Introducing the IBM Quantum Experience

Quantum computing is here. While today's quantum processors are modest in size, the complexity grows continuously. The time is ripe to build and engage a community of new quantum learners, so that we change the way we think about computing. Our goal with the IBM Quantum Experience is for you to learn about the quantum world by reading this User Guide, composing your own experiments, running them in simulation, and executing them on the world's first fully-controllable quantum processor through the IBM Cloud.

The IBM Quantum Experience consists of:

- a set of tutorials (this User Guide) that will lead you from the basics of simple single-qubit experiments (Section II (/qstage /#/tutorial?sectionId=71972f437b08e12d1f465a8857f4514c)) to more complicated multi-qubit experiments (Section III (/qstage /#/tutorial?sectionId=050edf961d485bfcd9962933ea09062b)), and then toward more advanced ideas in the area of quantum algorithms (Section IV (/qstage /#/tutorial?sectionId=8443c4f713521c10b1a56a533958286b)) and quantum error correction (Section V (/qstage /#/tutorial?sectionId=bfd2a30ad6c5c915da3a696d76c474d7));
- the ***Quantum Composer,*** which is a graphical user interface where you can create your very own quantum circuit (which we call a ***quantum score),*** much like a composer composes a music score;
- a simulator that can test your quantum scores;
- access to an actual quantum processor running in an IBM Quantum Computing lab, where your quantum scores will be executed; and
- a ***quantum community*** where your quantum scores, ideas, and experiences

can be shared and discussed.

Please note that the IBM Quantum Experience is a "living experiment," and we are constantly making updates to it. We hope our users will help us refine its functionality and improve the overall interface by providing feedback. In the community forum, please let us know your thoughts for improvement -- and please share any cool scores and results you come across! If you find any bugs, please report them with our bug tracker, accessible via the little bug icon in the lower right-hand corner of each page.

To make sure everyone has a chance to use the real device in our lab via the Cloud, we have established a **Units** currency system. If you have joined the IBM Quantum Experience as a **Standard User,** you have full access to our simulation capabilities and to previously-run cached results from the real device and a small number of Units to run real experiments on the quantum processor hardware. Once you have read through the User Guide you will be rewarded with extra Units to run more real-time experiments. This system allows our experiment queue to run smoothly. When your Units are used up, you will be replenished once you have viewed the results of the completed execution. We also invite Standard Users to request an upgrade of your User status to **Expert User**, which provides access to more Units and other advanced features as we introduce them. Contact us and let us know why you would like to become an Expert User.

The quantum processor in our lab requires frequent calibration; during these short periods, you will receive a "Down for Calibration" notice and if we need to perform maintenance a "Down for Maintenance" message will be displayed. In both cases the simulation will be available for you to keep learning and designing new experiments.

For those who wish to jump right into creating your own experiments, you can skip right to the Quantum Composer (/qstage /#/tutorial?sectionId=75a85f7e14ae3fd4329ad5c3e59466ea&pageIndex=3). If you wish to improve your understanding of the quantum world first, please continue reading this User Guide, starting with the Quantum World (/qstage /#/tutorial?sectionId=75a85f7e14ae3fd4329ad5c3e59466ea&pageIndex=1) section on the next page. Use the numbers at the top and bottom of each page to navigate through each section.

Prepare yourself for a wild and fascinating journey -- and it all starts with a qubit.

We would like to acknowledge the work done under the IARPA Multi-Qubit Coherent Operations program, the Logical Qubits program, the LPS Quantum Characterization Validation & Verification program, and the IBM Research Frontiers Institute. The research performed in those programs has contributed to making this Quantum Experience possible.

Thank you,

Jay Gambetta, Jerry Chow, and the IBM Quantum team

The Quantum World

Today's computers perform calculations and process information using the standard (or as a physicist would say, "classical") model of computation, which dates back to Turing (https://en.wikipedia.org/wiki/Alan_Turing) and von Neumann (https://en.wikipedia.org/wiki/John_von_Neumann). In this model, all information is reducible to bits, which can take the values of either 0 or 1 -- and all processing can be performed via simple logic gates (https://en.wikipedia.org/wiki/Logic_gate) (AND, OR, NOT, NAND) acting on one or two bits at a time. At any point in its computation, a classical computer's state is entirely determined by the states of all its bits, so that a computer with n bits can exist in one of 2^n possible states, ranging from $00...0$ to $11\ldots1$.

The power of the quantum computer, meanwhile, lies in its much richer repertoire of states. A quantum computer also has bits. But instead of 0 and 1, its quantum bits, or *qubits*, can represent a 0, 1, or both at once, which is a property known as **superposition**. This on its own is no special thing, since a computer whose bits can be intermediate between 0 and 1 is just an analog computer, scarcely more powerful than an ordinary digital computer. However, a quantum computer takes advantage of a special kind of superposition that allows for *exponentially many* logical states at once, all the states from $|00...0\rangle$ to $|11\ldots1\rangle$. This is a powerful feat, and no classical computer can achieve it. The vast majority of these quantum superpositions, and the ones most useful for quantum computation, are **entangled**—they are states of the whole computer that do not correspond to any assignment of digital or analog states of the individual qubits. While not as powerful as exponentially many classical computers, a quantum computer is significantly more powerful than any one classical computer -- whether it be deterministic, probabilistic, or analog. For a few famous problems (such as factoring large numbers), a quantum computer is clearly the winner over a classical computer. A working quantum computer could factor numbers in a day that would take a classical computer millions of years.

One might think that the difficulty in understanding quantum computing or quantum physics lies in "hard math"... but mathematically, quantum concepts are only a bit more complex than high school algebra. Quantum physics is hard because, like Einstein's theory of relativity, it requires internalizing ideas that are simple but counter-intuitive. What is strange about relativity is the concept that time and space are interconnected, when common sense tells us they should act independently. If you begin to explain relativity to a person new to the idea by jumping straight to time and space, you will likely be greeted by a blank stare. A better way to start is as Einstein did, by explaining that relativity follows from a simple physical principle: the speed of light is the same for all uniformly moving observers. This one modest idea then becomes extremely profound and leads, by inexorable logic, to Einsteinian spacetime.

The counter-intuitive ideas of quantum physics that one must accept are 1) *a physical system in a perfectly definite state can still behave randomly*, and 2) *two systems that are too far apart to influence each other can nevertheless behave in ways that, though individually random, are somehow strongly correlated*. Unfortunately, unlike relativity, there is no single simple physical principle from which these conclusions follow. The best we can do is to distill quantum mechanics down to a few abstract-sounding mathematical laws, from which all the observed behavior of quantum particles (and qubits in a quantum computer) can be deduced and predicted. And, as with relativity, we must guard against attempting to describe quantum concepts in classical terms.

Quantum Laws in Black, White, and Blackandwhite

Quantum laws are, as far as we know, the most fundamental physical laws; they are inviolable. Here is our version of quantum physics distilled into five key laws**.** If the math is new to you, just print yourself a copy of these laws and skip them for now; while you work through the rest of the tutorial, return to them every now and then to see how they tie into what you are learning.

Quantum is a system like everything else.

To each physical system there corresponds a Hilbert space (1) of dimensionality equal to the system's maximum number of reliably distinguishable states (2).

A quantum state is a configuration of the system.

Each direction (ray) in the Hilbert space corresponds to a possible state of the system (3), with two states being reliably distinguishable if and only if their directions are orthogonal (inner product is zero).

A quantum state changes; it naturally wants to evolve, but it can always be *undone*.

Evolution of a closed system is a unitary (4) transformation on its Hilbert space.

Scaling - how parts make a whole.

The Hilbert space of a composite system is the tensor product of the Hilbert space of the parts (5).

Quantum measurements are probabilistic.

Each possible measurement (6) on a system corresponds to a resolution of its Hilbert space into orthogonal subspaces $\{\Pi_j\}$ where $\sum_j \Pi_j = 1$. On state $|\psi\rangle$ the result j occurs with probability $P(j) = \langle\psi|\Pi_j|\psi\rangle$ and the state after the measurement is $|\psi_j\rangle = \Pi_j|\psi\rangle/\sqrt{P(j)}$.

These five principles are the foundation for the whole quantum world.

Clarifications

1. A Hilbert space is a linear vector space with complex coefficients and inner products $\langle\phi|\psi\rangle = \sum_i \phi_i^* \psi_i$.
2. For a single qubit, there are two standard orthogonal states (*computational basis states*) that are conventionally denoted $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.
3. Other qubit states include $|+\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $|-\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $|\circlearrowright\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix}$ and $|\circlearrowleft\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -i \end{pmatrix}$
4. *Unitary* means linear and inner-product-preserving.
5. A two-qubit system can exist in a product state such as $|00\rangle$ or $|0+\rangle$ but also in an *entangled* state $(|00\rangle + |11\rangle)/\sqrt{2}$, in which neither qubit has a definite state, even though the pair together does.
6. Measurement causes the system to behave probabilistically and forget its pre-measurement state, unless that state happens to lie entirely with one of the subspaces Π_j.

The Quantum Composer

The ***Quantum Composer*** is our graphical user interface for programming a quantum processor. Those familiar with quantum computing may recognize the composer as a tool to construct *quantum circuits* using a library of well-defined gates and measurements. For those not familiar, we will explain a few of the key parts.

When you first click on the "Composer" tab above, you will have a choice between running a *real* quantum processor or a *custom* quantum processor. In the custom processor, gates can be placed anywhere, whereas in the real processor, the topology is set by the physical device running in our lab (note that this restricts the usability of some of the two-qubit gates).

Once you are in the "Composer" tab, you can start making your very own quantum circuits!

With the Composer, you can create a *quantum score,* which is analogous to a musical score in several respects. Time progresses from left to right. Each line represents a qubit (as well as what happens to that qubit over time). Each qubit has a different frequency, like a different musical note. Quantum gates are represented by square boxes that play a frequency for different durations, amplitudes, and phases. Gates on just one line are called single-qubit gates. The gates made with vertical lines connecting two qubits together are known as CNOT gates; these two-qubit gates function like an exclusive OR gate in conventional digital logic. The qubit at the solid-dot end of the CNOT gate controls the whether or not the target qubit at the $\oplus$ -end of the gate is inverted (hence controlled NOT, or CNOT). Some gates, like the CNOT, have hardware constraints; the set of allowed connections is defined by the schematic of the device located below the Quantum Composer, along with recently calibrated device parameters.

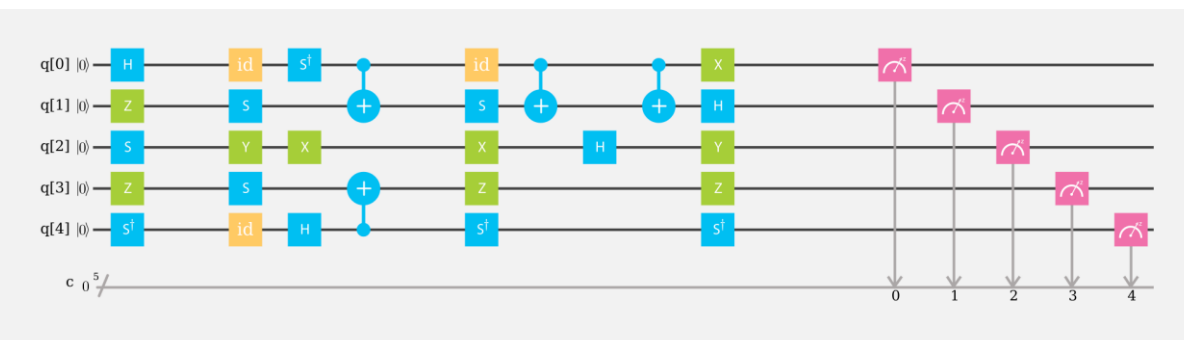
The Quantum Composer's library (located to the right of the qubit stave) contains many different classes of gates: single-qubit gates, such as the **yellow** idle operation; the **green** class of *Pauli operators*, which represent bit-flips (X, equivalent to a classical NOT); phase-flips (Z); and a combined bit-flip and phase-flip (Y). We also offer *Clifford operations*, which are the **blue** class of gates, such as H, S, and $S^\dagger$ gates for generating quantum superpositions, as well as the two-qubit entangling gate CNOT previously mentioned. The **red** gates are two-phase gates that are not in the Clifford group and are important for giving quantum computing its power. To measure the state of any qubit, use the **pink** standard measurement operation, which is a simple Z projection that is assigned to a classical bit in a classical bit register. If you ever need a reminder, hit the Help button (the question mark near the gates heading) for a quick summary of all available gates. A quantum algorithm (circuit) begins by preparing the qubits in well-defined states (here the ground state, $|0\rangle$, which we've automatically done for you), then executing a series of one- and two-qubit gates in time, followed by a measurement of the qubits.

If you are feeling brave, you can hit the **Advanced** button to view an additional set of gate operations and sub-routines; these are described in further detail here (https://quantumexperience.ng.bluemix.net/qstage /#/tutorial?sectionId=89ada8b1aa9e798ce6aa9a705feab237&pageIndex=0).

To use the Composer, simply drag the gate boxes onto the qubit stave to place them. Double-tap the boxes to delete, or drag them to the trash bin. To place a CNOT gate, drag first onto the target qubit (a ⊕ symbol will appear), then click on the control qubit (a solid dot will appear). Note that on the real quantum processor, you cannot add more gates to a circuit after placing a measurement; this feature will be added in the future.

Load the quantum score below and try out a simulation, or start composing your own!

My First Score



Running your Quantum Scores

Now that we've gotten familiar with the Composer, let's go over how to run a quantum score.

When you begin an experiment, you'll be prompted to give it a name, so that you can recognize it later. You will also see two choices: Real Quantum Processor, or Custom Topology. In both cases, you create your score by dragging gates onto the stave, adding a measurement, and then hitting "Run" for the score to execute.

Running on an Custom Quantum Processor

If you select "Custom Topology," your only option is to run your score in simulation. This is because the custom processor permits all-to-all connectivity; the real device, in contrast, is limited by physical connectivity. When you select Custom Topology, a dialogue box will ask you to select the number of qubits and classical bits assigned to different registers. We have set the maximum number of qubits to 20. The execution of your circuit happens immediately (unless the number of qubits is large) and the output can then be viewed in the Results (see the next section). Try out the "Single Qubit Measurement" below.

Running on a Real Quantum Processor (Requires Units)

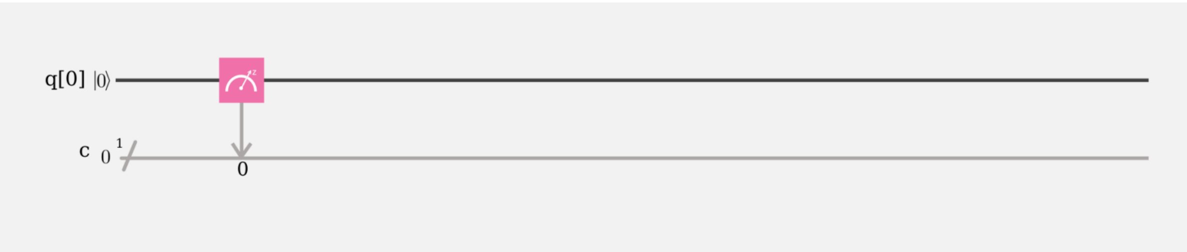
If you select "Real Quantum Processor," the score you compose will be placed into the experimental queue when you hit "Run," and you will be notified via email when it has been executed on the actual quantum computer in our lab.

You must have Units in your Quantum Experience account to use the Real Quantum Processor. If the score you are trying to run has previously been run, you will be given the choice of seeing the results from the cached execution right away (which costs no Units), or spending the Units to re-run the score as a new execution (meaning it will go into the experimental queue, and you will receive an email notification when it is done).

Note that many of the score examples found in this User Guide have previous executions available for you to view and experiment with!

Once you hit "Run," your score's progress will be visible in the "My Scores" tab, sorted by date executed. When your results are ready, you will be able to view them from this tab. You can also re-edit your score and execute it on the simulator while you wait for your results to return.

Single Qubit Measurement



Single Qubit Measurement (Real)

The Results

In our Quantum Experience, the results from launching your quantum scores can be visualized in two different ways: a standard histogram/bar graph, and as a Quantum Sphere, or QSphere. The QSphere, unique to the Quantum Experience, represents quantum circuit measurement outcomes in a visually striking and information-dense graphic.

After performing a quantum measurement, a qubit's information becomes a classical bit, and in our system (as is standard) the measurements are performed in the computational basis. For each qubit the measurement either takes the value 0 if the qubit is measured in state $|0\rangle$ and value 1 if the qubit is measured in state $|1\rangle$.

In a given run of a quantum circuit with n measurements, the result will be one of the 2^n possible n-bit binary strings. If the experiment is run a second time, even if the measurement is perfect and has no error, the outcome may be different due to the fundamental randomness of quantum physics. The results of a quantum circuit executed many different times can be represented as a distribution over the full 2^n possible outcomes. It is not scalable to represent all possible outcomes; therefore, we keep only those outcomes that happen in a given experiment and represent them in two different ways: as bars or as a Quantum Sphere.

Histogram representation (Bars)

The histogram/bar graph representation is the simplest to understand. The height of the bar represents the fraction of instances the outcome comes up in the different runs on the experiment. Only those outcomes that occurred with non-zero occurrences are included. If all the bars are small for visualization only (not if you download the data) they are collected into single bar called other values. In general this is not a problem as a good quantum circuit should not have many outcomes only circuits that have the final state in a large superposition will give many outcomes and these would take exponential measurements to measure.

The Quantum Sphere representation (QSphere)

The QSphere is our new alternative representation to visually show the same data as the bar graph neatly and strikingly. Each line from the center represents a possible outcome of the experiment, and the weight (darkness of the line) represents the likelihood of each outcome. As with the histogram, only those outcomes are included that occurred in a given experiment. The QSphere is divided into $n + 1$ levels, and each section represents the weight (total number of 1s) of the binary outcome. The top is the $|0 \ldots 0\rangle$ outcome, the next line is all the outcomes with a single 1 ($|10 \ldots 0\rangle$, $|01 \ldots 0\rangle$, etc), the line after that is all outcomes with two 1s, and so on until the bottom is the outcome $|1...1\rangle$.

For a single qubit there are two outcomes, and the sphere has only two levels; for two qubits, it has three sections with the middle section separated into two parts; for three qubits, it has four sections with the middle two being broken into three sections, and so on, following Pascal's triangle.

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

The usefulness of this representation is for distinguishing classical states from entangled states. A computational basis state will have a single line pointing in one direction. Under the assumption the state is pure, a superposition of two basis states will have two lines pointing in two directions of half weight. If these directions are on opposite sides of the QSphere we have a state that is maximally entangled (for $n > 1$) in the computation bases. Finally if there are faint lines in every direction we have made a uniform superposition state.

Superposition +++++