more about quantum computing technology.

In March 2020, due to the spread of COVID-19, my workplace instructed me to work from home. Until then, I had been traveling overseas often for my job. The change was hard, but without the sleep deprivation of business trips my health improved. I wanted to use this time productively, with a clear mind. I volunteered to start researching about quantum computing technologies. I became completely absorbed. However, it was difficult to clearly identify quantum technology's business value. Additionally, Japanese companies–especially in the manufacturing industry–have been criticized for being technologically advanced but lacking business insight. I have been seeking ways to excel in technology and business.

I am an optimistic person. Reflecting on my past as a technologist, I decided to enroll in business school to pursue a path in management. I wanted to become someone who could lead in technology and business, while also developing the skills to pursue aggressive management strategies.

I am a complete novice when it comes to quantum computing. However, my strong desire to pursue the truth has led me to take on work related to this cutting-edge technology at my company. Currently, our joint team from IBM Quantum and TEL is tackling several challenging issues in semiconductor manufacturing equipment. Although it is still unclear whether quantum computing will be successful in solving these problems, the team maintains a positive and upbeat atmosphere. My role is to support and guide the team in the right direction whenever needed.

What advice do you have for others getting into the field or looking to build careers in quantum computing?

My advice is very simple: As a development expert in the industry, I'd like you to imagine what your future could be like. I guess the answer will probably come to you naturally.

Q&A: Sarah Sheldon

How would you describe your role?

I lead our applied quantum science team. Our team is focused on what we are going to do with quantum computers from the near term into early fault tolerance. That includes everything from algorithm development to working with our partners on bringing algorithms to the applications that they have. And so my role is setting an agenda for this team and figuring out how to put it into action working with the technical leads.

What does that look like on a day-to-day basis?

I spend more time talking than I thought I would going into research! I spend a lot of time with the technical leads and first-line managers on my team and also coordinating with the other teams within quantum and IBM Research as well as our external partners.

We're situated in the middle of a lot of things, right? We use the hardware, we use the software, we work with clients, universities, industry partners, all of those things. We intersect with a lot of other missions. A lot of my role is coordinating with those other teams and giving our point of view—and also making sure we're aligned, and that we're all working together as best we can.

How did you end up in this role?

My PhD was on quantum control in nuclear magnetic resonance systems, which had a lot of overlap with what we do at IBM in terms of the control techniques we use. But the systems were very different.

I started at IBM right after grad school as a post doc, and I started on the hardware side. Initially my role was working on gate calibrations, trying to get higher-fidelity gates. And then that kept progressing further and further into the theory side. So gate calibration turned into more work on verification and validation techniques–how we know that we're performing the right operations that we're trying to perform, and that we're getting the right results out of our systems. That moved more into large scale, multi-qubit demonstrations. We were trying to see how to characterize things at a larger scale, and then, how we actually do something useful and get more out of the systems. That's when I started also managing more of the theory side.

Having a really tight connection between the algorithms and applications on one side and the capabilities of our systems on the other side felt really natural. So in my last role, I was managing our theory and capabilities team, which is really focused on error mitigation techniques as well as utility-scale demonstrations.

There's still a part of our org that's focused on the error mitigation techniques. But my role has shifted

Q&A: Sarah Sheldon

more into how will we use the systems for real applications, and that's expanded into partnership work as well as the algorithms focus.

How do you think about management?

The analogies that I relate to the most are from gardening. Gardening is a big hobby for me, and I connect with a lot of analogies about having people in the right place for them to grow without needing a lot of additional input. Everyone needs their own individualized approach and not everyone's the same. But my role as a manager of a research team that works very well independently, especially on the theory side, is just setting high-level direction, goals that people can align to.
We're all here to to do something together and get to useful quantum computing. So I think even for researchers, who can be very independent and who are very technically strong and don't need a lot of guidance all the time, still having that alignment as a team and good communication helps people to feel like they're contributing and that we're working together.

What excites you about quantum computing right now?

The closeness of quantum advantage is really exciting. I have been at IBM for 12 years, and in quantum for my PhD before that. Seeing things be so close, and the possibilities beyond quantum advantage, it's a point that we haven't been at before.

Also, I think we have a lot of ideas of what we're going to be doing in the next couple of years. But I think we'll also look back in five years and have a bunch of surprises. We're going to see things happen that we're not even imagining today. Being on the edge of that possibility is really exciting.

You may have partly answered this already, but what do you do outside of work?

Yeah, gardening. Although I will say I had a lot of travel this summer, which made gardening harder to keep up with, but I volunteer with a group that plants pollinator-friendly gardens around my village. I do like being outdoors in general. I did a trip to New Zealand earlier this year, where I did one of the "Great Walks" with some friends, and got to explore a lot of local hiking in New York getting ready for that. There's also a lot of days that I'm at my sister's house watching reality TV —I don't know if that's Coherence Times worthy though.

Phase Kickback

Quantum advantage is approaching fast, and many believe its first demonstrations will emerge from computational chemistry. That belief isn't just about clever theory; it reflects a long history of chemists pioneering new computing hardware as it emerged. From analog integrators to mainframes, from vector supercomputers to today's quantum-centric systems—every time a new compute architecture became available and reliable, a new wave of methods moved from theory to practice. In hindsight, computational chemistry's history reads like a dialogue between algorithms designed to enable new kinds of scientific inquiry and the machines that could finally run them.

The story starts in 1928, when Douglas Hartree introduced the self-consistent field (SCF) method, the first ab initio or "from first principles" framework for many-electron systems. Less remembered is that Hartree didn't just propose an algorithm; he built one, running SCF cycles on a differential analyzer assembled from Meccano parts—an analog integrator whose gears and shafts made self-consistency feasible before stored-program computers existed. Vladimir Fock's 1930 refinement, introducing a mathematical construct known as Slater determinants to enforce antisymmetry and exchange, pushed the formalism forward. By the early 1950s in Cambridge, S. F. Boys and colleagues were pushing configuration-interaction and basis-set work on EDSAC, one of the first stored-program machines in regular service. This was an early signal that quantum mechanics was becoming software.

Clemens Roothaan's 1951 recasting of Hartree–Fock (HF) as a matrix eigenvalue problem did more than present tidy theory; it aligned electronic structure with what computers at that time did well: linear algebra. That shift made HF programmable and, crucially, scalable across the growing university computing centers and IBM mainframes that turned eigenproblems into routine service calls. The release of the general-purpose chemistry software Gaussian 70 then packaged ab initio workflows for a much wider community. Even the field's vocabulary reflects the machine room: the dominance of Gaussian-type orbitals over Slater-type orbitals was a hardware-shaped choice, exploiting the fact that products of Gaussians are Gaussians, meaning that two-electron integrals collapse to closed forms that 1960s/70s CPUs could evaluate quickly and within tight memory budgets.

In addition to shaping the foundations of computational chemistry software, HF would also eventually inspire numerous "post-Hartree-Fock" methods, many of which are direct ancestors of important quantum computing methods we use today. The original HF formulation is based on an approximation that neglects the detailed interactions between electrons in quantum systems—also known as electron correlation. Post-HF methods were developed to approximate this correlation for more accurate calculations through a wide variety of approximation schemes. Examples of this include the coupled-cluster method, which would eventually serve as the basis of many important variational

Phase Kickback

quantum circuits, and density-matrix renormalization, which was based on what's known as a "tensor network approximation"—now a common technique for simulating quantum circuits on classical computers.

The story of vector machines, early supercomputers capable of operating on entire arrays (or "vectors") of numbers at once, is just as much a story of post Hartree–Fock methods. As vector pipelines, wide registers, and fast memory systems matured, integral builds and tensor contractions—which are the heavy kernels behind methods like MP2, CI, and coupled cluster—were reorganized to take full advantage of basic linear algebra kernels for matrix operations that were running at peak-performance on Cray- and IBM- vector architectures. What had been algorithmic curiosities became production tools: a series of methods allowed researchers to cut memory traffic, transformed two-electron integrals flowed through long contiguous loops, and transformed once heroic calculations into benchmarks for vectorized dense linear algebra. The same hardware logic nudged density matrix renormalization toward tensor network formulations with batched contractions, foreshadowing today's simulators of quantum circuits. The matching of electron correlation theory to the grain of the machine architecture, including longer vectors, better cache line use, and even more aggressive fusion of operations, is a constant motif that runs from HF-era kernels on classic vector supercomputers into modern many-vector descendants like GPUs and wide SIMD CPUs.

A different family of density-based methods would come into focus in 1964 when Hohenberg and Kohn showed that the ground-state density is a sufficient variable. Around the same time, Aneesur Rahman's molecular-dynamics simulation of 864 liquid argon atoms, step by step on a CDC 3600 mainframe, demonstrated how computational modeling could open entirely new questions in condensed-phase physics and chemistry. The vector machines era helped these ideas spread far and wide. Car and Parrinello's 1985 unification of DFT forces with molecular dynamics turned first-principles trajectories into a practical tool, and the watershed year 1988 brought Becke's exchange and the LYP correlation functional, which together lifted DFT accuracy and reach to larger systems. Just as important, the hardware matured. Vector supercomputers and early parallel systems accelerated plane-wave FFTs and dense linear algebra, and the IBM System/370 vector architecture and the 3090/ICAP experience enabled the development of scalable scientific computing for chemistry and beyond. Within a few years, large-scale Car–Parrinello implementations were being tuned and compared across multiprocessors like the IBM Blue Gene-class and Cray systems, making once-heroic simulations routine and redefining what "standard practice" meant for timescales and system sizes.

A third wave began when quantum information met chemistry. Feynman's argument that unitary time evolution, native to quantum mechanics, should be efficiently emulable on a quantum device provided the philosophical spark. A 2005 paper showed how to use time-evolution unitaries and phase

Phase Kickback

estimation to target molecular energies with polynomial scaling on a quantum computer, translating Hamiltonian structure into circuits. A decade later, the first variational quantum eigensolver (VQE) demonstration used a photonic processor to estimate ground-state energies, proving that hybrid quantum-classical loops could extract chemistry from very small devices. Low and Chuang then introduced optimal constructions for Hamiltonian simulation and phase estimation—qubitization and quantum signal processing (QSP)—pinning down the asymptotics that future devices would need to exploit.

The theory generally ran ahead of the machines during this period. Photonic and NMR platforms validated principles even as coherence limits and gate errors capped circuit depth. The target for hardware, however, had become clear. In 2024–2025, that blueprint turned into a system: a mid-scale superconducting processor linked tightly to a national-scale supercomputer. Using an IBM Quantum Heron processor coupled to thousands of classical nodes on RIKEN's Fugaku, teams reported chemically meaningful problems beyond exact diagonalization with circuits up to dozens of qubits and deep gate counts, along with certified upper bounds from hybrid estimators.

At the same time, another accelerator joined the stack: artificial intelligence (AI) algorithms taking advantage of GPUs and AI accelerators. A vivid proof-point is a machine-learned exchange–correlation function trained to satisfy fractional-charge and fractional-spin constraints; it materially improved DFT accuracy across broad benchmarks and corrected key delocalization pathologies. This is clear evidence that AI can change the chemistry, not just speed the pipeline.

The integration of AI and quantum-centric supercomputing shapes today's fourth act: the development of algorithms built for hybrid compute environments. Quantum Filter Diagonalization, multireference Krylov approaches, and their sampling-based cousins, SQD and SKQD, share a common design: use of the quantum processor to prepare and sample a compact subspace, then push the heavy linear algebra and statistical estimation to classical nodes before extracting ground and excited states. This is not just a clever trick; it is an algorithmic style built around the machines we actually have.

AI strengthens this playbook. Learning-based compilers and RL circuit synthesizers can discover shorter block encodings and optimized phase schedules for QSP and quantum singular value transformation (QSVT). Graph models can auto-partition commuting Hamiltonian terms to slash measurement variance. Bayesian shot allocation can steer sampling to where it most reduces eigenvalue uncertainty. Even on the classical side, AI-designed preconditioners, low-rank surrogates, and learned deflation can make Krylov and Lanczos iterations converge in fewer matrix vector multiplications. In short, AI learns sparsity, symmetry, and conditioning, compressing circuits, cutting samples, and speeding up the eigensolves that turn quantum measurements into chemistry.

The message echoes across a century: algorithmic methods mature and scale when hardware makes a core operation broadly available. The day chemistry became a machine problem wasn't a single date; it's the recurring moment where algorithms fit the available compute, and the available compute, in turn, sparks new algorithmic ideas.

Phase Kickback

Seminal works

1928

The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I, Hartree, D. R.

1930

Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems, Fock, V.

1951

New Developments in Molecular Orbital Theory, Roothaan, C. C. J.

1964

Inhomogeneous Electron Gas, Hohenberg, P. & Kohn, W. (November)

1964

Correlations in the Motion of Atoms in Liquid Argon, Rahman, A. (October)

1970

Gaussian 70 (Quantum Chemistry Program Exchange, Program No. 237, 1970), Hehre, W. J., et al.

1981

Simulating Physics with Computers, Feynman, R. P.

1985

Unified Approach for Molecular Dynamics and Density-Functional Theory, Car, R. & Parrinello, M.

1986

Engineering and scientific processing on the IBM 3090, Gibson, D. H., et al.

1988

Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density, Lee, C., et al. (January)

Phase Kickback

Seminal works

1988 — **Density-functional exchange-energy approximation with correct asymptotic behavior**, Becke, A. D. (September)

1988 — **lCAP/3090: Parallel processing for large-scale scientific and engineering problems**, Clementi, E., et al. (December)

1989 — **A Fifth-Order Perturbation Comparison of Electron Correlation Theories**, Raghavachari, K. & Trucks, G. W.

1994 — **A Hybrid Decomposition Parallel Implementation of the Car–Parrinello Method**, Wiggs, J. K. & Jónsson, H.

1997 — **The Development of Computational Chemistry in the United Kingdom**, Smith, S. J. & Sutcliffe, B. T.

1998 — **The Chemistry of Water on Alumina Surfaces: Reaction Dynamics from First Principles**, Hass, K. C., et al.

1999 — **Quantum chemical models**, (Nobel Lecture), Pople, J. A.

2005 — **Simulated Quantum Computation of Molecular Energies**, Aspuru-Guzik, A., et al.

2005 — **Car–Parrinello Molecular Dynamics on Massively Parallel Computers**, Hutter, J. & Curioni, A.

2010 — **Towards quantum chemistry on a quantum computer**, Lanyon, B. P., et al.

Phase Kickback

Seminal works

2014

A variational eigenvalue solver on a photonic quantum processor, Peruzzo, A., et al.

2015

Nicholas Charles Handy. 17 June 1941 – 2 October 2012, Clary, D. C., et al.

2016

Hamiltonian Simulation by Qubitization, Low, G. H. & Chuang, I. L. (October)

2016

Optimal Hamiltonian Simulation by Quantum Signal Processing, Low, G. H. & Chuang, I. L. (June)

2019

Quantum Filter Diagonalization: Quantum Eigendecomposition without Full Quantum Phase Estimation, Parrish, R. M. & McMahon, P. L. (September)

2019

A Multireference Quantum Krylov Algorithm for Strongly Correlated Electrons, Stair, N. H., et al. (November)

2021

Pushing the frontiers of density functionals by solving the fractional electron problem, Kirkpatrick, J., et al.

2025

Quantum-Centric Algorithm for Sample-Based Krylov Diagonalization, Yu, J. (January)

2025

Chemistry beyond the scale of exact diagonalization on a quantum-centric supercomputer, Robledo-Moreno, J., et al. (June)