Fourier Coefficient Behaviour in Noisy Gaussian Boson Sampling

Julia Miklas

4th Year Project Report
Computer Science and Physics
School of Informatics
University of Edinburgh

2024

Abstract

In the era of noisy intermediate-scale quantum (NISQ) computing, Gaussian boson sampling has emerged as one of the main avenues used to claim experimental quantum advantage. However, noise in these experiments may temporarily limit the validity of their claims. A team at the University of Edinburgh is currently developing a classical simulation method that aims to take advantage of noise by using Fourier analysis to obtain the probabilities of GBS outcomes.

In this project, I investigated the behaviour of Fourier coefficients under photon loss noise for GBS systems of 20 and 100 modes. I used a quantum hardware simulation library to obtain a uniform spread of output states from which I calculated Fourier coefficients using Wigner functions and density matrix representation. I analysed ideal systems to establish a baseline and then compared them to systems with varying levels of photon loss. For both cases, I also investigated the effects of squeezing used in the input states.

I found that both squeezing and loss significantly alter the distribution of Fourier coefficients, and for most loss amounts this results in a significant decrease in the complexity of classical simulation. I then derived a formula to adjust squeezing to losses and found that when this was used to calibrate the systems, higher losses resulted in faster rates of decay of higher-order Fourier coefficients, and that in some cases this decay was exponential-like.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Julia Miklas*)

Acknowledgements

I would like to express my gratitude to my supervisor, Raúl García-Patrón Sánchez, for the invaluable guidance and constant encouragement he offered me throughout this past year. His passion for his work is contagious and I am thankful to him for patiently introducing me to such a fascinating field.

I would also like to extend my appreciation to my family and friends who have supported me through this project. Jakub, Ailsa, Emma, and Beatrice, this would not have been possible without the kindness you have shown me. Thank you!

Table of Contents

Chapter 1

Introduction

1.1 Quantum advantage

Quantum computing is a rapidly developing field in Computer Science that uses principles of quantum mechanics to redefine the construction of computers and the algorithms they execute. Quantum mechanical properties like superposition and entanglement are consequences of the dual wave-particle nature of matter at a microscopic level. These properties can be exploited to create a quantum bit (qubit) that is no longer confined to the $\{0, 1\}$ computational states but can also exist in a superposition of both of these states simultaneously.

Unlocking the possibility of computing with superposed and entangled states and of utilizing wave interference properties to manipulate the interaction of said states can allow quantum computers to achieve quantum advantage in performing certain tasks. Quantum advantage is the ability of quantum machines to carry out computations that a classical computer cannot replicate in any feasible amount of time.[1] Part of the difficulty of quantum computing is finding specific implementations where the non-deterministic properties of qubits allow quantum computers to outperform their classical counterparts. It is important to note that not all quantum advantage is inherently useful—devising a problem with the sole purpose of proving quantum advantage might result in a problem so abstract that it does not have any practical applications.[2] At the same time, like in many emergent fields, it might not immediately be evident what the future applications of a task or algorithm that achieves quantum advantage might be. There is a fine line here that must be carefully threaded so as to not produce insubstantial results.

As of today, two main applications have produced experimental claims to quantum advantage: random quantum circuits and Gaussian boson sampling.

1.1.1 Random quantum circuits

Random quantum circuits (RQC) are a class of quantum circuits characterized by the random selection of gates or operations applied to qubits. Sampling the output of these circuits was the first avenue that led to a claim of quantum advantage. The qubit states that enter the circuit are entangled by the repeated application of one and two-qubit logical operations encoded in the gates. Due to the randomness of the circuit, the resulting states have no symmetries or predictable structure and can be spread over the entirety of the 2^n Hilbert space spanned by the n input qubits. The lack of symmetries makes the task of sampling from the output distribution of said circuits computationally expensive on classical computers, since the problem cannot be reduced to a simpler form.[3]

In July 2019, Google reported that they were able to sample the output of a random circuit using a processor with programmable superconducting qubits named Sycamore. The researchers sampled a million bitstrings in 200 seconds, and estimated that the same task performed by the most powerful classical computers would take approximately 10,000 years.[4]

This claim was followed by many studies that contested the large computation time disparity. In August 2022, a collaboration of several universities in China used a tensor network approach to replicate the Google experiment results' sample size and fidelity in 15 hours, proving that a classical algorithm is, at least for now, not necessarily faster but still feasible to implement.[5]

1.1.2 Gaussian boson sampling

A more recently emerging method for demonstrating computational advantage is Gaussian boson sampling—a variation of boson sampling (BS). In boson sampling experiments, non-classical light (photons) is passed through an interferometer (a type of linear optical network) to generate a highly entangled output state. The output is subsequently measured using either photocounters, which count the number of photons detected, or threshold photodetectors[1], which indicate whether at least one photon has been detected. Aaronson and Arkhipov, who first proposed BS, proved it to be classically hard by showing that sampling from the output distribution of states requires the calculation of the permanent of a matrix at least once per sample, a calculation that is itself #P-hard.[6]. Currently, the best-known exact simulation algorithm has a running time that is exponential in the number of input photons[7], making BS a good candidate for demonstrating quantum advantage.

However, the task of boson sampling on quantum computers is limited in its computational speed-up and scalability by the time required for photon sources to simultaneously emit n separate photons. To circumvent this technical difficulty, squeezed states were proposed as the non-classical light source for sampling experiments.[8] Squeezed states exhibit reduced uncertainty in either their momentum or spatial components, and thus an increased uncertainty in the other (due to Heisenberg's uncertainty relation $\Delta x \Delta p \geq \hbar/2$). Using a quantum optics process that converts one photon into two lower-energy photons (parametric down-conversion), photon generation efficiency is increased to the point that it is no longer a limiting factor to the expansion of the experiment.[9]

Gaussian boson sampling (GBS) makes use of squeezed states from parametric down-conversion to solve the boson sampling problem. The GBS protocol not only significantly enhances the photon generation probability compared to standard Boson

[1] Also known as on/off photodetectors and binary photodetectors.

sampling with single-photon Fock states (see Section 2.3), but also links to potential applications such as dense subgraph problems and molecular vibronic spectra.[10] The term Gaussian in the name refers to the Gaussian quasi-probability distributions of both the input and output states when plotted in phase space (see Section 2.4). Compared to the permanent for BS, classical simulations of GBS are bottlenecked by the calculation of a Torontonian—a matrix operation related to the Hafnian, both of complexity $O(n^3 \times 2^n)$ where n is the number of input qubits.[11]

Three notable results implementing GBS have been reported. In December 2020 a team at the University of Science and Technology of China (USTC) reported that they built Jiuzhang, a photonic quantum computer with a 100-mode interferometer.[12] They sent 50 squeezed states through the device and were able to sample over 3 million bit strings in 200 seconds. In July 2021 the same team reported improved results with the quantum computer Jiuzhang 2.0.[13] The 144-mode interferometer produced up to 113 photon detections at a sampling rate of the order 10^{24} faster than brute-force simulation on classical supercomputers. Finally, in November 2021 a group at Xanadu announced that they had built a 216-mode quantum processor named Borealis which detected up to 219 photons in 36 μs,[14] a feat that would take the best supercomputers more than 9000 years to simulate rigorously. Their novel approach used sequential pulses of light to store state information and optical fiber loops to implement gates, greatly decoupling the complexity and size of the quantum computer from the number of input photons.

1.2 Impact of noise on computational hardness of GBS

All quantum computers mentioned above fall under the category of noisy intermediate-scale quantum (NISQ) computers. The defining characteristic of this level of computing is the presence of a non-negligible amount of uncorrected noise in both the optical circuit and photodetector components of the computers. In this context, the term noise is synonymous with error, since corruptions (noise) in the experiment will lead to a variable amount of error which makes the sampling results diverge from ground truth.

Two of the largest sources of noise in GBS are photon loss (Section 1.2.1) and photon distinguishability (Section 1.2.1). Other imperfections include dark counts (false positive detection events) and unwanted variations in the settings of the interferometer[13].

The intrinsic error present in GBS experiments can change the requirements and methods needed to classically sample from GBS distributions. In many different ways, these noise artifacts may therefore result in a decrease in computational hardness of classical simulations.[15]

1.2.1 Photon loss

As light propagates through an interferometer, some photons will be scattered out of the circuit or absorbed by defects in the circuit materials. The Xanadu experiment detected a mean of 125 photons from an input of 160 photons and the latest USTC experiment reported an overall transmission efficiency of 78%.

Lower photon counts at the detectors reduce the dimension of the Hilbert space that the output distribution can span and therefore simplify the space and time complexity of classical simulation of lossy GBS experiments. It has been proven that losses will render GBS classically simulatable if the depth of the circuit is linear in the number of modes and if photons decay exponentially with depth, as is the case for optical fibers and integrated chips.[15]

1.2.2 Photon distinguishability

Photons are said to be indistinguishable when there is no way to differentiate between their wavefunctions. GBS relies on specific patterns of both constructive and destructive interference that only occur between indistinguishable photons. An example of this is the Hong-Ou-Mandel effect: when two indistinguishable photons enter a 50:50 beamsplitter from two separate input ports, they will always exit from the same output port.

Photon distinguishability is caused by inaccurate photon sources emitting photons with slightly different frequencies, spatial modes, or polarisations. If distinguishable photons pass through a circuit simultaneously, the effect will be similar to running two experiments at the same time but detecting the photons at the end as if only one experiment was running. This phenomenon can be used to break GBS simulations down into smaller subsystems that are easier to simulate classically. This simulation is done by creating virtual modes to contain distinguishable photons that do not interfere with other photons but contribute to photon detection at the end.[16]

1.3 Project motivation

Several papers have attempted to exploit noise to disprove quantum advantage using GBS. As a result, progressively stricter inequalities have been set on noise and experimental parameters that experiments must observe so as to not be classically simulatable. This reflects the constant competition between increasingly improving GBS experiments and newer, more efficient methods of classical simulation.

Currently, a team at the University of Edinburgh is investigating potential avenues to efficiently simulate noisy GBS. One of the techniques selected to do so is to represent the output probability distribution using Fourier series. One can use Boolean function Fourier analysis to describe the distribution since output samples of photodetectors are bitstrings, with one bit per mode of the simulated interferometer. The bit is 0 if no photons are detected in a given mode and 1 otherwise.

Due to the way that noise changes the underlying structure of the GBS output distribution, Fourier coefficients above a certain order become negligible. It is then possible to truncate the series without impacting the quality of the simulation of a noisy device. Here, order is defined as the Hamming weight (the number of ones) of the Fourier coefficient and relates to the number of output state modes involved in the calculation of the Fourier coefficient. Fourier coefficients have an analytical expression that can be easily derived, and for low order can be solved numerically within a practical timeframe.

It is predicted that this method can be implemented with a running time of $O(2^k \cdot k^3)$ for each coefficient, where k is the order we truncate at. The factor k^3 relates to the time required to compute the determinant of a submatrix of the covariance matrix and the factor of 2^k corresponds to the number of bitstrings with order k or lower that we need to consider in our coefficient calculation.

1.3.1 Context of the project in literature

There are several precedents that validate the use of Fourier series and their truncation to simulate quantum advantage claims on classical computers. In 2018, Xun Gao and Luming Duan showed that when a constant level of polarised noise is applied to the gates of a generalized quantum circuit, the coefficients in the Fourier series that describes the output decay fast enough that a good approximation can be obtained by truncating the series.[17] A tensor network representation is used to show this and the results are abstracted from GBS-specific implementations, yet the foundation of the method is the same.

More recently, Aharonov et al. published an algorithm that calculates the marginal probability of outcomes in noisy boson sampling on the basis of sparsity of non-suppressed Fourier coefficients under specific Gaussian noise models.[18] However, their algorithm was not directly applicable to the type of noise that is created by photon loss or distinguishability.

In addition to these two works, the inverse correlation between the rate of decay of coefficients and their order is further reinforced by results from the Jiuzhang 2.0 experiment where it was found that the correlation between theoretical and experimental results decreased as order increased. In their discussion, they go so far as to mention that using low-order marginal probabilities to approximate the outcome probability distribution is a potential avenue for competitors to efficiently simulate their experiment.[13]

Indeed, in 2022 the Google quantum research team published a classical sampling algorithm that built upon the idea of using marginals to simulate lossy GBS.[19] The method involves sampling from a distribution that approximates all one and two-mode marginal probabilities, which unlike RQC are non-uniform and therefore easy to compute efficiently. The greedy heuristic algorithm used can be expanded to obtain a probability distribution with approximations of marginals up to k^{th} order, with a running time exponential in order k (i.e. the number of clicks) but polynomial in the number of modes.

1.4 Objectives

The above findings support the idea that noise in Gaussian boson sampling may result in an accelerated decay of higher-order Fourier coefficients. The behaviour of increasing orders of Fourier coefficients of noisy systems will therefore determine the viability of classical simulation methods that rely on Fourier series truncation.

An efficient way of analysing this behaviour is to collect all Fourier coefficients of the same order and compute their squared average value. This measure is called the Fourier weight and it is the cornerstone of the analyses done in this project. Squaring the Fourier coefficients before taking the average is useful when dealing with Fourier coefficients that can be both positive and negative. For ideal random quantum circuits, it is known that the expectation value of the mean Fourier weight over many random circuits is $1/2^n$ for all orders (where n is the number of modes in the circuit).[20] It is also known that for noisy RQC the mean Fourier weight decays exponentially as order increases.

No such analysis has yet to be conducted in the context of GBS. Therefore, in this thesis, I investigated the behavior of Fourier coefficients under realistic noise in GBS experiments with on/off photodetectors. To do this, I computed the mean Fourier weights for all orders over many GBS output states.

In order to correctly interpret the effect of noise on the Fourier coefficients in GBS, we first need to understand the baseline behaviour of Fourier coefficients in noiseless systems. Therefore, I also investigated the expected value of Fourier coefficients of ideal, non-noisy GBS.

The type of noise I investigated is photon loss. I have assumed that this noise is both local and uniform for simplicity, meaning the noise is applied independently on each mode and uniformly distributed over the circuit. There are also other variables that impact the behavior of Fourier coefficients; notably the squeezing parameter chosen for the input states. I also looked at how this affects Fourier coefficients, both independently and combined with loss.

1.5 Report structure

Chapter 2 overviews the main topics in quantum optics required to model Gaussian boson sampling. It also introduces density matrix and Wigner function representations, and illustrates how the Gaussian nature of the states of light in GBS simplifies the notation and classical simulation method we use.

Chapter 3 introduces Fourier analysis of Boolean functions and contextualizes its use in Gaussian boson sampling. The derivation of Fourier coefficients and their integration into the classical simulation method are then outlined.

Chapter 4 covers methodology and implementation. It explains how we can use random unitaries to obtain a large sample of data and introduces the StrawberryFields library and how it is used in this project. It then covers the convergence of coefficients, its significance, as well as the required sample size for convergence to occur.

Chapter 5 presents the results of this project: an examination of the interplay between losses and squeezing on the potential decay of Fourier coefficients. This is accompanied by a comparison to the baseline behaviour of Fourier coefficients for ideal systems.

Chapter 6 summarizes my findings within the context of the research questions posed in the introduction and offers insights for future analyses.

Chapter 2

Quantum Optics

In this chapter, I describe the experiment whose results we are trying to simulate, and the mathematical and physical tools we will need to be able to study it.

2.1 Experimental set-up

There are three modules in a GBS device: input state preparation, interferometer, and output state measurement. There are different hardware components and techniques that can be used to achieve each step, but the mathematical descriptions of how the states are transformed are identical. We will take the Xanadu experiment as an example (see Figure 2.1) as it is the most recent one.

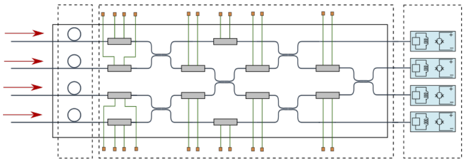

Figure 2.1: A four-mode Xanadu photonic chip.[21] The three modules (state preparation, interferometer, and measurement) are indicated with a dashed outline.

Xanadu uses a photonic chip made of silicon and silicon nitride to run GBS experiments from start to end. Input states for each mode of the circuit are prepared by sending classical laser light through squeezers, made of small ring resonators that generate the squeezed states. These are injected into an array of waveguides that guide the photons through the interferometer. The interferometer is composed of an alternating series of beam splitters and phase shifters that entangle the light states as they pass through. At the end of the circuit, each output mode is measured with a photodetector that returns a click[1] when at least one photon is detected in that mode.

[1] Clicks are represented by 1s, no clicks are represented by 0s.

2.2 The physical model

At the core of quantum optics is the concept of wave-particle duality. It tells us that light is made of photons, massless particles that carry a discrete amount of energy, and that this energy can be observed as an oscillating electromagnetic field. Therefore, we can rely on a model of an electromagnetic field (EMF) to describe the physics governing the behavior of light states in a GBS experiment.

In a classical EMF, the evolution of the system is described by a field equation and modes are defined as the discrete solutions to this equation. However, a classical framework will not suffice, as it does not include photons and it is not able to describe quantum effects such as superposition and entanglement. Therefore, the model we use is that of a quantum EMF, where each mode is represented as a quantum harmonic oscillator.[22] The specific properties of each mode or oscillator determine how many bosons (photons) can inhabit it. These modes correspond to the paths of a GBS circuit and the photodetectors measure how many photons are present in each mode.

2.3 Quantum Harmonic Oscillators and Fock States

The energy of a quantum harmonic oscillator is described by the Hamiltonian $\hat{H}$

$$\hat{H} = \frac{1}{2}(\hat{p}^2 + \omega^2\hat{x}^2) \tag{2.1}$$

where ω is the angular frequency of the oscillator and the relationship between the position operator $\hat{x}$ and momentum operator $\hat{p}$ is constrained via the commutator relationship $[\hat{x},\hat{p}] = \hat{x}\hat{p} - \hat{p}\hat{x} = i\hbar$.

We can also define two ladder operators, called the destruction ($\hat{a}$) and creation ($\hat{a}^\dagger$) operators, that modify the quanta of energy, i.e. the number of energy packets in the oscillator.

$$\hat{a} = \frac{1}{\sqrt{2\hbar\omega}}(\omega\hat{x} + i\hat{p}) \qquad \hat{a}^\dagger = \frac{1}{\sqrt{2\hbar\omega}}(\omega\hat{x} - i\hat{p}) \tag{2.2}$$

Using the commutator relation $[\hat{a},\hat{a}^\dagger] = 1$ and the definition of the number operator $\hat{n} = \hat{a}^\dagger\hat{a}$, we can rewrite the Hamiltonian as

$$\hat{H} = \hbar\omega(\hat{a}^\dagger\hat{a} + \frac{1}{2}) = \hbar\omega(\hat{n} + \frac{1}{2}) \tag{2.3}$$

The solutions (eigenstates) of the Hamiltonian are called Fock states, and they have a defined, quantized degree of excitation. We can obtain the energy and the degrees of excitation of the ground state $|0\rangle$ and the excited states $|1\rangle$, $|2\rangle$, etc. by applying the Hamiltonian and number operators respectively.

$$\hat{H}|n\rangle = \hbar\omega\left(n + \frac{1}{2}\right)|n\rangle \qquad \hat{n}|n\rangle = n|n\rangle \tag{2.4}$$

In quantum optics, a photon is synonymous with the degrees of excitation in a mode, so the number operator also tells us how many photons are in a given mode. Applying a creation or destruction operator on a Fock state will create or destroy a photon:

$$\hat{a}^{\dagger}|n\rangle = \sqrt{n+1}\,|n+1\rangle \qquad \hat{a}|n\rangle = \sqrt{n}\,|n-1\rangle \tag{2.5}$$

2.4 Phase space

There is an alternative representation that is better suited to describe the Gaussian states that are manipulated in a GBS experiment. Phase space is a two-dimensional space in which the two axes correspond to the position and momentum variables that describe a given state. For an n-mode system where each mode $m \in \{0, 1, ..n\}$ is represented by an oscillator with operators $\hat{a}_m^{\dagger}$ and $\hat{a}_m$, we can rearrange[2] for $\hat{x}$ and $\hat{p}$ to get operators for the corresponding phase space:

$$\hat{x}_m = \sqrt{\frac{\hbar}{2}}(a_m^{\dagger} + a_m) \qquad \hat{p}_m = i\sqrt{\frac{\hbar}{2}}(a_m^{\dagger} - a_m) \tag{2.6}$$

Due to Heisenberg's uncertainty principle which tells us that we cannot know both position and momentum simultaneously, quantum states cannot have well-defined points in phase space. Instead, states of light occupy regions of phase space characterized by quasi-probability distributions (quasi because negative values are permitted – see Section 2.5) over both momentum and position variables. As an example, the distributions of a single-mode vacuum state $|0\rangle$ over position and momentum space are

$$P^{|0\rangle}(x) = |\langle\hat{x}|0\rangle|^2 = \frac{1}{\sqrt{\hbar\pi}}e^{-x^2/\hbar} \qquad P^{|0\rangle}(p) = |\langle\hat{p}|0\rangle|^2 = \frac{1}{\sqrt{\hbar\pi}}e^{-p^2/\hbar} \tag{2.7}$$

See Appendix AA for the complete calculation. Both distributions in phase space are Gaussian. Similarly, a squeezed state also exhibits Gaussian distributions. However, these distributions will have different widths, reflecting the increased uncertainty in one variable and the decreased uncertainty in the other. The squeezing is evident in Figure 2.2 where the two states are visualized in position-momentum phase space.

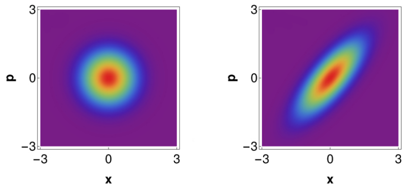

Figure 2.2: Phase space representation of a vacuum (left) and squeezed vacuum (right) state.[23]

Any operation on states in phase space can be interpreted as a manipulation of the phase space. The squeezing operator stretches one dimension of the space and compresses

[2] From here on I omit the angular frequency ω for simplicity, as it is an implicit property of the system and can be taken as a constant.

the other. The rotation operator can be viewed as a rotation of the space, effectively changing the orientation of the distributions. A beam splitter conserves the photon number but redistributes them among the output modes. This is reflected in phase space as changes in the shape and orientation of the distributions.

These operations are all linear transformations on the initial, already Gaussian set of vacuum states. Linear transformations preserve the Gaussian nature of the states, hence the name Gaussian Boson Sampling.

2.4.1 Statistical moments of Gaussian states

Gaussian states can be completely characterized by their first and second statistical moments[22], the mean vector $\bar{r}$, and the covariance matrix σ. The mean vector of n modes is given by the displacement of each mode from the origin (in terms of the $\hat{x}$ and $\hat{p}$ operators).

$$\bar{\mathbf{r}} = (\hat{x}_1, \hat{x}_2, ..., \hat{x}_n, \hat{p}_1, \hat{p}_2, ..., \hat{p}_n)^\intercal \tag{2.8}$$

Each element of the $2n \times 2n$ covariance matrix is then related to the mean square displacements of two states i and j.

$$\sigma_{i,j} = \frac{1}{2}\langle \Delta\mathbf{r}_i \Delta\mathbf{r}_j + \Delta\mathbf{r}_j \Delta\mathbf{r}_i \rangle \qquad \text{where} \quad \Delta\mathbf{r} = \hat{r} - \bar{\mathbf{r}} \tag{2.9}$$

The diagonal elements of the covariance matrix reduce to $\sigma_{i,i} = (\Delta x_i)^2$ and $\sigma_{i+n,i+n} = (\Delta p_i)^2$, i.e. the variances of the position and the momentum of that mode. The off-diagonal elements are related to the correlation between the position and momentum of the two different modes they refer to.

The mean vector for a single mode vacuum state is $\bar{\mathbf{r}} = (0,0)^\intercal$ as its fluctuations in position and momentum are centered around the origin. The covariance matrix is $\sigma = \frac{\hbar}{2}\mathbb{I}^2$ as each mode must respect the uncertainty principle $\Delta x \Delta p \geq \hbar/2$ and it is a state with minimal uncertainty so we have $\Delta x = \Delta p = \sqrt{\hbar/2}$. For a single-mode squeezed state, the mean vector is also $\bar{\mathbf{r}} = (0,0)^\intercal$, while the covariance matrix now reflects the difference in uncertainty between position and momentum. For a state squeezed in the position component, the covariance matrix is given by equation 2.10. For squeezing in the momentum variable, the signs in the exponents are exchanged for states squeezed in the momentum variable.

$$\sigma = \frac{\hbar}{2}\begin{pmatrix} e^{-2r} & 0 \\ 0 & e^{2r} \end{pmatrix} \tag{2.10}$$

2.5 Wigner functions

An important mathematical tool that describes states in phase space is the Wigner function. It provides a unified representation of the probability distributions in both $\hat{x}$ and $\hat{p}$, and a formula to calculate the expectation value of an operator in phase space that we will use in one of the steps of the classical simulation technique.

Before we introduce the Wigner function, we need to define the density operator $\hat{\rho}$: a general representation of mixed quantum states such as those formed by the superposition of multiple pure states. It is given by the sum over the projection operators of all basis states of the space being considered multiplied by the probability of measurement of the mixed state collapsing it into that basis state.[24]

$$\hat{\rho} = \sum_{i}^{n} p_i \left|\psi_i\right\rangle\left\langle\psi_i\right| \tag{2.11}$$

By applying the density operator to a state we obtain its density matrix representation. The density matrix of a pure state is simply given by $\hat{\rho} = |\psi\rangle\langle\psi|$.

The density operator has two properties that are useful to us. Firstly, it is normalized such that $\mathrm{Tr}(\hat{\rho}) = 1$. We can show this for a pure state using the cyclic property of the trace (which means that it is invariant under circular shifts).

$$\mathrm{Tr}(\hat{\rho}) = \mathrm{Tr}(|\psi\rangle\langle\psi|) = \mathrm{Tr}(\langle\psi|\psi\rangle) = 1 \tag{2.12}$$

Secondly, we can use $\hat{\rho}$ to obtain the expectation value of an arbitrary operator $\hat{A}$, once again taking advantage of the cyclic property of the trace and the fact that the trace of a scalar is the scalar itself.

$$\langle\hat{A}\rangle = \mathrm{Tr}(\hat{\rho}\hat{A}) = \mathrm{Tr}(|\psi\rangle\langle\psi|\hat{A}) = \mathrm{Tr}(\langle\psi|\hat{A}|\psi\rangle) = \langle\psi|\hat{A}|\psi\rangle \tag{2.13}$$

These properties extend and apply to the density operators of mixed states. The proofs for mixed states can be found in Appendix A

We can now define the Wigner function of an n-mode state as the Wigner transform of its density matrix, i.e. the mapping of a Hilbert space operator to a phase space function

$$W(\mathbf{x},\mathbf{p}) = \frac{1}{(2\pi\hbar)^n}\int_{\mathbb{R}^n} d^n\mathbf{y}\ \langle \mathbf{x} + \frac{1}{2}\mathbf{y}|\rho|\mathbf{x} - \frac{1}{2}\mathbf{y}\rangle\ e^{2i\mathbf{p}\cdot\mathbf{y}/\hbar} \tag{2.14}$$

where $\mathbf{x}$ and $\mathbf{p}$ are n-length vectors containing the eigenstates of the $\hat{x}_i$ and $\hat{p}_i$ operators of each mode.

Solving the integral gives the unique quasi-probability distribution of the state $\hat{\rho}$ which, like a probability distribution over phase space, is normalized to one, but unlike a probability distribution, can take on negative values. This is a sign of the non-classicality of the state and an indicator that it cannot be fully interpreted as a genuine probability distribution.

The Wigner transform can also be applied to operators to get their representation in phase space. Using this, we can derive an equality between the trace of the product of $\hat{\rho}$ and an arbitrary operator $\hat{A}$, and the integral of the product of their Wigner functions. See Appendix A for the full proof.

$$\mathrm{Tr}(\hat{\rho}\hat{A}) = 2\pi\hbar\int_{-\infty}^{\infty} d\mathbf{x}\int_{-\infty}^{\infty} d\mathbf{p}\, W_{\rho}(\mathbf{x},\mathbf{p})W_{A}(\mathbf{x},\mathbf{p}) \tag{2.15}$$

Combining this with equation 2.13, using the phase space displacement vector $\mathbf{r} = (x, p)^\intercal$, and generalizing to n modes, we obtain an equation for the expectation of an operator in phase-space using the Wigner function formalism:

$$\langle \hat{A} \rangle_\rho = \mathrm{Tr}(\hat{\rho}\hat{A}) = (2\pi\hbar)^n \int_{\mathbb{R}^{2n}} d\mathbf{r}\, W_\rho(\mathbf{r}) W_A(\mathbf{r}) \tag{2.16}$$

For Gaussian states, the Wigner function can be expressed in terms of the mean vector and covariance matrix of the state.[25]

$$W(\mathbf{r}) = \frac{\exp(-\frac{\hbar}{2}(\mathbf{r}-\bar{\mathbf{r}})^\intercal \sigma^{-1}(\mathbf{r}-\bar{\mathbf{r}}))}{(2\pi\hbar)^n \sqrt{\det \sigma}} \tag{2.17}$$

This new representation makes the calculation of the above integrals less complex. It can also lead to some simplified functions for some states, such as the one for a vacuum state $|0\rangle$ with the values of $\bar{\mathbf{r}}$ and σ defined above:

$$W_0 = \frac{\exp(-\mathbf{r}^\intercal \mathbb{I} \mathbf{r})}{2\pi\hbar\sqrt{\hbar}} = \frac{\exp(-x^2 - p^2)}{2\pi\hbar\sqrt{\hbar}} \tag{2.18}$$

2.6 GBS operations

Having seen how we can represent the input states, we can now look at the mathematical description for the squeezing of the input states, the interferometer transformation, and the measurement of the output state.

Squeezing is applied to each vacuum mode with the squeezing operator. For squeezing strictly in the position component, the operator $\hat{S}_x$ is defined in equation 2.19, but a more general squeezing operator that acts on any direction of the phase space can be obtained by rotating $\hat{S}_x$.

$$\hat{S}_x = \cosh(r) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \sinh(r) \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.19}$$

This is applied to the mean vector and covariance matrix of the state as shown in equation 2.20. The total amount of squeezing across all input states is what determines the average total photon number at the end of the circuit.[26]

The interferometer operation is represented by a unitary matrix $\hat{U}$ (a complex $n \times n$ square matrix with orthonormal rows and columns) determined by the parameters of the rotation and beamsplitter gates that make up the interferometer. This matrix is used to construct a $2n \times 2n$ orthogonal matrix $\hat{O}$ which applies the interferometer transformation on both the position and momentum variables of the n modes by acting on the mean vector and covariance matrix of the squeezed input states in the following manner:

$$\bar{\mathbf{r}}_{out} = \hat{O}\hat{S}_x \bar{\mathbf{r}}_{in} \qquad \sigma_{out} = \hat{O}\hat{S}_x \sigma_{in} \hat{S}_x{}^\intercal \hat{O}^\intercal \tag{2.20}$$

We can then use projective measurement to determine whether there are photons present in each mode. The on/off photodetectors used in these experiments allow for two possible outcomes: no photons present which can be measured with a projector to the vacuum

$|0\rangle\langle 0|$, and any other number of photons present which is measured with the projector $\mathbb{I} - |0\rangle\langle 0|$ instead of $|1\rangle\langle 1|$ to account for the fact that a click represents one or more photons. Projectors corresponding to each bit of an outcome represented by bitstring $\mathbf{x}$ can be assembled into an $n \times n$ diagonal matrix $P_\mathbf{x}$ to represent the measurement of all modes. We can apply the measurement by calculating the expectation value of $P_\mathbf{x}$ for output state ρ (with mean vector $\bar{\mathbf{r}}$ and covariance matrix σ') using formula 2.13

$$p(\mathbf{x}) = \mathrm{Tr}(\rho P_\mathbf{x}) \tag{2.21}$$

2.6.1 Applying noise

The models and representations we have described so far have not accounted for noise in the experimental procedure. The type of noise we will simulate is photon loss. This is assumed to be local to each mode and uniform in both directions, i.e. equal in each mode and constant throughout the length of the circuit. It is known that under these circumstances (uniform and local loss), losses can be compounded at the beginning or the end of the circuit without changing their effects on the output state.[27;28] I use this to apply losses to the interferometer output state. Photon loss re-scales the mean vector of the state by $1/\sqrt{T}$, but in our case, the mean vector is zero so it does not modify it. It does however change our state's covariance matrix.

Photon loss is quantified by the transmission rate T. The loss rate is then $(1-T)$ and the covariance matrix after losses is given by

$$\tilde{\sigma} = T\sigma + (1-T)\mathbb{I} \tag{2.22}$$

where $0 \leq T \leq 1$.

Chapter 3

Fourier Analysis for GBS

In this chapter, I will go over some basic topics in Fourier analysis of Boolean functions and see how we can combine these with concepts from Chapter 2 to derive an analytical expression for Fourier coefficients. I will then introduce Fourier weights and explain how they are used in the context of this project.

3.1 Fourier analysis of Boolean functions

In its most general definition, Fourier analysis consists of decomposing complex functions into a series, or expansion, of simpler functions. Specifically, we are interested in decomposing real-valued Boolean functions—functions that map a binary bitstring vector of length n to a real number.[29]

$$f : \{0,1\}^n \to \mathbb{R}$$

This mapping represents the sampling process we are trying to simulate; taking a bitstring $\{0,1\}^n$ that represents the measurement outcome of each mode of the circuit and mapping it to the probability of obtaining that specific combination of outputs.

The set of all functions $f : \{0,1\}^n \to \mathbb{R}$ forms a 2^n-dimensional vector space $\mathbf{V}$, where a function in this space is represented by a 2^n column vector of all values of $f(\mathbf{x})$. The spanning set of this space is the set of parity functions χ that compute the exclusive-OR (XOR) for all bits $(x_i)_{i \in S}$.

$$\chi_{\mathbf{s}}(\mathbf{x}) = \prod_{i \in \mathbf{s}} \chi(x_i) = (-1)^{\mathbf{x} \cdot \mathbf{s}} \qquad \forall \mathbf{s} \in \{0,1\}^n \tag{3.1}$$

The definition of the inner product of a pair of functions f, g in $\mathbf{V}$ is given by

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{\mathbf{x} \in \{0,1\}^n} f(\mathbf{x}) g(\mathbf{x}) \tag{3.2}$$

where a scaling factor equal to the dimension of the space is introduced to guarantee that functions in this space are unit vectors, i.e. $\langle f, f \rangle = 1$. The inner product definition can be used to prove that the parity functions form an orthonormal basis for $\mathbf{V}$.

$$\langle \chi_{\mathbf{s}}, \chi_{\mathbf{t}} \rangle = \frac{1}{2^n} \sum_{\mathbf{x} \in \{0,1\}^n} \chi_{\mathbf{s}}(\mathbf{x}) \chi_{\mathbf{t}}(\mathbf{x}) = \frac{1}{2^n} \sum_{\mathbf{x} \in \{0,1\}^n} (-1)^{\mathbf{x} \cdot (\mathbf{s}+\mathbf{t})} = \delta_{\mathbf{s},\mathbf{t}} \tag{3.3}$$

With an orthonormal basis, we can Fourier expand any function in $\mathbf{V}$, i.e. express it as a linear combination of the parity functions.

$$f(\mathbf{x}) = \sum_{\mathbf{s} \in \{0,1\}^n} \hat{f}(\mathbf{s}) \chi_{\mathbf{s}}(\mathbf{x}) \tag{3.4}$$

where the Fourier coefficients are determined by the inner product of the function with the corresponding parity function, which can be interpreted as the projection of the function onto the basis of the parity function $\chi_{\mathbf{s}}$.

$$\hat{f}(\mathbf{s}) = \langle f, \chi_{\mathbf{s}} \rangle = \frac{1}{2^n} \sum_{\mathbf{x} \in \{0,1\}^n} f(\mathbf{x}) \chi_{\mathbf{s}}(\mathbf{x}) \tag{3.5}$$

A useful classification for Fourier coefficients is the order of a coefficient, defined as the Hamming weight of the bitstring $\mathbf{s}$. This is given by the number of non-zero bits in the bitstring: $|\mathbf{s}| = \sum_i x_i$.

3.2 Fourier coefficients

We can obtain the probability of a given outcome encoded by a bitstring $\mathbf{x}$ by replacing $f(\mathbf{x})$ with $p(\mathbf{x})$ in equation 3.4.

$$p(\mathbf{x}) = \sum_{\mathbf{s} \in \{0,1\}^n} \hat{p}(\mathbf{s}) (-1)^{\mathbf{x} \cdot \mathbf{s}} \quad \text{where} \quad \hat{p}(\mathbf{s}) = \frac{1}{2^n} \sum_{\mathbf{x} \in \{0,1\}^n} p(\mathbf{x}) (-1)^{\mathbf{x} \cdot \mathbf{s}} \tag{3.6}$$

The bitstring $\mathbf{s}$ is the “label” of the Fourier coefficient and indicates which bits of the outcome $\mathbf{x}$ are taken into consideration in the calculation of that specific coefficient. For a one-mode system, the expanded calculation of the two Fourier coefficients is

$$\hat{p}(0) = \frac{1}{2}[p(0)(-1)^{0 \cdot 0} + p(1)(-1)^{1 \cdot 0}] = \frac{1}{2}[p(0) + p(1)]$$
$$\hat{p}(1) = \frac{1}{2}[p(0)(-1)^{0 \cdot 1} + p(1)(-1)^{1 \cdot 1}] = \frac{1}{2}[p(0) - p(1)]$$

The measurement outcomes for a single mode are either zero or one, so by definition, the probabilities sum up to one: $p(0) + p(1) = 1$. Therefore, we find that $\hat{p}(0)$, the Fourier coefficient for $\mathbf{s} = 0$, is $1/2$. Conversely, for $\mathbf{s} = 1$, the Fourier coefficient $\hat{p}(1)$ is computed as the difference between the probability of the mode yielding outcome zero and the probability of the mode yielding outcome one.

This procedure for bits in **s** equal to one extends to coefficients of larger systems. To illustrate this, let's consider a Fourier coefficient of a two-mode system:

$$\begin{aligned}\hat{p}(10) &= \frac{1}{4}\left[p(00)(-1)^{00\cdot 10} + p(01)(-1)^{01\cdot 10} + p(10)(-1)^{10\cdot 10} + p(11)(-1)^{11\cdot 10}\right] \\ &= \frac{1}{4}\left[p(00) + p(01) - (p(10) + p(11))\right] \\ &= \frac{1}{4}\left[p(0\mathrm{X}) + p(0\mathrm{X}) - (p(1\mathrm{X}) + p(1\mathrm{X}))\right]\end{aligned}$$

In the last line, the second bit is replaced with an 'X' to draw attention to how the sign of each probability term is determined by its first bit, namely that only probabilities of outcomes with a click in the first mode are negative. This effectively corresponds to calculating the difference between the marginal probabilities of the first mode yielding outcomes zero and one.

We see that the calculation of the Fourier coefficient is governed by the values of the bits in **s**. Calculating a Fourier coefficient corresponds to the calculation of the difference in marginal probabilities for all bits for which $s_i = 1$, scaled by a factor of $1/2^n$. We see this in both the calculation of $\hat{p}(1)$ and $\hat{p}(10)$. Conversely, $s_i = 0$ indicates that those bits do not contribute to the calculation of the coefficient, as we see for $\hat{p}(10)$ where the second bit is not relevant, and in $\hat{p}(0)$ which is just 1 scaled by $1/2^n$.

We can obtain this exact behaviour by applying the Pauli-Z operator to modes corrisponding to $s_i = 1$, and the identity matrix to modes with $s_i = 0$. Indeed, the Pauli-Z operator comprises the difference between the projective measurements to $|0\rangle$ (vacuum) and to $|1\rangle$ (one photon).

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1| \tag{3.7}$$

For each **s** can create a composite operator $Z_{\mathbf{s}}$ which contains Z for each mode for which $s_i = 1$ and $\mathbb{I}$ for each mode for which $s_i = 0$. For example, the bitstring $\{101\}$, has

$$Z_{\mathbf{s}} = Z_1 \otimes \mathbb{I}_2 \otimes Z_3$$

The Fourier coefficient for **s** is then given by the expectation value of $Z_{\mathbf{s}}$ for the output state ρ, divided by 2^n.

$$\hat{p}(\mathbf{s}) = \frac{1}{2^n} \sum_{\mathbf{x} \in \{0,1\}^n} p(\mathbf{x})\,(-1)^{\mathbf{x}\cdot\mathbf{s}} = \frac{1}{2^n}\langle Z_{\mathbf{s}}\rangle_{\rho} \tag{3.8}$$

3.2.1 Fourier coefficients with on/off photodetectors

In the context of GBS, the use of $|1\rangle\langle 1|$ is not entirely accurate because the on/off photodetectors return a one (i.e. a click) for any photon number that is not zero. Instead, we can use a projection to all states that are not $|0\rangle$ as introduced in Section 2.6, and redefine $\hat{Z}$ as

$$Z' = |0\rangle\langle 0| - (\mathbb{I} - |0\rangle\langle 0|) = 2|0\rangle\langle 0| - \mathbb{I} \tag{3.9}$$

The operator $Z_\mathbf{s}$ in the example with bitstring $\mathbf{s} = \{101\}$ then becomes

$$Z_\mathbf{s} = Z_1' \otimes \mathbb{I}_2 \otimes Z_3' = (2|0\rangle\langle 0|_1 - \mathbb{I}_1) \otimes \mathbb{I}_2 \otimes (2|0\rangle\langle 0|_3 - \mathbb{I}_3)$$

The Fourier coefficient equation (3.8) is otherwise unchanged. Using equation 2.13, we can calculate the expectation value $\langle Z_\mathbf{s}\rangle_\rho$ with the trace of $\rho Z_\mathbf{s}$. For a three-mode system, this looks like:

$$\begin{aligned}\hat{p}(101) &= \frac{1}{8}\mathrm{Tr}[\rho(2|0\rangle\langle 0|_1 - \mathbb{I}_1) \otimes \mathbb{I}_2 \otimes (2|0\rangle\langle 0|_3 - \mathbb{I}_3)] \qquad (3.10)\\ &= \frac{1}{8}\mathrm{Tr}[\rho(2|0\rangle\langle 0|_1) \otimes \mathbb{I}_2 \otimes (2|0\rangle\langle 0|_3)] + \frac{1}{8}\mathrm{Tr}[\rho(2|0\rangle\langle 0|_1) \otimes \mathbb{I}_2 \otimes (-\mathbb{I}_3)]\\ &+ \frac{1}{8}\mathrm{Tr}[\rho(-\mathbb{I}_1) \otimes \mathbb{I}_2 \otimes (2|0\rangle\langle 0|_3)] + \frac{1}{8}\mathrm{Tr}[\rho(-\mathbb{I}_1) \otimes \mathbb{I}_2 \otimes (-\mathbb{I}_3)]\\ &= \frac{1}{8}[4\,\mathrm{Tr}[\rho|00\rangle\langle 00|_{13}] - 2\,\mathrm{Tr}[\rho|0\rangle\langle 0|_1] - 2\,\mathrm{Tr}[\rho|0\rangle\langle 0|_3] + 1]\end{aligned}$$

where we use equation 2.12 to simplify the last term. Generalizing to any bitstring $\mathbf{s}$ we obtain a formula to calculate any Fourier coefficient

$$\hat{p}(\mathbf{s}) = \frac{1}{2^n} \sum_{c \in \mathcal{P}(s_i=1)} (-1)^{k-|c|} 2^{|c|} \mathrm{Tr}[\rho(\textstyle\bigotimes_{i\in c} |0\rangle\langle 0|_i)] \qquad (3.11)$$

where k is the order of $\mathbf{s}$ and the summation is taken over the power set $\mathcal{P}$ of all indices i for which $s_i = 1$. A power set is a set that includes all possible subsets, including the original set and an empty set. The case of the empty set corresponds to the projection operator to no modes which is given by $\mathbb{I}$.

Each trace term corresponds to the projective measurement of a specific subsystem of the modes specified by the vacuum projection operators. We can convert each trace into an integral of Wigner functions using equation 2.16. Both the vacuum projector and final state ρ are Gaussian. We know that the vacuum state has mean vector $\bar{r} = (0,0)^\mathsf{T}$, but so does the output state, as neither the squeezing operator nor the operations within the interferometer displace the modes. This indicates that the result will not depend on $\bar{r}$. Recall that the covariance matrix of the vacuum is $\mathbb{I}$.[1] With this information we can obtain the numerical value of each trace term.[30]

$$Tr[\rho|0\rangle\langle 0|_c] = \iint d\mathbf{x}d\mathbf{p}W_\rho W_0 = \frac{2^{|c|}}{\sqrt{det(\sigma_c + \mathbb{I}_c)}} \qquad (3.12)$$

The determinant is taken over the submatrix of σ_ρ with the rows and columns specified by set c. Similarly, $\mathbb{I}_c$ will be the same size as the submatrix. Using this result we construct a formula for the numerical value of any Fourier coefficient.

$$\hat{p}(\mathbf{s}) = \frac{1}{2^n} \sum_{c \in \mathcal{P}(s_i=1)} (-1)^{k-|c|} \frac{2^{2|c|}}{\sqrt{det(\sigma_c + \mathbb{I}_c)}} \qquad (3.13)$$

[1] It is common and convenient to set $\hbar$ to a particular value. From here on I use $\hbar = 2$ for calculations. Therefore, $\sigma = (\hbar/2)\mathbb{I}$ reduces to $\sigma = \mathbb{I}$.

3.3 Fourier weights

The Fourier weight is a useful method of grouping together Fourier coefficients of the same order. For each order k, the Fourier weight is given by[2]

$$W^k(p) = \sum_{\mathbf{s}:|\mathbf{s}|=k} \hat{p}(\mathbf{s})^2 \tag{3.14}$$

This descriptor forms the basis of the analysis conducted in this project as it allows us to observe the collective behavior of Fourier coefficients of varying orders. By computing the Fourier weights of all orders from 1 to n—where n is the number of modes in a given system—for both ideal and lossy systems, we can compare any trends that emerge and determine whether Fourier coefficients decay as order increases in noisy systems without having to compare each Fourier coefficient individually.

However, the varying number of Fourier coefficients per order would make it challenging to infer relative trends. For instance, mid-order Fourier weights would be biased to higher values simply because there are more Fourier coefficients to sum over. To obtain a more insightful measure, I normalise the Fourier weight by dividing it by the number of Fourier coefficients of the order. This count is determined using the combinatorial formula $\binom{n}{k}$. Any references to Fourier weight from now on will assume that it has been re-scaled.

With the aims of capturing general trends and analysing the average-case behaviour of Fourier coefficients in mind, it will not suffice to look at the Fourier weights of a single system. Instead, I calculate the mean Fourier weights of a large number of output states. To ensure that the mean of the output states considered is representative of all possible output states, I generate random unitary matrices for the interferometer transformation that produces the output states (see Section 4.1 for more details on randomization).

3.3.1 Convergence

If we want to calculate the Fourier weights for each order of a system, we will need to compute all possible Fourier coefficients. The number of these increases exponentially (2^n) with the number of modes in the system n.

For a relatively small system of 10 modes, calculating the 1024 Fourier coefficients took approximately 10 minutes of computing time. Upon scaling the system to 20 modes, the number of coefficients increases to over a million (1,048,576). Additionally, it is important to note that computation time does not scale linearly with the number of Fourier coefficients. There is also an exponential increase in the computation time for a single Fourier coefficient, due to the determinant calculation involved. Therefore, the task of calculating Fourier weights for each order per unitary quickly becomes impractical for system sizes above 10, especially considering that this calculation will be repeated for numerous output states.

[2] Note that W^k refers to the Fourier weight of order k, whereas W_ρ refers to the Wigner function of state ρ.

Fortunately, the mean of Fourier coefficients of the same order converges to the same value when calculated over multiple random unitaries. We can understand why we expect this with the proof outlined below.

Consider the task of calculating the same Fourier coefficient with modes i and j yielding outcome one for all possible output states resulting from all possible unitaries U. The expression for the calculation would be

$$\int d\bar{\Phi}\, p(\bar{\Phi})\mathrm{Tr}(U_{\bar{\Phi}}\,\rho\, U^{\dagger}_{\bar{\Phi}} A_{ij}) \tag{3.15}$$

The integration is over all possible sets of parameters $\bar{\Phi}$ which define the different unitaries and their probability $p(\bar{\Phi})$ of occurring. The unitaries transform the initial state ρ to obtain the output state, and the operator A_{ij} represents the operation on modes i, j that gives the Fourier coefficient. Consider now a new set of unitaries $U_{\bar{\nu}} = PU_{\bar{\Phi}}$ obtained by permuting the modes of the $\bar{\Phi}$ unitaries. The distribution of probabilities of these unitaries does not change when we shift the labels of the rows and columns of the matrices, hence $\int d\bar{\Phi} p(\bar{\Phi}) = \int d\bar{\nu} p(\bar{\nu})$. We can use this equality, the fact that the permutation operator is Hermitian, and the linear and cyclical properties of the trace to rearrange expression 3.15.

$$\begin{aligned}
\mathrm{Tr}\left[\int d\bar{\Phi} p(\bar{\Phi})\, U_{\bar{\Phi}}\,\rho\, U^{\dagger}_{\bar{\Phi}} A_{ij}\right] &= \mathrm{Tr}\left[\int d\bar{\nu} p(\bar{\nu})\, P U_{\bar{\nu}}\,\rho\, U^{\dagger}_{\bar{\nu}} P^{\dagger} A_{ij}\right] \\
&= \mathrm{Tr}\left[\int \underbrace{d\bar{\nu} p(\bar{\nu})\, U_{\bar{\nu}}\,\rho\, U^{\dagger}_{\bar{\nu}}}_{= d\bar{\Phi} p(\bar{\Phi})\, U_{\bar{\Phi}}\,\rho\, U^{\dagger}_{\bar{\Phi}}} P\, \underbrace{A_{ij} P^{\dagger}}_{= A_{xy}}\right]
\end{aligned}$$

This indicates that when summing over a large number of unitaries, the value of the Fourier coefficient of modes i, j will converge to that of modes x, y. Extrapolating from this, we infer that the values of Fourier coefficients of the same order will all converge to the same value.

Convergence is helpful in two ways. Firstly, we can deduce that the mean of the Fourier weight will also converge to a single value when taken over sufficient unitaries, given that it is the average of all of the coefficients of a given order. Secondly, it simplifies the calculation of mean Fourier weights by allowing us to use just one coefficient per order per unitary. This is possible because when coefficients converge, summing over the same value x times and dividing by x returns the original value. Note that these simplifications only hold when we are calculating the mean Fourier weights over numerous unitaries. If we instead wanted to calculate the Fourier weight of order 2 of a 10-mode system, we could not take advantage of convergence and would have to compute all 45 Fourier coefficients to obtain the Fourier weight. A visual representation of the convergence of Fourier coefficients is shown in Section 4.3 and the method for determining the number of required random unitaries for convergence is described in Section 4.3.1.

Chapter 4

Implementations and tests

In this chapter, I will discuss how I set up the analysis, what techniques I use, and what tests I perform to guarantee their functionality. I also go over Python implementations of some key functions I used for this project.

4.1 Random unitaries

As mentioned in the previous chapter, we need to calculate a large number of Fourier coefficients to analyze general trends in the behaviors of Fourier coefficients of different orders. The output states used to calculate these should span the entire phase space to ensure that the conclusions we derive are robust and widely applicable.

This is reflected in the choice of the unitary matrices used to perform the interferometer transformation. Knowing that an interferometer can be deconstructed into one or two qubits gates described by unitary matrices, the next logical step is to randomly sample rotation angles for the rotation matrices and reflection, as well as transmission coefficients for the beamsplitter matrices, from their respective ranges of possible values to create the $n \times n$ unitary matrix. The issue with this approach is that sampling parameters from a flat uniform distribution does not result in a uniform distribution of unitary matrices and consequentially, of output states in n-dimensional phase space.[31] This can be corrected by introducing a Haar measure—a correction factor that reweighs the parameters based on their contributions to the final distribution in phase space. However, this correction needs to be added for every parameter in the $n \times n$ unitary, and this becomes both complex and inefficient as the dimensions of the system increase.

A quicker way to obtain Haar-random unitaries is to use QR decomposition on complex-valued matrices. The procedure I used to obtain random unitary matrices is based on a 2006 paper on generating random matrices from classical compact groups:[31]

1. Create a $n \times n$ matrix Z of random complex numbers obtained by sampling the real and imaginary parts from a normal distribution.
2. Preform a QR decomposition of Z using the `scipy.linalg` library to obtain a an orthonormal matrix Q and the upper triangular matrix R.

3. Extract the diagonal elements from R and construct a new diagonal matrix P where each element is given by the ratio between the corresponding element of R and its absolute value: `p = np.diag(np.diag(R)/np.absolute(np.diag(R)).`

4. Multiply the matrices to obtain the unitary Haar-random matrix: $U = QPQ$.

4.2 Calculating Fourier coefficients

To calculate the Fourier coefficients, we have to use random unitaries to transform the input state and then extract the covariance matrix of the final state. This can be done with the `StrawberryFields` Python library[21;32] created by Xanadu Quantum Technologies, which allows users to run programs on their virtual quantum computer backends.

Using this library, these are the steps I took to implement the GBS circuit:

1. Create a circuit (a `Program` object) with n input states.
2. Apply squeezing gates with a squeezing parameter r (`Sgate(r)`) to the input states.
3. Apply the interferometer operation with unitary matrix U obtained with the procedure described above.
4. Initialise an `Engine` with Guassian backend to execute the program on the Strawberry Fields state simulator which returns an output `State` object.

Then, using the `Walrus` library, I extracted the covariance matrix from the `State` object in the form of a `numpy` matrix, which I used to calculate the Fourier coefficients. Pseudocode for my function can be found in Algorithm 1.

To calculate the desired Fourier coefficients using equation 3.13, I extract the rows and columns using `numpy.take`. The row and column indices are specified by the set c (where $c \in \mathcal{P}$). As an example, the power set for the Fourier coefficient of the bitstring `10010` is $\mathcal{P} = \{\{1,4\},\{1\},\{4\},\{\}\}$.

Note that the covariance matrix will be of size $2n \times 2n$ as it contains the momentum and position variance for each mode, so the rows and columns taken will be given by $c' = c \cup \{x+n|x \in c\}$.

To make the algorithm more efficient, in the `if-else` statements I skip the bulk of the operations if the current set is empty because the contribution to the summation is just one, as we see in the last term in equation 3.10.

4.2.1 Verification

To verify the correctness of this procedure and of the Fourier coefficient values I obtained, I compared them with the actual Fourier coefficients of GBS output states. `StrawberryFields` does not provide GBS Fourier coefficients, but we can obtain them by inserting the probabilities generated with the function `threshold_detection_prob` from the `Walrus` library into the Fourier coefficient formula 3.6.

Algorithm 1

```
Input: cov: array, bits: list of bits in s
function FourierCoefficient(cov, bits)
    onebits ← [i for (i, x) in enumerate(bits) if x =1]
    order ← length of onebits
    combinations ← PowerSet(onebits)
    coef ← 0
    for combination ∈ combinations do
        size ← length of combination
        sign ← $(-1)^{order-size}$
        if size = 0 then
            coef ← coef + sign
        else
            n ← (length of cov) / 2
            indices ← combination + [x + n for x in combination]
            submatrix ← TakeRowsAndColumns(cov, indices)
            det ← Determinant(submatrix + Identity(size * 2) )
            temp ← $2^{2*size}$ / Sqrt(det)
            coef ← coef + (sign * temp)
    return coef
```

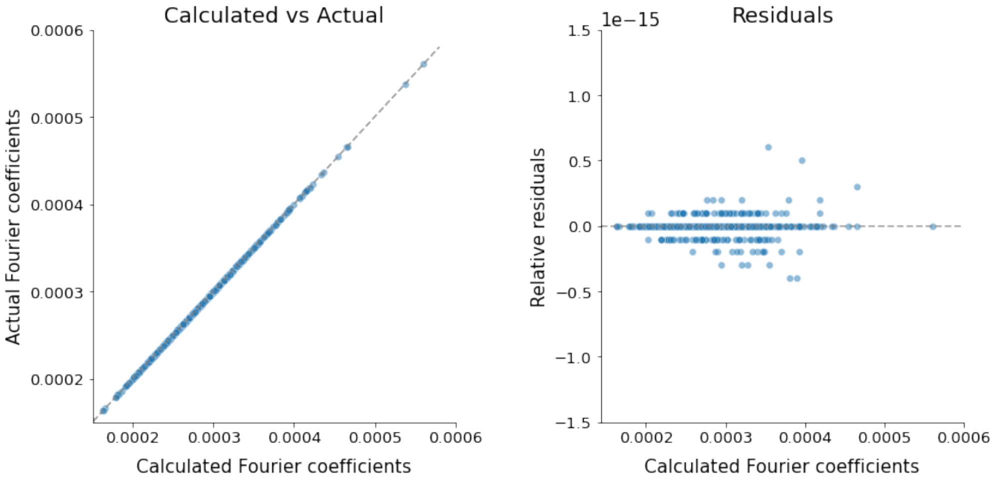

Figure 4.1: Comparison of calculated Fourier coefficients with coefficients obtained from `StrawberryFields` probabilities. 10-mode system (1024 Fourier coefficients).

In Figure 4.1, I compare the Fourier coefficients I calculated for a 10-mode system with coefficients obtained from `StrawberryFields` probabilities. In the left plot, we see good agreement between the two sets of values. The residual plot (right) confirms this, with no relative residual larger than 10^{-15}. A numerical analysis further confirms these observations, finding that the average distance of the calculated coefficients from their true value is $9.1 \times 10^{-12}\%$.

From this plot, we can also garner the typical magnitude of a Fourier coefficient. To

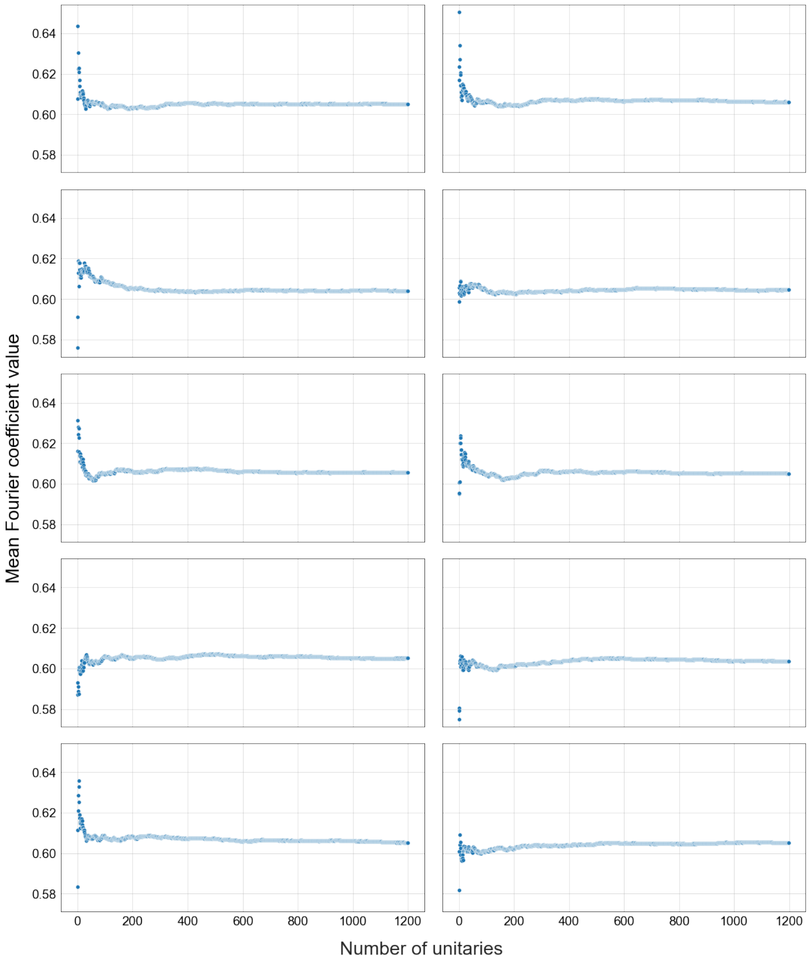

Figure 4.2: Cumulative mean of the ten first-order Fourier coefficients (`1000000000`, `0100000000`, ...) of a 10-mode system over 1200 different random unitaries.

avoid working with decimal numbers with multiple leading zeros or with scientific notation, I will re-scale the coefficients by 2^n, essentially excluding the denominator in equation 3.5. This will not impact the relative behavior of the Fourier coefficients and this factor can be reintroduced if any quantitative analysis regarding the values of single coefficients is required.

4.3 Visualising and quantifying convergence

In Chapter 3, I mentioned how we can use convergence of the mean value of Fourier coefficients of the same order to simplify the calculation of mean Fourier weights. To gain a better intuition of what convergence looks like, Figure 4.2 shows the cumulative mean of the ten first-order coefficients of a 10-mode system as we add new values from new random unitaries.

Despite varying behaviors in the first 100 unitaries, the cumulative means slowly converge to one value as expected. The standard deviation of the mean Fourier coefficients after 1200 unitaries is 6.11×10^{-4}—less 0.1% of the values.

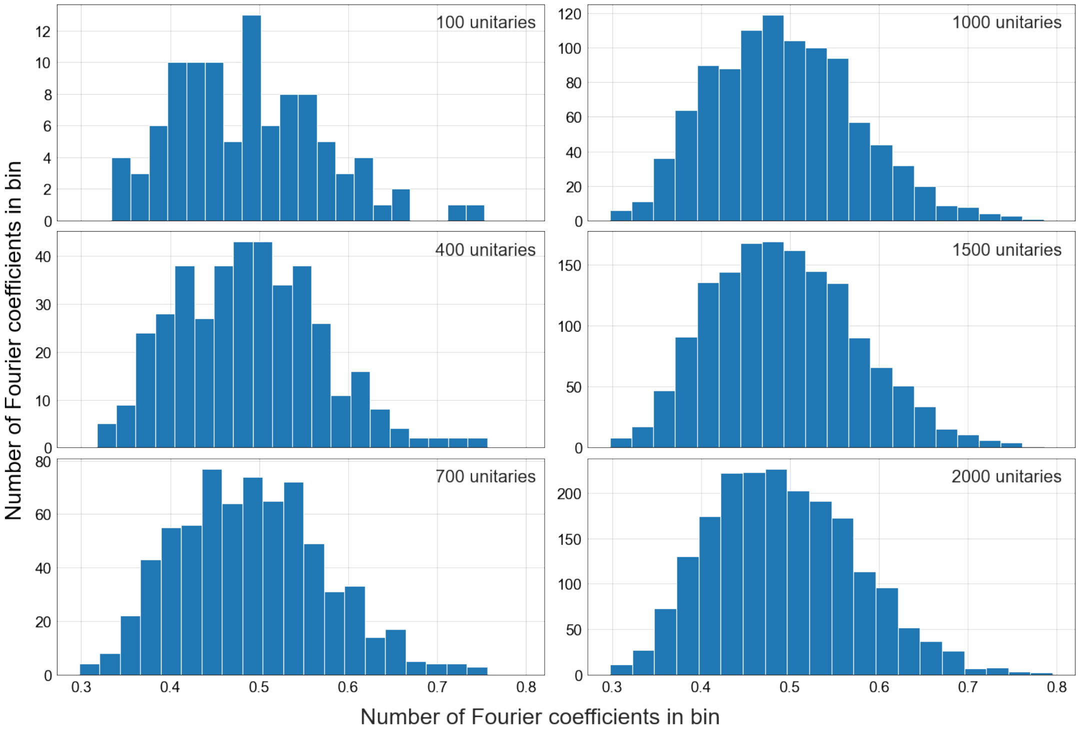

Figure 4.3: Distribution of Fourier coefficients as sample size increases for the $k = 3$, $n = 5$ bitstring `00111`.

When we calculate the first-order mean Fourier weight of a 10-mode system, we can see that calculating all 10 Fourier coefficients would be redundant. We can instead pick one Fourier coefficient value and square it to obtain the mean Fourier weight. This is only possible if the mean is taken over sufficient unitaries—from the plot, we see that with a small number of unitaries, the mean Fourier coefficient values are not the same.

4.3.1 Determining sample size

In order to calculate a single coefficient per Fourier weight, I needed to to determine a sufficient sample size (i.e. number of random unitaries) such that the coefficients converge and the simplification holds. We expect both the value at which Fourier coefficients converge and how long they take to converge to vary with the number of modes n and the order k of the coefficient's bitstring. Therefore, I needed a metric to evaluate convergence which can vary accordingly.

Plotting the distribution of Fourier coefficients of a single bitstring (in this case `00111`) for different output states (see Figure 4.3), we notice that the distribution slowly becomes more Gaussian-like. Knowing this, we can measure both the mean and the standard deviation as sample size increases, and once the fractional change in both of these measures drops below a certain threshold and stays below it for multiple sample size increases, we can be certain that the mean has converged. The threshold for the two fractional changes is fixed for coefficients of all sizes and order, as the fractional change already accounts for the variations caused by these variables.

Using 0.005 for the thresholds of fractional changes in mean and standard deviation respectively, and requiring the fractional changes to remain under said threshold for 50 consecutive increases in sample size (i.e. 50 additional coefficients calculated from new unitaries), I obtained the sample sizes required for convergence for 10-mode Fourier coefficients. The values vary slightly when repeating the algorithm so Table 4.1 shows the averages over 5 runs for each order.

Order	# of unitaries	Order	# of unitaries
1	426	6	437
2	440	7	480
3	476	8	448
4	446	9	471
5	441	10	440

Table 4.1: Number of unitaries required for convergence in a 10-mode system.

I used the highest value for all orders for convenience. Therefore, for a 10-mode system, I used 480 unitaries. Following this procedure, I also found that the sample size required for 20 modes is 496 unitaries.

For 100-mode systems, the computation time of higher order single Fourier coefficients increases beyond what is feasible for this project because of both the number of terms in the summation and the size of the matrices we take the determinant of, so I only explored the Fourier weights of the lower orders. The limit for what was achievable with my computer (AMD Ryzen 7 3700U, 16 GB RAM) within a 72-hour window was computing the first 12 orders of Fourier coefficients. The sample size required for convergence for these orders was 485 unitaries.

I verified that I had obtained a satisfying level of convergence by cross-checking with the data used in Figure 4.2. The cumulative means of the 10 Fourier coefficients after 480 unitaries have a standard deviation of 1.12×10^{-3}, confirming that all 10 values are very similar. Furthermore, the difference between the mean of all the means at 480 unitaries and 1200 unitaries is less than 0.03%, meaning that the value that the 10 coefficients converge to is practically the same after applying 480 and 1200 unitaries. This confirms sufficient convergence after 480 unitaries.

Chapter 5

Results

In this chapter, I present the results of my analyses; first looking at an ideal system to establish a baseline, and then introducing noise to observe its impact. For each case, I also assess the influence of the squeezing parameter utilized.

5.1 Ideal case

I studied the behaviour of Fourier weights for varying system sizes. The presented results are categorized based on system size to explicitly illustrate how trends evolve as we transition to more realistic experimental system sizes.

5.1.1 20 modes

To determine the mean Fourier weights for a 20-mode system, I used 500 random unitaries—rounding up from the sample size of 496 found in Section 4.3.1. The results are plotted against the order of the Fourier weight in Figure 5.1. Note that the order zero weight is not included as Fourier coefficients of order zero are always equal to 1 (re-scaled to the normalization of the probability distribution with a factor of $1/2^n$).

Higher values of Fourier weights at small and large orders suggest a non-uniform contribution to the probability calculation (equation 3.4) from different Fourier coefficient orders. This is in stark contrast to Fourier coefficients of an ideal random quantum circuit probability distribution, where all Fourier coefficients are found to have a value of $1/2^n$.[20] Uniformity of these coefficients in RQCs is attributed to the aformentioned lack of structure in the distribution, resulting in an even span across the Hilbert space. This characteristic is what renders classical simulation of random quantum circuits computationally expensive. Conversely, the concentration of Fourier coefficients shown here supports the notion of a discernible structure in GBS distributions, potentially facilitating an efficient classical simulation of GBS.

It is important to emphasize that the Fourier coefficients are squared before averaging when calculating the Fourier weights. Therefore, very low Fourier weight values in the

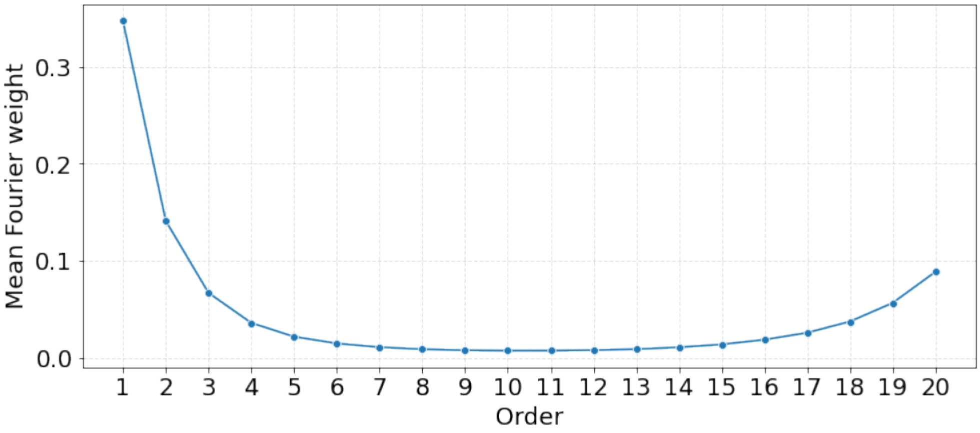

Figure 5.1: Mean Fourier weights for an ideal 20-mode system (squeezing $r = 0.5$, 500 random unitaries).

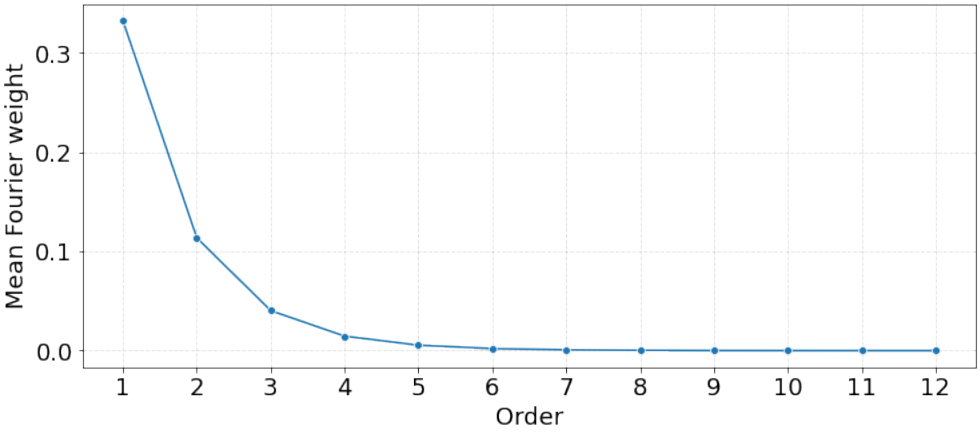

Figure 5.2: Mean Fourier weights for an ideal 100-mode system (with squeezing $r = 0.5$, 500 random unitaries).

middle orders are not attributable to negative and positive coefficients canceling out and instead are indicative of a distinct behavior inherent to GBS.

5.1.2 100 modes

As mentioned in Section 4.3.1, the computation time for Fourier coefficients grows exponentially with order. This makes the 20-mode system the biggest system size for which calculating the Fourier weights for all orders is practical. However, even for this size, computing the Fourier weights for the first ten orders takes around 20 minutes, while the last three orders require over 4 hours.

For a 100-mode system, I therefore only calculate the first twelve orders. Extrapolating from the 20-mode system, this is where the majority of the concentration of coefficients is found. I once again round up from the sample size of 480 found in Section 4.3.1 and use Fourier coefficients from 500 unitaries to compute the mean Fourier weights.

Since we cannot know the Fourier weights of the last few orders, we cannot be entirely

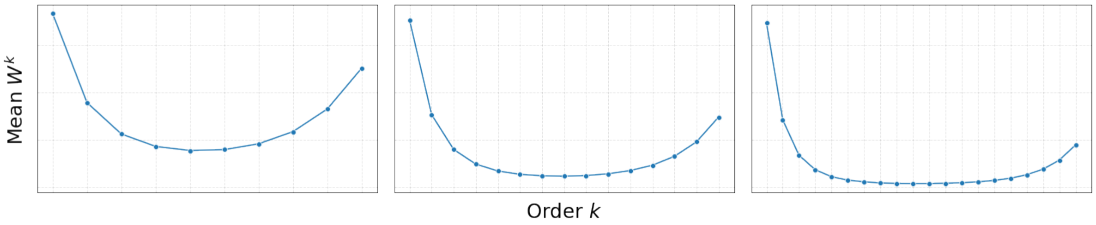

Figure 5.3: Left to right: mean Fourier weights for ideal systems with 10, 15, and 20 modes (squeezing $r = 0.5$).

sure that the 100-mode system exhibits the same U-shaped curve as smaller-sized systems. We can however extrapolate from the differences between 10-mode, 15-mode, and 20-mode graphs in Figure 5.3. These all exhibit a U-shaped curve with a decrease in the values of the central orders as system size increases. The 100-mode data seems to follow this pattern, approaching zero around order 7. We can assume that the 100-mode curve would also have very small Fourier weights in the central orders and a left tail that increases for the last few orders but is much smaller than the right one.

5.1.3 Ideal case with varying squeezing

As mentioned in Section 2.6, total squeezing across modes is directly related to the mean number of photons in output states. In experiments, larger squeezing parameters can be used to increase the number of photodetector clicks.[33] As an example, Table 5.1 shows the mean click numbers $\bar{c}$ obtained by different runs of the 144-mode Jiuzhang 2.0 interferometer and the mean squeezing parameter $\bar{r}$ used. These offer an intuition into the relationship between squeezing and clicks, but it is important to remember that there are other systematics, notably noise, that also influence the mean click number.

$\bar{r}$	$\bar{c}$
0.52	8
0.83	19
1.26	43

Table 5.1: Jiuzhang squeezing and mean clicks.

To assess the impact of squeezing on Fourier coefficients, I examined the variation in Fourier weights across experiments initialized with different parameters; plotted in Figure 5.4. The parameters analysed range from $r = 0.25$ to $r = 1.5$, because average squeezing parameters from experimental data rarely exceed 1.5. I conducted this analysis for 20 modes, since I was able to simulate the whole system.

We find that higher squeezing parameters result in a lower mean Fourier weight for large orders. This preliminarily suggests that high squeezing parameters can cause a concentration of Fourier coefficients towards lower orders. Consequently, by reducing the contribution of higher orders in the Fourier expansion in equation 3.6, high squeezing parameters can contribute to making a system easier to classically simulate.

Clearly, this is only applicable to certain squeezing parameter values. With small squeezing like $r = 0.25$, the Fourier series would have to be calculated in its entirety as the high-order Fourier weights imply that there would be non-zero Fourier coefficients in the highest orders of the expansion.

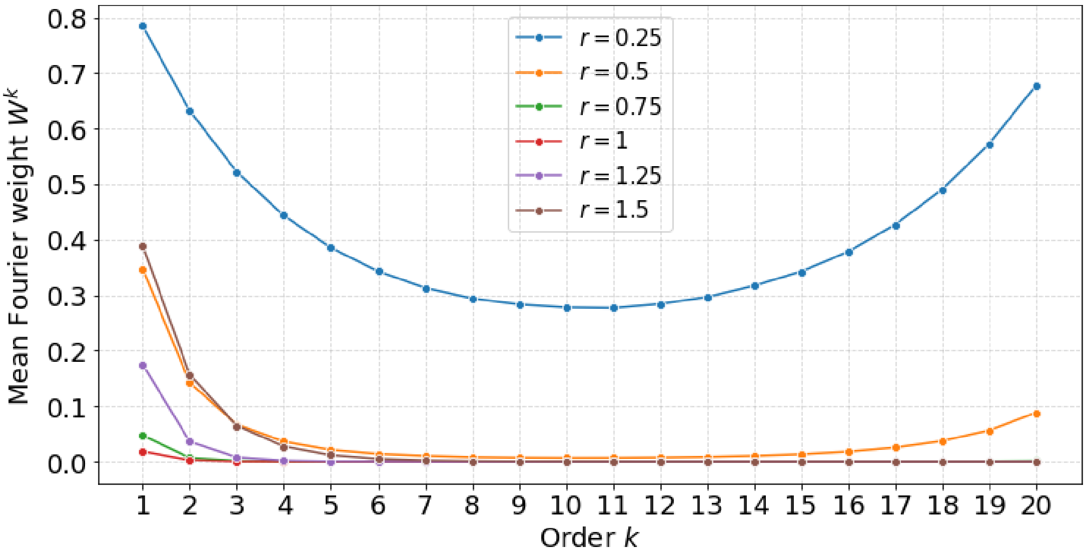

Figure 5.4: Mean Fourier weights for ideal 20-mode systems with varying squeezing parameters r.

It is interesting to note that with squeezing parameters greater than $r = 1$ we obtain a slow re-emergence of the Fourier weights in orders 2-5, indicating that there is a threshold squeezing value, after which increasing squeezing further will not continue to decrease the classical complexity of the simulation.

5.2 Effects of photon loss

Now that we have looked at the trends that emerge in lossless scenarios, we can move on to noisy systems. These systems are more representative of current GBS experiments. The amount of noise introduced in each system is quantified by the transmission coefficient T—a higher T value indicates lower losses. Loss is applied with the procedure described in Section 2.6.

I compared systems initialised with the same squeezing parameter to specifically isolate the effect of photon loss. The squeezing I chose is $r = 0.5$ because it is the highest squeezing parameter that yields non-zero high-order Fourier weights, as shown in Figure 5.4.

The plot shows two edge cases. In the case where the transmission coefficient is zero, the output state will be n vacuum modes. If $\mathbf{x}$ is a bitstring of all zeros, then the sign term in equation 3.6 will always be positive. Every Fourier coefficient will then be the sum over probabilities of all possible outcomes, which is one by definition. This is indeed what we see in the plot. Subsequent plots will not include this trivial edge case. The other edge case is when we have $T = 1$. If all photons are transmitted, we simply recover the ideal case. This is included to provide a comparison to lossy Fourier weights.

We can identify two main behaviour trends from the data:

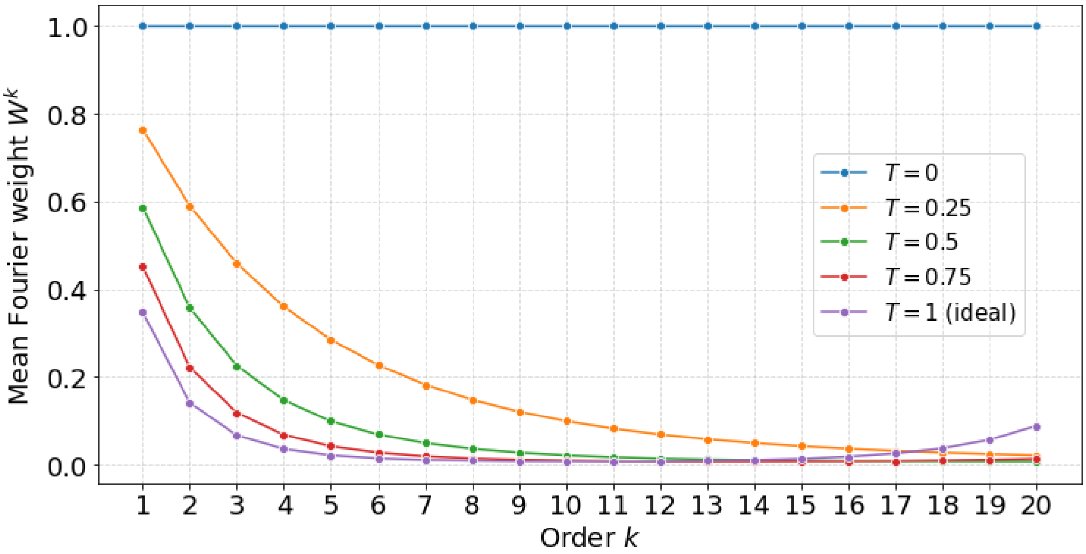

Figure 5.5: Mean Fourier weights for noisy 20-mode systems with varying losses (squeezing $r = 0.5$, 500 random unitaries).

1. **The right tail of the curve disappears when we add photon loss to the system.**

   This favours the classical simulation method based on Fourier series truncation. Higher-order Fourier weights result in larger power sets, thereby increasing the number of terms in the summation in equation 3.13. They also result in larger matrices in the determinant calculation (which is $O(n^3)$ with matrix size), thereby significantly increasing computation time. The disappearance of the left tail means that these higher-order Fourier coefficients do not need to be calculated.

2. **The distribution of coefficients is altered, resulting in the lower-order Fourier weights taking longer to approach zero.**

   This reduces the speed-up created by the first one because it adds extra non-zero terms in the summation, albeit at lower orders. Moreover, we see that as losses decrease (corresponding to higher transmission), the lower-order Fourier weights take longer to approach zero. This results in an inverse correlation between the amount of loss and the reduction in complexity of the classical simulation.

However, when considered together, they have a net positive effect on the classical simulatability of GBS—it is less computationally intensive to calculate the first 10 orders of a Fourier series compared to the first 7 and the final 3, as would be required in the ideal 20-mode case.[1]

In summary, I found that for most loss amounts, we can truncate the series and significantly speed up the classical simulation of lossy systems. On top of this, I have also established that when comparing systems with different amounts of noise, the improvement in the time complexity of their classical simulation is correlated to the decrease in photon loss.

[1] Note that even calculating all Fourier coefficients up to order 10 may be very hard for a high number of modes.

5.3 Adjusting squeezing to loss

We have seen how the choice of squeezing parameter and photon loss throughout the circuit changes the distribution of coefficients in the system. However, as we might expect, photon loss also has a large impact on the average photon number in the final state. To visualise this, I plot the probability of obtaining a click in one mode as the transmission coefficient decreases in Figure 5.6. A clear inverse relationship between transmission and click probability emerges.

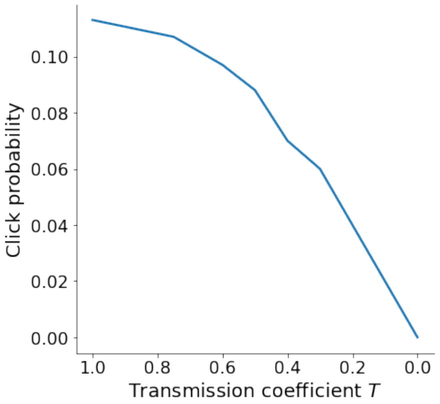

Figure 5.6: Mean probability of obtaining a click in one mode (100 unitaries per T value)

Lower photon numbers can pose issues for experiments as interference effects and the statistical significance of an experiment decrease with a decrease in photons detected.

Because of the effect squeezing has on the output state, we can counteract the decay in photon number by adjusting the squeezing parameter given the loss we expect (or in our case, we know) there to be. Furthermore, from the plot data, we see that even with 100% transmission, the mean probability of obtaining a click is only 0.113. To fix this, we can also use the squeezing parameter to calibrate the system to have an equal probability of obtaining a zero or a one in each mode and maintain that probability even if losses are introduced in the system. When the probabilities of the two outputs are balanced, we gain access to a larger space of outputs compared to scenarios where there is a high probability of predominantly 0s or 1s.

For these purposes, I derived a formula for the relationship between loss and squeezing for one mode. With equations 3.6 and 3.13, we find that the probability of obtaining a 0 in one mode is given by:

$$p(0) = \frac{2}{\sqrt{\det(\sigma + \mathbb{I})}} \tag{5.1}$$

If we want to ensure—at least until the interferometer is applied—that there is a 50/50 chance of obtaining a 1 or a 0 in a specific mode, we can set this equation equal to $1/2$. The covariance matrix of a squeezed state before the interferometer effects are applied is

$$\sigma = \begin{bmatrix} e^{2r} & 0 \\ 0 & e^{-2r} \end{bmatrix}$$

Applying uniform and independent losses before the interferometer unitary, we obtain

$$\tilde{\sigma} = T\sigma + (1-T)\mathbb{I} = \begin{bmatrix} Te^{2r} + (1-T) & 0 \\ 0 & Te^{-2r} + (1-T) \end{bmatrix}$$

We add the identity matrix to $\tilde{\sigma}$ and then take its determinant, which results in a large polynomial that can be factorized:

$$\det(\tilde{\sigma} + \mathbb{I}) = T^2(2 - e^{2r} - e^{-2r}) + T(2e^{2r} + 2e^{-2r} - 4) + 4$$

Using the trigonometric substitution $\cosh(x) = (e^x + e^{-x})/2$ and solving for r, we obtain the relationship

$$r = \frac{1}{2}\cosh^{-1}\left(1 - \frac{6}{T^2 - 2T}\right) \tag{5.2}$$

5.3.1 Verification

The squeezing/loss pairs obtained with the derived formula are shown in Table 5.2.

Loss (T)	Squeezing parameter (r)
1 (ideal)	1.31
0.75	1.35
0.5	1.44
0.25	1.69

Table 5.2: Squeezing parameters to correct loss effects

We can confirm these values by checking if the output states of one-mode systems with these parameters give outcomes 0 and 1 with equal probabilities. The probabilities are calculated with the `Walrus.threshold_detection_prob` function. For each pair of values, we get a probability of 0.498 of measuring zero photons in the mode, meaning there is an almost exact $50/50$ split between the probabilities of the two possible outcomes. Therefore, I can confirm that this formula works for creating an equal probability of obtaining 0 and 1 and maintaining this for all loss amounts.

For the one-mode case, this shows that different system parameters and properties can be used to achieve the same outcome and that when interpreting experimental results we must be careful not to correlate results that are not directly connected.

The way this scales up to a multi-mode system is not trivial, but we can once again use `threshold_detection_prob` to help validate whether the parameter pairs computed for a one-mode system are suitable approximations for larger system sizes. In a 5-mode system, if the probabilities of obtaining a click or not are equal for each mode, then the probability of any outcome should be $1/2^5$. Calculating these probabilities (before the interferometer is applied), we indeed find this to be the case.

5.3.2 Effects of adjusting squeezing

We can now visualise the effect of adjusting the squeezing parameter to losses in Figure 5.7. Here, I compare the mean Fourier weights of:

- a ideal system calibrated with the squeezing parameter $r = 1.31$;
- a noisy system with this same squeezing parameter ($T = 0.5$ not adjusted);
- a noisy system with the same amount of loss but with an adjusted squeezing parameter.

We can see that adjusting r to loss doubles the Fourier weight values from the non-adjusted noisy case, but they remain about half of the ideal ones. However, both the

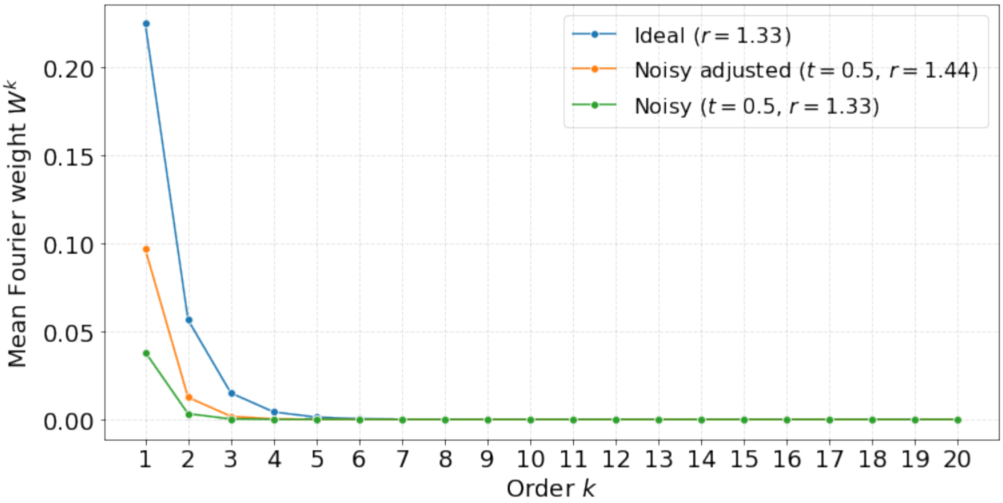

Figure 5.7: Comparison of mean Fourier weights for three different 20-mode systems.

non-adjusted and adjusted scenarios approach zero at the same order, meaning that with the classical simulation method under consideration, the Fourier series could be truncated at the same order and the simulations of the two systems would have the same time complexity.

This occurs because there is a difference between useful and non-useful photons. When squeezing and losses are high, some photons do not significantly contribute to the entanglement of the final state but their clicks are still counted.[33] If we adjust the input squeezing parameter, we do not actually prevent decoherence and increase the quantum resources (the amount of final entanglement). These are limited by the system's transmission, and we see this by the fact that even when squeezing is adjusted, the system is still equally simulatable. These findings are in excellent agreement with a 2023 paper by Oh et al.[33] that found that in lossy GBS, most of the output photons are classical and do not contribute to the high computational costs of classical simulation.

A typical way that experiments estimate the cost of classical simulations of their systems is based on the output photon number as the complexity of some of the best-known simulation methods[34;35] significantly increases with photon number. Figure 5.7 shows that this is not always the case (or at least not always applicable to Fourier series truncation methods) because the underlying dynamics are greatly influenced by the effects of squeezing and losses; and therefore that experiments might be overestimating the hardness of their results.

5.3.3 20 and 100 modes adjusted

We can now explore the trends that emerge when all loss values are adjusted with their respective squeezing parameter (as given in Table 5.2). In the left plot of Figure 5.8, we plot Fourier weights for 20-mode systems with the same losses as in Figure 5.5 but with adjusted squeezing parameters. We can see that the behavior is much more uniform across different loss values, further confirming that the adjustment formula is correct and that all the systems are calibrated equally.

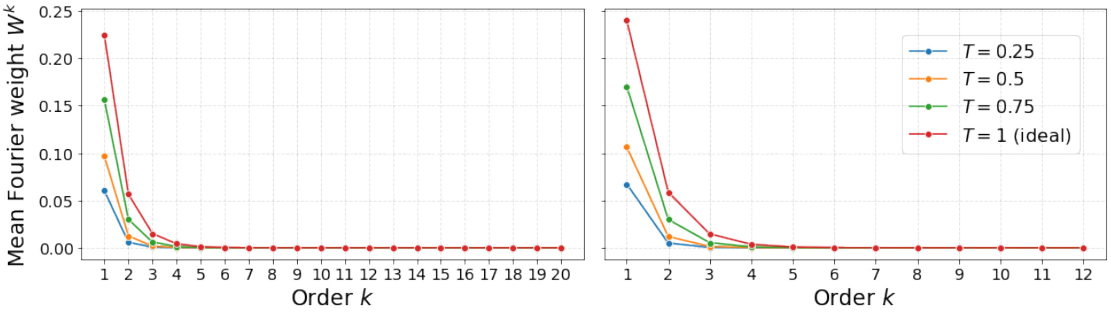

Figure 5.8: Mean Fourier weights for 20-mode (left) and 100-mode (right) noisy systems with adjusted squeezing parameters, taken over 500 random unitaries.

Furthermore, we find that with adjusted values, the rate of decay of the Fourier weights is now positively correlated with the amount of photon loss. This is opposite to the behaviour we saw for the non-adjusted noisy systems in Figure 5.5. This is also more in line with what we would expect, considering that loss degrades the entanglement and interference effects present in the system.

This plot also offers a good visualisation of how effective a simulation method involving truncating a Fourier series can be. It is evident that the computation time required to calculate Fourier coefficients beyond the fifth order is not compensated for by the minimal increase in precision we would get in return.

We observe the same uniformity in behavior across loss amounts for a 100-mode system. Interestingly, the orders at which the mean Fourier weight approaches zero are very similar to those of the 20-mode system. Although this may make it seem like the time complexity of the classical simulation of these systems is the same, it is important to remember that the amount of Fourier coefficients for each order in a 100-mode system is exponentially larger than those in a 20-mode system (e.g. there are 190 second-order Fourier coefficients in a 20-mode system and 4950 in a 100-mode one).

Although we can extrapolate from the trends we see from noisy 20-mode systems in figures 5.5, 5.7, and 5.8, this analysis does not entirely exclude the existence of non-zero mean Fourier weights at the highest orders of 100-mode systems.

5.3.4 Exponential decay

Finally, I present a result that I found when I adjusted squeezing to losses. Figure 5.9 displays the log-scaled plots for 20 and 100-modes systems. Exponential curves become straight lines in plots with a log-scaled y-axis. The lines on these plots suggest that the decay of Fourier coefficients with order is incredibly close to exponential, if not entirely so.

There seems to be oscillatory behaviour at very high orders (and correspondingly very low Fourier weights), with even-ordered Fourier weights deviating from the straight line. The origin of this behaviour is unknown, but it is important note that the values at which they happen are minuscule (less than 10^{-8}) and the Fourier series would be truncated well before these higher orders.

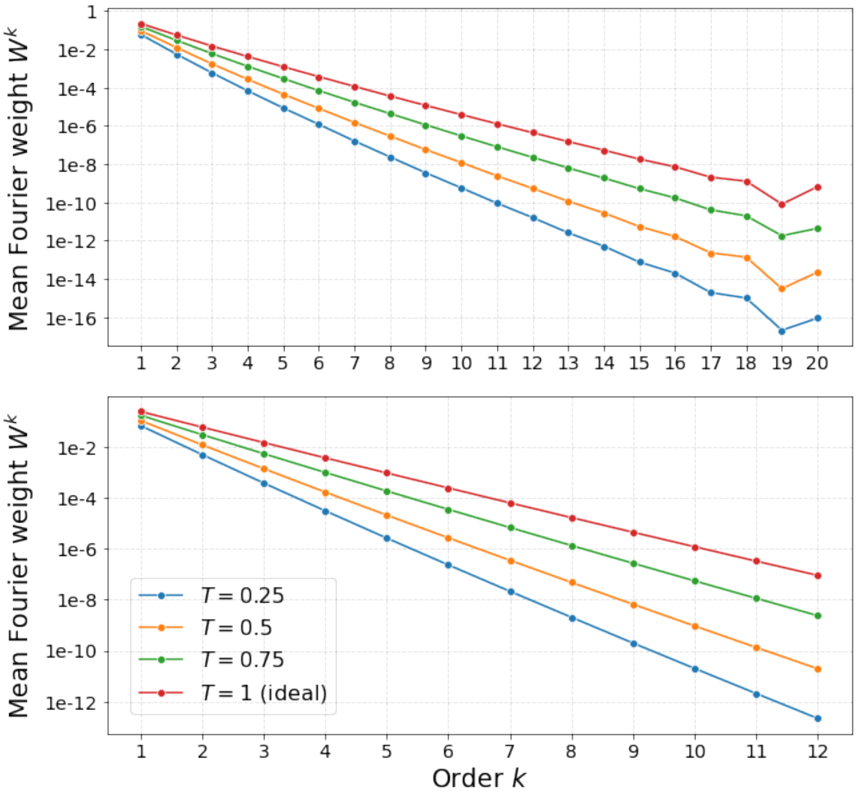

Figure 5.9: Log-scaled mean Fourier weights for noisy 20 and 100-systems with adjusted squeezing parameters, taken over 500 random unitaries.

We do not see exponential-like decay for any loss values in the non-adjusted noisy system (Figure 5.5). The fact that we only see it for adjusted systems indicates that Fourier weights approach zero faster compared to equally lossy systems without adjusted squeezing parameters. This indicates that increasing the squeezing parameters to adjust for losses makes noisy systems easier to simulate classically, as the series can be truncated at earlier orders. Log-scaled versions of plots 5.5 and 5.7 are provided in Appendix B.

These plots also offer an excellent visualisation of the inverse relationship between transmission coefficients and rates of decay. I performed linear regression to find the slopes of the 100-mode lines. These are shown in Table 5.3.

Transmission	Slope
0.25	-1.04
0.5	-0.88
0.75	-0.71
1 (ideal)	-0.58

Table 5.3: Slope of lines in log-scaled 100-mode plot. R^2 rounds to 1 in all cases.

R^2 quantifies the goodness of fit with a value between 0 and 1, where 1 is a perfect fit. The R^2 values for all fits are greater than 0.999. The slopes correspond to the exponent of the decay rate—a more negative exponent corresponds to a faster decay. The plots and the calculated slopes offer further confirmation that Fourier coefficients decay faster as we increase photon loss when the squeezing parameters have been adjusted.

When we truncate a Fourier series, there will be some error due to the fact that we are not including all orders. This error is proportional to the size of Fourier weights past the truncation order. The exponential decay we see in these plots allows us to quantify the progressive decrease in error as we increase the number of orders we decide to keep in the series. For each increasing order we keep, we will have an exponential decrease in error of our simulation compared to a non-exponential increase in extra Fourier coefficients that need to be calculated. This behaviour is very favourable to an efficient and accurate classical simulation of GBS using Fourier series.

Finally, these plots seem to suggest that ideal systems also exhibit exponential decay at high enough squeezing. Going back to our ideal system analysis, we find that the Fourier weights for the two highest squeezing parameters in Figure 5.4 also exhibit exponential decay in an ideal system. This is significant, as it would mean that even some ideal GBS experiments could be simulated efficiently, in stark comparison to ideal random quantum circuits. A log-scaled version of Figure 5.4 is also available in Appendix B.

Chapter 6

Conclusions

In this project, I analysed the behavior of Fourier coefficients of the output distributions of Gaussian boson sampling experiments, and how this behaviour is affected by squeezing and losses in the context of a classical simulation method based on Fourier series truncation. To this end, I implemented the procedure for calculating Fourier coefficients, investigated the best method to obtain random unitaries, and developed a technique to calculate the sample size required for the convergence of Fourier coefficients of the same order.

In the analysis, I first looked at ideal GBS systems. I found that the Fourier weights of these systems resemble a U-shaped curve when plotted against Fourier weight order. This points to a structure underpinning the distribution, which suggests that the classical simulation of GBS may be easier compared to random quantum circuits. I also determined that high squeezing suppresses the right tail of this curve (high orders), which corresponds to a possible reduction of the time complexity of the simulation.

I then analysed lossy systems and found that, in general, loss changes the distribution of Fourier coefficients in two ways. Firstly, it suppresses high-order Fourier weights. This significantly reduces the time complexity of classical simulation as it allows the truncation of the Fourier series. Secondly, loss causes lower-order Fourier coefficients to take longer to approach zero, which moderately reduces the speed-up achieved by the truncation of the series as it requires more orders to be calculated. Even so, the net effect of loss is a reduction in the hardness of simulation. Moreover, the extent of this reduction varies; the data shows that the increase in classical simulatability is inversely correlated to the amount of loss.

Finally, I derived a relationship between squeezing parameters and transmission coefficients that ensured an equal probability of outcomes in each mode at varying amounts of losses. Utilising the values calculated with this formula, I found that adjusting the squeezing does not increase the order at which the Fourier weights go to zero compared to equally lossy systems without adjusted squeezing. Consequently, experiments that increase the squeezing parameter to counteract losses are not also increasing the complexity of their classical simulation, despite a potential increase in output photon number.

I also found that adjusting the squeezing parameter to losses inversed the relationship between Fourier weight decay and losses that was present in the noisy non-adjusted systems. Systems with higher losses present a quicker Fourier weight decay with order, making them easier to simulate.

Plotting the log-scaled versions of the plots with adjusted squeezing, I found that these systems exhibit exponential-like decay, similar to what we know occurs in RQC. The straight lines have some fluctuations at very high orders where the Fourier weights have very small values. I found that this exponential trend is present for all transmission amounts including full transmission, which seemingly implies that with specific parameters, this Fourier series truncation method could also be used to simulate ideal GBS systems.

6.1 Research limitations and future research

6.1.1 Optimisation

The size of the systems we can analyze is currently limited by the time complexity of the Fourier coefficient calculation. I attempted to format my code so that the `@njit` decorator from the `numba` library could be used. This library translates numeric calculations in Python into optimized machine code at runtime. Since the bottleneck of my program is the `numpy` array determinant calculation, my code was an ideal candidate for `numba`. However, it only supports a subset of `numpy` functions. For it to work with my code, I had to break some steps down to use simpler functions. This reformatting ended up canceling out most of the speed-up achieved by the library, so I did not use it.

An avenue that would be worth considering for the type of optimization required for the task at hand is GP-GPU computing (GPUs modified to support general purpose calculation), a type of computing that increases performance by parallelising thousands of tasks. This would require a GP-GPU computer (available at the University of Edinburgh) and significant reformatting of the code, but if implemented, all Fourier coefficients could be calculated at once in the time required to compute a single one on a regular computer.

6.1.2 Output state decomposition

In the case of lossy Gaussian boson sampling, the final output state can be decomposed into a quantum and classical part. More specifically, the covariance matrix of the output state can be decomposed like so:

$$\sigma = \sigma_p + N_{\geq 0} \tag{6.1}$$

where σ_p is the covariance of a pure Gaussian state with no losses, and $N_{\geq 0}$ can be interpreted as a Gaussian distribution of displacements acting on the quantum state of σ_p.

This is useful because displacements are local operations and should not increase the entanglement and complexity of the classical simulation. Exploiting this with the Fourier series truncation method is not trivial, but for the scope of this project, it is interesting that this decomposition separates the “useful” photons discussed in section (which are encoded in σ_p) from the classical ones.[33] Analyzing the Fourier coefficients for these two covariance matrices separately could help expand the analysis of the interplay between squeezing and losses done in Section 5.3.2, and in general could give great additional insight into the classical simulatability of lossy GBS experiments.

Bibliography

[1] Nils Herrmann, Daanish Arya, Marcus W. Doherty, Angus Mingare, Jason C. Pillay, Florian Preis, and Stefan Prestel. Quantum utility – definition and assessment of a practical quantum advantage. In *2023 IEEE International Conference on Quantum Software (QSW)*. IEEE, July 2023. doi: 10.1109/qsw59989.2023.00028. URL `http://dx.doi.org/10.1109/QSW59989.2023.00028`.

[2] Torsten Hoefler, Thomas Häner, and Matthias Troyer. Disentangling hype from practicality: On realistically achieving quantum advantage. *Communications of the ACM*, 66(5), Apr 2023. doi: 10.1145/3571725.

[3] Alexander Moos Dalzell. *Random Quantum Circuits and their simulation complexity: An analysis with statistical mechanics*. PhD thesis, 2022.

[4] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. Brandao, David A. Buell, and et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, Oct 2019. doi: 10.1038/s41586-019-1666-5.

[5] Feng Pan, Keyang Chen, and Pan Zhang. Solving the sampling problem of the sycamore quantum circuits. *Physical Review Letters*, 129(9), August 2022. ISSN 1079-7114. doi: 10.1103/physrevlett.129.090502. URL `http://dx.doi.org/10.1103/PhysRevLett.129.090502`.

[6] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. *Theory of Computing*, 9(4):143–252, 2013. doi: 10.4086/toc.2013.v009a004. URL `https://theoryofcomputing.org/articles/v009a004`.

[7] Peter Clifford and Raphaël Clifford. The classical complexity of boson sampling, 2017.

[8] Craig S. Hamilton, Regina Kruse, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex. Gaussian boson sampling. *Physical Review Letters*, 119 (17), Oct 2017. doi: 10.1103/physrevlett.119.170501.

[9] Han-Sen Zhong, Li-Chao Peng, Yuan Li, Yi Hu, Wei Li, Jian Qin, Dian Wu, Weijun Zhang, Hao Li, Lu Zhang, and et al. Experimental gaussian boson sampling. *Science Bulletin*, 64(8), Apr 2019. doi: 10.1016/j.scib.2019.04.007.

[10] Craig S. Hamilton, Regina Kruse, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex. Gaussian boson sampling. *Phys. Rev. Lett.*, 119:170501,

Oct 2017. doi: 10.1103/PhysRevLett.119.170501. URL `https://link.aps.org/doi/10.1103/PhysRevLett.119.170501`.

[11] Nicolás Quesada, Juan Miguel Arrazola, and Nathan Killoran. Gaussian boson sampling using threshold detectors. *Phys. Rev. A*, 98, Dec 2018. doi: 10.1103/PhysRevA.98.062322. URL `https://link.aps.org/doi/10.1103/PhysRevA.98.062322`.

[12] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, and et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, Dec 2020. doi: 10.1126/science.abe8770.

[13] Han-Sen Zhong, Yu-Hao Deng, Jian Qin, Hui Wang, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Dian Wu, Si-Qiu Gong, Hao Su, and et al. Phase-programmable gaussian boson sampling using stimulated squeezed light. *Physical Review Letters*, 127(18), Oct 2021. doi: 10.1103/physrevlett.127.180502.

[14] Lars S. Madsen, Fabian Laudenbach, Mohsen Falamarzi. Askarani, Fabien Rortais, Trevor Vincent, Jacob F. Bulmer, Filippo M. Miatto, Leonhard Neuhaus, Lukas G. Helt, Matthew J. Collins, and et al. Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81, Jun 2022. doi: 10.1038/s41586-022-04725-x.

[15] Haoyu Qi, Daniel J. Brod, Nicolás Quesada, and Raúl García-Patrón. Regimes of classical simulability for noisy gaussian boson sampling. *Physical Review Letters*, 124(10), Mar 2020. doi: 10.1103/physrevlett.124.100502.

[16] Junheng Shi and Tim Byrnes. Effect of partial distinguishability on quantum supremacy in gaussian boson sampling. *npj Quantum Information*, 8(1), May 2022. doi: 10.1038/s41534-022-00557-9.

[17] Xun Gao and Luming Duan. Efficient classical simulation of noisy quantum computation, 2018.

[18] Changhun Oh, Liang Jiang, and Bill Fefferman. On classical simulation algorithms for noisy Boson Sampling. 1 2023.

[19] Benjamin Villalonga, Murphy Yuezhen Niu, Li Li, Hartmut Neven, John C. Platt, Vadim N. Smelyanskiy, and Sergio Boixo. Efficient approximation of experimental gaussian boson sampling, 2022.

[20] Dorit Aharonov, Xun Gao, Zeph Landau, Yunchao Liu, and Umesh Vazirani. A polynomial-time classical algorithm for noisy random circuit sampling. *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, Jun 2023. doi: 10.1145/3564246.3585234.

[21] Thomas R Bromley, Juan Miguel Arrazola, Soran Jahangiri, Josh Izaac, Nicolás Quesada, Alain Delgado Gran, Maria Schuld, Jeremy Swinarton, Zeid Zabaneh, and Nathan Killoran. Applications of near-term photonic quantum computers: Software and algorithms. *Quantum Science and Technology*, 5(3), May 2020. doi: 10.1088/2058-9565/ab8504.

[22] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J. Cerf, Timothy C. Ralph, Jeffrey H. Shapiro, and Seth Lloyd. Gaussian quantum information. *Reviews of Modern Physics*, 84(2), May 2011. doi: 10.1103/revmodphys.84.621.

[23] J. Rivera-Dean, P. Stammer, E. Pisanty, Th. Lamprou, P. Tzallas, M. Lewenstein, and M. F. Ciappina. New schemes for creating large optical schrödinger cat states using strong laser fields. *Journal of Computational Electronics*, 20(6), Oct 2021. doi: 10.1007/s10825-021-01789-2.

[24] Jon Brogaard. Wigner function formalism in quantum mechanics. URL `https://nbi.ku.dk/english/theses/bachelor-theses/jon-brogaard/Jon_Brogaard_Bachelorthesis_2015.pdf`.

[25] S. Olivares. Quantum optics in the phase space. *The European Physical Journal Special Topics*, 203(1), Apr 2012. doi: 10.1140/epjst/e2012-01532-4.

[26] Abhinav Deshpande, Arthur Mehta, Trevor Vincent, Nicolás Quesada, Marcel Hinsche, Marios Ioannou, Lars Madsen, Jonathan Lavoie, Haoyu Qi, Jens Eisert, and et al. Quantum computational advantage via high-dimensional gaussian boson sampling. *Science Advances*, 8(1), Jan 2022. doi: 10.1126/sciadv.abi7894.

[27] Raúl García-Patrón, Jelmer J. Renema, and Valery Shchesnovich. Simulating boson sampling in lossy architectures. *Quantum*, 3, August 2019. ISSN 2521-327X. doi: 10.22331/q-2019-08-05-169. URL `http://dx.doi.org/10.22331/q-2019-08-05-169`.

[28] Michał Oszmaniec and Daniel J Brod. Classical simulation of photonic linear optics with lost particles. *New Journal of Physics*, 20(9), September 2018. ISSN 1367-2630. doi: 10.1088/1367-2630/aadfa8. URL `http://dx.doi.org/10.1088/1367-2630/aadfa8`.

[29] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[30] Jonatan Bohr Brask. Gaussian states and operations – a quick reference, 2022.

[31] Francesco Mezzadri. How to generate random matrices from the classical compact groups, 2006.

[32] Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook. Strawberry fields: A software platform for photonic quantum computing. *Quantum*, 3, Mar 2019. doi: 10.22331/q-2019-03-11-129.

[33] Changhun Oh, Minzhao Liu, Yuri Alexeev, Bill Fefferman, and Liang Jiang. Classical algorithm for simulating experimental gaussian boson sampling, 2023.

[34] Nicolás Quesada, Rachel S. Chadwick, Bryn A. Bell, Juan Miguel Arrazola, Trevor Vincent, Haoyu Qi, and Raúl García−Patrón. Quadratic speed-up for simulating gaussian boson sampling. *PRX Quantum*, 3:010306, Jan 2022. doi: 10.1103/PRXQuantum.3.010306. URL `https://link.aps.org/doi/10.1103/PRXQuantum.3.010306`.

[35] Jacob F. F. Bulmer, Bryn A. Bell, Rachel S. Chadwick, Alex E. Jones, Diana Moise, Alessandro Rigazzi, Jan Thorbecke, and Haus et. al. The boundary for quantum advantage in gaussian boson sampling. *Science Advances*, 8(4), January 2022. ISSN 2375-2548. doi: 10.1126/sciadv.abl9236. URL `http://dx.doi.org/10.1126/sciadv.abl9236`.

Appendix A

Proofs

Here we present proofs that were omitted from the main text for the sake of brevity.

A.1

Equation.

$$P^{|0\rangle}(x) = |\langle\hat{x}|0\rangle|^2 = \frac{1}{\sqrt{\hbar\pi}}e^{-\frac{x^2}{\hbar}}$$

Proof. To obtain the distribution of the vacuum in position space, we must find the position representation of the ground state of a quantum harmonic oscillator, i.e. express the ground state in terms of the x variable.

$$\langle\hat{x}|0\rangle = \psi_0(x)$$

The ground state of quantum harmonic oscillator satisfies the equation $\hat{a}\psi_0 = 0$ as applying the lowering operator on the lowest state of a harmonic oscillator gives zero. Recall the definition of the lowering operator:

$$\hat{a} = \frac{1}{\sqrt{2\hbar}}(\hat{x} + i\hat{p})$$

To obtain its position representation, we express the momentum operator in terms of position $\hat{p} = -i\hbar\frac{\partial}{\partial x}$ (a common definition in quantum mechanics). Simplifying, we obtain the following differential equation

$$(x + \hbar\,\frac{\partial}{\partial x})\psi_0 = 0$$

The solution to this is:

$$\psi_0(x) = Ce^{-\frac{x^2}{2\hbar}}$$

where C is a constant that we can find by imposing that the distribution over the entire position space is normalised to 1, i.e. that $\int_{-\infty}^{\infty}\psi_0(x)\,\psi_0(x)\,dx = 1$. We find that $C = (\pi\hbar)^{-1/4}$ and we plug it into ψ_0 to find:

$$P^{|0\rangle}(x) = |\psi_0(x)|^2 = \frac{1}{\sqrt{\hbar\pi}}e^{-\frac{x^2}{\hbar}}$$

By symmetry, we can find $P^{|0\rangle}(y)$ in the same way.

A.2

Equation.

$$\mathrm{Tr}(\hat{\rho}) = 1$$

Proof. For the density matrix of a mixed state, by linearity and using the proof for the pure state, we have:

$$\mathrm{Tr}(\hat{\rho}) = \mathrm{Tr}(\sum_i p_i |\psi_i\rangle\langle\psi_i|) = \sum_i p_i \mathrm{Tr}(|\psi_i\rangle\langle\psi_i|) = \sum_i p_i = 1$$

Here, we take the normalisation of the sum of probabilities to one as a postulate of quantum mechanics.

A.3

Equation.

$$\mathrm{Tr}(\hat{\rho}\hat{A}) = \sum_i p_i \langle\psi_i|\hat{A}|\psi_i\rangle$$

Proof. For the density matrix of a mixed state, by linearity we have:

$$\begin{aligned}\mathrm{Tr}(\hat{\rho}\hat{A}) &= \mathrm{Tr}(\sum_i p_i |\psi_i\rangle\langle\psi_i|\hat{A}) = \mathrm{Tr}(\sum_i p_i |\psi_i\rangle\langle\psi_i|\hat{A})\\ &= \sum_i p_i \mathrm{Tr}(|\psi_i\rangle\langle\psi_i|\hat{A}) = \sum_i p_i \mathrm{Tr}(\langle\psi_i|\hat{A}|\psi_i\rangle)\\ &= \sum_i p_i \langle\psi_i|\hat{A}|\psi_i\rangle\end{aligned}$$

A.4

Equation.

$$\mathrm{Tr}[\hat{\rho}\hat{A}] = 2\pi\hbar \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dp W_{\rho}(x,p) W_A(x,p)$$

Proof. We start with the integral over all x and p of the product of the Wigner transform of $\hat{\rho}$ and $\hat{A}$ for one mode.

$$I = \frac{1}{(2\pi\hbar)^2} \int\int\int\int dx dp dy dy' e^{-ip(y+y')/\hbar} \langle x + \frac{y}{2}|\hat{\rho}|x - \frac{y}{2}\rangle \times \langle x + \frac{y'}{2}|\hat{A}|x - \frac{y'}{2}\rangle$$

Using the integral identity $\int dp e^{ipy/\hbar} = 2\pi\hbar\delta(y)$ we evaluate the p integral and the resulting delta function tells us that the only y' value that gives a non-zero result is $y' = y$.

$$\begin{aligned}I &= \frac{2\pi\hbar}{(2\pi\hbar)^2} \int\int\int dx dy dy' \langle x + \frac{y}{2}|\hat{\rho}|x - \frac{y}{2}\rangle \langle x + \frac{y'}{2}|\hat{A}|x - \frac{y'}{2}\rangle \delta(y + y')\\ &= \frac{1}{2\pi\hbar} \int\int dx dy \langle x + \frac{y}{2}|\hat{\rho}|x - \frac{y}{2}\rangle \langle x - \frac{y}{2}|\hat{A}|x + \frac{y}{2}\rangle\end{aligned}$$

Now we can take the change of variables $u = x - \frac{y}{2}, v = x = \frac{y}{2}$ and $dudv = dxdy$ to get

$$I = \frac{1}{2\pi\hbar} \int \int dudv \langle u|\hat{\rho}|v\rangle \langle v|\hat{A}|u\rangle$$

Recognising the identity formula $\int dv|v\rangle\langle v| = 1$ we can simplify this further.

$$I = \frac{1}{2\pi\hbar} \int du \langle u|\hat{\rho}\hat{A}|u\rangle$$

Finally, recognizing the formula for the trace of an operator $\text{Tr}[\hat{O}] = \int \langle n|\hat{O}|n\rangle$ we obtain

$$I = \frac{1}{2\pi\hbar} \text{Tr}[\hat{\rho}\hat{A}]$$

Appendix B

Additional plots

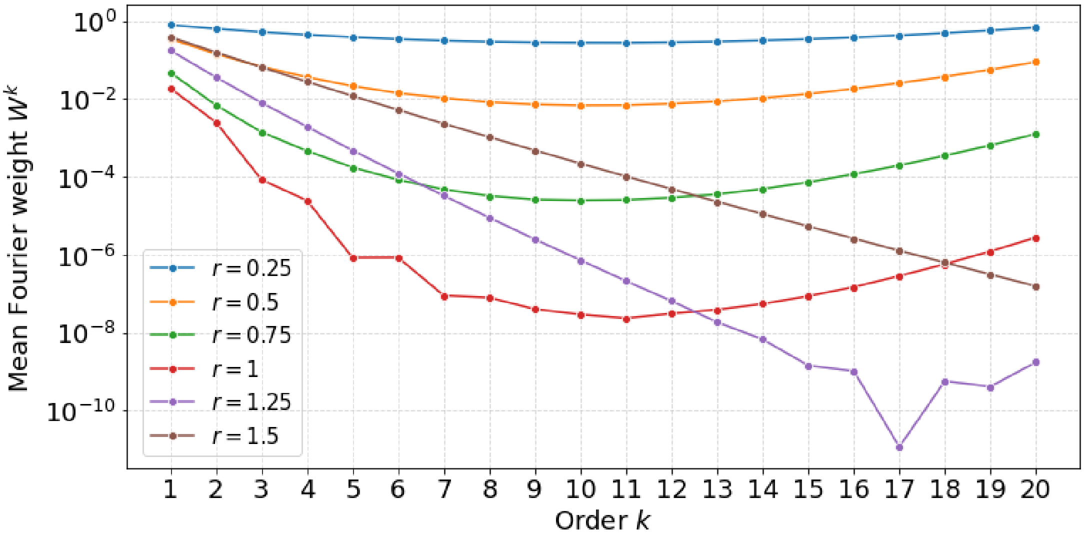

Figure B.1: Log-scaled version of 5.4: Fourier weights for an ideal 20 modes with varying squeezing. For $r = 1.5$ and for the first 12-13 orders of $r = 1.25$ we see exponential-like decay in this ideal system.

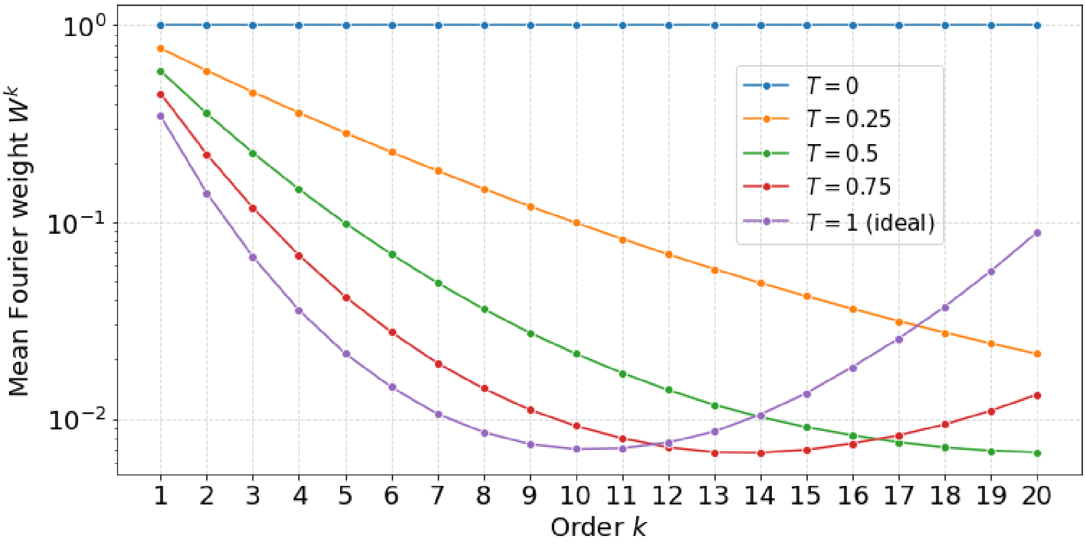

Figure B.2: Log-scaled version of 5.5: Fourier weights for noisy 20 modes with varying losses. Here we do not see exponential decay – even the $T = 0.25$ line is not quite straight.

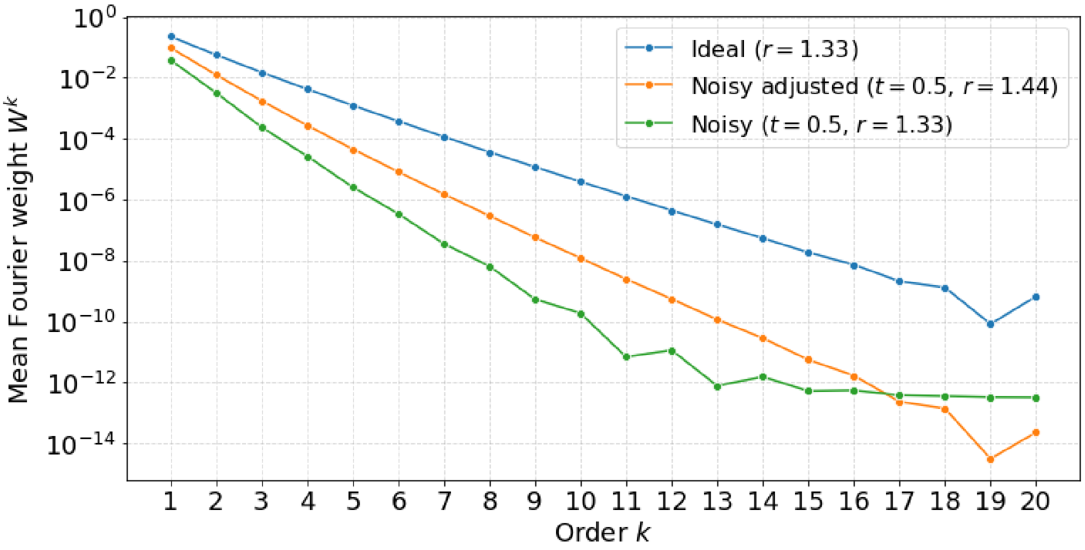

Figure B.3: Log-scaled version of 5.7: Comparison of mean Fourier weights for ideal, noisy, and adjusted noisy 20-mode systems. Here we see that the lines for the two adjusted systems are straight up until the last few orders, while the line for the non-adjusted lossy system is not.