Quantum gate synthesis

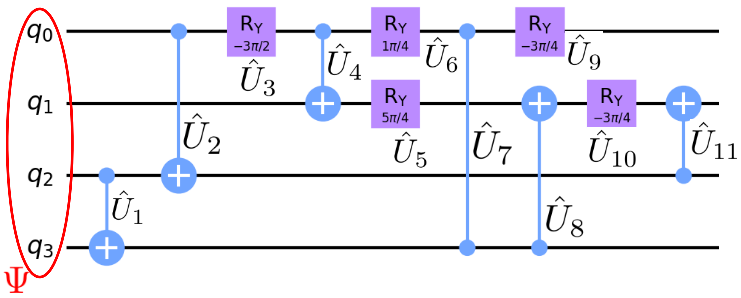
Péter Rakyta[1,2]

[1]Department of Physics of Complex Systems, Eötvös Loránd University, Budapest, Hungary
[2]Quantum Information National Laboratory, Hungary

1 Introduction

The compilation of quantum programs in terms of elementary operations is essential to run quantum algorithms on real quantum hardware. In our recent work we reported on a novel quantum gate approximation algorithm based on the application of parametric two-qubit gates in the synthesis process [1]. The utilization of these parametric two-qubit gates in the circuit design allows us to transform the discrete combinatorial problem of circuit synthesis into an optimization problem over continuous variables. The circuit is then compressed by a sequential removal of two-qubit gates from the design, while the remaining building blocks are continuously adapted to the reduced gate structure by iterated learning cycles. We implemented the developed algorithm in the SQUANDER software package [2] and benchmarked it against several state-of-the-art quantum gate synthesis tools. Our numerical experiments revealed outstanding circuit compression capabilities of our compilation algorithm providing the most optimal gate count in the majority of the addressed quantum circuits.

$$\hat{U} = \hat{U}_{11} \cdot \hat{U}_{10} \cdot \hat{U}_9 \cdot \hat{U}_8 \cdot \hat{U}_7 \cdot \hat{U}_6 \cdot \hat{U}_5 \cdot \hat{U}_4 \cdot \hat{U}_3 \cdot \hat{U}_2 \cdot \hat{U}_1$$



2 Description of the project

In this project we will examine how the choice of the optimization cost function influences the performance of the quantum gate synthesis process in the SQUANDER package. We compare our results to other synthesis tools (like the QISKIT package) and execute the compiled programs on the IBM quantum computers. The objective of the project is to

1. Learn the basic concept of quantum logical gates.
2. Learn the usage of the SQUANDER package, interfacing python and C++ codes to improve computational performance.
3. Develop python-side scripts to decompose unitary matrices in terms of single- and two-qubit gates.
4. execute the compiled program on the IBM quantum computers.

The students choosing this project are expected to be familiar with

- Quantum Mechanics

- Python programming language.
- Linux operating systems. (recommended to use for the development environment)

References

[1] Péter Rakyta and Zoltán Zimborás. Efficient quantum gate decomposition via adaptive circuit compression, 2022.

[2] Sequential quantum gate decomposer. https://github.com/rakytap/sequential-quantum-gate-decomposer, 2021.