

# Project Proposal: A classical simulator for quantum circuits dominated by Clifford gates

Patrick Rall and Iskren Vankov, advised by David Gosset  
California Institute of Technology - February 24, 2016

## Introduction

Simulation of quantum systems is in general intractable on classical computers. This is a central reason for our interest in quantum devices. However, as we build systems with more and more qubits, verification of their correct operation using a classical computer will be necessary. Classical algorithms that are exponential in the number of qubits are unsuitable for this purpose.

A recent paper by Sergey Bravyi and David Gosset [1] describes an algorithm for quantum circuit simulation that is polynomial in the number of qubits. By utilizing the Gottesmann-Knill theorem to efficiently simulate Clifford gates, the procedure is only exponential in the number of  $T$ -gates used in the circuit. A preliminary Matlab implementation was capable of simulating a hidden shift quantum algorithm with 40 qubits and 50  $T$ -gates on a laptop.

The goal of this project is to develop an open-source application that permits other physicists to use this algorithm. It will employ a massively parallel architecture, so it can be run on many CPUs in a computer cluster. Our stretch goal is to implement the algorithm on Graphics Processing Units (GPUs) present in many laptops and desktops, which could be useful for physicists with limited access to a cluster.

In addition to creating a useful tool, this project will be an excellent learning experience. Primarily, we will learn algorithms to

manipulate stabilizer states by representing them in terms of affine spaces and quadratic forms, permitting us to e.g., take their inner products or sample random states efficiently. We will also gain experience in writing distributed software for several architectures, and how to automatically manage highly parallel code across many computers.

## Algorithm Information

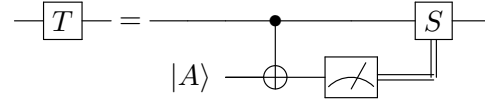


Figure 1: The  $T$ -gate gadget. By replacing all occurrences of  $T$  with this gadget we can avoid using non-Clifford gates in the simulation. However, we now require several non-stabilizer states  $|A\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle)$ .

The paper [1] describes two algorithms. An algorithm for computing the output probability of a string  $x$  runs in

$$\text{poly}(n, m) + 2^{0.5t}t^3.$$

Another algorithm for sampling a string  $x$  from the output distribution runs in

$$\text{poly}(n, m) + 2^{0.23t}t^3w^4,$$

where  $n$  is the number of qubits,  $m$  the total number of gates,  $t$  the number of  $T$ -gates and  $w$  the size of the string  $x$ .

## Time plan

We will employ an ‘agile’ development strategy which through regular meetings allows us to be flexible with our goals while maximizing productivity. The following time plan ensures that even if we encounter unexpected delays the product will still be useful to many physicists.

**Before Spring break** we will meet regularly with David Gosset, who will teach us the algorithm in detail. We will also read papers and also review [4] to evaluate challenges of a GPU implementation.

**By the beginning of Spring term** we will implement stabilizer state manipulation subroutines detailed in the appendix of [1], likely in C++.

**By mid-March** we will have both algorithms detailed above running in a non-parallel environment. Transitioning from MATLAB to C++ may already yield a performance boost.

**By mid-April** we aim to prepare the existing non-parallel code to run in a massively parallel environment. This involves separating parallel and non-parallel code, scanning a network of computers for resources, and distributing information among these resources.

**By mid-May** we intend to port the parallel portions of our code to run efficiently on GPU. These devices have a different architecture which can be beneficial for parallel applications. Iskren will be taking *CS179 GPU Programming* (textbook: [4]).

**By June 10** we intend to build a graphical user interface that will permit physicists to specify circuits with ease. Gates like the Toffoli gate can be implemented using the Clifford+T gate set, but doing so requires a non-trivial circuit (Figure 3 in [1]). A tool to simplify the process of entering and testing circuits will make the software more popular.

## References

- [1] S. Bravyi, D. Gosset. Jan 29, 2016. “Improved classical simulation of quantum circuits dominated by Clifford gates”. <http://arxiv.org/abs/1601.07601>
- [2] S. Bravyi, G. Smith, J. Smolin. Jun 3, 2015. “Trading classical and quantum computational resources”. <http://arxiv.org/abs/1506.01396>
- [3] A. Aaronson, D. Gottesman. Jun 25 2004. “Improved Simulation of Stabilizer Circuits”. <http://arxiv.org/abs/quant-ph/0406196>
- [4] N. Wilt. Jun 12, 2013, Addison-Wesley Professional. “The CUDA Handbook: A Comprehensive Guide to GPU Programming”. <http://www.cudahandbook.com/>