An introduction to implementing quantum algorithms

A basic Qiskit tutorial

Lindon Roberts, University of Melbourne (lindon.roberts@unimelb.edu.au)

MathQuEST Day, University of Western Australia

30 Nov 2026

Overview

This presentation is intended to be a gentle introduction to executing quantum algorithms using IBM's Qiskit software.

- Assumes basic Python knowledge
- Assumes minimal quantum computing knowledge

Overview

This presentation is intended to be a gentle introduction to executing quantum algorithms using IBM's Qiskit software.

- Assumes basic Python knowledge
- Assumes minimal quantum computing knowledge

Caveat

I am also a beginner at this, apologies to those of you who are not: please contribute any comments/feedback as we go!

Overview

This presentation is intended to be a gentle introduction to executing quantum algorithms using IBM's Qiskit software.

- Assumes basic Python knowledge
- Assumes minimal quantum computing knowledge

Caveat

I am also a beginner at this, apologies to those of you who are not: please contribute any comments/feedback as we go!

The goals of this talk are:

- Give a minimum/baseline knowledge of executing quantum algorithms
  - End point: factorize 15 using Shor's algorithm
- Provide a starting point for future MathQuEST tutorials

Qiskit

Qiskit (Quantum Information Software Kit) is an open-source software development kit for creating quantum circuits/programs. `https://www.ibm.com/quantum/qiskit`

- Originally developed by IBM in 2017
- Open source: `https://github.com/Qiskit/qiskit`
- Implemented in Python & Rust
- Used primarily through Python (also has a C interface for low-level functions)

Qiskit

Qiskit (Quantum Information Software Kit) is an open-source software development kit for creating quantum circuits/programs. `https://www.ibm.com/quantum/qiskit`

- Originally developed by IBM in 2017
- Open source: `https://github.com/Qiskit/qiskit`
- Implemented in Python & Rust
- Used primarily through Python (also has a C interface for low-level functions)

The broader qiskit ecosystem has modules for executing circuits via simulators (on classical hardware) or by connecting to quantum hardware from IBM, Amazon, IonQ, ...

Alternatives

Qiskit isn't the only quantum SDK...

- PennyLane
- cuQuantum (NVIDIA)
- Cirq (Google)
- Classiq

Alternatives

Qiskit isn't the only quantum SDK...

- PennyLane
- cuQuantum (NVIDIA)
- Cirq (Google)
- Classiq

...but it is the most popular by a wide margin (69% of quantum community used in 2025; next most popular is PennyLane 22%).

Unitary Foundation, 2025 Quantum Open Source Survey ($N = 997$),
`https://unitaryfoundation.github.io/survey-2025/`

Getting Started

Qiskit can be installed using the Python package manager (Windows, Mac, Linux all supported):

```
# Create & activate new Python virtual environment
python3 -m venv .venv
source .venv/bin/activate

# Or, using Anaconda
conda --create name myenv
conda activate myenv

# Install Qiskit and relevant dependencies
pip install qiskit                          # <-- main Qiskit package
pip install qiskit-ibm-runtime qiskit-aer   # <-- IBM runtime, Aer simulator
pip install jupyter matplotlib pylatexenc   # <-- for development/plotting
```

Hardware Access

To run quantum programs on real hardware, you need to be able to connect to the relevant hardware service and have a login/compute budget.

Hardware Access

To run quantum programs on real hardware, you need to be able to connect to the relevant hardware service and have a login/compute budget.

I have an IBM Cloud account through Melbourne's IBM Quantum Hub (gives 400 minutes of compute time per month) `https://quantum.cloud.ibm.com/`

Tutorial

Tutorial

If you want to participate, you can download the Jupyter notebook (and 1 JSON file storing some saved results) and follow along.

`https://lindonroberts.github.io/talks/mathquest2026`

I will upload full 'solutions' to this page after the session.