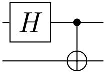
Quantum Circuits Project

Ulrik de Muelenaere

December 15, 2022

Project Goal

- To write a tool that converts a quantum circuit to a unitary matrix, and vice versa



$$\Longleftrightarrow \quad \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

- Circuit can then be simulated by matrix-vector multiplication

$$|\psi'\rangle = U\,|\psi\rangle$$

Implementation Overview

- Implemented in Python using NumPy
- `State` class wraps a complex vector with useful methods
- `Gate` class wraps a unitary matrix with useful methods
- `Circuit` contains a list of `Gates`, and is itself a subclass of `Gate`
- `DecomposedCircuit` is a `Circuit` consisting of only CNOT and single-qubit gates, constructed from an arbitrary unitary matrix

Defining Gates

```
S = Gate(
    name="S",
    num_qubits=1,
    matrix=[
        [1, 0],
        [0, 1j],
    ],
)

Sdagger = S.inverse()

CS = S.control()
```

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

$$S^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$$

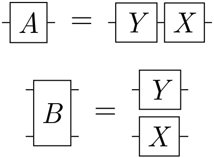
$$CS = \begin{bmatrix} I & 0 \\ 0 & S \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$$

Two Ways of Combining Gates

```
A = X @ Y
B = X.tensor_product(Y)
```

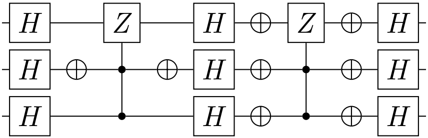
$$A = XY$$

$$B = X \otimes Y$$



Grover's Algorithm Circuit

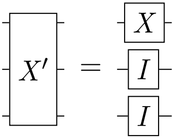
```
circuit = Circuit(
    name="Grover",
    num_qubits=3,
)

circuit.add(HHH)

circuit.add(X, [1])
circuit.add(CCZ)
circuit.add(X, [1])

circuit.add(HHH)
circuit.add(XXX)
circuit.add(CCZ)
circuit.add(XXX)
circuit.add(HHH)
```



Adding a Gate to a Circuit

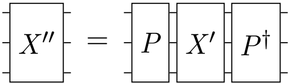
- Consider `circuit.add(X, [1])` from the previous slide
- X must first be padded to a three-qubit gate

$$X' = X \otimes I \otimes I$$



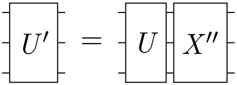
- Qubits must be permuted so that X is applied to qubit 1

$$X'' = P^\dagger X' P$$



- Now the gate matrix can be multiplied with the previous circuit matrix U

$$U' = X'' U$$



Simulating the Circuit

- The Grover circuit has unitary matrix

$$U = \begin{bmatrix} -0.177 & 0.177 & 0.53 & 0.177 & 0.177 & 0.53 & 0.177 & 0.53 \\ -0.177 & -0.53 & 0.53 & -0.53 & 0.177 & -0.177 & 0.177 & -0.177 \\ -0.177 & 0.177 & -0.177 & -0.53 & 0.177 & 0.53 & -0.53 & -0.177 \\ -0.177 & -0.53 & -0.177 & 0.177 & 0.177 & -0.177 & -0.53 & 0.53 \\ -0.177 & 0.177 & 0.53 & 0.177 & -0.53 & -0.177 & -0.53 & -0.177 \\ -0.884 & 0.177 & -0.177 & 0.177 & 0.177 & -0.177 & 0.177 & -0.177 \\ -0.177 & 0.177 & -0.177 & -0.53 & -0.53 & -0.177 & 0.177 & 0.53 \\ -0.177 & -0.53 & -0.177 & 0.177 & -0.53 & 0.53 & 0.177 & -0.177 \end{bmatrix}$$

- When applied to initial state $|000\rangle$, the final state is

$$\begin{aligned} U\,|000\rangle = &-0.177\,|000\rangle - 0.177\,|001\rangle - 0.177\,|010\rangle - 0.177\,|011\rangle \\ &-0.177\,|100\rangle - 0.884\,|101\rangle - 0.177\,|110\rangle - 0.177\,|111\rangle \end{aligned}$$

- `State` contains a method to plot a histogram of the probability distribution

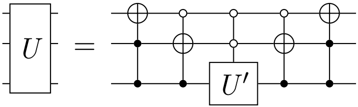
- The search has correctly amplified the probability of $|101\rangle$

Decomposition

- Single-qubit and CNOT gates are universal, so we can decompose any unitary matrix to a circuit of those gates
- First, decompose into a product of two-level matrices, which only act non-trivially on two components
- For example, this two-level matrix only acts on $|000\rangle$ and $|111\rangle$

$$U = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 & 0 & c \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ b & 0 & 0 & 0 & 0 & 0 & 0 & d \end{bmatrix}$$

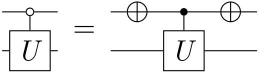
- Swap components using multi-controlled-NOT gates to get multi-controlled single-qubit gate



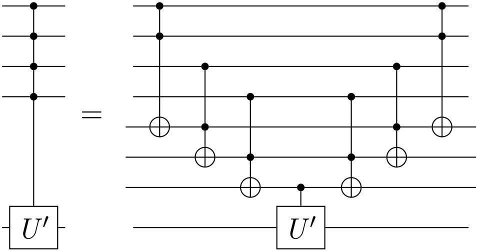
where $U' = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$

Decomposition

- Open circles represent negative controls

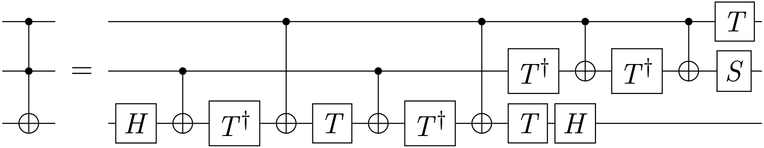


- Decompose multi-controlled-gates to Toffoli and single-controlled gates using ancilla qubits

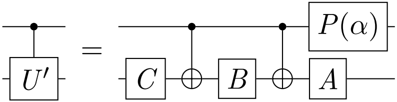


Decomposition

- Decompose Toffoli gates



- Decompose remaining controlled gates except CNOT, by finding α, β, γ and δ such that $U' = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta)$, then



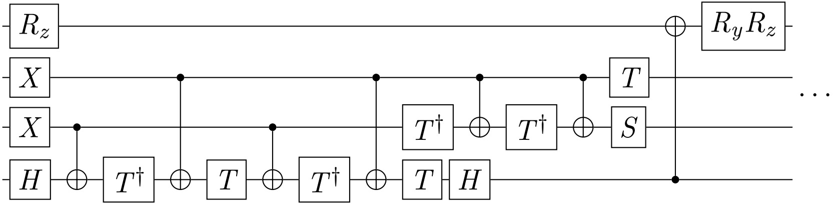
where $A = R_z(\beta) R_y(\gamma/2)$, $B = R_y(-\gamma/2) R_z(-(\delta+\beta)/2)$ and $C = R_z((\delta-\beta)/2)$

Decomposition Testing

- `DecomposedCircuit` asserts that the absolute difference between its matrix and the original is at most 10^{-4}
- Any single-qubit unitary matrix can be written as $e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta)$ for $\alpha, \beta, \gamma, \delta \in [0, 2\pi)$, so I sampled the entire space of single-qubit matrices at regular intervals
- I tested with larger matrices, including the Grover circuit

Decomposition Results

- The Grover circuit decomposes to 2520 gates



- In general, the size of the decomposed circuit is exponential in the number of qubits

Lessons Learned

- I knew decomposition to CNOT and single-qubit gates was possible, but had to learn how
- I learned that decompositions of general unitary matrices grow exponentially in the number of qubits
- Qiskit has a more efficient and practical approach, where the definition of each gate includes a decomposition

Potential Uses

- Decomposition would be useful if it was not so inefficient
- Building a unitary matrix from a circuit could be useful, but other tools can already to it
- Likely most useful as an educational tool to understand the mapping between quantum circuits and unitary matrices

References and Links

- Decomposition based on Sections 4.2, 4.3 and 4.5 of:
  Nielsen, M. A., & Chuang, I. L. (2010). *Quantum Computation and Quantum Information* (10th Anniversary ed.). Cambridge University Press.
- Code available at:
  `https://github.com/ulrikdem/cse60932-project`