ESTIMATOR REGIONS IN QUANTUM STATE TOMOGRAPHY

by

Yi-Lin SEAH

Under the supervision of:

Berthold-Georg ENGLERT
Hui Khoon NG
David NOTT

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science with Honours in Physics

National University of Singapore
2014

Abstract

When making estimates of a quantum state, point estimators express one's best guess of the true state, but estimator regions are important to express the uncertainty associated with the estimate. Employing the Bayesian approach, we discuss the use of smallest credible regions (SCRs), the smallest possible regions for given credibilities, as optimal estimator regions. However, constructing the SCRs involves evaluating multidimensional integrals, which can be done by Monte Carlo integration. Therefore, we explore several methods that can be exploited to obtain the sample points required for the Monte Carlo integration.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Prof. Berthold-Georg Englert for giving me this opportunity to be part of the group. His patience and sense of humor, which is a stark contrast to his authoritative appearance, has made this project an enjoyable one.

I would also like to thank co-supervisors Asst. Prof. Ng Hui Khoon and Assoc. Prof. David Nott. Through her insightful comments during discussions, Asst. Prof. Ng has enabled me to look at concepts from different perspectives and gaining a deeper understanding of them, while A/P Nott's profound wisdom in statistics has helped with the progress of this project and allowed me to gain much knowledge in statistics.

Lastly, I would like to thank Dr. Shang Jiangwei, who mentored me over the duration of the project. His patient explanations helped resolve my doubts, while his constant encouragement kept my morale up the whole time.

Contents

List of Figures

Chapter 1

Introduction

Quantum state tomography is a process where one attempts to reconstruct a quantum state, using data from measurements performed on identically prepared copies of the state. Typically, this is used to characterize a source of quantum information carriers.

In the typical scenario, an unknown quantum state ρ is sent through a probability-operator measurement (POM), which will result in a click in one of K detectors. A K-outcome POM is characterized by K positive operators $\{\Pi_1, \Pi_2, \ldots, \Pi_K\}$ which satisfy

$$\sum_{k=1}^{K} \Pi_k = 1\,. \tag{1.1}$$

Upon sending the state ρ through the POM, the probability of the kth detector clicking, p_k, is given by

$$p_k = \mathrm{tr}\left\{\Pi_k \rho\right\} = \langle \Pi_k \rangle\,. \tag{1.2}$$

The fact that ρ is positive semi-definite and has unit trace, together with (1.1), ensures that we have

$$\sum_{k=1}^{K} p_k = 1,\ p_k \geq 0\,. \tag{1.3}$$

Data is obtained by performing multiple measurements of identically prepared copies of the state ρ, and counting the number of times each detector clicks.

In statistics, the most popular estimator is the maximum-likelihood estimator (MLE) [1]. This is true in quantum tomography as well, where the MLE is the state that gives the highest likelihood of observing the data obtained [2, 3, 4]. However, such point estimators are of very limited utility when it comes to making inferences. Afterall, all estimates have some associated uncertainty (otherwise they would not be called estimates), and these uncertainties must be expressed in some way. In problems of single-dimensional parameter space, these uncertainties are expressed as error bars. For higher dimensional problems, error regions will be required. In this thesis, we discuss how estimator regions can be uniquely characterized, and also how such regions can be constructed.

Chapter 2

Optimal Estimator Regions

When making estimates, a good estimator region should have the following properties:

1. The region should be small, narrowing the possibilities of where the true value of the parameter being estimated could be.

2. The region should contain the true value of the parameter being estimated with a high probability.

An optimal estimator region should therefore be as small as possible, yet as likely as possible to contain the true value of the parameter being estimated. In this chapter, we discuss how to characterize and construct optimal error regions.

2.1 Confidence regions

In the frequentists' view of statistics, the parameters being estimated are fixed. Confidence regions are regions chosen such that if the experiments are repeated multiple times, each time giving a different set of data, at least a certain proportion of the confidence regions will contain the true value of the parameter being estimated, this proportion being the level of confidence. This is demonstrated in Fig. 2.1 on page 5. In the figure, 100 sets of random data are generated based on a parameter x, which has value 2.0. From each

set of data, a 95% confidence region of x is constructed. These regions take the form of intervals, since x is a single-dimensional parameter. With a level of confidence being 95%, we expect that in the long run, at least 95% of these intervals will contain the true value of parameter x.

It should be noted that there is no unique way of constructing the confidence regions. For an experiment that consists of certain measurements, there may be an infinite number of ways to construct regions for any level of confidence. Furthermore, there is no meaningful way to define the size of the confidence regions. Without a measure of size, there is no way to decide which set of confidence regions are optimal. As such, we look to another alternative, the credible regions.

2.2 Credible regions

Before moving on to define optimal credible regions in the Bayesian framework of statistics, we first introduce the concepts of prior distribution, size, and confidence.

2.2.1 Prior and size

In Bayesian statistics, the underlying state of the system being studied is assumed to be probabilistic, drawn from a distribution whose probability density function is called the prior density. We denote $(\mathrm{d}\rho)$ to represent the probability of the system being in the infinitesimal vicinity of state ρ before any data is observed. The state can be parameterized by a set of parameters. Here, we use $\theta = \{\theta_1, \theta_2, \ldots, \theta_K\}$. In this parametrization, we have

$$(\mathrm{d}\rho) = w(\theta)(\mathrm{d}\theta), \tag{2.1}$$

where $(\mathrm{d}\theta)$ is the infinitesimal volume element in the parameter space, given by

$$(\mathrm{d}\theta) = \mathrm{d}\theta_1 \mathrm{d}\theta_2 \cdots \mathrm{d}\theta_K, \tag{2.2}$$

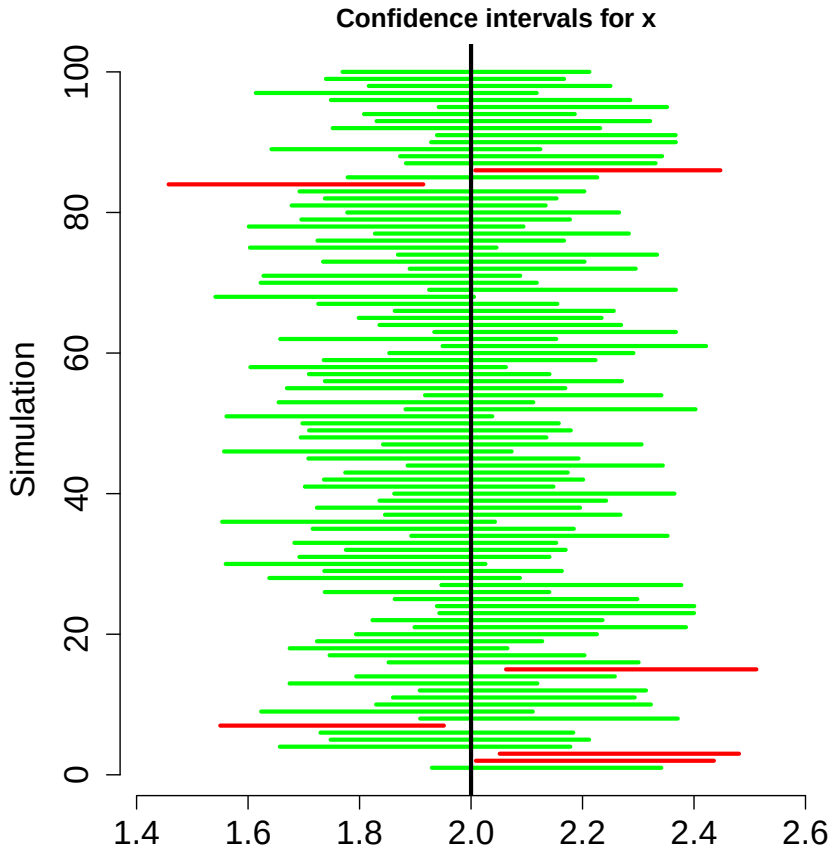


Figure 2.1: Visualisation of confidence intervals. Here, the true value of the parameter x is 2.0. 100 sets of random data are generated, and for each set, a 95% confidence interval is constructed. Intervals that contain the true value of x are colored green, while those that do not contain the true value are colored red. In the long run, 95% of these regions are expected to contain the true value of the parameter x.

and $w(\theta)$ is the prior density of θ. Generally, the permissible values of θ may not cover the entire $\mathbb{R}^K$. As a result, $w(\theta)$ should contain a constraint factor $w_{\text{cstr}}(\theta)$, which assigns a prior value of 0 for nonpermissible θ, and takes the form

$$w_{\text{cstr}}(\theta) = \begin{cases} 1 & \text{if } \theta \text{ is permissible} \\ 0 & \text{otherwise} \end{cases}. \tag{2.3}$$

Under reparameterization $\theta \to \theta'$, the prior transforms according to

$$w(\theta') = w(\theta) \left| \frac{\partial \theta}{\partial \theta'} \right|, \tag{2.4}$$

where $\left|\frac{\partial \theta}{\partial \theta'}\right|$ is the determinant of the Jacobian matrix.

Since the prior represents the probability density of the state before any data is observed, the prior should express indifference, namely, assigning equal probability to all possible states. At first glance, a seemingly obvious choice of prior would therefore be the primitive prior, which is constant over all permissible θ. Written explicitly, the primitive prior is given by

$$w(\theta) \propto w_{\text{cstr}}(\theta). \tag{2.5}$$

However, it must be noted that the form of the prior depends on the parametrization used. In other words, the primitive prior density will generally not remain flat under re-parametrization, and demanding the prior to be constant results in a prior that depends on the parametrisation used. In order to have a parameter-independent prior, a popular non-informative prior is the Jeffreys prior [5, 6, 7], which takes the form

$$w(\theta) \propto \sqrt{\det \mathcal{I}(\theta)}, \tag{2.6}$$

where $\mathcal{I}(\theta)$ is the Fisher information matrix. It can be verified that such a prior preserves its form under reparametrisation. For the rest of this thesis, we will be using the primitive prior and Jeffreys prior, although the methods discussed later can be applied to any other choice of prior.

Finally, for a region $\mathcal{R}$, we define the size of the region $s_{\mathcal{R}}$ to be the prior content of the region, that is, the probability that the state of the system lies within the region, prior to observing any data. The size is therefore given by

$$s_{\mathcal{R}} = \int_{\mathcal{R}} (\mathrm{d}\rho) = \int_{\mathcal{R}} w(\theta)\,(\mathrm{d}\theta). \tag{2.7}$$

Following this definition, the size of the entire state space $\mathcal{R}_0$ is

$$s_{\mathcal{R}_0} = \int_{\mathcal{R}_0} (\mathrm{d}\rho) = 1\,. \tag{2.8}$$

By defining the size in this manner, we ensure that the size of a region is independent of the parametrization used to describe the state space.

2.2.2 Posterior and credibility

The posterior density of a state ρ is defined to be the probability density that the system is in state ρ, given that data D were observed. Mathematically, the posterior density is given by

$$g\left(\rho|D\right) = \frac{L\left(D \mid \rho\right)}{L\left(D\right)} \propto L\left(D \mid \rho\right), \tag{2.9}$$

or in terms of parameterization θ,

$$g\left(\theta|D\right) = \frac{L\left(D \mid \theta\right) w\left(\theta\right)}{L\left(D\right)} \propto L\left(D \mid \theta\right) w\left(\theta\right). \tag{2.10}$$

Here, $L\left(D|\rho\right)$ is the likelihood of observing data D given the system is in state ρ, and $L\left(D\right)$ is the prior likelihood of observing the data D, given by

$$L\left(D\right) = \int_{\mathcal{R}_0} L\left(D \mid \rho\right)(\mathrm{d}\rho) = \int_{\mathcal{R}_0} L\left(D \mid \theta\right) w\left(\theta\right)(\mathrm{d}\theta)\,. \tag{2.11}$$

For a region $\mathcal{R}$, we define the credibility of the region $c_{\mathcal{R}}$ to be the posterior content of the region, that is, the probability that the state of the system lies within the region, given the observed data. The credibility is therefore given by

$$c_{\mathcal{R}} = \frac{1}{L\left(D\right)} \int_{\mathcal{R}} L\left(D \mid \rho\right)(\mathrm{d}\rho) = \frac{1}{L\left(D\right)} \int_{\mathcal{R}} L\left(D \mid \theta\right) w\left(\theta\right)(\mathrm{d}\theta)\,. \tag{2.12}$$

2.2.3 Smallest credible region

The smallest credible region (SCR) is defined as the smallest possible region that gives a desired level of credibility. In order to find such a region, it is convenient to first introduce the concept of a bounded-likelihood region (BLR) [8].

A BLR $\mathcal{R}_\lambda$ is a region that contains all states in the state space $\mathcal{R}_0$ with likelihood $L(D \mid \rho)$ exceeding a certain threshold value. We specify this threshold value as a fraction λ of the maximum value $L(D \mid \hat{\rho}_{\text{MLE}})$, where $\hat{\rho}_{\text{MLE}}$ is the maximum likelihood estimator for the state. Such a region can therefore be described by

$$\chi_\lambda(\rho) = \eta\left(L(D \mid \rho) - \lambda L(D \mid \hat{\rho}_{\text{MLE}})\right), \tag{2.13}$$

where $\eta()$ is the Heaviside unit step function. The size of such a region is given by

$$s_\lambda = \int_{\mathcal{R}_0} (\mathrm{d}\rho)\, \chi_\lambda(\rho) = \int_{\mathcal{R}_\lambda} (\mathrm{d}\rho), \tag{2.14}$$

and its credibility is

$$c_\lambda = \frac{1}{L(D)} \int_{\mathcal{R}_0} (\mathrm{d}\rho)\, \chi_\lambda(\rho)\, L(D \mid \rho) = \frac{1}{L(D)} \int_{\mathcal{R}_\lambda} (\mathrm{d}\rho)\, L(D \mid \rho). \tag{2.15}$$

The size and credibility of a BLR are related by

$$L(D) \frac{\partial}{\partial \lambda} c_\lambda = L(D \mid \hat{\rho}_{\text{MLE}})\, \lambda \frac{\partial}{\partial \lambda} s_\lambda . \tag{2.16}$$

As a consequence, if we know s_λ is a function of λ, we can obtain c_λ using

$$c_\lambda = \frac{\lambda s_\lambda + \int_\lambda^1 \mathrm{d}\lambda' s_{\lambda'}}{\int_0^1 \mathrm{d}\lambda' s_{\lambda'}} . \tag{2.17}$$

For each BLR, it is impossible to find another region with a smaller size but the same credibility – each BLR is also a SCR, with credibility c_λ. Therefore, the problem of finding a SCR with credibility c is equivalent to finding a BLR with $c_\lambda = c$.

2.2.4 Construction of SCRs

We now present a systematic approach for constructing SCRs [9].

1. Vary λ from 0 to 1, and for each value of λ, find the size s_λ of the corresponding BLR using (2.14).

2. Create a subroutine that takes in λ and returns c_λ, making use of (2.17) to evaluate c_λ. The integrals of s_λ are to be evaluated using the values of s_λ obtained in step 1.

3. Using a suitable root-finding algorithm, vary λ to find a λ^* such that $c_{\lambda^*} = c$. The subroutine from step 2 is used to provide the required values of s_λ.

4. The SCR with credibility c is the BLR with threshold λ^*, or the set of points where the likelihood exceeds $\lambda^* L\left(D \mid \hat{\rho}_{\text{MLE}}\right)$.

It is straightforward to evaluate the integrals of s_λ in step 2. Being single-dimensional integrals, they can be done using quadratures such as Simpson's rule. It is advisable, however, to use adaptive quadratures instead [10], as s_λ tends not to be very smooth, especially the area near $\lambda = 0$. Typically, the most computationally expensive part in this algorithm is step 1, which is in general a multidimensional integral over the prior. In Chapter 5, we will explore various methods to deal with such integrals.

Chapter 3

Coin Flip

3.1 Overview

Before moving to quantum systems, we will first look at a classical analog of a two-outcome POM – a coin which lands heads with probability p and tails with probability $(1-p)$ when flipped. The data will be in the form of $D = \{n_1, n_2\}$, with n_1 and n_2 being the number of heads and tails obtained respectively. Given p, the likelihood of obtaining data D is given by

$$L(D \mid p) = p^{n_1} (1-p)^{n_2} . \tag{3.1}$$

As for the prior, since p must be within the range $[0, 1]$, the constraint factor is given by

$$w_{\text{cstr}} = \eta(p)\, \eta(1-p) . \tag{3.2}$$

The primitive prior on p then takes on the form

$$w_{\text{primitive}}(p) = \eta(p)\, \eta(1-p) . \tag{3.3}$$

As for the Jeffreys prior, the Fisher information for a binomial distribution is given by

$$\mathcal{I}(p) = \frac{n}{p(1-p)} . \tag{3.4}$$

Hence, the normalised Jeffreys prior is given by

$$w_{\text{Jeffreys}}(p) = \frac{1}{\pi\sqrt{p(1-p)}}\eta(p)\,\eta(1-p)\,. \tag{3.5}$$

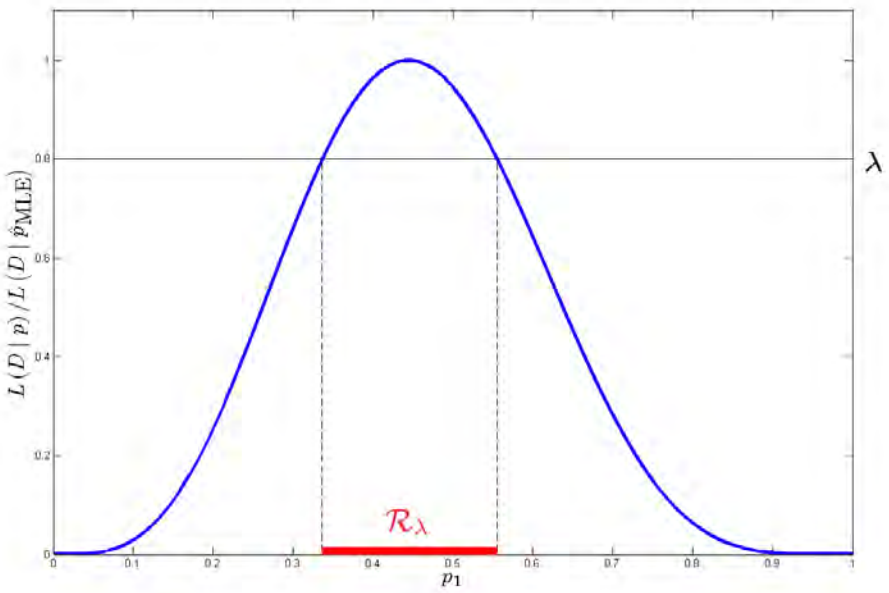


Figure 3.1: An illustration of a BLR. In this plot, $L(D \mid p)/L(D \mid \hat{p}_{\text{MLE}})$ is plotted against p for a coin that landed heads four times out of nine flips. $\mathcal{R}_\lambda$ is the BLR obtained for the given λ.

3.2 Constructing SCRs

In order to find a SCR with a given credibility c, we use the method described in 2.2.4.

3.2.1 Obtaining the size

The first thing we need to do is to find the size of BLRs s_λ for the range $\lambda \in [0, 1]$. This is relatively simple for the coin flip, since the state space

has dimension 1. Since the likelihood function is unimodal, the BLR $\mathcal{R}_\lambda$ takes the form $\{p|a \leq p \leq b\}$, where a and b are the two solutions $\{x|L(D \mid x) = \lambda L(D \mid \hat{p}_{\text{MLE}})\}$, which can be found numerically with any suitable root-finding algorithm. The size is then given by the integral

$$s_\lambda = \int_a^b w(p)(\mathrm{d}p). \tag{3.6}$$

This results in the size being $(b-a)$ for the primitive prior, and $\frac{2}{\pi}\left[\arcsin\left(\sqrt{b}\right) - \arcsin\left(\sqrt{a}\right)\right]$ for the Jeffreys prior.

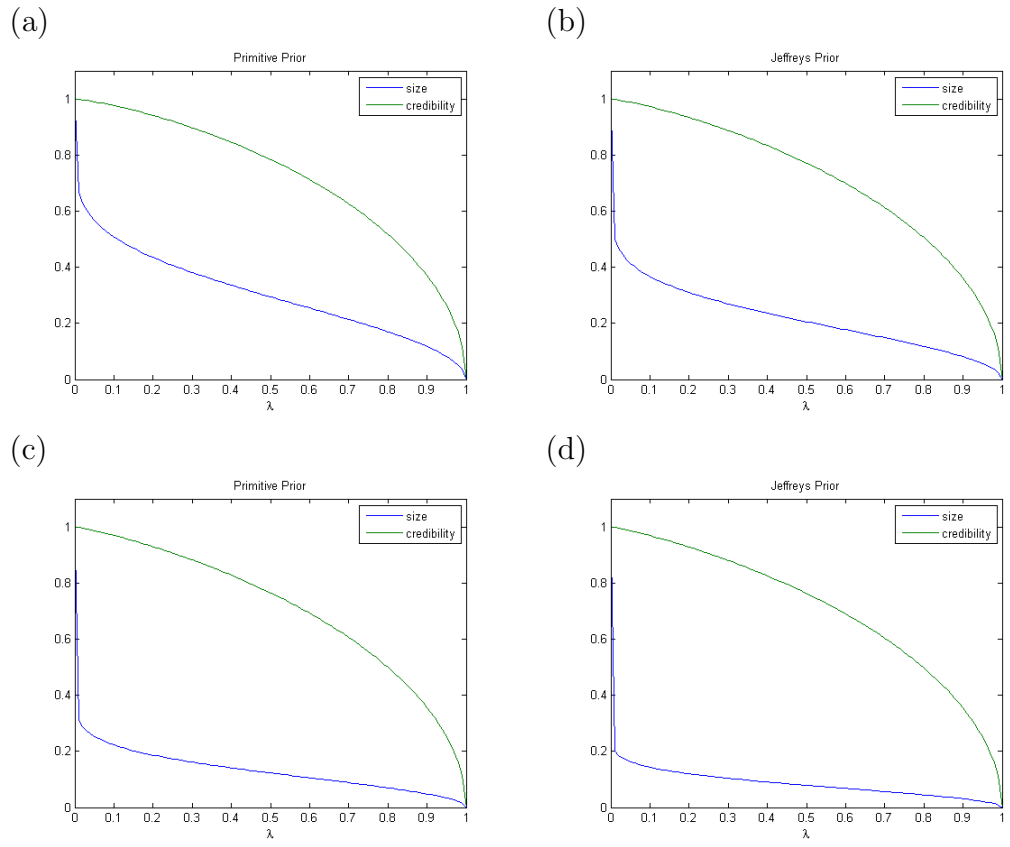


Figure 3.2: Size and credibility varying with λ. In plots (a) and (b), the data used was 4 heads and 5 tails, with the primitive and Jeffreys priors respectively. For plots (c) and (d), the data used was 40 heads and 50 tails, with the primitive and Jeffreys priors respectively.

The credibility of each BLR is computed using (2.17), with the integrals

being done numerically. However, as can be seen in Fig. 3.2 on page 13, s_λ is in general not a very smooth function of λ, especially when λ is close to 0. As a result, quadratures with evenly spaced intervals are likely to perform very poorly, and adaptive quadratures should be used [10]. Next, it should be noted that the integral in the denominator of (2.17) is independent of λ. If c_λ is to be determined repeatedly for varying values of λ, this integral should be evaluated just once and its value stored in the memory, and recalled each time c_λ is calculated. In order to evaluate the integral in the denominator of (2.17), it may be useful to consider the relation

$$\int_0^1 \mathrm{d}\lambda' s_{\lambda'} = \int_0^1 \lambda(p)\, w(p)\, (\mathrm{d}p)\,, \tag{3.7}$$

where $\lambda(p)$ defined to be

$$\lambda(p) = \frac{L(D \mid p)}{L(D \mid \hat{p}_{\mathrm{MLE}})}\,. \tag{3.8}$$

This is useful as the integrand only contains terms that are known analytically, as opposed to the original integral where the integrand contains $s_{\lambda'}$, which must be found numerically, hence improving the speed and accuracy when performing this integral.

Now that we are able to find the credibility c_λ of a BLR with threshold λ, we find the value λ^* where c_{λ^*} is equal to our target credibility c. With this value of λ^*, we proceed to find the two solutions $\{a^*, b^*\} = \{x | L(D \mid x) = \lambda^* L(D \mid \hat{p}_{\mathrm{MLE}})\}$. These can all be done numerically using suitable root-finding algorithms. The SCR is then the region bounded by a^* and b^*.

3.3 SCRs as confidence regions

In this section, we use the example of the coin flip to compare the notions of credibility and confidence. We do so by considering our SCRs as confidence regions, and finding out how the level of confidence relates to the credibility of the regions.

3.3.1 Covering probability

In order to find the level of confidence we get from our regions, we first introduce a quantity called the covering probability. Given a scheme $\mathbb{S}$ used to construct estimator regions, we define the covering probability $\gamma(\rho; \mathbb{S})$ to be the probability that the constructed region contains the true state, given the true state is ρ. Here, the data D is the random variable, and the state ρ is fixed. For SCRs with credibility c, the covering probability takes the form

$$\gamma(\rho; c) = \sum_D L(D|\rho)\, I(D, c, \rho), \tag{3.9}$$

where $I(D, c, \rho)$ is an indicator variable that we define as

$$I(D, c, \rho) = \begin{cases} 1 & \text{if the SCR with credibility } c \text{ and data } D \text{ contains } \rho \\ 0 & \text{otherwise} \end{cases}. \tag{3.10}$$

We now proceed to see how the covering probability varies with p in our coin flip model. For each value of p, we use (3.9) to determine $\gamma(p; c)$. The sum over D will have $N + 1$ entries, where N is the number of coin flips observed. For each D, the likelihood $L(D \mid p)$ is given by (3.1). In order to get the value of the indicator variable $I(D, c, p)$, the SCR must first be constructed as described in Section 3.2, and then checked if it contains p. In Fig. 3.3 on page 16, we show a few plots of the covering probability against the probability of heads of a coin.

3.3.2 Confidence level

If we were to use the SCRs as confidence regions, the level of confidence we obtain is the minimum value of the covering probability. If we look at Fig. 3.3 on page 16, it is worth noting that there are points where the plot is discontinuous, and these occur at the edges of the SCRs. It also turns out that all local minima occur at these points. Additionally, the plot is symmetrical about $p = 0.5$. Hence, when looking for the minimum value of the covering probability, one only has to check the edges of the SCRs within

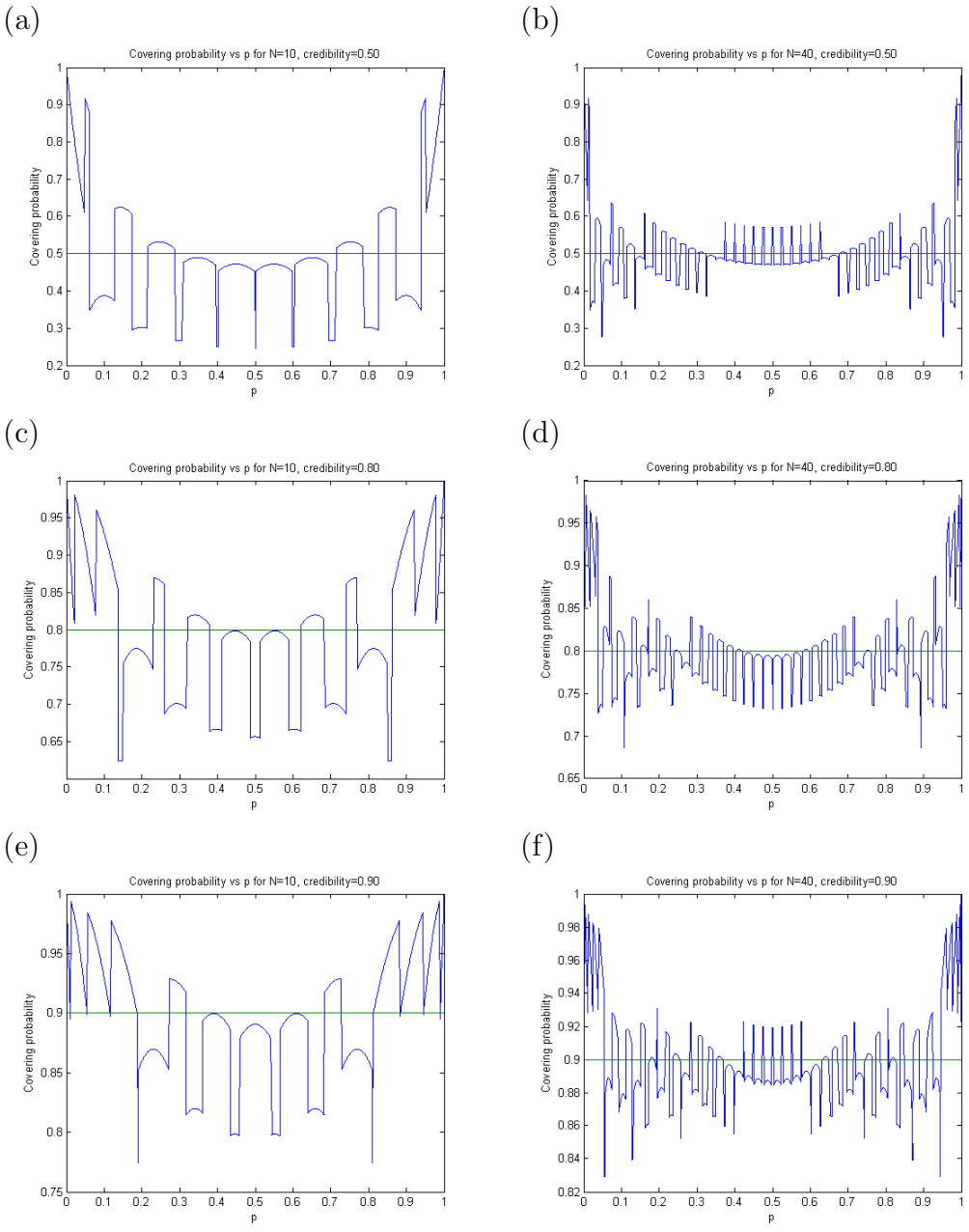


Figure 3.3: Covering probability varying with p for a coin. For these plots, the primitive prior was used. The number of coinflips observed were 10 and 40, with credibilities 0.5, 0.8 and 0.9. In the plots, the horizontal green line shows the credibility.

the range $0 < p < 0.5$, rather than sweeping through the entire range of p.

In order to see how the confidence varies with the credibility of our regions, we vary the credibility and find the level of confidence that we get for each value of the credibility. This is done for various values of N, for both the primitive and Jeffreys prior, and the results are shown in Fig. 3.4 on page 18. From the plots, a few observations can be made. Firstly, we see that given the same credibility, the confidence level of the SCRs can be very different for different priors. Additionally, we notice that the level of confidence is always significantly lower than the credibility of our regions. This emphasizes the fact that confidence and credibility mean very different things, and one has to be clear about which it is they are using when talking about estimator regions.

3.3.3 Expanding SCRs

As we saw in Fig. 3.4 on page 18, our SCRs have significantly levels of confidence compared to their credibilities. In an attempt to see if we can modify our SCRs to make them useful as confidence regions, we try expanding them slightly to see if we can raise the level of confidence to match the original credibility. This is done by first specifying an expansion factor E. Then, we transform each region's boundaries $\{a, b\} \to \{a_E, b_E\}$ using

$$a_E = a - \frac{b-a}{2}E\,, \tag{3.11a}$$

$$b_E = b + \frac{b-a}{2}E\,. \tag{3.11b}$$

Using various values of E, we plot the level of confidence against the credibility in Fig. 3.5 on page 19 to see the effect the expansion has on the level of confidence of our SCRs. We see that the expansion did not help much for the Jeffreys prior, with the level of confidence consistently below the credibility even when lengths of the intervals were doubled ($E = 1$). As for the primitive prior, the expansion did improve the levels of confidence. However, we see that we have to expand the regions by a significant amount before

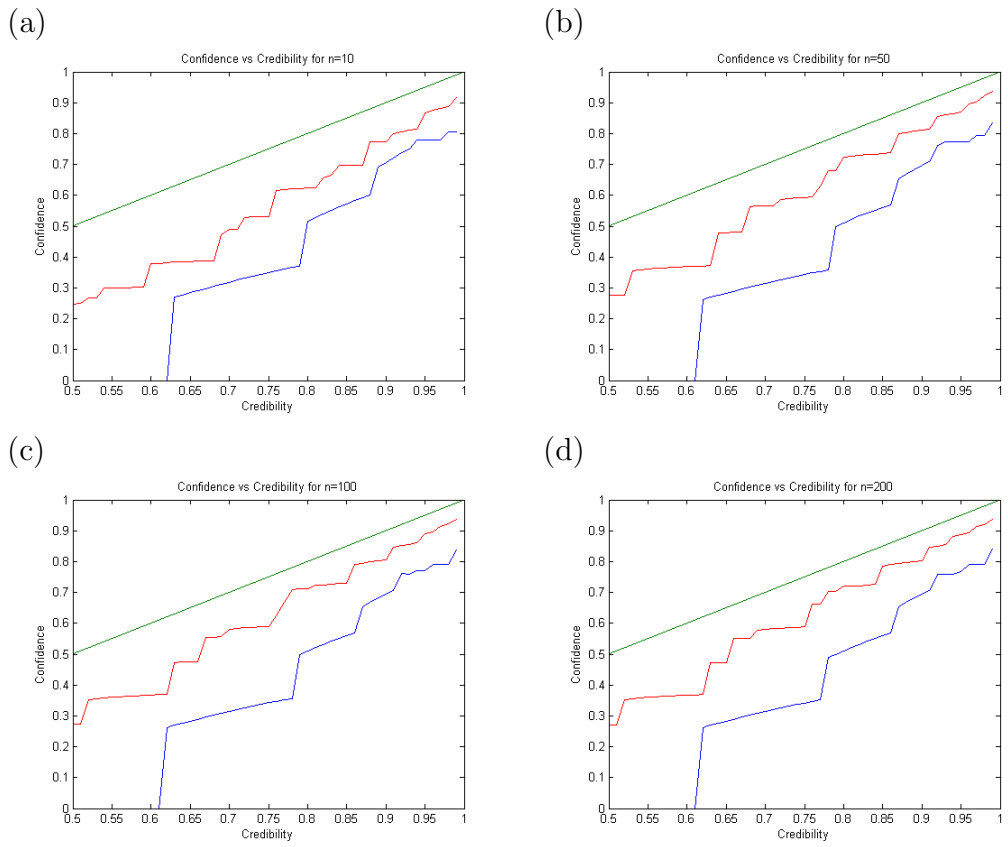


Figure 3.4: Confidence level of SCRs. The number of observed coin flips N was 10, 50, 100 and 200 for plots (a), (b), (c) and (d) respectively. In the plots, the red line shows the results for the primitive prior, the blue line shows the results for the Jeffreys prior, and the green line is where the confidence is equal to the credibility. We see that for all the plots, the red and blue lines are always under the green one, indicating that the confidence is always lower than the credibility for our SCRs.

achieving the corresponding levels of confidence. Therefore, in order to use our SCRs as confidence regions, we have to either choose a credibility higher than the desired level of confidence, perform expansions on the SCRs, or a combination of both.

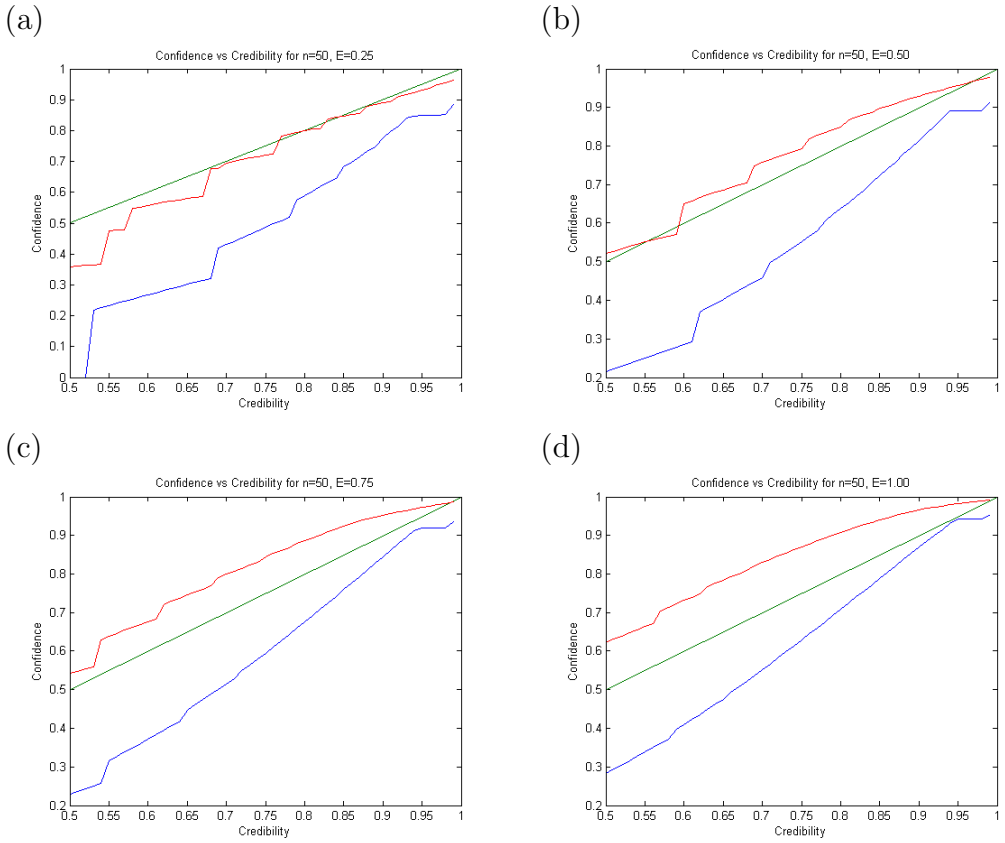


Figure 3.5: Effect of expanding SCRs on their confidence level. The number of coin flips N was fixed at 50, and the expansion factors used were 0.25, 0.5, 0.75 and 1 for plots (a), (b), (c) and (d) respectively. The horizontal axis is the credibility before the regions were expanded. The red and blue lines represent the primitive and Jeffreys priors respectively, while the green line is where the confidence is equal to the credibility. For comparison, the result when $E = 0$ can be seen from plot (b) of Fig. 3.4 on page 18.

Chapter 4

Moving to Quantum Systems

4.1 Likelihood and prior

After measuring N identical copies of the state, the data $D = \{n_1, n_2, \ldots, n_K\}$ is obtained, where n_k is the number of times detector k clicked, with $\sum_{k=1}^{K} n_k = N$. Since D has a multinomial distribution for a fixed ρ, $L(D \mid \rho)$ takes on the form

$$L(D \mid \rho) = p_1^{n_1} p_2^{n_2} p_3^{n_3} \cdots p_K^{n_K} . \tag{4.1}$$

As for the prior, since each of the probabilities p_k must be non-negative and normalized, $w(p)$ contains a factor $w_0(p)$ given by

$$w_0(p) = \eta(p_1)\,\eta(p_2) \cdots \eta(p_K)\,\delta\left(\sum_K p_K - 1\right), \tag{4.2}$$

with $\delta()$ being the Dirac delta function. Generally, there are additional constraints on the probability space owing to the fact that the probabilities must describe realizations of the POM acting on physical quantum states. Therefore, (4.2) is insufficient to describe the constraint factor $w_{cstr}(p)$, which takes the form

$$w_{\text{cstr}}(p) = \begin{cases} 1 & \text{if } p \text{ is physically permissible} \\ 0 & \text{otherwise} \end{cases} . \tag{4.3}$$

For a multinomial distribution with probabilities $\{p_1, p_2, \ldots, p_K\}$, the Jeffreys prior is proportional to $\frac{1}{\sqrt{p_1 p_2 \cdots p_K}}$. As such, the primitive and Jeffreys priors for our probabilities p will be

$$w_{\text{primitive}}(p) \propto w_{\text{cstr}}(p), \tag{4.4}$$

$$w_{\text{Jeffreys}}(p) \propto \frac{1}{\sqrt{p_1 p_2 \cdots p_K}} w_{\text{cstr}}(p). \tag{4.5}$$

4.1.1 Example: prior for trine measurement on single qubit

The trine measurement is a three-outcome measurement characterized by the POM

$$\Pi_1 = \frac{1}{3}(1 + \sigma_x), \tag{4.6a}$$

$$\Pi_2 = \frac{1}{6}\left(2 - \sigma_x + \sqrt{3}\sigma_y\right), \tag{4.6b}$$

$$\Pi_3 = \frac{1}{6}\left(2 - \sigma_x - \sqrt{3}\sigma_y\right). \tag{4.6c}$$

With this POM, the probabilities of each outcome are simply

$$p_1 = \frac{1}{3}(1 + x), \tag{4.7a}$$

$$p_2 = \frac{1}{6}\left(2 - x + \sqrt{3}y\right), \tag{4.7b}$$

$$p_3 = \frac{1}{6}\left(2 - x - \sqrt{3}y\right), \tag{4.7c}$$

where x and y represent $\langle\sigma_x\rangle$ and $\langle\sigma_y\rangle$ respectively. If we sum the squares of the probabilities, we find

$$p^2 = \sum_{k=1}^{3} p_k^2 = \frac{1}{6}\left(2 + x^2 + y^2\right). \tag{4.8}$$

Since the state of the qubit lies within the Bloch sphere, where $(x^2 + y^2) \leq 1$, we find that $p^2 \leq \frac{1}{2}$. Combining this constraint with w_0 of (4.2), we get

$$w_{\text{cstr}}(p) = \eta(p_1)\,\eta(p_2)\,\eta(p_3)\,\delta(p_1 + p_2 + p_3 - 1)\,\eta\left(\frac{1}{2} - p^2\right). \tag{4.9}$$

With this, along with (4.4) and (4.5), we get the complete form of the primitive and Jeffreys priors to be

$$w_{\text{primitive}}(p) \propto \eta(p_1)\,\eta(p_2)\,\eta(p_3)\,\delta(p_1 + p_2 + p_3 - 1)\,\eta\left(\frac{1}{2} - p^2\right), \tag{4.10}$$

$$\begin{aligned} w_{\text{Jeffreys}}(p) \propto{}& \frac{1}{\sqrt{p_1 p_2 p_3}}\eta(p_1)\,\eta(p_2)\,\eta(p_3) \\ &\times \delta(p_1 + p_2 + p_3 - 1)\,\eta\left(\frac{1}{2} - p^2\right). \end{aligned} \tag{4.11}$$

4.2 Determining the physicality of p

For priors (4.4) and (4.5), we had a factor $w_{cstr}(p)$ which was given by (4.3), and in 4.1.1, we showed an example of the form $w_{cstr}(p)$ might take. In general, however, for more complicated systems and measurements, the physicality of p cannot be easily expressed with a collection of delta and step functions. We need a systematic way to determine the physicality of p that will work regardless of the system being measured and the POM being used. This can be done using the following steps.

1. Find the maximum likelihood estimator $\hat{\rho}_{\text{MLE}}$ that maximizes the likelihood of obtaining data D with relative frequencies $n_k/N = p_k$. How this can be done will be discussed shortly.

2. Compute $\hat{p}$ corresponding to $\hat{\rho}_{\text{MLE}}$ using $\hat{p}_k = \text{tr}\{\Pi_k \hat{\rho}_{\text{MLE}}\}$.

3. Find the distance between $\hat{p}$ and p, $d = \sqrt{\sum_{k=1}^{K}(\hat{p}_k - p_k)^2}$.

4. If $d < \varepsilon$ for a small tolerance ε, we conclude that p is physical. If $d > \varepsilon$, then we say that p is not physical.

Iterative methods can be used to find $\hat{\rho}_{\text{MLE}}$ numerically. In this thesis, we discuss the use of the direct-gradient and conjugate-gradient methods.

4.2.1 Finding $\hat{\rho}_{\text{MLE}}$ using direct-gradient method

The direct-gradient method, or steepest ascent, aims to maximize the likelihood function by moving along the direction of steepest ascent. This can be used for finding $\hat{\rho}_{\text{MLE}}$ [4], and the algorithm is as follows:

1. Start with $j = 1$, a small tolerance value ε, a fixed constant α, and the maximally mixed state $\rho_1 = 1/d$, where d is the dimensionality of the Hilbert space of ρ.

2. Compute R_j using
$$R_j = \sum_{k=1}^{K} \frac{p_k}{\operatorname{tr}\{\rho_j \Pi_k\}} \Pi_k \,. \tag{4.12}$$

3. If $\operatorname{tr}\{|R_j \rho_j - \rho_j|\} \leq \varepsilon$, escape the loop and jump to step 6. Otherwise, proceed on to step 4.

4. Compute ρ_{j+1} using
$$\rho_{j+1} = \frac{\left[1 + \frac{\alpha}{2}(R_j - 1)\right] \rho_j \left[1 + \frac{\alpha}{2}(R_j - 1)\right]}{\operatorname{tr}\left\{\left[1 + \frac{\alpha}{2}(R_j - 1)\right] \rho_j \left[1 + \frac{\alpha}{2}(R_j - 1)\right]\right\}} \,. \tag{4.13}$$
Here, α characterizes how far we change ρ along the direction of steepest ascent.

5. Update $j = j + 1$, and go back to step 2.

6. Return ρ_j as the result for $\hat{\rho}_{\text{MLE}}$.

In [11], rather than using a fixed constant α in step 4, Teo advocates using a line search for finding an optimum value of α in each iteration. A line search is performed as follows:

1. Using two trial values of α, compute $\rho_{j+1}(\alpha)$ using (4.13).

2. For each of these $\rho_{j+1}(\alpha)$, along with $\rho_{j+1}(0) = \rho_j$, determine $L(D \mid \rho_{j+1}(\alpha))$.

3. Find the quadratic function that interpolates these three points, and determine α that maximizes this quadratic function.

4. Use this value of α to generate ρ_{j+1}.

Performing a line search here has the benefit of reducing the number of iterations required for the algorithm to converge. However, it has many drawbacks. Firstly, it significantly increases the computational cost of each iteration. More importantly, as the algorithm approaches the optimal state $\hat{\rho}_{\text{MLE}}$, the target function (the likelihood function in this case) becomes rather flat. With the target function varying very slightly, numerical errors become very significant when performing the interpolation, giving rise to cases where the algorithm may return errors or even fail to converge. As such, using a constant value of α may be preferred. When using a constant value of α, it should be noted that too small a value will result in the algorithm converging very slowly, while too large a value can result in algorithm circling about the peak, and failing to converge altogether. Hence, should the user decide against using a line search, care must be taken to choose an appropriate value of α that works best for the particular situation.

The benefits of the direct-gradient method are that it is straightforward to understand and apply. Additionally, individual iterations are relatively fast. However, it is known to have a slower convergence compared to other optimization algorithms. As such, we will take a look at an alternative, the conjugate-gradient method.

4.2.2 Finding $\hat{\rho}_{\text{MLE}}$ using conjugate-gradient method

In the conjugate-gradient method, the "zig-zag" path that often results from the direct-gradient method is avoided by attempting to make the direction vectors in each iteration conjugate to the rest [12]. Applied to our search for $\hat{\rho}_{\text{MLE}}$, the conjugate-gradient method is as follows:

1. Start with $j = 1$, a small tolerance value ε, a fixed constant α, another fixed constant ξ, identity matrix $A_1 = 1$, and the maximally mixed state $\rho_1 = 1/d$.

2. Compute R_1 using (4.12).

3. Set $G_1 = R_1 - 1$, $H_1 = G_1$.

4. If $\operatorname{tr}\{|R_j\rho_j - \rho_j|\} \leq \varepsilon$, escape the loop and jump to step 12. Otherwise, proceed on to step 5.

5. Set $A_{j+1} = A_j + \alpha H_j$.

6. Compute $\rho_{j+1} = \frac{A_{j+1}^\dagger A_{j+1}}{\operatorname{tr}\left\{A_{j+1}^\dagger A_{j+1}\right\}}$.

7. Compute R_{j+1} using (4.12).

8. Set $G_{j+1} = A_{j+1}\,(R_{j+1} - 1)$.

9. Compute $\gamma_j = \max\left\{\frac{\operatorname{tr}\left\{G_{j+1}^\dagger (G_{j+1} - \xi G_j)\right\}}{\operatorname{tr}\left\{G_j^\dagger G_j\right\}}, 0\right\}$.

10. Set $H_{j+1} = G_{j+1} + \gamma_j H_j$.

11. Update $j = j + 1$, and go back to step 4.

12. Return ρ_j as the result for $\hat{\rho}_{\mathrm{MLE}}$.

Instead of using a constant α in step 5, a line search can be performed to find the optimal α, similar to what was described for the direct-gradient method. The benefits and drawbacks of using such a line search are also the same as mentioned for the direct-gradient method. As for the ξ parameter that is used in step 9, it should be chosen from the range $[0, 1)$. Smaller values typically lead to quicker convergence, although Teo mentions that if set to 0, the algorithm may not converge. However during our tests, we did not encounter any problems with setting ξ to 0. Having it at 0 indeed gave us the best convergence rate, as shown in Fig. 4.1 on page 27. One suggestion is to set it to 0, and then raise it slightly should it run into problems.

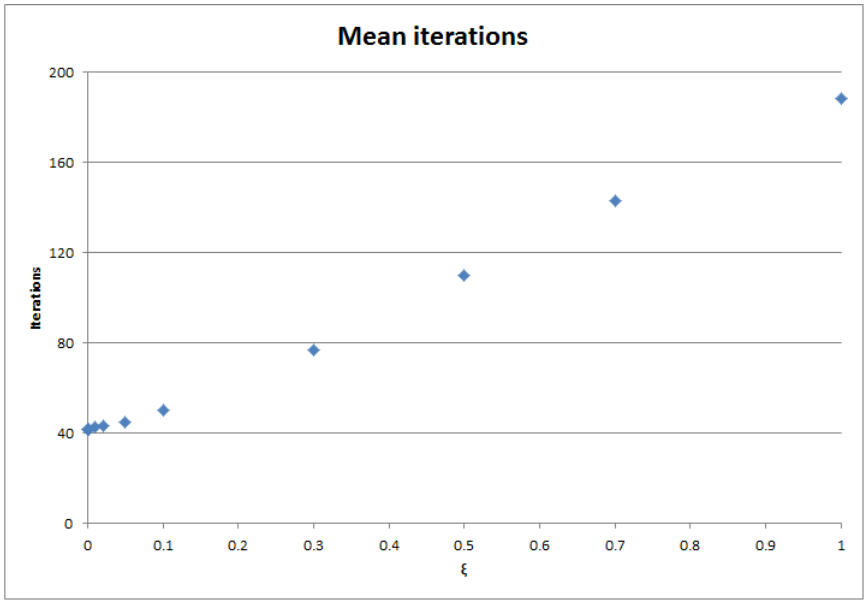


Figure 4.1: Mean number of iterations required for the conjugate-gradient algorithm to converge, as ξ is varied. We see that performance improves for smaller values of ξ. The scenario used for this test is a nine outcome POM, comprising of a double trine measurement of two qubits.

Compared to the direct-gradient method, each iteration of the conjugate-gradient method is slightly more expensive computationally. However, the number of iterations required for convergence is much smaller for the conjugate-gradient method, as shown in Fig. 4.2 on page 28. The overall effect is that the CPU time required for the conjugate-gradient method to converge is a fraction of that required by the direct-gradient method.

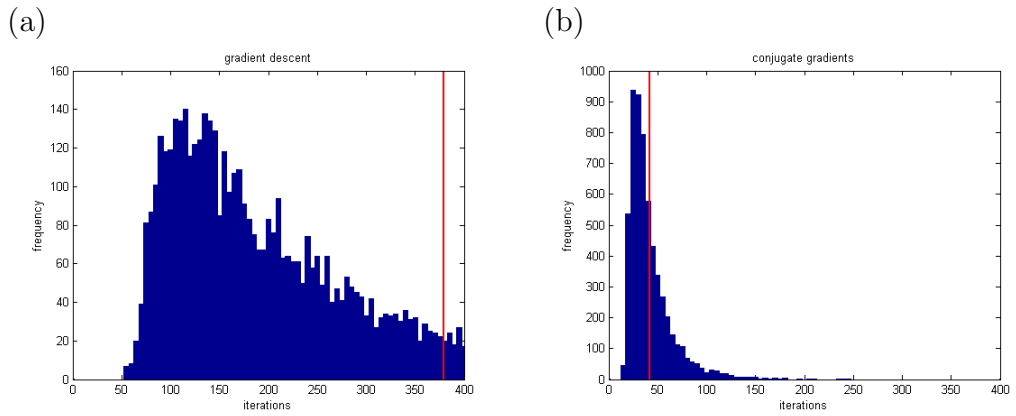


Figure 4.2: Number of iterations taken using the direct-gradient method (a) and conjugate-gradient method (b). The verticle red line represents the mean number of iterations for each algorithm. For the direct-gradient method, the mean large mean was due to a long and heavy tail which extended to the range of thousands of iterations. The testing was done using a 9-dimensional trine-antitrine measurement on a qubit pair, and the CPU time required for the conjugate-gradient method to converge was an eighth of that required by the direct-gradient method.

Chapter 5

Constructing SCRs for Quantum Systems

In order to construct SCRs for our quantum systems, we return to the method described in 2.2.4. However, we run into problems when attempting to evaluate s_λ. Written in terms of p, s_λ takes the form

$$s_\lambda = \int_{\mathcal{R}_\lambda} w\left(p\right)\left(\mathrm{d}p\right). \tag{5.1}$$

Since $(\mathrm{d}p) = \mathrm{d}p_1\mathrm{d}p_2\cdots\mathrm{d}p_K$, this is a K-dimensional integral that has to be evaluated. The performance of integration schemes that rely on grids over the variables deteriorate exponentially as dimensionality of the problem increases. An improvement is to use sparse grids [13], but its reliability too deteriorates quickly as dimensionality is increased. Our approach, therefore, is to use Monte-Carlo integration [14].

In Monte-Carlo integration, points within the integration range are generated randomly. The integrand is evaluated at each of these points, and a weighted sum over these values gives the estimated result for the integral. Unlike integration over grids, the performance of Monte-Carlo integration only depends on the number of sample points, and not on the dimensionality of the problem. For the evaluation of (5.1), our general approach will be to randomly generate sample points that are distributed according to our prior

$w(p)$. After generating m points, our estimate for the integral will be given by

$$s_\lambda = \sum_{j=1}^{m} \frac{\eta \left(L\left(D \mid p^{(j)} \right) - \lambda L\left(D \mid \hat{\rho}_{\mathrm{MLE}} \right) \right)}{m}, \tag{5.2}$$

where $p^{(j)}$ is the set of probabilities of the jth sample, D is the set of data obtained, and $\hat{\rho}_{\mathrm{MLE}}$ is the state that maximizes the likelihood of data D, which can be found using the direct-gradient method described in 4.2.1, or conjugate-gradient method described in 4.2.2. What is left now is to generate samples distributed according to $w(p)$. There are many methods for generating such samples [15, 16], and in this chapter, we discuss how some of them can be used for our situation.

5.1 Independence sampling

In independence sampling, sample points are randomly generated independent of each other, as its name suggests. In general, it is not straightforward to sample directly from the target distribution ($w(p)$ in this case). However, as long as we can sample over the parameter space with some known distribution, we can make use of such a sample by means of rejection sampling or importance sampling [17]. For the case of probabilities p, the parameter space is the simplex described by the simplex (1.3). Sampling uniformly over the simplex is easy, and we provide two algorithms for doing so.

The first uses the idea that K exponential random variables, when normalized, reduce to a Dirichlet distribution [18]. Together with the fact that a uniform sample over the simplex is simply a Dirichlet distribution with its α parameters all equal to one, we can sample uniformly over the simplex as follows:

1. Generate K random numbers $\{x_1, x_2, \cdots, x_K\}$ uniformly over the interval $(0, 1)$.

2. Transform each x_k using $y_k = -\log(x_k)$. Each y_k obtained this way is drawn from an exponential distribution with $\lambda = 1$.

3. Obtain p_k using $p_k = \frac{y_k}{\sum_{k=1}^{K} y_k}$. The $p = \{p_1, p_2, \cdots, p_K\}$ obtained will be drawn uniformly over the simplex.

The other method, which is analyzed in [19], is as follows:

1. Start with $x_0 = 0$, $x_K = 1$.

2. Draw $K - 1$ random numbers uniformly over the interval $(0, 1)$, and sort the list from smallest to largest, to obtain a sorted list $\{x_1, x_2, \cdots, x_{K-1}\}$.

3. Obtain p_k using $p_k = x_k - x_{k-1}$.

Now that we are able to sample uniformly over the simplex, we will discuss how we can use rejection sampling and importance sampling to evaluate (5.1) with such a sample.

5.1.1 Rejection sampling

In rejection sampling, we draw sample points from a convenient distribution, then reject some sample points such that the remaining sample points are distributed according to the target distribution. For target density $w(p)$ and sampling density $f(p)$, we accept each sample point $p^{(j)}$ with probability

$$a = \frac{w\left(p^{(j)}\right)}{f\left(p^{(j)}\right) M}, \tag{5.3}$$

where M is a covering constant, with value

$$M = \max\left\{\frac{w(p)}{f(p)}\right\}. \tag{5.4}$$

For our case, since our sampling distribution is uniform, we have $f(p) \propto 1$, and our acceptance ratio becomes

$$a = \frac{w\left(p^{(j)}\right)}{\max\{w(p)\}}. \tag{5.5}$$

For the primitive prior, this reduces to the acceptance ratio taking the value of 1 if $p^{(j)}$ is physical, and 0 otherwise. However, we run into a problem for the Jeffreys prior. With the Jeffreys prior, $\max\{w(p)\}$ is typically very large, or even infinite in some cases. This results in an extremely small acceptance rate, or even 0 if $\max\{w(p)\} = \infty$. As such, we should avoid rejection sampling for sharply peaked priors. Additionally, the physical region can be a very tiny subregion of the entire simplex. As dimensionality of the problem increases, the volume of the physical region relative to the entire simplex drops exponentially. For a 3-outcome trine measurement on one qubit, $\frac{\pi}{3\sqrt{3}} = 60.5\%$ of the generated points will be physical, but for a 9-outcome trine-antitrine measurement on a qubit pair, it is found that only 9.3% of the points are physical [9]. Therefore, such a sampling scheme will likely not be practical for cases of higher dimensions.

5.1.2 Importance sampling

In importance sampling, after drawing sample points from the sampling distribution, we assign weights to these points to compensate for the difference between the sampling and target distributions. For each sample point, we calculate the weight factor

$$W_j = \frac{w\left(p^{(j)}\right)}{f\left(p^{(j)}\right)} = w\left(p^{(j)}\right). \tag{5.6}$$

With our uniform sampling distribution, we have $f(p) \propto 1$, and the weight factors become

$$W_j = w\left(p^{(j)}\right). \tag{5.7}$$

Our estimate for the integral (5.1) is then

$$s_\lambda = \frac{\sum_{j=1}^{m} w\left(p^{(j)}\right) \eta\left(L\left(D \mid p^{(j)}\right) - \lambda L\left(D \mid \hat{\rho}_{\mathrm{MLE}}\right)\right)}{\sum_{j=1}^{m} w\left(p^{(j)}\right)}. \tag{5.8}$$

At first glance, it may seem that importance sampling solves all the problems of rejection sampling when working with priors that are sharply peaked. This

is not true, however. For such priors, our uniform sample over the simplex is likely to generate very few points that lie within the peak(s). Since these few points carry a very large weight, the value of s_λ is dominated by these few points, resulting in a large variability in our estimate of s_λ. Furthermore, the problem of most points being unphysical will still be present here. The unphysical points will carry 0 weight, and will therefore not contribute to the evaluation of (5.8).

5.2 Markov-chain Monte Carlo

In the independence sampling schemes, we find that as the dimensionality of the problem gets larger, the acceptance rate becomes very low due to the physical region being a very small subregion in the entire simplex. Furthermore, for sharply peaked priors, we either end up with extremely low acceptance rates for rejection sampling, or for importance sampling, we get very few points in the peaks where they matter most. These problems can be resolved using Markov-chain Monte Carlo (MCMC), where sample points are generated by means of a Markov-chain random walk, with the position of each point depending on the position of the previous [20, 21].

Ideally, we want our Markov-chain's stationary distribution to be our target distribution, or the prior in this case. In order to achieve this, we use the Metropolis-Hastings criteria [22, 23, 24] when performing the random walk. The general algorithm for such a random walk over parameter θ and target density $f(\theta)$ is as follows:

1. Choose a proposal-generating density, $q(x|y)$, which describes the probability density of proposing point x if the current point is y. A common example is a multivariate normal distribution, with a constant covariance matrix and mean at y.

2. Choose an arbitrary starting point $\theta^{(1)}$ and $j = 1$.

3. Randomly generate a proposal θ^* with density $q\left(\theta^*|\theta^{(j)}\right)$.

4. Compute the acceptance ratio

$$a = \min\left\{\frac{f\left(\theta^*\right) q\left(\theta^{(j)}|\theta^*\right)}{f\left(\theta^{(j)}\right) q\left(\theta^*|\theta^{(j)}\right)}, 1\right\}. \tag{5.9}$$

5. Draw a random number U uniformly from the range $(0, 1)$. If $a > U$, set $\theta^{(j+1)} = \theta^*$. Otherwise, set $\theta^{(j+1)} = \theta^{(j)}$.

6. Set $j = j + 1$.

7. For target number of sample points m, escape the loop if $j = m$, otherwise return to step 3.

For the proposal-generating density, it is convenient to choose one that is symmetric, whereby $q\left(x|y\right) = q\left(y|x\right)\ \forall\left\{x, y\right\}$. This proves difficult, however, for bounded parameter spaces such as our probabilities p. Fortunately, there are ways to reparameterize p that allow us to overcome this obstacle. We define an auxiliary variable $x = \{x_1, x_2, \cdots, x_K\}$, with $p_k = x_k^2$. From (1.3), we get $\sum_{k=1}^{K} x_k^2 = 1$, indicating that the parameter space of x is the surface of the unit $(K-1)$- sphere centered about the origin. From point x, we then propose a new point by drawing a K-dimensional multivariate normal random variable with mean x and variance s^2, and normalize it back to unit length. The symmetry of such a proposal distribution is guaranteed due to the spherical symmetry of the multivariate normal distribution.

There are a few more things we need to take note of before we can begin generating our random walk. First of all, we will need the prior under our new parameterization. Using (2.4), we find that $w\left(x\right) \propto xw\left(p\right) = \sqrt{p}w\left(p\right)$. Next, it should be noted that our starting point must be physical. This can be done by picking a state ρ (the maximally mixed state, for instance), then getting the initial $p^{(1)}$ using (1.2), and finally getting $x^{(1)}$ using $x_k^{(1)} = \sqrt{p_k^{(1)}}$. Put together, our algorithm for performing the Metropolis-Hastings random walk is as follows:

1. Start with an arbitrary state ρ, a suitable constant step size s, and $j = 1$.

2. Obtain $p^{(1)}$ using (1.2) and $x^{(1)}$ using $x_k^{(1)} = \sqrt{p_k^{(1)}}$.

3. Randomly generate Δx, a K-dimensional multivariate random variable with 0 mean and variance s^2.

4. Compute $x^* = \frac{x^{(j)} + \Delta x}{\left| x^{(j)} + \Delta x \right|}$.

5. Determine p^* using $p^* = (x^*)^2$.

6. Compute the acceptance ratio $a = \min\left\{ \frac{\sqrt{p^*} w(p^*)}{\sqrt{p^{(j)}} w\left(p^{(j)}\right)}, 1 \right\}$.

7. Draw a random number U uniformly from the range $(0, 1)$. If $a > U$, set $x^{(j+1)} = x^*$. Otherwise, set $x^{(j+1)} = x^{(j)}$.

8. Obtain $p^{(j+1)}$ using $p_k^{(j+1)} = \left(x_k^{(j+1)}\right)^2$.

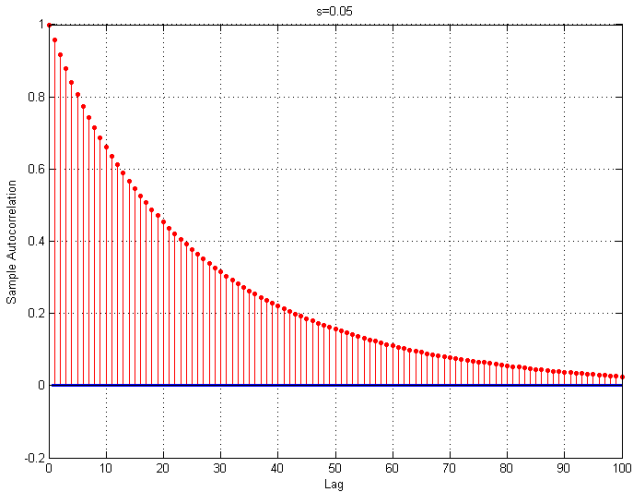
9. Set $j = j + 1$.

10. For target number of sample points m, escape the loop if $j = m$, otherwise return to step 3.

Some attention should be brought to the choice of step size s in step 1 of the above algorithm. When carrying out the Metropolis-Hastings random walk, if the step size is too large, acceptance rates tend to be very low. On the other hand, if the step size is too small, the random walk will take a long time before it can sample the entire space. Therefore, the step size has to be chosen carefully. In the literature, it has been shown that under various conditions, the optimal acceptance rate was found to be 23.4% [25, 26, 27, 28]. This gives a good rule of thumb when finding an optimal step size s. Another way of finding an optimal step size s would be to calculate the autocorrelations at different lags, and to use a step size where autocorrelations decay most quickly. This is illustrated in Fig. 5.1 on page 36.

While the Metropolis-Hastings criteria ensures all the sample points are within the permissible region, we find that the points generated are highly correlated. As the dimensionality increases, smaller step sizes are required to maintain an acceptance rate close to 23.4%. The result is that increasingly long chains are required before the entire space is sampled.

(a)



(b)

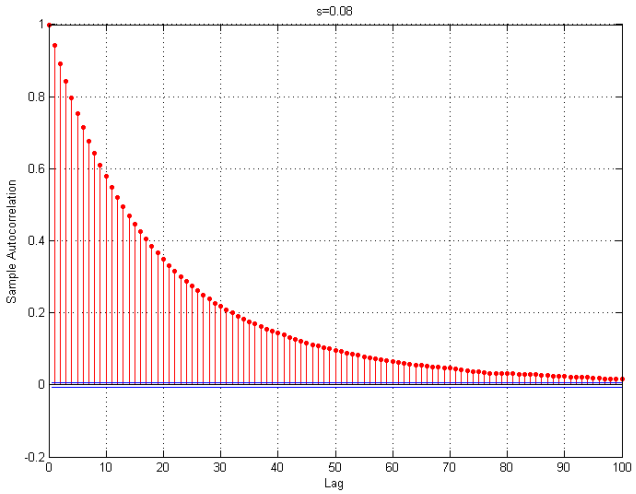


Figure 5.1: Autocorrelations at various lags using the Metropolis-Hastings random walk. These are the autocorrelation plots obtained from doing a Metropolis-Hastings random walk for the 9-outcome trine-antitrine measurement on a qubit pair. The step sizes used were 0.05 and 0.08 for (a) and (b) respectively. The acceptance rates obtained were about 45% for step size 0.05 and 25% for step size 0.08. Based on this, our rule of thumb (target acceptance rate of 23.4%) favours the the step size of 0.08. We see also that for the step size of 0.08, the autocorrelations decay off faster, which agrees with what we concluded based on our rule of thumb.

5.3 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC), or Hybrid Monte Carlo, is a Markov-chain Monte Carlo method that allows us to take large steps while still ensuring an acceptance rate close to 1 [29, 16]. This is done by first treating the parameters as the position q of the system. At each step, we give it a random momentum $\dot{q}$ and evolve the system over a time interval T according to Hamiltonian dynamics. Here, the Hamiltonian H of the system is given by

$$H(q, \dot{q}) = \frac{\dot{q}^2}{2} + U(q), \tag{5.10}$$

where the potential energy U is given by

$$U(q) = -\log[w(q)], \tag{5.11}$$

where $w(q)$ is the density function of the target distribution. For evolving the system over time T, we use the leapfrog method, which is as follows:

1. Start with $t = 1$, initial position $q(0)$, initial momentum $\dot{q}(0)$, number of sub-steps L, and time sub-interval $\varepsilon = T/L$.
2. Compute $\dot{q}\left(t - \frac{1}{2}\right) = \dot{q}(0) - \frac{\varepsilon}{2}\nabla U(q(t-1))$.
3. Compute $q(t) = q(t-1) + \varepsilon\dot{q}\left(t - \frac{\varepsilon}{2}\right)$.
4. Compute $\dot{q}(t) = \dot{q}\left(t - \frac{1}{2}\right) - \frac{\varepsilon}{2}\nabla U(q(t))$.
5. If $t = L$, escape the loop. Otherwise, set $t = t + 1$ and return to step 2.

With this, we proceed to the HMC algorithm.

1. Start with $j = 1$, an arbitrary starting point $q^{(1)}$, time step T, and momentum scaling s.
2. Generate $\dot{q}^{(j)}$ from a multivariate normal distribution with variance s.
3. Using the leapfrog method, evolve $\left\{q^{(j)}, \dot{q}^{(j)}\right\}$ over time T to obtain $\{q^*, \dot{q}^*\}$.

4. Calculate the acceptance ratio

$$a = \min\left\{\exp\left[H\left(q^{(j)}, \dot{q}^{(j)}\right) - H\left(q^*, \dot{q}^*\right)\right], 1\right\}. \tag{5.12}$$

5. Draw a random number U uniformly from the range $(0, 1)$. If $a > U$, set $q^{(j+1)} = q^*$. Otherwise, set $q^{(j+1)} = q^{(j)}$.

6. Set $j = j + 1$.

7. For target number of sample points m , escape the loop if $j = m$, otherwise return to step 2.

If the evolution of the system is done exactly, one finds that the acceptance ratio is always 1 due to conservation of energy. As a result, there is a direct relation between the number of sub-steps L in the leapfrog method, and the acceptance rate. As for the momentum scaling s, if its value is too small, one finds that the algorithm takes a long time to converge, while too large a value of s requires a large L to maintain a reasonable acceptance rate. Similarly with time step T, too small a value results in slow convergence, while too large a value results in a drop in the acceptance rate.

5.3.1 HMC on quantum systems

At first glance, one might be tempted to use the probabilities p as the position q. This will not work though, because the form of the potential (5.11) demands that the prior $w(q)$ must be non-zero for all values of q[‡]. Since our prior contains the factor w_{cstr} given by (4.3), we need a parameterisation that is physical for the entire parameter space. As such, we look to parameterize ρ directly.

We first consider the case where the POM we are working with is informationally complete. For an n-level quantum system, ρ can be writen in the form

$$\rho = A^\dagger A\,. \tag{5.13}$$

[‡]For continuous q, the prior is allowed to reach zero for countably infinitely many points. This is because in practice, it is very unlikely for our numerical algorithms to land precisely on any of these points.

Here, A is an upper-triangular $n \times n$ complex matrix with real diagonal entries, and satisfying

$$\sum_{j=1,k\geq j}^{n} |A_{jk}|^2 = 1\,. \tag{5.14}$$

From (5.14), we see that the moduli of the elements of A are points on a $\left(\frac{n(n-1)}{2}-1\right)$-sphere, which can be parameterized by $\left(\frac{n(n+1)}{2}-1\right)$ angles using a spherical coordinate system. In addition, we need another $\frac{n(n-1)}{2}$ angles to describe the arguments of the off-diagonal terms. All of these angles are unbounded, which is what we want. If we consider a tetrahedron measurement on a single qubit, we have $n = 2$, which results in a total of 3 angles required to parameterize ρ. In terms of the angles, we have

$$A = \begin{pmatrix} \cos\theta_1 & \sin\theta_1 \sin\theta_2 e^{i\theta_3} \\ 0 & \sin\theta_1 \cos\theta_2 \end{pmatrix}, \tag{5.15}$$

$$\rho = A^\dagger A = \begin{pmatrix} \cos^2\theta_1 & \frac{1}{2}\sin(2\theta_1)\sin\theta_2 e^{i\theta_3} \\ \frac{1}{2}\sin(2\theta_1)\sin\theta_2 e^{-i\theta_3} & \sin^2\theta_1 \end{pmatrix}. \tag{5.16}$$

It is also worth determining the expectation values of the Pauli matrices,

$$x = \langle\sigma_x\rangle = \sin(2\theta_1)\sin\theta_2\cos\theta_3\,, \tag{5.17a}$$

$$y = \langle\sigma_y\rangle = \sin(2\theta_1)\sin\theta_2\sin\theta_3\,, \tag{5.17b}$$

$$z = \langle\sigma_z\rangle = \cos(2\theta_1)\,. \tag{5.17c}$$

On the other hand, if the POM we are working with is not informationally complete, we should not parameterize the entire state space. Rather, we should be parameterizing just the reconstruction space, which is a set of ρ that contains exactly one ρ for each permissible p, otherwise we will not be able to find the correct Jacobian factor relating these parameterizations. An example will be the trine measurement on the xz-plane of a single qubit, where a possible reconstruction space is simply the $y = 0$ plane of the Bloch sphere. We can then simply reuse the parameterization described above for the full qubit, but with $\theta_3 = 0$. It should be noted, however, that there is no

general way to find a parameterization that gives us a reconstruction space.

5.3.2 Example: sampling the prior of a tetrahedron measurement

The tetrahedron measurement is a symmetric, informationally complete (SIC) POM on a single qubit, which takes the form

$$\Pi_1 = \frac{1}{4}\left(1 + \sqrt{\frac{1}{3}}\sigma_x + \sqrt{\frac{2}{3}}\sigma_y\right), \tag{5.18a}$$

$$\Pi_2 = \frac{1}{4}\left(1 + \sqrt{\frac{1}{3}}\sigma_x - \sqrt{\frac{2}{3}}\sigma_y\right), \tag{5.18b}$$

$$\Pi_3 = \frac{1}{4}\left(1 - \sqrt{\frac{1}{3}}\sigma_x - \sqrt{\frac{2}{3}}\sigma_z\right), \tag{5.18c}$$

$$\Pi_4 = \frac{1}{4}\left(1 - \sqrt{\frac{1}{3}}\sigma_x + \sqrt{\frac{2}{3}}\sigma_z\right). \tag{5.18d}$$

Using the parameterization described in 5.3.1, the probabilities of each outcome take the form

$$\begin{aligned} p_1 &= \frac{1}{4}\left(1 + \sqrt{\frac{1}{3}}x + \sqrt{\frac{2}{3}}y\right) \\ &= \frac{1}{4}\left(1 + \sqrt{\frac{1}{3}}\sin(2\theta_1)\sin\theta_2\cos\theta_3 + \sqrt{\frac{2}{3}}\sin(2\theta_1)\sin\theta_2\sin\theta_3\right), \end{aligned} \tag{5.19a}$$

$$\begin{aligned} p_2 &= \frac{1}{4}\left(1 + \sqrt{\frac{1}{3}}x - \sqrt{\frac{2}{3}}y\right) \\ &= \frac{1}{4}\left(1 + \sqrt{\frac{1}{3}}\sin(2\theta_1)\sin\theta_2\cos\theta_3 - \sqrt{\frac{2}{3}}\sin(2\theta_1)\sin\theta_2\sin\theta_3\right), \end{aligned} \tag{5.19b}$$

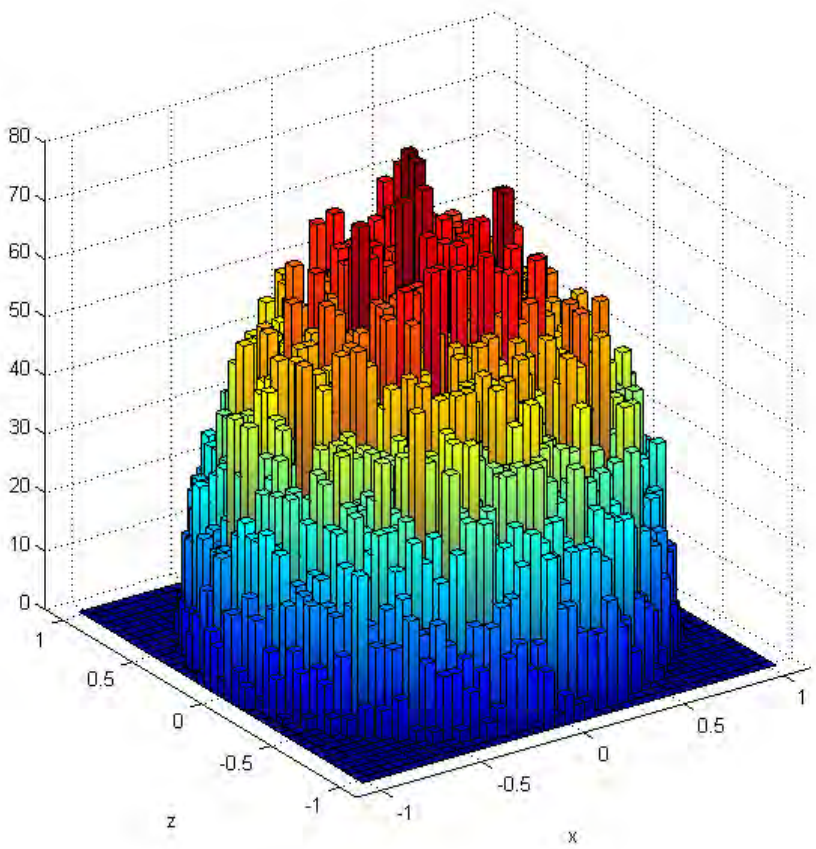


Figure 5.2: Random sample from the prior distribution generated using HMC. The POM used was a tetrahedron measurement on a single qubit. The sampling was done over the primitive prior with 50,000 sample points. The points are then projected onto the xz-plane for the purpose of illustration. We see that the density of points is higher near the middle, which is what we expect when we project a random sample over the volume of a sphere onto a flat plane. This indicates that the HMC algorithm has indeed successfully converged to our target distribution.

$$p_3 = \frac{1}{4}\left(1 - \sqrt{\frac{1}{3}}x - \sqrt{\frac{2}{3}}z\right)$$
$$= \frac{1}{4}\left(1 - \sqrt{\frac{1}{3}}\sin(2\theta_1)\sin\theta_2\cos\theta_3 - \sqrt{\frac{2}{3}}\cos(2\theta_1)\right), \tag{5.19c}$$

$$p_4 = \frac{1}{4}\left(1 - \sqrt{\frac{1}{3}}x + \sqrt{\frac{2}{3}}z\right)$$
$$= \frac{1}{4}\left(1 - \sqrt{\frac{1}{3}}\sin(2\theta_1)\sin\theta_2\cos\theta_3 + \sqrt{\frac{2}{3}}\cos(2\theta_1)\right). \tag{5.19d}$$

Using (2.4) to transform the prior, we get the primitive prior in our parameterization to be

$$w_0(\theta) = w_0(p)\left|\frac{\partial p}{\partial \theta}\right| \propto \left|\frac{\partial p}{\partial \theta}\right| = \left|\sin^3(2\theta_1)\sin(2\theta_2)\right|. \tag{5.20}$$

With this, we use (5.11) to obtain the potential for our Hamiltonian,

$$U(\theta) = 3\log|\sin(2\theta_1)| + \log|\sin 2(\theta_2)|, \tag{5.21}$$

as well as $\nabla U(\theta)$ which we need for performing the leapfrog algorithm, given by

$$\frac{\partial U}{\partial \theta_1} = 6\cot(2\theta_1), \tag{5.22a}$$

$$\frac{\partial U}{\partial \theta_2} = 2\cot(2\theta_2), \tag{5.22b}$$

$$\frac{\partial U}{\partial \theta_3} = 0. \tag{5.22c}$$

We now proceed to perform the HMC sampling over the prior of θ. Fig. 5.2 on page 41 shows the distribution of 50,000 sample points generated using the HMC algorithm. The density of the points are higher around the middle, which is what we expect for points distributed over the volume of a sphere and then projected onto a flat plane, hence verifying that the HMC algotithm

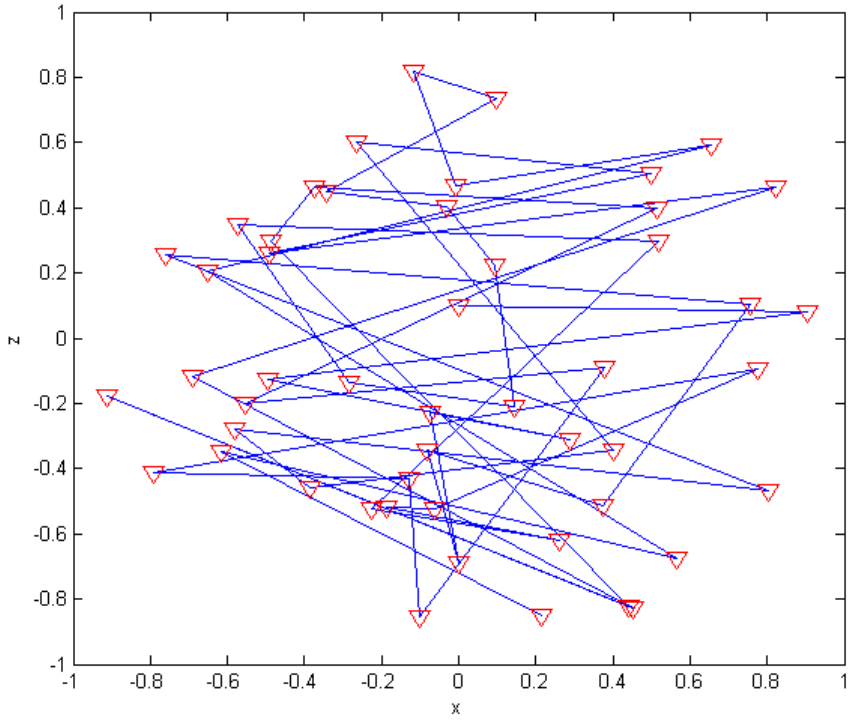


Figure 5.3: Trajectory taken by the HMC algorithm. The POM is the same as that used in Fig. 5.2 on page 41, but only 50 sample points were generated for this plot. Here, the blue lines join consecutive points to show the trajectory taken by the HMC algorithm. We can see that the sample points do not stay close together, unlike what we might expect from other MCMC random walks.

has successfully produced a sample according to our target distribution. In Fig. 5.3 on page 43, only 50 sample points are generated, with consecutive points joined by a blue line. This allows us to see the path taken by the HMC algorithm. From the plot, we can see that the path does not resemble what we might expect from a random walk, where consecutive points are always close together. Hence, HMC overcomes the problem of long autocorrelation times found in the Metropolis-Hastings random walk. Together with a high acceptance rate (about 95% in this example), HMC is able to converge on the target distribution much faster than the Metropolis-Hastings random walk. Furthermore, since all proposed points are guaranteed to be physical, we do not have to check them for physicality, therefore saving much time.

5.3.3 Sampling the posterior

In 2.2.4, we used the s_λ and (2.17) to obtain c_λ. There is, however, a more direct method, which is to compute c_λ using (2.15). By obtaining m sample points distributed according to the posterior distribution, we can estimate c_λ using

$$c_\lambda = \sum_{j=1}^{m} \frac{\eta\left(L\left(D \mid \rho^{(j)}\right) - \lambda L\left(D \mid \hat{\rho}_{\text{MLE}}\right)\right)}{m} \,. \tag{5.23}$$

For the other sampling methods, this would not work as the posterior tends to be sharply peaked, resulting in extremely low acceptance rates. HMC however, allows us to still achieve high acceptance rates when sampling such distributions. Therefore, we have the flexibility to decide either sampling the prior and computing c_λ as described in 2.2.4, or sampling the posterior and computing c_λ using (5.23).

A benefit of sampling from the posterior is that by simply setting $D = \{0, 0, \ldots, 0\}$, we get back the prior distribution, meaning that an algorithm for sampling the posterior will also be able to sample from the prior. Furthermore, for the multinomial likelihood, the Jeffreys prior happens to be a conjugate prior. This results in the posteriors from the Jeffreys and

primitive priors being related by

$$g_{\text{Jeffreys}}\left(\rho|D\right) = g_{\text{primitive}}\left(\rho|D - \frac{1}{2}\right), \tag{5.24}$$

where $\left(D - \frac{1}{2}\right) = \left\{n_1 - \frac{1}{2}, n_2 - \frac{1}{2}, \ldots, n_K - \frac{1}{2}\right\}$. As such, having a scheme to sample from the posterior of the primitive prior will enable us to sample both the prior and the posterior distributions, for both the Jeffreys prior and the primitive prior.

We shall now work out the posterior in our parameterization of the reconstruction space. Combining (2.10), (4.1) and (5.20), the posterior density takes the form

$$g\left(\theta|D\right) \propto p_1^{n_1} p_2^{n_2} \cdots p_K^{n_K} \left|\frac{\partial p}{\partial \theta}\right|. \tag{5.25}$$

5.3.4 Example: sampling the posterior of a trine measurement

For a trine measurement on the xz-plane of a single qubit, the POM takes the form

$$\Pi_1 = \frac{1}{3}\left(1 + \sigma_x\right), \tag{5.26a}$$

$$\Pi_2 = \frac{1}{6}\left(2 - \sigma_x + \sqrt{3}\sigma_z\right), \tag{5.26b}$$

$$\Pi_3 = \frac{1}{6}\left(2 - \sigma_x - \sqrt{3}\sigma_z\right). \tag{5.26c}$$

Using the parameterization described in 5.3.1 with $\theta_3 = 0$, the probabilities of each outcome take the form

$$p_1 = \frac{1}{3}\left(1 + x\right) = \frac{1}{3}\left(1 + \sin\left(2\theta_1\right)\sin\theta_2\right), \tag{5.27a}$$

$$p_2 = \frac{1}{6}\left(2 - x + \sqrt{3}z\right) = \frac{1}{6}\left(2 - \sin\left(2\theta_1\right)\sin\theta_2 + \sqrt{3}\cos\left(2\theta_1\right)\right), \tag{5.27b}$$

$$p_3 = \frac{1}{6}\left(2 - x - \sqrt{3}z\right) = \frac{1}{6}\left(2 - \sin\left(2\theta_1\right)\sin\theta_2 - \sqrt{3}\cos\left(2\theta_1\right)\right). \tag{5.27c}$$

From (5.20), we find that the primitive prior takes the form

$$w_0(\theta) = \left|\frac{\partial p}{\partial \theta}\right| \propto \sin^2(2\theta_1)\,|\cos\theta_2|\,. \tag{5.28}$$

Using (5.25), the posterior becomes

$$\begin{aligned} g(\theta|D) \propto & \left[\frac{1}{3}\left(1+\sin(2\theta_1)\sin\theta_2\right)\right]^{n_1} \\ & \times \left[\frac{1}{6}\left(2-\sin(2\theta_1)\sin\theta_2+\sqrt{3}\cos(2\theta_1)\right)\right]^{n_2} \\ & \times \left[\frac{1}{6}\left(2-\sin(2\theta_1)\sin\theta_2-\sqrt{3}\cos(2\theta_1)\right)\right]^{n_3} \\ & \times \sin^2(2\theta_1)\,|\cos\theta_2|\,. \end{aligned} \tag{5.29}$$

Next, we use (5.11) to obtain the potential of our Hamiltonian,

$$\begin{aligned} U(\theta) = & -n_1\log\left(1+\sin(2\theta_1)\sin\theta_2\right) \\ & - n_2\log\left(2-\sin(2\theta_1)\sin\theta_2+\sqrt{3}\cos(2\theta_1)\right) \\ & - n_3\log\left(2-\sin(2\theta_1)\sin\theta_2-\sqrt{3}\cos(2\theta_1)\right) \\ & - 2\log|\sin(2\theta_1)| - \log|\cos\theta_2| + c\,, \end{aligned} \tag{5.30}$$

where c is a constant that has no effect on the dynamics of the system. Finally, we need $\nabla U(\theta)$ in order to perform the leapfrog algorithm. These take the form

$$\begin{aligned} \frac{\partial U}{\partial \theta_1} = & -n_1\frac{2\cos(2\theta_1)\sin\theta_2}{1+\sin(2\theta_1)\sin\theta_2} \\ & + n_2\frac{2\cos(2\theta_1)\sin\theta_2+2\sqrt{3}\sin(2\theta_1)}{2-\sin(2\theta_1)\sin\theta_2+\sqrt{3}\cos(2\theta_1)} \\ & + n_3\frac{2\cos(2\theta_1)\sin\theta_2-2\sqrt{3}\sin(2\theta_1)}{2-\sin(2\theta_1)\sin\theta_2-\sqrt{3}\cos(2\theta_1)} \\ & + 2\left(\tan\theta_1-\cot\theta_1\right), \end{aligned} \tag{5.31a}$$

$$\begin{aligned}\frac{\partial U}{\partial \theta_2} &= -n_1 \frac{\sin(2\theta_1)\cos\theta_2}{1+\sin(2\theta_1)\sin\theta_2} \\ &\quad + n_2 \frac{\sin(2\theta_1)\cos\theta_2}{2-\sin(2\theta_1)\sin\theta_2+\sqrt{3}\cos(2\theta_1)} \\ &\quad + n_3 \frac{\sin(2\theta_1)\cos\theta_2}{2-\sin(2\theta_1)\sin\theta_2-\sqrt{3}\cos(2\theta_1)} \\ &\quad + \tan\theta_2. \end{aligned} \tag{5.31b}$$

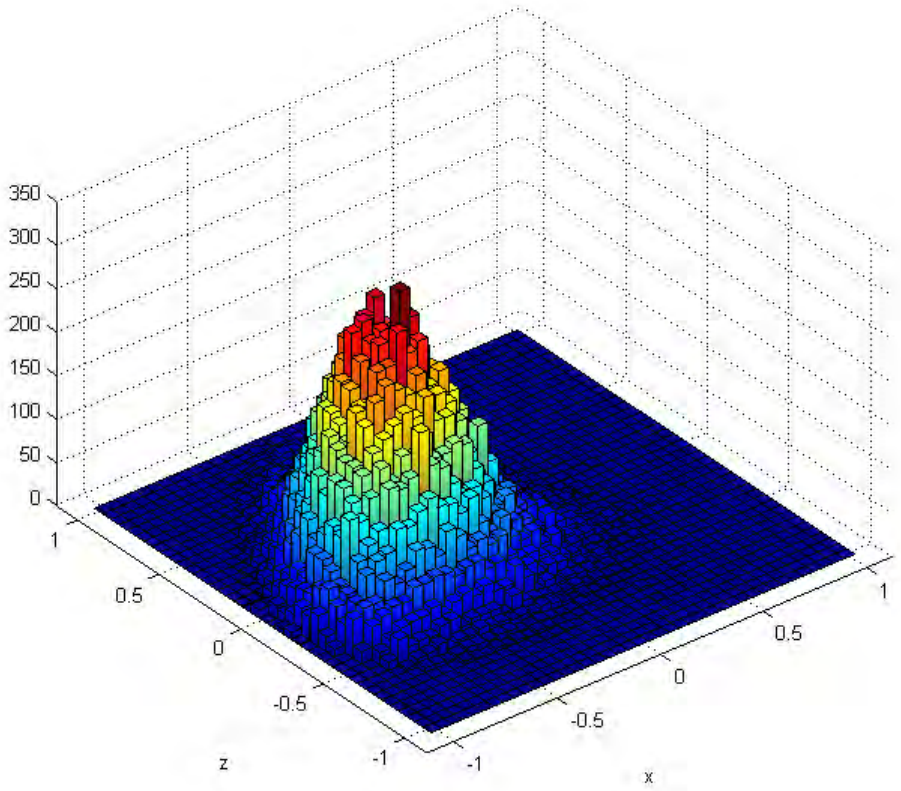


Figure 5.4: Random sample from the posterior distribution generated using HMC. The POM used was a trine measurement on the xz-plane of a qubit, the simulated data was $\{8, 5, 11\}$, and the number of sample points was 50,000. From this, we see that the HMC algorithm is indeed able to successfully converge to the target distribution.

We can now perform the HMC sampling over the posterior of θ. Fig. 5.4 on page 47 shows the distribution of 50,000 sample points obtained using the HMC method. Here, the data used was $\{8, 5, 11\}$. This verifies that the HMC algorithm is able to successfully sample from the posterior. We then proceed to compute c_λ for varying λ using (5.23). Fig. 5.5 on page 48 shows the results of computing c_λ using two methods, the first being sampling the posterior and using (5.23) to obtain c_λ, and the second being sampling the prior, obtaining s_λ using (5.2), and then using (2.17) to compute c_λ. We see that the results from both methods are in agreement with each other, indicating that both methods are able to reliably produce c_λ.

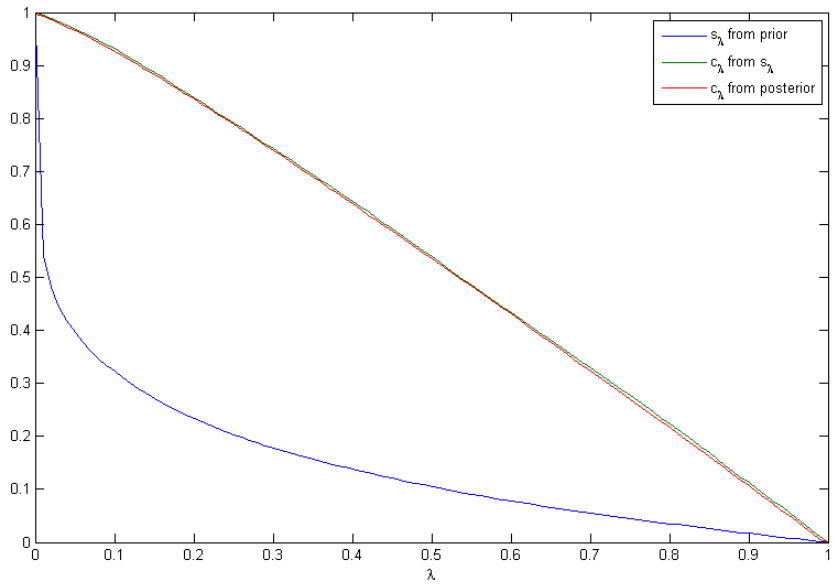


Figure 5.5: Obtaining c_λ from sampling the prior and posterior. The red line represents c_λ obtained by sampling from the posterior, and then using (5.23). The blue line represents s_λ, obtained by sampling the prior and then using (5.2). (2.17) is then used to compute c_λ from s_λ, and these values for c_λ are represented by the green line. We see that the values of c_λ obtained from both methods agree with each other, as they should.

HMC does, however, have its drawbacks. The most important drawback is that in general, for informationally incomplete POMs, there is no straightforward parameterization that will give a reconstruction space. In addition,

as the dimensionaility of the problem increases, we find that the potential (5.30) and its derivatives (5.31a)(5.31b) become increasingly complicated, making HMC difficult to implement.

Chapter 6

Conclusion

In this thesis, we have discussed how optimal estimator regions can be characterized and constructed. We argued since there is no meaningful way to measure the size of confidence intervals, there is no way to define the optimality of a set of confidence regions. We therefore move to the Bayesian approach, and define optimal estimator regions to be SCRs, regions with the smallest size for a given credibility. Here, the size and credibility of a region refers its prior and posterior content respectively.

We then use the case of a coin flip with an unknown success rate to demonstrate how SCRs are constructed. We use this example to draw constrast between credible regions and confidence regions, and also show that for this case, credible regions do not make good confidence regions.

Moving on to quantum systems, the first problem we encounter is that checking if a set of probabilities p is physical given the POM. Fortunately, this can be done using numerical methods [30]. By comparing the direct-gradient and conjugate gradient methods, we found the conjugate-gradient method to be the better of the two, being able to converge in a fraction of the time of the direct-gradient method.

The remaining problem is then to find a way to evaluate the multidimensional integral (5.1). In order to evaluate the integral using the method of Monte Carlo integration, we need to sample the parameter space according to the prior. To do this, we propose three methods, each with its own benefits

and drawbacks. The first is independence sampling, where we generate random points uniformly over the simplex and either accept of scale the points according to the prior. Independence sampling has the benefits of being easy to implement, and also gives uncorrelated sample points. However, for high dimensional problems, most of the sample points will be unphysical, resulting in a very small proportion of sample points being useful. The second method is MCMC using the Metropolis-Hastings random walk. While this gives us points that are all physical, the sample points are highly autocorrelated, resulting in a large number of sample points required before the sampling distribution converges onto the target distribution. Finally, we have HMC, which enables us to achieve larger step sizes compared to the Metropolis-Hastings random walk. This allows us to obtain sample points that are all physical, and with small autocorrelations. Hence among the three methods, HMC is able to converge using the fewest sample points. Furthermore, in HMC, all the proposal points are physical, hence saving us the trouble of checking them, which is a computationally expensive procedure. Its drawbacks however, are that it is difficult to implement, and also it requires a parameterization of a reconstruction space, which is not straightforward to find for informationally incomplete POMs.

Currently, we are trying to implement the HMC method to two qubit systems, and possibly even more complicated systems in the future.

Bibliography

[1] R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, pp. 309–368, 1922.

[2] C. W. Helstrom, "Quantum detection and estimation theory," *Journal of Statistical Physics*, vol. 1, no. 2, pp. 231–252, 1969.

[3] M. Paris and J. Řeháček, *Quantum state estimation*, vol. 649. Springer, 2004.

[4] J. Řeháček, Z. Hradil, E. Knill, and A. Lvovsky, "Diluted maximum-likelihood algorithm for quantum tomography," *Physical Review A*, vol. 75, no. 4, p. 042108, 2007.

[5] R. E. Kass and L. Wasserman, "The selection of prior distributions by formal rules," *Journal of the American Statistical Association*, vol. 91, no. 435, pp. 1343–1370, 1996.

[6] R. L. Winkler, "The assessment of prior distributions in bayesian analysis," *Journal of the American Statistical association*, vol. 62, no. 319, pp. 776–800, 1967.

[7] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.

[8] L. A. Wasserman *et al.*, "A robust bayesian interpretation of likelihood regions," *The Annals of Statistics*, vol. 17, no. 3, pp. 1387–1393, 1989.

[9] J. Shang, H. K. Ng, A. Sehrawat, X. Li, and B.-G. Englert, "Optimal error regions for quantum state estimation," *New Journal of Physics*, vol. 15, no. 12, p. 123026, 2013.

[10] W. M. McKeeman, "Algorithm 145: Adaptive numerical integration by simpson's rule," *Communications of the ACM*, vol. 5, no. 12, p. 604, 1962.

[11] Y. S. Teo, "Numerical estimation schemes for quantum tomography," *preprint arXiv:1302.3399*, 2013.

[12] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," *Carnegie Mellon University*, 1994.

[13] T. Gerstner and M. Griebel, "Numerical integration using sparse grids," *Numerical algorithms*, vol. 18, no. 3-4, pp. 209–232, 1998.

[14] G. Peter Lepage, "A new algorithm for adaptive multidimensional integration," *Journal of Computational Physics*, vol. 27, no. 2, pp. 192–203, 1978.

[15] J. Albert, *Bayesian computation with R*, vol. 747389981. Springer, 2007.

[16] R. Neal, "Mcmc using hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, pp. 113–162, 2011.

[17] J. S. Liu, *Monte Carlo strategies in scientific computing*. Springer, 2008.

[18] L. Devroye, "Sample-based non-uniform random variate generation," in *Proceedings of the 18th conference on Winter simulation*, pp. 260–265, ACM, 1986.

[19] N. A. Smith and R. W. Tromble, "Sampling uniformly from the unit simplex," *Johns Hopkins University, Tech. Rep*, 2004.

[20] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*. CRC Press, 2011.

[21] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of Monte Carlo Methods*, vol. 706. John Wiley & Sons, 2011.

[22] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 2004.

[23] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[24] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[25] M. Bedard and J. S. Rosenthal, "Optimal scaling of metropolis algorithms: Heading toward general target distributions," *Canadian Journal of Statistics*, vol. 36, no. 4, pp. 483–503, 2008.

[26] G. O. Roberts, A. Gelman, W. R. Gilks, *et al.*, "Weak convergence and optimal scaling of random walk metropolis algorithms," *The annals of applied probability*, vol. 7, no. 1, pp. 110–120, 1997.

[27] G. O. Roberts, J. S. Rosenthal, *et al.*, "Optimal scaling for various metropolis-hastings algorithms," *Statistical science*, vol. 16, no. 4, pp. 351–367, 2001.

[28] Y. F. Atchadé, G. O. Roberts, and J. S. Rosenthal, "Towards optimal scaling of metropolis-coupled markov chain monte carlo," *Statistics and Computing*, vol. 21, no. 4, pp. 555–568, 2011.

[29] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid monte carlo," *Physics letters B*, vol. 195, no. 2, pp. 216–222, 1987.

[30] Z. Hradil, J. Řeháček, J. Fiurášek, and M. Ježek, "Maximum-likelihood methods in quantum mechanics," in *Quantum state estimation*, pp. 59–112, Springer, 2004.