DEPARTMENT OF COMPUTER SCIENCE

A multi-core CPU implementation of the classical Boson Sampling algorit

Manan Vaswani

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering.

Thursday 9th May, 2019

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of MEng in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Manan Vaswani, Thursday 9th May, 2019

Contents

Executive Summary

A compulsory section, of at most 1 page

This section should précis the project context, aims and objectives, and main contributions (e.g., deliverables) and achievements; the same section may be called an abstract elsewhere. The goal is to ensure the reader is clear about what the topic is, what you have done within this topic, *and* what your view of the outcome is.

The former aspects should be guided by your specification: essentially this section is a (very) short version of what is typically the first chapter. Note that for research-type projects, this **must** include a clear research hypothesis. This will obviously differ significantly for each project, but an example might be as follows:

> My research hypothesis is that a suitable genetic algorithm will yield more accurate results (when applied to the standard ACME data set) than the algorithm proposed by Jones and Smith, while also executing in less time.

The latter aspects should (ideally) be presented as a concise, factual bullet point list. Again the points will differ for each project, but an might be as follows:

- I spent 120 hours collecting material on and learning about the Java garbage-collection sub-system.
- I wrote a total of 5000 lines of source code, comprising a Linux device driver for a robot (in C) and a GUI (in Java) that is used to control it.
- I designed a new algorithm for computing the non-linear mapping from A-space to B-space using a genetic algorithm, see page 17.
- I implemented a version of the algorithm proposed by Jones and Smith in [6], see page 12, corrected a mistake in it, and compared the results with several alternatives.

Supporting Technologies

0.1 Blue Crystal 4

I made use of the Blue Crystal Phase 4 supercomputer (henceforth abbreviated as BC4) to run most of my code. BC4 is one of the fastest and most advanced supercomputing facilities in the UK, made available by the University of Bristol High Performance Computing group [15]. Phase 4 has 525 Lenovo nx360 m5 compute nodes, each of which has two 14 core 2.4 GHz Intel E5-2680 v4 (Broadwell) CPUs, and 128 GiB of RAM. Note that such a CPU with 14 physical cores would have 28 logical cores.

0.2 C++ libraries/APIs

0.2.1 OpenMP

OpenMP is an application programming interface(API) that implements multithreading, which is a method of parallelising where a master thread divides tasks among a team of threads. Concurrently, each of these threads is run individually on a single logical core. Each of the threads has an associated id, indexed starting from 0. It is possible to obtain the number of threads, or the thread index using a collection of special OpenMP functions. In C++, the OpenMP functions are included in a header file 'omp.h'. OpenMP is used with the help of #pragmas in C++, which is a compiler-specific preprocessor directive, and is used to carry out operations such as creating a team of threads or automatically dividing iterations of a loop between threads [6]. I used OpenMP to parallelise the algorithm in the paper.

0.2.2 Armadillo

In the implementation, I make extensive use of the C++ Armadillo library[21]. Armadillo is a high quality linear algebra library that provides us with efficient implementations of matrix and vector operations. It greatly simplifies operations such as adding vectors, calculating cumulative products, and accessing matrix columns to name a few. Armadillo supports complex numbers as well, which was a necessary requirement for the project. In addition, it supports the use of OpenMP for parallelisation, and automatically uses it in some cases to speed up computationally expensive operations.

0.3 Intel VTune

I used the Intel VTune 2018 profiler to analyse my code and identify bottlenecks. It was available to use on BC4. Code profilers like VTune dynamically analyse executable code that is running, and identify bottlenecks by giving a detailed break down of information about CPU usage, and time taken to run individual lines of the code. Intel VTune also comes with a graphical user interface which lets the user look at bottom-up approaches of performance hotspots, enabling them to easily locate time-consuming areas in the code. [23].

0.4 Miscellaneous Tools

0.4.1 Python, Matplotlib

I used the Python to process the timing data that I had obtained, and subsequently used the Matplotlib library to plot the graphs representing this data.

0.4.2 Godbolt Compiler Explorer

Godbolt is an online tool that shows what high level language (such as C++) code ends up looking like after compilation [7]. I used it to gain a deeper understanding of how some sections of my code ran.

Acknowledgements

I would like to thank my supervisor, Dr. Raphael Clifford, for his help and support throughout the year and for introducing me to the Boson Sampling problem.

Chapter 1

Contextual Background

A compulsory chapter, of roughly 5 pages

The advances made in the field of quantum algorithms in the past 25 years has given rise to questions about whether a number of classical hypotheses still hold true, in light of modern quantum research. The most notable example is the Extended Church-Turing Thesis which, in simple terms says that all computational problems that are efficiently solvable by realistic physical devices are also efficiently solvable by a probabilistic Turing machine [10]. However, Shor's algorithm [22] potentially disproves this. Informally, Shor's discovery stated that if a classical computer could accurately simulate a quantum-mechanical experiment in probabilisitic polynomial time, then it must be able to factor integers in polynomial time as well. The problem of factoring integers is known to have no polynomial-time algorithm on a classical computer. It is so widely accepted that factoring integers is a hard problem that a number of cryptographic systems are built on its hardness, including the famous RSA algorithm [19]. Hence, Shor's algorithm has a strong implication: If a classical computer is able to simulate quantum experiments efficiently, it would be able to break widely used cryptosystems such as RSA.

The topic of classically simulating quantum systems is a central idea in the field of quantum supremacy. Quantum supremacy is the phenomena that there exist quantum experiments that cannot be accurately and efficiently simulated using classical systems [**?**]. Numerous attempts have been made to demonstrate quantum supremacy but the current physical limitations of quantum systems has made that an extremely challenging problem. Additionally, what makes quantum supremacy so hard to demonstrate is that it is not defined in terms of the ability of a quantum system to solve a particular problem, but rather it requires showing that classical computers cannot solve a problem. In a way, it could be likened to classifying computational problems into complexity classes [16]. In the same way that a number of complexity theory related theorems and conjectures rely on a number of theoretic assumptions, it is necessary for quantum supremacy to rely on similar such assumptions as well, as it would be nearly impossible to unconditionally prove such statements [8]. Linking back to Shor's algorithm and its potential to demonstrate quantum supremacy and simultaneously disprove the Extended-Church Turing thesis, there are two significant drawbacks which prevent it from doing so.

Firstly, the problem of factoring integers, while widely believed to be an $\mathsf{NP}-$hard problem, its hardness has not been formally proved. The assumption is not strong enough to verify that Shor's algorithm does indeed achieve quantum supremacy.

The second, and more obvious drawback to Shor's algorithm is that a large-scale physical implementation of a quantum computer to run it is beyond the current reach of technology. While there have been several experiments that successfully factor small numbers like 15 [13] and 21 [25] with high accuracy, we are not at risk of RSA breaking just yet!

This has motivated the search for other such quantum experiments that can be run physically, but not simulated classically. However, an important requirement is that it should be possible to model these experiments as computational problems in order to study their hardness.

There are a handful of problems that satisfy this criteria, such as the problem of sampling the output distribution of random quantum circuits, which has been one of the most popular problems in the field of quantum supremacy in the past few years due to a lot of interest from big companies like Google and IBM. The best known classical algorithm for simulating the task of sampling bitstrings from the output of random quantum circuits is exponential in the number of qubits in the bitstring, leading one

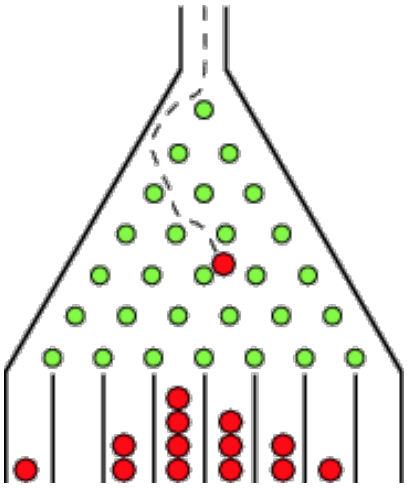

Figure 1.1: A Galton board, used to demonstrate sampling from a binomial distribution [2]

group of researchers in 2016 to estimate that a physical experiment with ≈ 50 qubits could be enough to demonstrate quantum supremacy [3]. However, more recently a classical simulation of the problem with 56 qubits was carried out successfully [17]. This pushed the imminence of quantum supremacy for the circuit sampling problem even further away.

The other famous problem in the field of quantum supremacy, of a similar flavour, is the Boson Sampling problem, which is the main focus of this paper. While the problem reduces down to a sampling problem like circuit sampling, as the name suggests, it actually belongs to a paradigm of Quantum Computation different to the ones discussed before, called Linear Optical Quantum Computation (or LOQC). The protocol by Knill, Laflamme and Milburn in 2001 [11] demonstrated that a scalable quantum computer with linear optical elements could be robustly implemented. In LOQC, a photon is used to represent a single qubit.

Aaronson and Arkhipov in their paper on the classical complexity of linear optics showed that it was possible to build a quantum computation model within the LOQC paradigm which could not be efficiently simulated by classical computers [1]. They define a model for 'Boson Sampling' in which n identical photons are passed through a linear optical network and then measured to determine their location. This linear optical network consists of a collection of simple optical elements called beam splitters and phase shifters.

The model could be thought of as being a parallel to a Galtons Board, shown in figure 1.1, in which n balls are dropped into a vertical lattice of wooden pegs, each of which randomly scatters an incoming ball to one of two other pegs with equal probability. The input to this system is the exact arrangement of the pegs, while the output is the number of balls that have landed at each location on the bottom. The output could alternatively be thought of as a sample from the distribution of all possible output arrangements (which is the binomial distribution in this case).

In the Aaronson and Arkhipov Boson Sampling model, the balls are replaced with identical photons, and the pegs arranged in a lattice are replaced with an arbitrary arrangement of optical elements. Also, the photons can be dropped from different starting locations as opposed to just a single position. The Boson-Sampling problem involves sampling the output distribution using photon-number discriminating detectors. They argued that if there is a polynomial time classical algorithm that exactly samples from the Boson Sampling distribution, the complexity class $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$ which would collapse the polynomial hierarchy to the third level by Toda's theorem [24].

In order to actually compare the physical model with a classical computational model, the paper reduces the Boson Sampling problem to a problem in purely mathematical terms. Photons are a type of boson, which is one of the two main types of particles in the world. The link between bosons and permanents of matrices has been known since 1953, from the work by Caianiello [4] which showed that the amplitudes of n-boson process can be written as the permanents of $n \times n$ matrices. Hence it was possible to represent the Boson Sampling distribution in terms of the permanents of matrices, and this was shown and proved in the paper.

Carrying out a physical experiment for Boson Sampling is not a trivial task, and increasing the number of photons or the number of output modes to large enough values is extremely difficult. The current experimental record is with 5 photons and 9 output modes. However, this does show the possibility of

potentially having larger-scale implementations in the future.

The Aaronson and Arkhipov paper provoked much interest in the study of classical computation algorithms for Boson Sampling. Much of the work is focussed around fast implementations for the calculation of permanents of large matrices, with recent works showing efficient results for matrices as large as 54×54 [20, 12]. There have been a few other approaches to simulating the Boson sampling problem such as classically modelling the linear optical network itself, and sampling from the output as opposed to computing the purely mathematical equivalent of the problem [18]. There have also been attempts at using Markov Chain Monte Carlo (MCMC) methods to sample from the Boson Sampling distribution by trying to identify it as the equilibrium distribution of a Markov chain, following the method by Hastings [9]. However, estimating how accurately these methods approximate to the true Boson Sampling experiment has been difficult. Nonetheless, the MCMC method proposed by Neville et. al. [14] gave strong numerical evidence that classical Boson Sampling may be feasible for large input sizes.

In terms of actual implementations of the Boson Sampling algorithm, there have been few, with the most significant benchmark being the test of the simulation run on the Tianhe-2 supercomputer [26] which managed to simulate Boson Sampling for 50 photons with a runtime of approximately 100 minutes.

However, in the groundbreaking paper by Clifford and Clifford [5], a new significantly faster algorithm was proposed which showed promise of carrying out classical Boson Sampling in an even shorter time. The algorithm proposed in this paper is central to our work. We give a highly optimised implementation of this algorithm with the aim of setting a new benchmark for the classical Boson Sampling problem, and in the process pushing away the imminence of quantum supremacy in the near future.

Building on the content discussed in this section, the concrete aims of the project are:

1. Research and analyse literature on the Boson Sampling problem, and its relation to quantum supremacy.

2. Demonstrate a strong understanding of the Boson Sampling problem, and the algorithms proposed in [5].

3. Study potential techniques to optimise an implementation of the algorithm, and provide a high-speed CPU implementation of the algorithm.

4. Summarise the results obtained by optimising the implementation, and state its implications.

Chapter 2

Technical Background

A compulsory chapter, of roughly 10 **pages**

This chapter is intended to describe the technical basis on which execution of the project depends. The goal is to provide a detailed explanation of the specific problem at hand, and existing work that is relevant (e.g., an existing algorithm that you use, alternative solutions proposed, supporting technologies).

Per the same advice in the handbook, note there is a subtly difference from this and a full-blown literature review (or survey). The latter might try to capture and organise (e.g., categorise somehow) *all* related work, potentially offering meta-analysis, whereas here the goal is simple to ensure the dissertation is self-contained. Put another way, after reading this chapter a non-expert reader should have obtained enough background to understand what *you* have done (by reading subsequent sections), then accurately assess your work. You might view an additional goal as giving the reader confidence that you are able to absorb, understand and clearly communicate highly technical material.

Chapter 3

Project Execution

A topic-specific chapter, of roughly 15 **pages**

This chapter is intended to describe what you did: the goal is to explain the main activity or activities, of any type, which constituted your work during the project. The content is highly topic-specific, but for many projects it will make sense to split the chapter into two sections: one will discuss the design of something (e.g., some hardware or software, or an algorithm, or experiment), including any rationale or decisions made, and the other will discuss how this design was realised via some form of implementation.

This is, of course, far from ideal for *many* project topics. Some situations which clearly require a different approach include:

- In a project where asymptotic analysis of some algorithm is the goal, there is no real "design and implementation" in a traditional sense even though the activity of analysis is clearly within the remit of this chapter.
- In a project where analysis of some results is as major, or a more major goal than the implementation that produced them, it might be sensible to merge this chapter with the next one: the main activity is such that discussion of the results cannot be viewed separately.

Note that it is common to include evidence of "best practice" project management (e.g., use of version control, choice of programming language and so on). Rather than simply a rote list, make sure any such content is useful and/or informative in some way: for example, if there was a decision to be made then explain the trade-offs and implications involved.

3.1 Example Section

This is an example section; the following content is auto-generated dummy text. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit.

Figure 3.1: This is an example figure.

foo	bar	baz
0	0	0
1	1	1
⋮	⋮	⋮
9	9	9

Table 3.1: This is an example table.

Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

3.1.1 Example Sub-section

This is an example sub-section; the following content is auto-generated dummy text. Notice the examples in Figure 3.1, Table 3.1, Algorithm 3.1 and Listing 3.1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan

for $i = 0$ **upto** n **do**
  $t_i \leftarrow 0$
end

Algorithm 3.1: This is an example algorithm.

```
for( i = 0; i < n; i++ ) {
  t[ i ] = 0;
}
```

Listing 3.1: This is an example listing.

bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Example Sub-sub-section

This is an example sub-sub-section; the following content is auto-generated dummy text. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est,

iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Example paragraph. This is an example paragraph; note the trailing full-stop in the title, which is intended to ensure it does not run into the text.

Chapter 4

Critical Evaluation

A topic-specific chapter, of roughly 15 **pages**

This chapter is intended to evaluate what you did. The content is highly topic-specific, but for many projects will have flavours of the following:

1. functional testing, including analysis and explanation of failure cases,
2. behavioural testing, often including analysis of any results that draw some form of conclusion wrt. the aims and objectives, and
3. evaluation of options and decisions within the project, and/or a comparison with alternatives.

This chapter often acts to differentiate project quality: even if the work completed is of a high technical quality, critical yet objective evaluation and comparison of the outcomes is crucial. In essence, the reader wants to learn something, so the worst examples amount to simple statements of fact (e.g., "graph X shows the result is Y"); the best examples are analytical and exploratory (e.g., "graph X shows the result is Y, which means Z; this contradicts [1], which may be because I use a different assumption"). As such, both positive *and* negative outcomes are valid *if* presented in a suitable manner.

Chapter 5

Conclusion

A compulsory chapter, of roughly 5 pages

The concluding chapter of a dissertation is often underutilised because it is too often left too close to the deadline: it is important to allocation enough attention. Ideally, the chapter will consist of three parts:

1. (Re)summarise the main contributions and achievements, in essence summing up the content.
2. Clearly state the current project status (e.g., "X is working, Y is not") and evaluate what has been achieved with respect to the initial aims and objectives (e.g., "I completed aim X outlined previously, the evidence for this is within Chapter Y"). There is no problem including aims which were not completed, but it is important to evaluate and/or justify why this is the case.
3. Outline any open problems or future plans. Rather than treat this only as an exercise in what you *could* have done given more time, try to focus on any unexplored options or interesting outcomes (e.g., "my experiment for X gave counter-intuitive results, this could be because Y and would form an interesting area for further study" or "users found feature Z of my software difficult to use, which is obvious in hindsight but not during at design stage; to resolve this, I could clearly apply the technique of Smith [7]").

Bibliography

[1] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 333–342, New York, NY, USA, 2011. ACM.

[2] Margherita Barile and Eric W Weisstein. Galton board. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/GaltonBoard.html.

[3] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. Characterizing Quantum Supremacy in Near-Term Devices. *arXiv e-prints*, page arXiv:1608.00263, Jul 2016.

[4] E. R. Caianiello. On quantum field theory — i: explicit solution of dyson's equation in electrodynamics without use of feynman graphs. *Il Nuovo Cimento (1943-1954)*, 10(12):1634–1652, Dec 1953.

[5] Peter Clifford and Raphaël Clifford. The classical complexity of boson sampling. *CoRR*, abs/1706.01260, 2017.

[6] Leonardo Dagum and Ramesh Menon. Openmp: an industry standard api for shared-memory programming. *Computational Science & Engineering, IEEE*, 5(1):46–55, 1998.

[7] Matt Godbolt. Godbolt compiler explorer. https://godbolt.org/, 2012.

[8] Aram W. Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203–209, Sep 2017.

[9] W. Larry Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[10] Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An Introduction to Quantum Computing*. Oxford University Press, Inc., New York, NY, USA, 2007.

[11] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, 2001.

[12] P. H. Lundow and K. Markström. Efficient computation of permanents, with applications to boson sampling and random matrices. *arXiv e-prints*, page arXiv:1904.06229, Apr 2019.

[13] Thomas Monz, Daniel Nigg, Esteban A. Martinez, Matthias F. Brandl, Philipp Schindler, Richard Rines, Shannon X. Wang, Isaac L. Chuang, and Rainer Blatt. Realization of a scalable shor algorithm. *Science*, 351(6277):1068–1070, 2016.

[14] Alex Neville, Chris Sparrow, Raphaël Clifford, Eric Johnston, Patrick M. Birchall, Ashley Montanaro, and Anthony Laing. Classical boson sampling algorithms with superior performance to near-term experiments. *Nature Physics*, 13:1153 EP –, 10 2017.

[15] University of Bristol ACRC. Blue crystal phase 4. https://www.bristol.ac.uk/acrc/high-performance-computing/bluecrystal-tech-specs/.

[16] Christos H. Papadimitriou. Computational complexity. In *Encyclopedia of Computer Science*, pages 260–265. John Wiley and Sons Ltd., Chichester, UK, 2003.

[17] Edwin Pednault, John A. Gunnels, Giacomo Nannicini, Lior Horesh, Thomas Magerlein, Edgar Solomonik, Erik W. Draeger, Eric T. Holland, and Robert Wisnieff. Breaking the 49-Qubit Barrier in the Simulation of Quantum Circuits. *arXiv e-prints*, page arXiv:1710.05867, Oct 2017.

[18] Saleh Rahimi-Keshari, Timothy C. Ralph, and Carlton M. Caves. Sufficient Conditions for Efficient Classical Simulation of Quantum Optics. *Physical Review X*, 6(2):021039, Apr 2016.

[19] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[20] Wojciech Roga and Masahiro Takeoka. Classical simulation of boson sampling with sparse output. *arXiv e-prints*, page arXiv:1904.05494, Apr 2019.

[21] Conrad Sanderson. Armadillo c++ linear algebra library, June 2016.

[22] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *arXiv e-prints*, pages quant–ph/9508027, Aug 1995.

[23] Intel Software. Intel vtune amplifier 2018. Software available at https://software.intel.com/en-us/vtune, 2018.

[24] Seinosuke Toda. Pp is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20:865–877, 10 1991.

[25] Juha J. Vartiainen, Antti O. Niskanen, Mikio Nakahara, and Martti M. Salomaa. Implementing Shor's algorithm on Josephson charge qubits. *Phys. Rev. A*, 70(1):012319, Jul 2004.

[26] Junjie Wu, Yong Liu, Baida Zhang, Xianmin Jin, Yang Wang, Huiquan Wang, and Xuejun Yang. A benchmark test of boson sampling on Tianhe-2 supercomputer. *National Science Review*, 5(5):715–720, 07 2018.

Appendix A

An Example Appendix

Content which is not central to, but may enhance the dissertation can be included in one or more appendices; examples include, but are not limited to

- lengthy mathematical proofs, numerical or graphical results which are summarised in the main body,
- sample or example calculations, and
- results of user studies or questionnaires.

Note that in line with most research conferences, the marking panel is not obliged to read such appendices.