A BAYESIAN MODEL FOR THE UNLABELLED SIZE-AND-SHAPE ANALYSIS

ANAMUL HAQUE SAJIB

School of Mathematical Science

University of Nottingham

NG7 2RD, United Kingdom

A thesis submitted for the degree of

Doctor of Philosophy

May 21^{st}, 2018

Abstract

This thesis considers the development of efficient MCMC sampling methods for Bayesian models used for the pairwise alignment of two unlabelled configurations. We introduce ideas from differential geometry along with other recent developments in unlabelled shape analysis as a means of creating novel and more efficient MCMC sampling methods for such models. For example, we have improved the performance of the sampler for the model of Green and Mardia (2006) by sampling rotation, $\boldsymbol{A} \in \mathbb{S}O(3)$, and matching matrix using geodesic Monte Carlo (MCMC defined on manifold) and Forbes and Lauritzen (2014) matching sampler, developed for finger print matching problem, respectively.

We also propose a new Bayesian model, together with implementation methods, motivated by the desire for further improvement. The model and its implementation methods proposed exploit the continuous nature of the parameter space of our Bayesian model and thus move around easily in this continuous space, providing highly efficient convergence and exploration of the target posterior distribution. The proposed Bayesian model and its implementation methods provide generalizations of the existing two models, Bayesian Hierarchical and regression models, introduced by Green and Mardia (2006) and Taylor, Mardia and Kent (2003) respectively, and resolve many shortcomings of existing implementation methods; slow convergence, traps in local mode and dependence on initial starting values when sampling from high dimensional and multi-modal posterior distributions. We illustrate our model and its implementation methods on the alignment of two proteins and two gels, and we find that the performance of proposed implementation methods under proposed model is better than current implementation techniques of existing models in both real and simulated data sets.

Acknowledgement

At first, my immense gratitude to Allah for enabling me to complete this PhD thesis successfully, and then a very special thanks from the core of my heart to both of my supervisors Dr. Christopher Fallaize and Dr. Christopher J. Brignell, Lecturer, School of Mathematical Science, University of Nottingham, for their excellent supervision, informative guidelines and continuous encouragement that helped me to complete this thesis successfully.

I also want to give a big thanks to Professor Andrew Wood and Professor Ian Dryden, School of Mathematical Science, University of Nottingham, who gave me the opportunity to discuss my PhD area with them when I needed, which was really helpful and motivating as well. My two internal examiners Professor Huiling Le and Assistant Professor Karthik Bharath helped me for the duration of my PhD career thus far by giving their valuable feedback which was key for me to stay in the right direction.

I would also like to thank the funding authority of University of Nottingham and School of Mathematical Science for providing me with the funding and academic freedom to carry out this research. Finally, I am indebted to my wife, Afruja Akter, and my parents for their continuous mental support and sacrifice.

Anamul Haque Sajib

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this Chapter we will mainly focus on the content of this PhD thesis and give a general picture about how this PhD thesis is organized. This PhD thesis is closely related to two areas of applied statistics: (i) Bayesian statistics and (ii) Statistical Shape Analysis. In addition to these two areas it also uses some tools from differential geometry. We organize this Chapter as follows: In section 1 we introduce Bayesian Statistics and discuss some of its terminology in detail where it is used directly in this thesis, while an introduction of Statistical Shape Analysis with various applied problems and how it is linked with Bayesian Statistics are covered in section 2.

Section 3 discusses some ideas from differential geometry which are helpful to understand some of the parts (especially Chapter 4) of this PhD thesis. We haven't gone through on these terminology in depth as this is not the focus of this thesis but we emphasize on understanding the contents which have been used in this thesis.

The final section of this Chapter narrate how this PhD thesis is organized.

1.1 The Bayesian Paradigm

Suppose we have the observed data $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$ from some population with unknown parameter $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_d)$, and our aim is to know about the unknown parameter $\boldsymbol{\theta}$ based on this observed sample $\boldsymbol{y}$. Both frequentist and Bayesian approaches are based on specifying a probability distribution for each of the observed data y given $\boldsymbol{\theta}$, say $\pi(y_i \mid \boldsymbol{\theta})$. Based on the independent identical distribution (IID) assumptions, we can construct the likelihood function $\pi(\boldsymbol{y} \mid \boldsymbol{\theta})$ which is in fact the product of n probability densities of $\boldsymbol{y}$ i.e. $\prod_{i=1}^{n} \pi(y_i \mid \boldsymbol{\theta})$. In the frequentist approach, the unknown parameter $\boldsymbol{\theta}$ is considered as fixed quantity and can be estimated by directly using the likelihood function. For example, maximum likelihood estimates of $\boldsymbol{\theta}$ can be obtained by maximizing $\pi(\boldsymbol{y} \mid \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

On the other hand, in the Bayesian approach, $\boldsymbol{\theta}$ is considered as a random variable which has a probability distribution $\pi(\boldsymbol{\theta})$ known as prior distribution in Bayesian literature. Using Bayes' theorem both the information from observed data and prior distribution are combined, resulting in the following posterior distribution $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$, which is the basis for any inference regarding $\boldsymbol{\theta}$:

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{\pi(\boldsymbol{y} \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})}{\int_{\Omega} \pi(\boldsymbol{y} \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}}\,, \tag{1.1}$$

where Ω denotes the parameter space of $\boldsymbol{\theta}$. The quantity $c(y) = \int_{\Omega} \pi(\boldsymbol{y} \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$ in the denominator of equation (1.1) normalizes the posterior density $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$, and is known as the marginal distribution of the observed data. For any function $h(\boldsymbol{\theta})$, a function of $\boldsymbol{\theta}$, we can define the following integral

$$I = \int_{\Omega} h(\boldsymbol{\theta})\, \pi(\boldsymbol{\theta} \mid \boldsymbol{y})\, \mathrm{d}\boldsymbol{\theta}, \tag{1.2}$$

which is simply the posterior expectation of $h(\boldsymbol{\theta})$, $E_{\pi(\boldsymbol{\theta}|\boldsymbol{y})}\left[h(\boldsymbol{\theta})\right]$, and calculat-

ing the expectation of function $h(\boldsymbol{\theta})$ with respect to the posterior distribution is essentially the same as evaluating the integral defined in equation (1.2). We could evaluate this integral numerically by using quadrature method which offers a great accuracy in lower dimension, but accuracy decreases rapidly with the dimensionality of the problem. Therefore, for a high dimensional statistical model there is little use of the quadrature method, and Monte Carlo methods which require samples from $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ (see subsection 1.1.1) are an effective way to evaluate this integral (Metropolis and Ulam (1949)).

For most complex statistical models, $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ defined in (1.1) has no closed form because of the unavailability of a normalizing constant $c(y)$. For example, consider a Bayesian posterior for the variance parameter σ^2 of a normal distribution with known mean parameter μ. Then according to equation (1.1), the posterior for σ^2 is

$$\pi(\sigma^2 \mid \boldsymbol{y}, \mu) = \frac{\pi(\boldsymbol{y} \mid \mu, \sigma^2)\, \pi(\sigma^2)}{\int_0^\infty \pi(\boldsymbol{y} \mid \mu, \sigma^2)\, \pi(\sigma^2)\, \mathrm{d}\sigma^2}. \tag{1.3}$$

Suppose we consider a non-conjugate prior for σ^2 which has the following distributional form

$$\pi(\sigma^2) = \frac{1}{\sigma^2\, \tau_0\, \sqrt{2\pi}} \exp\Big\{ - \frac{(ln\sigma^2 - \theta_0)^2}{2\tau_0^2} \Big\}. \tag{1.4}$$

The distribution defined in equation (1.4) is known as log-normal distribution with parameters θ_0 and ${\tau_0}^2$. Now the product of the likelihood and prior, $\pi(\boldsymbol{y} \mid \mu, \sigma^2)\, \pi(\sigma^2)$, is

$$\pi(\boldsymbol{y} \mid \mu, \sigma^2)\, \pi(\sigma^2) = \frac{1}{\sigma\, \sqrt{2\pi}} \exp\Big\{ - \frac{(\boldsymbol{y} - \mu)^2}{2\sigma^2} \Big\} \frac{1}{\sigma^2\, \tau_0\, \sqrt{2\pi}} \exp\Big\{ - \frac{(ln\sigma^2 - \theta_0)^2}{2\tau_0^2} \Big\}$$

$$\propto \frac{1}{\sigma^3} \exp\Big\{ - \frac{(\boldsymbol{y} - \mu)^2}{2\sigma^2} \Big\} \exp\Big\{ - \frac{(ln\sigma^2 - \theta_0)^2}{2\tau_0^2} \Big\}.$$

The posterior defined in equation (1.3) can't be evaluated explicitly as the integral $\int_0^\infty \pi(\boldsymbol{y} \mid \mu, \sigma^2)\, \pi(\sigma^2)\, \mathrm{d}\sigma^2$ involved in the denominator of its right hand side is algebraically intractable. An enormous literature in computational statistics is available to tackle this issue i.e. how to sample from these types of posterior distributions and evaluate the normalizing constant which involves a high dimensional integral problem. Markov Chain Monte Carlo (MCMC) methods can be an effective method to simulate from such high dimensional posterior distributions, which don't require knowledge of the normalizing constant of the posterior distribution. This is discussed further in subsection 1.1.2.

1.1.1 Monte Carlo Integration

Suppose we want to evaluate the following integral defined in equation (1.2)

$$I = \int_\Omega h(\boldsymbol{\theta})\, \pi(\boldsymbol{\theta} \mid \boldsymbol{y})\, \mathrm{d}\boldsymbol{\theta},$$

where $\boldsymbol{\theta}$ is our d dimensional random variable with pdf $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$. We also suppose that here we can generate samples from $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ easily for the sake of illustrating Monte Carlo integration clearly, and $\boldsymbol{\theta}^{(\mathrm{i})} = (\theta_{i1}, \theta_{i2}, \cdots, \theta_{id})$ consists of a sample of size n from $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ where $i = 1, 2, \cdots, n$. Then we can approximate the above integral I by defining a Monte Carlo estimator $\hat{I}$ as

$$\hat{I} = \frac{1}{n} \sum_{i=1}^{n} h(\boldsymbol{\theta}^{(\mathrm{i})}). \tag{1.5}$$

The above estimator $\hat{I}$ is an unbiased estimator of I and converges to the true integral almost surely (proofs are omitted here). In addition to these properties, according to the central limit theorem $\hat{I}$ follows $\mathcal{N}(I, \frac{\sigma^2}{n})$, where $\sigma^2 = \mathbb{V}ar(h(\hat{\boldsymbol{\theta}}^{(\mathrm{i})}))$ and can be estimated as follows

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(h(\boldsymbol{\theta}^{(\mathrm{i})}) - \hat{I}\right)^2. \tag{1.6}$$

The advantage of using Monte Carlo Integration to estimate I is the covergence speed of $\hat{I}$ does not depend on the dimension of I provided that the samples are independent. However, generating independent samples from a high dimensional and strongly correlated probability distribution is a tricky task and we will discuss one MCMC technique called Hamiltonian Monte Carlo in chapter 4 to draw samples from such distributions which are minimally correlated.

Illustrative Example

Suppose we wish to evaluate the integral $I = \int_0^{\pi/2} (\sin\theta + \cos\theta)\,\mathrm{d}\theta$ using Monte Carlo method. Here we consider θ which is one dimensional to simply explain the implementation of Monte Carlo integration. To evaluate I using Monte Carlo integration at first we need to write down I in the form of an expectation of some function $h(\theta)$ of a random variable θ with pdf $\pi(\theta)$ i.e.

$$I = \int_0^{\pi/2} \frac{\pi}{2}(\sin\theta + \cos\theta)\frac{2}{\pi}\,\mathrm{d}\theta = \int_0^{\pi/2} h(\theta)\,\pi(\theta)\,\mathrm{d}\theta, \tag{1.7}$$

where $h(\theta) = \frac{\pi}{2}(\sin\theta + \cos\theta)$ and $\pi(\theta) = \frac{2}{\pi}$. In the second step a sample of size n is required to draw from $\pi(\theta) = \frac{2}{\pi}$ which is uniform $(0,\, \pi/2)$, and suppose we have a samples $(\theta_1, \theta_2, \cdots, \theta_n)$ in our hand from this distribution. Finally, we can estimate I by defining Monte Carlo estimator $\hat{I}$ which uses generated samples directly as follows

$$\hat{I} = \frac{h(\theta_1) + h(\theta_2) + \cdots + h(\theta_n)}{n}. \tag{1.8}$$

The true value of the integral I defined in equation (1.7) is 2, and the Monte

Carlo estimates of I based on the sample sizes $n = 100, 1000$ and 10000 are 2.0018, 1.9983 and 2.0025, respectively.

1.1.2 Markov Chain Monte Carlo Methods

Before we begin the main discussion about MCMC, we first review some key facts about Markov chains which are helpful to understand MCMC.

Markov chain. Suppose we have two countable sets; one for state space $S = \{1, 2, \cdots, r\}$ and other for time space $T = \{1, 2, \cdots, t, \cdots, n\}$. Then a discrete time and discrete state space Markov chain is a sequence $X_1, X_2, \cdots, X_{n-1}, X_n$ of random variables taking values in S at time T which follows the following Markov property

$$\Pr(X_n \in S \mid X_1 = x_1, \cdots, X_{n-1} = x_{n-1}) = \Pr(X_n \in S \mid X_{n-1} = x_{n-1}), \quad (1.9)$$

i.e. the probability of moving to the next state only depends on its immediate past state and not the other previous states. If the state space S is continuous then we call the above Markov chain discrete time continuous state space Markov chain. The state space S can be discrete or continuous or mix of both.

Transition function. Transition function, also known as transition kernel characterizes the dynamics of the Markov chain. In other words, it is the conditional probability of Markov chain moving to a new state from a given initial state. For example, suppose the Markov chain is in currently at $\boldsymbol{\theta}$, then the probability of moving to new state $\boldsymbol{\theta}^\star$ from the initial state $\boldsymbol{\theta}$ is given by the transition function $Q(\boldsymbol{\theta}^\star | \boldsymbol{\theta})$ with the following form

$$Q(\boldsymbol{\theta}^\star | \boldsymbol{\theta}) = f_{\boldsymbol{X}_n | \boldsymbol{X}_{n-1}}(\boldsymbol{\theta}^\star | \boldsymbol{\theta}), \quad (1.10)$$

where f is a some function defined on $f : \mathbb{R}^d \to \mathbb{R}$, and we will discuss its mathematical form in the MCMC discussion section. This $Q(\boldsymbol{\theta}^\star | \, \boldsymbol{\theta})$ plays the same role as the transition probability matrix in discrete time and discrete state space Markov chain.

Irreducible Markov chain. A Markov chain is said to be irreducible if it can move from any given state to any other state in a finite number of steps. For example, suppose the Markov chain is in at state x at time t, then there exist a $w \in \mathbb{N}$ such that $\Pr(X_{t+w} \in S \mid X_t = x) > 0$.

Periodicity of Markov chain. If a Markov chain is currently at state $X_t = x$ at time t and suppose that it returns to the same state after w steps, where $w \in \mathbb{N}$, then the period of this state can be defined as

$$k = \text{gcd}\{w > 0 : \Pr(X_{t+w} = x \mid X_t = x\}, \tag{1.11}$$

where "gcd" is the greatest common divisor of $(t, t+w)$. Period of any state can be defined only when (1.11) is defined i.e. the set is not empty. For example, suppose $w = \{6, 9, 12, 18\}$ i.e. state x_t returns to x_t again after 6, 9, 12 and 18 time steps then period of x_t is 3 as the greatest common divisor of these number is k=3. If the value of k is greater than 1 then the period of this state is called periodic otherwise it is called aperiodic. All states of a Markov chain need to be aperiodic state to be an aperiodic Markov chain while an irreducible Markov chain only needs one aperiodic state (implying all states are aperiodic) to be an aperiodic Markov chain. Suppose we have the Markov chain with state space $S = \{a, b, c\}$ and transition matrix P given below:

$$P = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} 0 & 1/3 & 2/3 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array},$$

where the states a, b and c communicate each other, and hence this chain becomes an irreducible Markov chain. All states of this irreducible Markov chain belong to one communicating class and have the same period (all states in the same communicating class have the same period). Therefore, aperodicity of any state of irreducible Markov chain implies the aperodicity of the remaing states. For example, the period of state a of this Markov chain is the "gcd" of 2 and 3 which is 1, aperiodic, as state a can be reached after starting from state a again after 2 and 3 steps respectively. As the states a, b and c belong to the same class and the state a is aperiodic, so the states b and c are also aperiodic.

Introduction of MCMC

In this section, we will cover some important theoretical methodology of MCMC and some of the most commonly used MCMC techniques in Bayesian Statistics like Metropolis-Hastings and Gibbs Sampling algorithms. In addition, MCMC implementing issues are discussed, which plays an important role for inference procedure.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953) first used the Markov chain to simulate samples from an arbitrary probability distribution (which need only be known up to a normalizing constant), and the samples produced by this Markov chain are essentially the correlated samples from the target density we are interested in. As the Markov chain produces correlated samples which affects the variance of our Monte Carlo estimator, the dependence of samples needs to be properly accounted for in a valid inference. Therefore, our aim is to use a Markov chain Monte Carlo method that

reduces this dependence as much as possible.

MCMC starts working by firstly defining a proposal density $T(. \mid .)$ by which the Markov chain explores the parameter space, and the efficiency of most of the Markov chain is determined by how efficiently the proposal density explores the parameter space. Therefore, to make an efficient MCMC algorithm our aim is to choose a proposal density which proposes new parameter values sufficiently far from the current parameter values with high probability of acceptance, achieving quick convergence and less correlated samples (explained in Figure 1.1). The Metropolis-Hastings (MH) algorithm which gives us a mechanism to construct the transition function $Q(. \mid .)$ is defined as follows The above algorithm is

Algorithm 1: Standard Metropolis-Hastings (MH) Algorithm

Data: Current value of $\boldsymbol{\theta}$.

Result: Simulated value from the target density $\pi(\boldsymbol{\theta})$.

begin

1: Propose new state $\boldsymbol{\theta}^\star$ from the proposal density $T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta})$

2: Calculate the (MH) acceptance ratio

$$\alpha(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta}) = \min\left[1, \frac{\pi(\boldsymbol{\theta}^\star)T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star)}{\pi(\boldsymbol{\theta})T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta})}\right]$$

3: **begin**

3a: Start accept reject steps

3b: Draw $U \sim \text{Uniform}\big[0, 1\big]$

3c: **if** $U < \alpha(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta})$ **then**

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^\star$

else

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}$

3d: Stop accept reject steps

known as the generalized version of the original Metropolis algorithm, proposed by Metropolis et al. (1953), in which the proposal density is required to be symmetric. But this generalized version made by W. Hastings (1970) allows any

normalized probability density function as a proposal density $T(. \mid .)$. Indeed, W. Hastings (1970) was the first person who described the above algorithm from a statistical point of view i.e. in terms of a Markov chain to simulate samples from an arbitrary probability distribution which needs to be known up to normalizing constant, although its application conducted by physicists had been seen firstly in statistical mechanics.

We can now show that the chain produced by the above MH algorithm does converge to the required stationary distribution (target density we are interested in) with the following transition function $Q(\boldsymbol{\theta}^{\star} | \, \boldsymbol{\theta})$ defined in (1.10) where

$$Q(\boldsymbol{\theta}^{\star} | \, \boldsymbol{\theta}) = T(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta}) \, \alpha(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta}), \tag{1.12}$$

and $\alpha(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta})$ is defined in Algorithm 1. The above equation can be seen as the product of two probabilities; (i) probability of proposing a point $\boldsymbol{\theta}^{\star}$ from $T(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta})$ and (ii) the probability of accepting the proposed $\boldsymbol{\theta}^{\star}$ as a new state. To converge, a Markov chain requires the following conditions hold (Robert and Casella (2004)):

$$\pi(\boldsymbol{\theta}^{\star}) = \int_{\Omega} \pi(\boldsymbol{\theta}) \, Q(\boldsymbol{\theta}^{\star} | \, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}, \quad \forall \, \boldsymbol{\theta}^{\star}. \tag{1.13}$$

To understand the meaning of equation (1.13) let's explain first its right hand side which is simply the average probability of moving to a new point $\boldsymbol{\theta}^{\star}$ from any initial point $\boldsymbol{\theta}$ in the parameter space. Therefore we can say a Markov chain has converged when the average probability of moving to a new point $\boldsymbol{\theta}^{\star}$ from any initial point $\boldsymbol{\theta}$ is equal to the probability of $\boldsymbol{\theta}^{\star}$ itself. Apart from equation (1.13), the convergence of a Markov chain can also be explained under the following, more restrictive, condition known as detailed balance equation

$$\pi(\boldsymbol{\theta})\, Q(\boldsymbol{\theta}^{\star}|\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}^{\star})\, Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{\star}). \tag{1.14}$$

The detailed balance equation defined in equation (1.14) says the probability of moving to point $\boldsymbol{\theta}^{\star}$ from $\boldsymbol{\theta}$ is equal to the probability of moving to point $\boldsymbol{\theta}$ from $\boldsymbol{\theta}^{\star}$, and the chains that satisfy this symmetry constraint are called reversible Markov chains. We can obtain equation (1.13) from equation (1.14) by integrating both sides of equation (1.14) with respect to $\boldsymbol{\theta}$ i.e. assuming detailed balance can guarantee equation (1.13) holds

$$\begin{aligned}
\int \pi(\boldsymbol{\theta})\, Q(\boldsymbol{\theta}^{\star}|\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} &= \int \pi(\boldsymbol{\theta}^{\star})\, Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{\star})\, \mathrm{d}\boldsymbol{\theta} \\
&= \pi(\boldsymbol{\theta}^{\star}) \int Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{\star})\, \mathrm{d}\boldsymbol{\theta} \\
&= \pi(\boldsymbol{\theta}^{\star}) \times 1 \\
&= \pi(\boldsymbol{\theta}^{\star}).
\end{aligned} \tag{1.15}$$

In the above calculation $\int Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{\star})\, \mathrm{d}\boldsymbol{\theta}$ is equal to 1 as $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{\star})$ is a normalized probability density function of random variable $\boldsymbol{\theta}$. The detailed balance equation clearly holds when $\boldsymbol{\theta} = \boldsymbol{\theta}^{\star}$, and in the following we will show that it also holds when $\boldsymbol{\theta} \neq \boldsymbol{\theta}^{\star}$:

Case 1: $\dfrac{\pi(\boldsymbol{\theta}^{\star})T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{\star})}{\pi(\boldsymbol{\theta})T(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta})} \leq 1$

Assuming $\pi(\boldsymbol{\theta}^{\star})T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{\star}) \leq \pi(\boldsymbol{\theta})T(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta})$ gives us $\alpha(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta}) = \pi(\boldsymbol{\theta}^{\star})T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{\star})/\pi(\boldsymbol{\theta})T(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta})$ while $\alpha(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{\star})$=1 when $\pi(\boldsymbol{\theta})T(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta}) \geq \pi(\boldsymbol{\theta}^{\star})T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{\star})$.

Now we have

$$\begin{aligned}
\pi(\boldsymbol{\theta})\, Q(\boldsymbol{\theta}^\star|\boldsymbol{\theta}) &= \pi(\boldsymbol{\theta})\, T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta})\, \alpha(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta}) \\
&= \pi(\boldsymbol{\theta})\, T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta})\, \frac{\pi(\boldsymbol{\theta}^\star) T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star)}{\pi(\boldsymbol{\theta}) T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta})} \\
&= \pi(\boldsymbol{\theta}^\star) T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star) \\
&= \pi(\boldsymbol{\theta}^\star) T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star)\, \alpha(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star) \qquad \left[\text{as } \alpha(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star) = 1\right] \\
&= \pi(\boldsymbol{\theta}^\star)\, Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star) \qquad (1.16)
\end{aligned}$$

Case 2: $\dfrac{\pi(\boldsymbol{\theta}^\star) T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star)}{\pi(\boldsymbol{\theta}) T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta})} \geq 1$

Like as previous case assuming $\pi(\boldsymbol{\theta}^\star) T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star) \geq \pi(\boldsymbol{\theta}) T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta})$ gives us $\alpha(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta}) = 1$ while $\alpha(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star) = \pi(\boldsymbol{\theta}) T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta}) / \pi(\boldsymbol{\theta}^\star) T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star)$ when $\pi(\boldsymbol{\theta}) T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta}) \leq \pi(\boldsymbol{\theta}^\star) T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star)$. Now we have

$$\begin{aligned}
\pi(\boldsymbol{\theta}^\star)\, Q(\boldsymbol{\theta}|\boldsymbol{\theta}^\star) &= \pi(\boldsymbol{\theta}^\star)\, T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star)\, \alpha(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star) \\
&= \pi(\boldsymbol{\theta}^\star)\, T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star)\, \frac{\pi(\boldsymbol{\theta}) T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta})}{\pi(\boldsymbol{\theta}^\star) T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^\star)} \\
&= \pi(\boldsymbol{\theta}) T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta}) \\
&= \pi(\boldsymbol{\theta}) T(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta})\, \alpha(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta}) \qquad \left[\text{as } \alpha(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta}) = 1\right] \\
&= \pi(\boldsymbol{\theta})\, Q(\boldsymbol{\theta}^\star \mid \boldsymbol{\theta}) \qquad (1.17)
\end{aligned}$$

Apart from satisfying these conditions a Markov chain also needs to be irreducible and aperiodic to achieve convergence (Robert and Casella (2004)), which means that there is a non-zero probability of reaching any state of a Markov chain in parameter space from any other state within a finite number of steps, and that the chain does not get stuck in any loops such that it repeatedly visits the same location. Consider a Markov chain with the following transition matrix

$$P = \begin{array}{c} \\ a \\ b \end{array} \begin{array}{c} \begin{array}{cc} a & b \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array},$$

where a and b are the two states of this Markov chain which is irreducible and periodic. Clearly the periodic behvaiour of this Markov chain will interfere the convergence to an equilibrium distribution as

$$Pr\,(X_t = b|X_0 = b) = \begin{cases} 1 & \text{for even values of t,} \\ 0 & \text{for odd values of t,} \end{cases}$$

which implies that the probability cann't converge to any single value as $t \to \infty$.

Gibbs Sampler

One of the best known MCMC algorithms in Bayesian computational literature is the Gibbs sampler which was invented by Geman and Geman (1984). The main idea in the Gibbs sampler is to simulate from the distribution of each parameter in turn conditioned on the remaining parameters and the data, known as the full conditional distribution. In most complex situations, sampling from the overall high dimensional target density (posterior density) is quite challenging (sometimes even impossible) because of its intractable form. Fortunately, for these type of target densities full conditionals distributions can often be derived easily and easy to sample from them in most of the cases which altogether make this method so popular.

Suppose our aim is to simulate from $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ where $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_d)$ is a d dimensional parameter vector and $\boldsymbol{y}$ is our observed data. In some situations the parameter vector $\boldsymbol{\theta}$ may contain the missing variable as well, and we will see this type of scenario in our proposed model in chapter 4. The basic schemes of the Gibbs sampler to simulate from any target density $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ are given as follows:

Gibbs Sampling Algorithm

1. Start from an arbitrary point $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \cdots, \theta_d^{(0)})$ and set i=0
2. Generate $\boldsymbol{\theta}^{(i+1)} = (\theta_1^{(i+1)}, \cdots, \theta_d^{(i+1)})$ given $\boldsymbol{\theta}^{(i)} = (\theta_1^{(i)}, \cdots, \theta_d^{(i)})$ as follows:
   - Generate $\theta_1^{(i+1)}$ from $\pi(\theta_1 | \theta_2^{(i)}, \theta_3^{(i)}, \cdots, \theta_d^{(i)})$
   - Generate $\theta_2^{(i+1)}$ from $\pi(\theta_2 | \theta_1^{(i+1)}, \theta_3^{(i)}, \cdots, \theta_d^{(i)})$
   - Generate $\theta_3^{(i+1)}$ from $\pi(\theta_3 | \theta_1^{(i+1)}, \theta_2^{(i+1)}, \theta_4^{(i)}, \cdots, \theta_d^{(i)})$

     $\vdots$
   - Generate $\theta_{(d-1)}^{(i+1)}$ from $\pi(\theta_{(d-1)} | \theta_1^{(i+1)}, \theta_2^{(i+1)}, \theta_3^{(i+1)}, \cdots, \theta_{d-2}^{(i+1)}, \theta_d^{(i)})$
   - Generate $\theta_d^{(i+1)}$ from $\pi(\theta_d | \theta_1^{(i+1)}, \theta_2^{(i+1)}, \theta_3^{(i+1)}, \cdots, \theta_{d-1}^{(i+1)})$
3. Set $i = i + 1$ and go back to step 2

Repeat the above cycle N times and all the $\{\boldsymbol{\theta}^{(i)}; i = 1, 2, \cdots, N\}$ values are the required samples from our target density $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$. As the successive values of $\boldsymbol{\theta}$ are correlated, special care needs to be taken before using these samples for inference, which is discussed in the section on MCMC implementing isssues. When direct generation from each full conditional distribution is available, Gibbs sampler is quite fast and easy to implement, otherwise MH algorithm can be used to simulate from the full conditional distributions for which direct generation is not directly available. Sometimes MH algorithm can be used to simulate from any distribution although direct generation from it is available in order to achieve better proposals and mixing, which is shown in the following example:

Illustrative Example

Suppose we wish to sample from a bivariate normal distribution which has the following form

$$\pi(\boldsymbol{X}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^2 |\boldsymbol{\Sigma}|}} \exp\{-\frac{1}{2}(\boldsymbol{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{X} - \boldsymbol{\mu})\},$$

where $\boldsymbol{\mu} = (0, 0)^T$, $\boldsymbol{\Sigma} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ with $\rho = 0.98$, and $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$. Under Bayesian settings, we can use the Gibbs sampler to sample from the above density by directly sample from the follwing full conditional distributions of X_1 and X_2 where $\boldsymbol{X} = (X_1, X_2)$:

$$\pi(X_1|X_2) \sim \mathrm{N}(\rho X_2, (1 - \rho^2))$$
$$\pi(X_2|X_1) \sim \mathrm{N}(\rho X_1, (1 - \rho^2)).$$

We can also use the Metropolis-Hastings based sampler to draw samples from these full conditional distributions instead of direct generation using the Gibbs sampler in order to bring better proposal and good mixing, as shown in Figure 1.1. The first row of Figure 1.1 shows the first 50 iterations of MCMC output of Gibbs and Hamiltonian Monte Carlo (HMC), Metropolis-Hastings based sampler, which is discussed in chapter 4 in detail. From this row we can see that HMC has explored the target density very quickly, even in first 50 iterations, while Gibbs sampler hasn't explored yet. From the plot of first 500 iterations, it is obvious that Gibbs sampler has explored slowly the parameter space of the target density, which is shown by the second row of Figure 1.1. The ACF plots, shown in third row of Figure 1.1, show the quality of MCMC output of both the samplers, and from these plots it is obvious that the quality of MCMC output based on HMC is less correlated compared to the output of MCMC output based on Gibbs sampler.

In this example, as the target distribution is highly correlated the Gibbs sampler faces difficulty to explore its most constrained direction, whilst HMC explores the most constraint direction efficiently, proposing a new state far from the current state based on first order geometric information. The proposed state then accepted or rejected based on Metropolis-Hastings acceptance probability. Motivated by this problem we will use Metropolish Hastings based sampler to generate

samples from the full conditional distributions of rotation and matching matrix in chapter 4 although direct generation from each of the full conditional distribution is available.

MCMC Implementing Issues

The invention of MCMC method not only revolutionized statistical analysis but made Bayesian method applicable in more challenging and complex problems. However, this method has several limitations along with some implementing issues that need to be accounted for properly before making any inference based on MCMC outputs. Here we discuss some of the important issues regarding MCMC and their possible remedy.

Dependent Samples. Samples produced by Markov chain are no longer independent as successive observations are correlated which needs to take into account for making any valid inference. Suppose the Markov chain produces the samples $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \cdots, \boldsymbol{\theta}^{(P)}, \boldsymbol{\theta}^{(P+1)}, \boldsymbol{\theta}^{(P+2)}, \cdots, \boldsymbol{\theta}^{(N)}$ where N is the sample size, and it is also assumed that the chain is converged after P iterations. Here P is known as burn in period, samples up to P point are discarded. The Monte Carlo estimator based on $\boldsymbol{\theta}^{(P+1)}, \boldsymbol{\theta}^{(P+2)}, \cdots, \boldsymbol{\theta}^{(N)}$ is still an unbiased estimator, and to get an an accurate estimate N should be large enough as Markov chain moves slowly when successive values are correlated. However, the variance of the Monte Carlo estimator can't be calculated using standard methods as samples are not independent. As a possible remedy of successive correlation, samples are thinned, retaining only every k^{th} observation, so that resulting samples are close to independent. Then standard methods can be used to calculate the variance of Monte Carlo estimator. Hamiltonian Monte Carlo proposed by Neal (2011), discussed in Chapter 4 in detail, is another way to get less correlated samples (close to independent) compared to samples produced by usual MCMC methods.

Global Convergence. When the target density is multimodal, very common

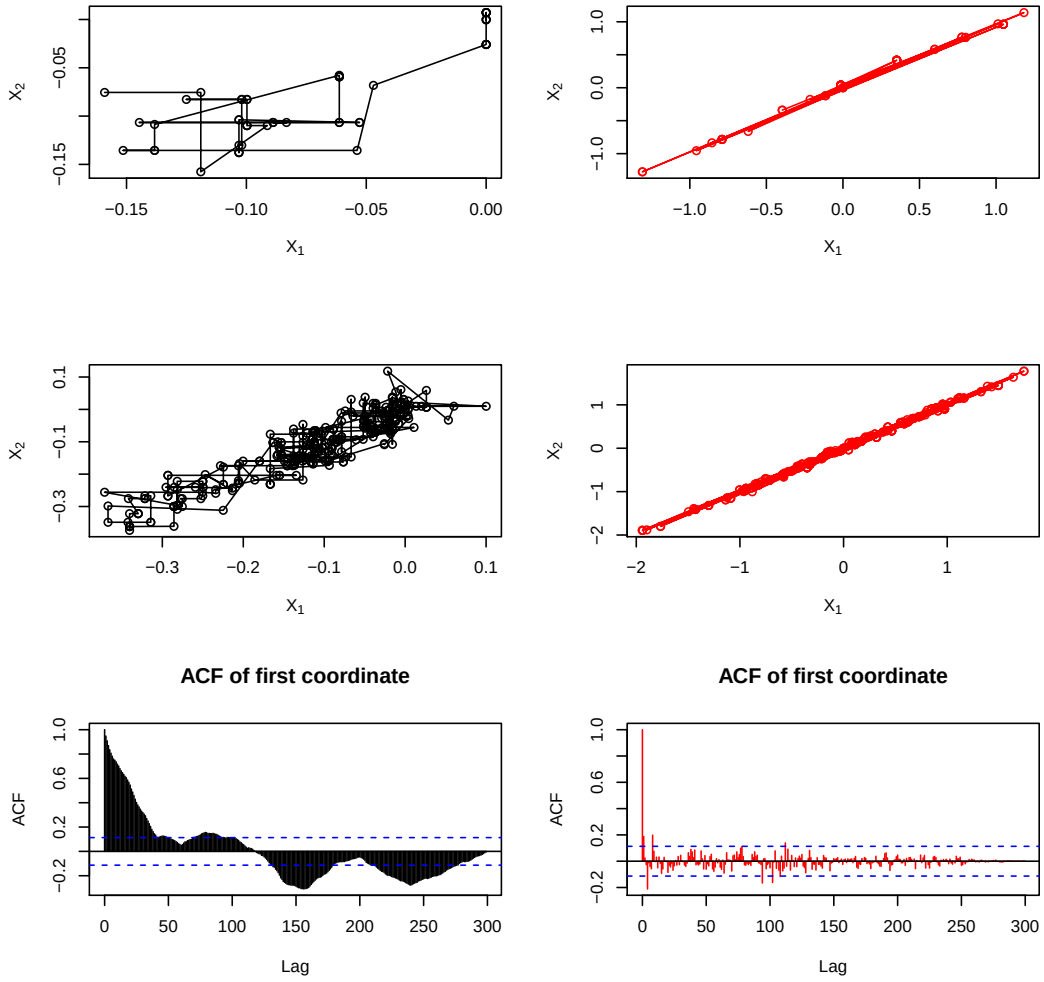

Figure 1.1: MCMC output based on the Gibbs and the MH based samplers; the left panel (black) and the right panel (red) of Figure 1.1 are used to represent the MCMC output obtained under the Gibbs and MH based sampler respectively

in protein matching problems considered in this thesis, exploring all modes with correct frequency (probability) from any random starting point is quite challenging as chain may get stuck in one particular mode. One of our aims is to use an efficient sampler that can visit all modes with correct frequency without becoming stuck in one particular mode. As a possible solution it is often useful to use parallel tempering proposed by Geyer (1991); Liu (2008) which helps the Markov chain to explore all modes with correct frequencies by introducing some tempered distributions of target density. Suppose $\pi(\boldsymbol{\theta})$ is our target distribution, then a heated version of this target distribution is $[\pi(\boldsymbol{\theta})]^{1/R}$, tempered distributions of target distributions, where R ($R \geq 1$) is called the temperature of the chain. The greater the value of R the flatter the target density. The idea behind introducing R in the target distribution is that a heated chain can readily move parameter space as the density is flattened relative to the cold (R=1) chain and attempts swapping states between the chains. Therefore, the cold chain will successfully swap states with a heated chain occasionally that may be exploring another mode in the sample space. This allows the cold chain to effectively move one mode to another mode.

For example, suppose the target density $\pi(\theta)$ (considered one dimensional for the sake of explanation) has the following form which is proportional to

$$\pi(\theta) \propto \exp\left\{-16\left(\theta^2-1\right)^2\right\}; \quad -2 \leq \theta \leq 2.$$

This distribution is bimodal and its two modes occur at $\theta = -1$ and $\theta = +1$. The role of introducing R in $\pi(\theta)$, known as temepred distributions of target density, is best illustrated through Figure 1.2, where the unnormalized densities of $[\pi(\theta)]^{1/R}$ are plotted for 6 different values of $R = 1, 2, 4, 8, 12, 16$. Tempered distribution with $R = 1$ is the same as our target density. From Figure 1.2, we can see that as temperatures are increased; the shape of the tempered distribu-

tions become relatively flat compare to the distribution with $R = 1$. Here we restrict the range of temperatures to $R \geq 1$ as we don't want to sample from the distribution which is more peaked than the target density defined in above.

We used Metropolis-Hastings algorithm to sample from this tempered distribution for 6 different values of R, and the MCMC output for these 6 sampler are shown in Figure 1.3. From these Figures we can see that MH sampler gets stuck at one mode when the distribution is highly peaked (when R is 1 and 2) while the sampler can easily explore the two modes when the distribution becomes more flatter (when R is 4, 8, 12 and 16) compared to the tempered distributions with $R = 1$ and 2. To sample from the target density, tempered distribution with R =1, we introduce pararell tempering within the MH sampler which is summarised as follows: Figure 1.4 shows the MCMC output of MH sampler with parallel tempering, where 6 chains are considered. From this Figure we can see that sampler explore the two modes successesfully of the target distribution although there is a scope to improve the mixing which depends on the number of chains chosen in parallel temering and effective proposal density of MH sampler. We haven't covered this mixing issue here as our aim is to explain only the algorithm of parallel tempering.

Choice of Proposal. The success of MCMC algorithm depends a lot how proposal density is chosen. A good MCMC algorithm should have a proposal density which helps the Markov chain to explore the parameter space quickly with a good acceptance rate. However, having a Markov chain which possess these two criteria together is quite challenging, and most of the time a trade - off is made between these two criteria. For example, a Markov chain with a proposal density where large proposals (big jump from current point) are made in order to explore the parameter space quickly may finally have low acceptance rate because of rejection of too many proposals, and the resulting chain often stays in the same place. On the other hand, a Markov chain with a proposal

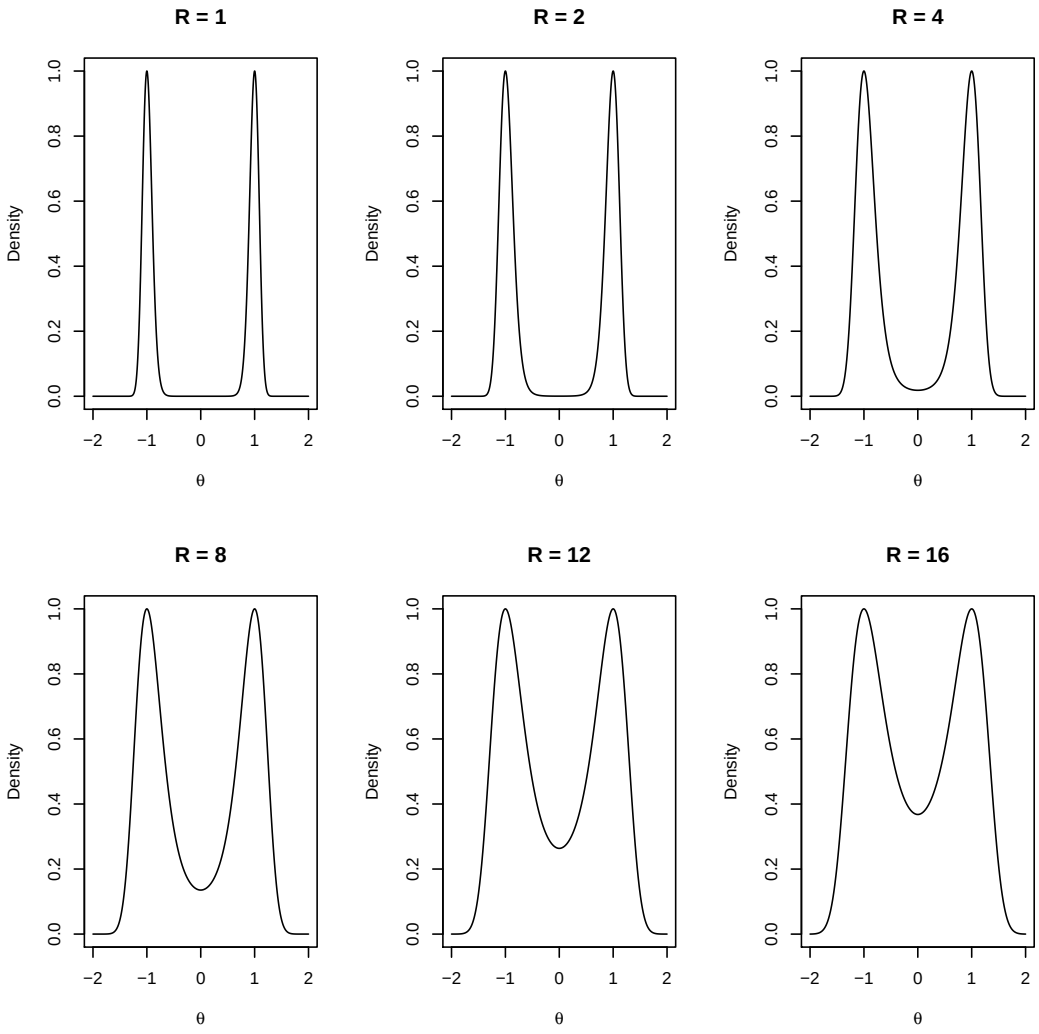

Figure 1.2: Density plots of tempered distribution for 6 different values of R

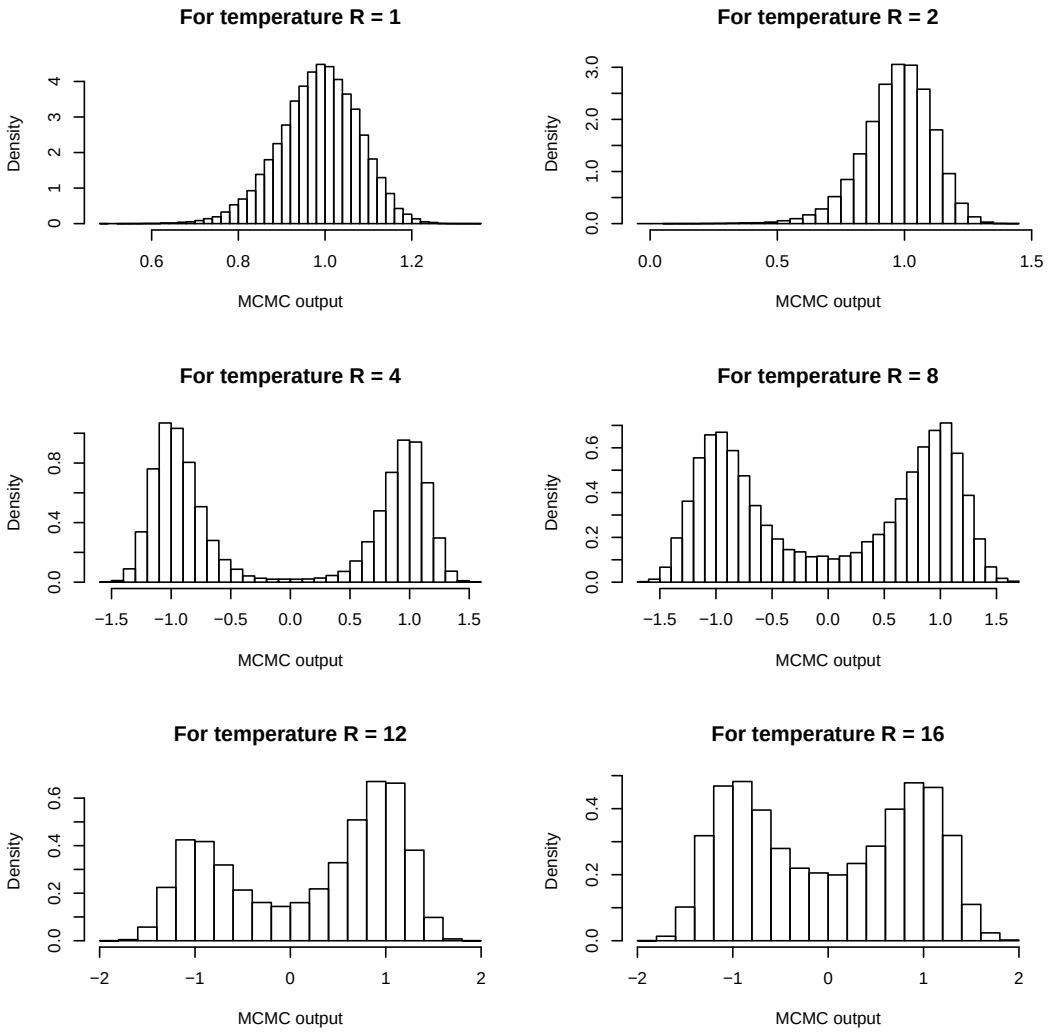

Figure 1.3: MCMC output of MH sampler for 6 different values of R

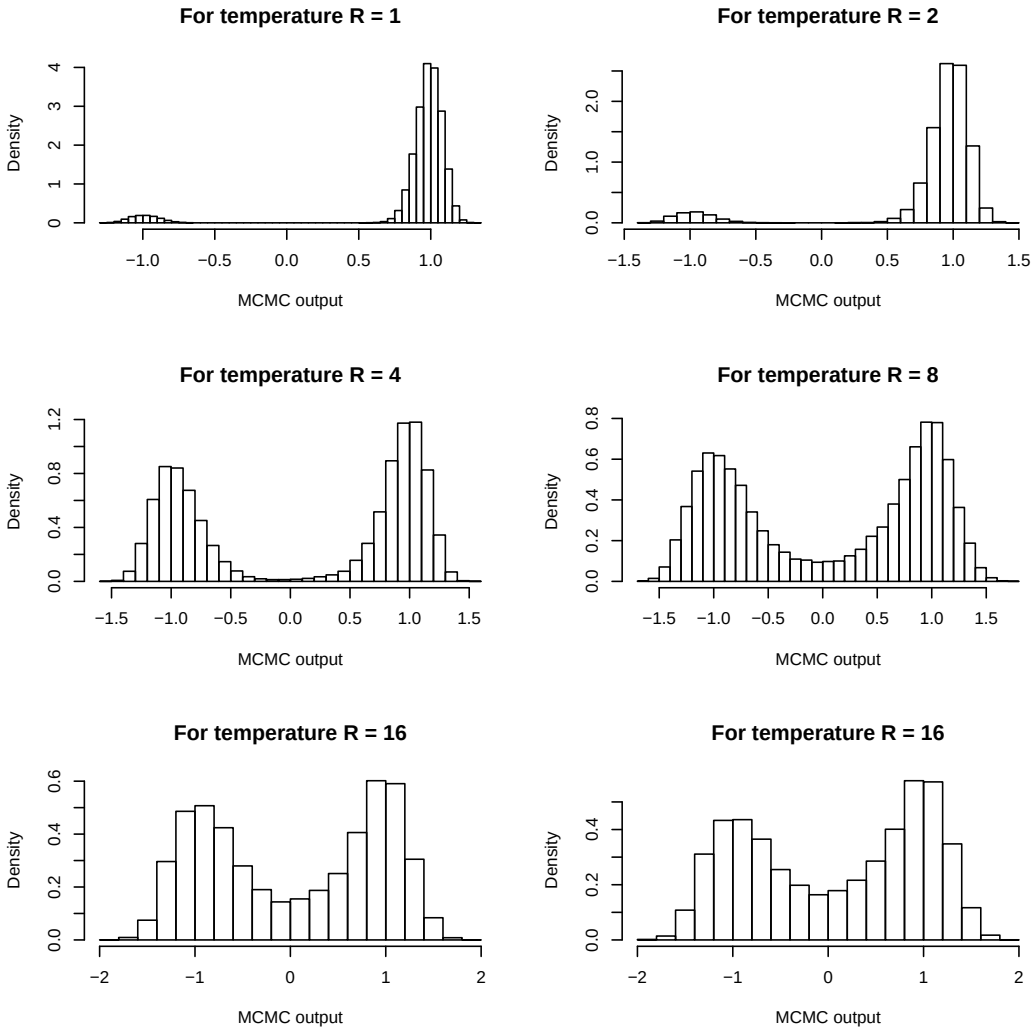

Figure 1.4: MCMC output of MH sampler with parallel tempering for 6 different values of R (6 chains)

Algorithm 2: Parallel Tempering Algorithm

Data: Current value of $\boldsymbol{\theta}$ for all the chains R (R =1, 2, $\cdots$, n).

Result: Simulated value from the target density $\pi(\boldsymbol{\theta})$.

begin

for R = *1, 2,* $\cdots$, n **do**

1: Propose new state $\boldsymbol{\theta}^{\star}$ from the proposal density $T(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta})$

2: Calculate the (MH) acceptance ratio

$$\alpha(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta}) = \min\left[1, \frac{\pi(\boldsymbol{\theta}^{\star})T(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{\star})}{\pi(\boldsymbol{\theta})T(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta})}\right]$$

3: **begin**

3a: Start accept reject steps

3b: Draw $U \sim \text{Uniform}\big[0, 1\big]$

3c: **if** $U < \alpha(\boldsymbol{\theta}^{\star} \mid \boldsymbol{\theta})$ **then**

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{\star}$

else

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}$

3d: Stop accept reject steps

4: **begin**

4a: Start the step of swapping states

4b: Select two chains randomly, and suppose j^{th} and k^{th} chains are selected

4c: Attempt to swap the states $\boldsymbol{\theta}_j \Rightarrow \boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_k \Rightarrow \boldsymbol{\theta}_j$

4d: Calculate $\text{P}_{\text{swap}} = \dfrac{[\pi(\boldsymbol{\theta}_k)]^{1/R_j}\,[\pi(\boldsymbol{\theta}_j)]^{1/R_k}}{[\pi(\boldsymbol{\theta_j})]^{1/R_j}\,[\pi(\boldsymbol{\theta_k})]^{1/R_k}}$

4e: Draw $U \sim \text{Uniform}\big[0, 1\big]$

4f: **if** $U < P_{swap}$ **then**

accept the swapping

else

stay at previous states

4g: Stop accept reject steps

5: The chain with R=1 gives the required sample

density where small proposals (small jumps from the current point) are made in order to maintain good acceptance rate may explore the parameter space very slowly although proposals are accepted frequently (high acceptance rate).

Assessing MCMC Performance. An MCMC algorithm might explore an unimportant part of the parameter space for a long time before achieving convergence, resulting in a slow convergence to the target density. For example, from the first and second rows of Figure 1.1, it is clear that the chain has converged to the true posterior distrubution quickly under Hamiltonian Monte Carlo (HMC) based Metropolis-Hastings sampler compared to the Gibbs sampling framework. In the MCMC literature this criterion (how many iterations a Markov chain requires as a burn in period) is used to assess the performance of a Markov chain. In Chapter 5, we will use this criterion to compare the performances of different MCMC algorithms. In addition to this criterion, there are several other criteria namely mixing qualilty, acceptance rate, and randomness of the simulated values of parameter are simultaneously used to assess the performance of a Markov chain. From Figure 1.4, we have seen that although the Markov chain explores the two modes correctly but not mixing well. This situation can arise especially in multimodal case. The acceptance rate of any parameter updates under any Metropolis-Hastings step of a Markov chain should also be in a desirable range, and we will monitor this acceptance rate along with the randomness property of simulated values of parameter using the ACF plot when we compare the performances of different MCMC alogorithms in Chapter 5.

1.2 Introduction to Statistical Shape Analysis

In every day life, we are surrounded by a lot of natural and man made objects, and we often are interested to know about these objects. For example, in a medical study medical images (scans) are used to determine a disease like the

way faces are recognized by analyzing image in automated face recognition field. Statistical shape analysis is an analysis of the geometric properties of some given shape by statistical methods. Before discussing in detail about statistical shape analysis we briefly discuss some of the important terminology used in statistical shape analysis which are considered helpful to understand this area, and we have used the text book Dryden and Mardia (1998) to prepare these terminologies.

Shape. The word 'shape' usually means the appearance of an object in everyday language. However, in statistical shape analysis it has some intuitive meaning. According to Kendall (1997), the geometric information of an object which is invariant under euclidean similarity transformations of translation, rotation and scaling is called shape. Two objects are considered as same shape objects if they can be matched exactly by translating, rotating and rescaling to each other. For example, in Figure 1.5, triangles A and B have the same shape as they can be matched perfectly on each other when translation, rotation and rescaling are performed on them while the shape of triangle D is different than the shape of A and B (no transformation is suitable to match D on them or vice versa).

Size-and-shape. Sometimes scale (size) information is retained along with shape information of an object, and in this case a joint analysis of size and shape is required which is also known as form analysis in statistical shape literature. Two objects are considered have same size-and-shape if they can be matched exactly by translating and rotating (rigid body transformation) to each other. Triangles A and C in Figure 1.5 have the same size-and-shape as they can be matched perfectly on each other when translation and rotation are performed on them.

Landmarks. An object can be described by locating a finite number of points (m) on it, known as landmarks, in d dimension ($m > d$) and thus the object can be represented mathematically as $m \times d$ point configuration. All triangles in Figure 1.5 are reduced to 3-point configurations as only three landmarks are

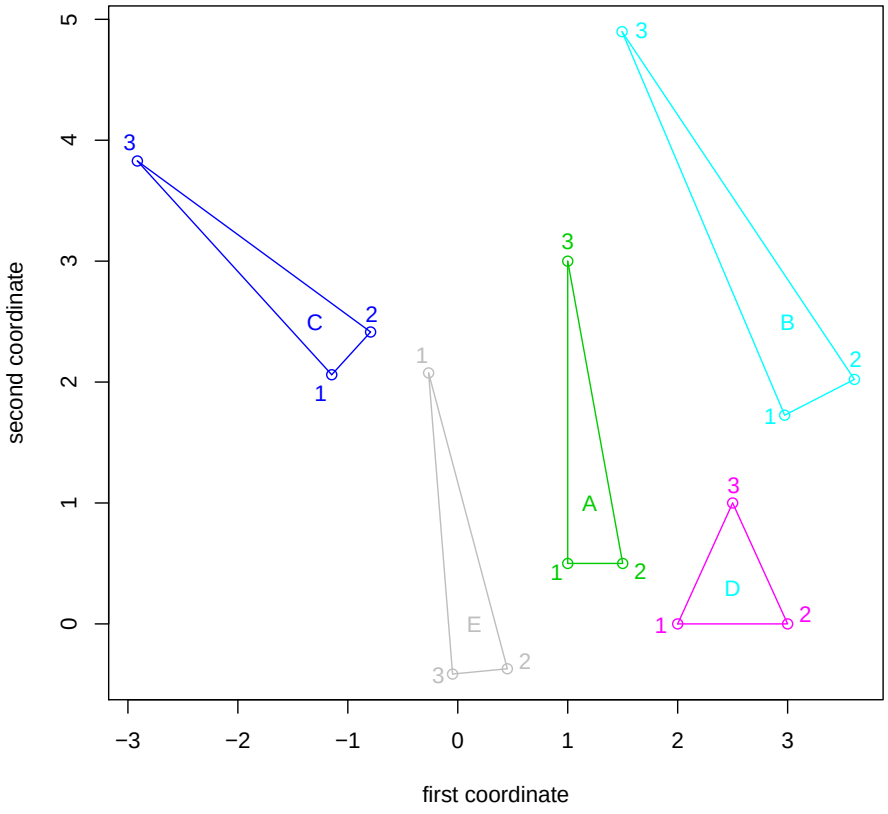

Figure 1.5: Five labelled triangles: Triangles A and C have the same size and labelled shape while B has the same labelled shape as A and C but larger size (require scaling to match). Triangle E has a different labelled shape compared to triangles A, B and C but its label can be permutated to give the same unlabelled shape. Triangle D has a completely different shape compared to triangles A, B, C and E.

used to describe their shapes respectively. Usually three basic types of landmarks are commonly seen in most of applications: scientific, mathematical and pseudo-landmarks.

A landmark on an object is called a **scientific landmark** when it is assigned by an expert on the basis of some scientific background which is also known as anatomical landmarks in biological applications. For example, the corner of an eye or the meeting of two sutures on a skull are assigned based on some biological background. We can also see the use of scientific landmarks in the protein matching problem, discussed in Chapter 5 in detail, where α-carbon (c_α) atom of an amino acid on a protein backbone is located based on the structure of chemical bond of the corresponding protein which determines the shape of that protein.

When points on an object are defined according to some mathematical or geometrical property of the figure then these points are called **mathematical landmarks**. For example, shapes of all triangles in Figure 1.5 are described by positioning landmarks on the vertices (points at which sides of an angle meet). On the other hand points constructed either around the outline or in between anatomical or mathematical landmarks of an object are called **pseudo landmarks** which is particularly useful in matching surfaces.

Label. Landmarks on an object are named using different numbers or names, called labels, with the aim of identifying which pairs of landmarks correspond when comparing two objects. Landmarks associated with labels are called labelled landmarks. For example, landmarks of all triangles in Figure 1.5 are named using labels 1, 2 and 3.

Labelled shape. In labelled shape one-to-one correspondence between the labelled landmarks on two objects is made in some meaningful way as labelling is done for both objects according to some criteria, and the correspondence is known when two objects are compared. For example, in Figure 1.5, landmark 1

of triangles A, B and C is located where a right angle is formed while landmarks 2 and 3 are located at the meeting points of adjacent and hypotenuse and opposite and hypotenuse respectively. Thus the triangles A and C are the same size and same labelled shape while triangle B has same labelled shape as A and B but larger in size with correspondences $1 \leftrightarrow 1$, $2 \leftrightarrow 2$ and $3 \leftrightarrow 3$ in both cases.

Unlabelled shape. In unlabelled shape one-to-one correspondence between the labelled landmarks on two objects is unknown as the labelling is arbitrary. For example, in Figure 1.5, the way labelling is set in triangles A, B and C the same rules are not applied to label triangles A and E. Therefore, it's not sensible to consider $1 \leftrightarrow 1$, $2 \leftrightarrow 2$ and $3 \leftrightarrow 3$ correspondences in triangles A and E like triangles A, B and C when they are compared. Unknown correspondence from all possible combinations of labelling, $1 \leftrightarrow 1$, $1 \leftrightarrow 2$ and $1 \leftrightarrow 3$, $2 \leftrightarrow 1$, $2 \leftrightarrow 2$ and $2 \leftrightarrow 3$ and $3 \leftrightarrow 1$, $3 \leftrightarrow 2$ and $3 \leftrightarrow 3$, along with translation, rotation and scaling need to be considered (estimated) to match triangles A and E. In this case triangles A and E have different label size-shape but they are considered as same unlabelled size-shape when labellings are ignored.

We will consider two practical problems of unlabelled shape from bioinformatics, gel and protein matching problems, in Chapter 5 where landmarks in each protein are labelled arbitrarily and the number of landmarks in each protein is not equal to each other in protein data. The aim is to match these proteins and gels, and to do this we need to estimate the unknown one to one correspondence along with translation and rotation (rigid body transformation). In this thesis we are interested only in the unlabelled shape problem.

1.2.1 Statistical Models for the Unlabelled Size-and-Shape

Applications of shape matching (unlabelled) are quite common in Bioinformatics and Image Analysis. In the following some important applications of unlabelled size-and-shape analysis in different fields are mentioned:

- Match proteins to determine their shape in Bioinformatics with the as-

sumptions that their biological function will be the same if they are similar shaped objects (Eid-hammaer et al, 2004; Green & Mardia, 2006; Mardia, Taylor & Westhead, 2003).

- Match images such as houses from different views in Object Recognition (Cross & Hancock, 1998; Glasbey & Mardia, 2001).

- Match location of stars in galaxies to identify resemblances and differences between their shape in Astronomy (Valdarnini, 2001).

- Match different boreholes in a mining study to identify the fractures line and map in Earth Science (Mardia et al., 2007).

To solve such matching problems, statistical models can be used along with some non-parametric methods (some distance based shape measure), and in this section we review two existing statistical models in detail proposed by Green and Mardia (2006)) and Taylor, Mardia and Kent (2003). These two models will play a building block role in deriving our proposed model and its inference technique presented in Chapter 3 and 4 respectively. A general set up for both of the models is outlined below:

In statistical shape analysis, objects are often represented by a set of points, known as landmarks, in d dimensions and thus can be represented as $m \times d$ point configurations, where m is the number of landmarks. Suppose we have one such configuration of points $\boldsymbol{X}$; the points on $\boldsymbol{X}$ are $\boldsymbol{x}_j$, $j = 1, 2, \ldots, m$, where $\boldsymbol{x}_j \in \mathbb{R}^d$ are the rows of $\boldsymbol{X}$, with each row therefore giving the coordinates of point $\boldsymbol{x}_j$. Here the aim is to align one configuration, such as $\boldsymbol{X}$ above, with another configuration, say, $\boldsymbol{Y}$, where the rows of $\boldsymbol{Y}$ are $\boldsymbol{y}_k \in \mathbb{R}^d$, $k = 1, 2, \ldots, n$, giving the coordinates of point $\boldsymbol{y}_k$ like before, under pairwise alignment problem, to compare the shapes of $\boldsymbol{X}$ and $\boldsymbol{Y}$.

Labelled shape analysis assumes a known, one-to-one correspondence between the points on $\boldsymbol{X}$ and $\boldsymbol{Y}$, labelled such that $\boldsymbol{x}_j$ matches with one point of $\boldsymbol{Y}$, say,

$\boldsymbol{y}_\alpha$, $j = 1, 2, \ldots, m$ and $\alpha \in \{1, 2, \ldots, n\}$. In labelled shape analysis the number of landmarks in both configurations is the same i.e. $m = n$. Therefore, in case of labelled shape analysis, one needs to only find the optimal transformation (translation, rotation and scaling) between $\boldsymbol{X}$ and $\boldsymbol{Y}$ to align them.

A much more challenging problem, which has been the subject of recent research interest, is that of unlabelled shape analysis, where the correspondence between landmarks is not known. For instance, any point of $\boldsymbol{X}$, say, $\boldsymbol{x}_j$, could be matched with any point of $\boldsymbol{Y}$, say, $\boldsymbol{y}_k$ and $m \neq n$ in general. A matching matrix $\boldsymbol{M}$ (m_{jk}) is introduced to denote the correspondence between landmarks, where m_{jk}=1 if $\boldsymbol{x}_j$ is matched to $\boldsymbol{y}_k$ and 0 otherwise. Thus, in the unlabelled case, one needs to simultaneously estimate the matching matrix $\boldsymbol{M}$ as well as to solve the alignment problem mentioned above for the labelled case. Multiple matches are not usually allowed, so there is at most one non zero entry in each row and in each column of $\boldsymbol{M}$ i.e. $\sum_j m_{jk} \leq 1$ for all k and $\sum_k m_{jk} \leq 1$ for all j. The total number of matched pairs, $\sum_{jk} m_{jk} = L$, is not known, and varies between $0 \leq L \leq \min(m, n)$. Thus, even for relatively small m and n, the number of possible matchings given by $\boldsymbol{M}$ is large, which makes the problem very challenging. Therefore, searching over all possible $\boldsymbol{M}$ and optimizing over transformation parameters to find a global solution is not computationally feasible.

Green and Mardia (2006) Model

For a pair of configurations of points in d dimensions, say, $\boldsymbol{X}$ and $\boldsymbol{Y}$, where $\boldsymbol{X}$ consists of m points and $\boldsymbol{Y}$ of n points which can be expressed as $m \times d$ and $n \times d$ matrices, respectively, then Green and Mardia (2006) assumed both observed configurations, $\boldsymbol{X}$ and $\boldsymbol{Y}$, are noisy realisations from a true point configuration, $\boldsymbol{\mu}$, in some space $\boldsymbol{V}$ of volume v to develop their model . Under the above settings, it is also considered that the true location $\{\mu_i\}$ is in the $\boldsymbol{X}$ - space, and configuration $\boldsymbol{Y}$ is transformed (using translation $\boldsymbol{\tau} \in \mathbb{R}^d$ and $d \times d$

matrix $\boldsymbol{A}$ which can be a rotation or any non-singular matrix) to bring into the $\boldsymbol{X}$ - space. The mapping between the observed locations $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$ with true location $\{\mu_i\}$ is indexed by labelings ψ and η respectively; $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$ are considered as a match pair when both are generated by the same point of $\boldsymbol{\mu}$, and it occurs when $\psi_j = \eta_k$. Mathematically, it can be expressed as

$$\boldsymbol{x}_j = \boldsymbol{\mu}_{\psi_j} + \boldsymbol{\epsilon}_{1j}, \tag{1.18}$$

$$\boldsymbol{A}\,\boldsymbol{y}_k + \boldsymbol{\tau} = \boldsymbol{\mu}_{\eta_k} + \boldsymbol{\epsilon}_{2k}, \tag{1.19}$$

where $\boldsymbol{\epsilon}_{1j}$ and $\boldsymbol{\epsilon}_{2j}$ represent the errors in the true points configurations and have the probability densities f_1 and f_2, respectively, for $j = 1, 2, \ldots, m$, and $k = 1, 2, \ldots, n$. All $\boldsymbol{\epsilon}_{1j}$ and $\boldsymbol{\epsilon}_{2j}$ are independent of each other, and independent of the $\boldsymbol{\mu}_i$. The unknown mapping between $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$ is one of the parameters of interest about which to draw inference, and can be represented by a $m \times n$ matching matrix $\boldsymbol{M}$ with elements $m_{jk} = I(\psi_j = \eta_k)$, where $I(.)$ is the indicator function. In this case multiple matches are excluded i.e. each point on $\boldsymbol{X}$ may be matched to at most one point on $\boldsymbol{Y}$ and vice versa. Therefore, each row and column of $\boldsymbol{M}$ may contain at most one non-zero entry. Note that when $\boldsymbol{M}$ is known, the unlabelled shape problem simply becomes the labelled shape problem.

Formulating Prior for M Using Poisson Process

Green and Mardia (2006) assumed a homogeneous Poisson process for the true point locations $\{\mu_i\}$ which occurs over a region $\boldsymbol{V} \subset \mathbb{R}^d$ of volume ν with rate λ, and considered exactly N points are realised in this region. Among these N points, assumed only L points give rise to both $\boldsymbol{X}$ and $\boldsymbol{Y}$ points, $(m-L)$ points give rise to only $\boldsymbol{X}$ but not $\boldsymbol{Y}$, $(n-L)$ points give rise to only $\boldsymbol{Y}$ but not $\boldsymbol{X}$ and $\{N-(m-L)-(n-L)-L\}$ points give rise for none of the $\boldsymbol{X}$ or $\boldsymbol{Y}$ points independently with probabilities $\rho p_x p_y$, p_x, p_y, $1-p_x-p_y-\rho p_x p_y$ respectively, where ρ is a prior probability for points to be matched here, i.e. the Poisson

process with parameter λ for the true point locations $\{\mu_i\}$ is thinned (split) into four independent Poisson processes with parameters $\lambda\,\nu\,\rho p_x p_y$, $\lambda\,\nu\,p_x$, $\lambda\,\nu\,p_y$, $\lambda\,\nu\,(1-p_x-p_y-\rho p_x p_y)$ respectively. Observed configurations $\boldsymbol{X}$ and $\boldsymbol{Y}$ become dependent because of this random thinning but they are independent from point to point. Given m and n, the prior probability distribution of L is therefore proportional to

$$\frac{\mathrm{e}^{-\lambda\,\nu\,p_x}(\lambda\,\nu\,p_x)^{m-L}}{(m-L)!}\times\frac{\mathrm{e}^{-\lambda\,\nu p_y}(\lambda\,\nu\,p_y)^{n-L}}{(n-L)!}\times\frac{\mathrm{e}^{-\lambda\,\nu\,\rho p_x\,p_y}(\lambda\,\nu\,\rho p_x\,p_y)^{L}}{(L)!}$$

so that

$$\pi(L)\propto\frac{(\rho/\lambda\,\nu)^L}{(m-L)!\,(n-L)!\,L!}, \tag{1.20}$$

for $L=0,1,2,\ldots\ \min(m,n)$. As multiple matches are ruled out so there is at most one 1 in each row and in each column of $\boldsymbol{M}$ i.e. $\sum_j m_{jk}\leq 1$ for all k and $\sum_k m_{jk}\leq 1$ for all j. It is also noted that total number of matches is L i.e. $\sum_{jk} m_{jk}=L$. Since the number of ways of selecting L from m and n are $\binom{m}{L}$ and $\binom{n}{L}$ respectively, and there are $L!$ different ways to arrange L, for a given L there are $L!\binom{m}{L}\binom{n}{L}$ different $\boldsymbol{M}$ matrices which are equally likely i.e. $\boldsymbol{M}$ is a priori uniform distribution with the following form:

$$\pi(\boldsymbol{M}\,|L)=\frac{1}{L!\binom{m}{L}\binom{n}{L}} \tag{1.21}$$

The joint density of $\boldsymbol{M}$ and L, denoted as $\pi(\boldsymbol{M})$, can be obtained by combining the equations (1.20) and (1.21) which is written as follows:

$$\begin{aligned}\pi(\boldsymbol{M})&=\pi(L)\,\pi(\boldsymbol{M}\,|L)\\&\propto\frac{(\rho/\lambda\nu)^L}{(m-L)!(n-L)!L!}\left\{L!\binom{m}{L}\binom{n}{L}\right\}^{-1}\\&\propto(\rho/\lambda\nu)^L\end{aligned} \tag{1.22}$$

Likelihood of the data

From (1.18) and (1.19), the densities of $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, conditional on $\boldsymbol{A}$, $\boldsymbol{\tau}$, $\boldsymbol{\mu}_i$, ψ_j and η_k, are $f_1(\boldsymbol{x}_j - \boldsymbol{\mu}_{\psi_j})$ and $|\boldsymbol{A}| f_2(\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu}_{\eta_k})$, where $|\boldsymbol{A}|$ is the determinant of matrix $\boldsymbol{A}$. The edge effects can be ignored by assuming that v is large enough relative to the support of f_1 and f_2. As hidden point locations $\{\mu_i\}$ are integrated out, considering $\{\mu_i\}$ without subscripts in the following calculations is the same as considering $\{\mu_i\}$ with subscripts. In the following calculations, we ignore the subscripts of $\{\mu_i\}$. Then the likelihood contribution of the $(m - L)$ unmatched $\boldsymbol{X}$ points is, say l_1

$$l_1 = \prod_{j:m_{jk=0\,\forall k}} \nu^{-1} \int_{\boldsymbol{V}} f_1(\boldsymbol{x}_j - \boldsymbol{\mu})\, \mathrm{d}\boldsymbol{\mu} \approx \nu^{-(m-L)}.$$

Similarly, the contribution of the $n-L$ unmatched $\boldsymbol{Y}$ and L matched points are, say l_2 and l_3 respectively

$$l_2 = \prod_{k:m_{jk=0\,\forall j}} \nu^{-1} \int_{\boldsymbol{V}} |\boldsymbol{A}|\, f_2(\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu})\, \mathrm{d}\boldsymbol{\mu} \approx \nu^{-(n-L)} |\boldsymbol{A}|^{(n-L)},$$

and

$$\begin{aligned}
l_3 &= \prod_{j,k:m_{jk=1}} \nu^{-1} \int_{\boldsymbol{V}} f_1(\boldsymbol{x}_j - \boldsymbol{\mu})\, |\boldsymbol{A}|\, f_2(\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu})\, \mathrm{d}\boldsymbol{\mu} \\
&= \nu^{-L} |\boldsymbol{A}|^L \prod_{j,k:m_{jk=1}} \int_{\boldsymbol{V}} f_1(\boldsymbol{x}_j - \boldsymbol{\mu})\; f_2(\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu})\, \mathrm{d}\boldsymbol{\mu} \\
&= \nu^{-L} |\boldsymbol{A}|^L \prod_{j,k:m_{jk=1}} \int_{\boldsymbol{V}} f_1\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau}) + (\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu})\}\; f_2(\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu})\, \mathrm{d}\boldsymbol{\mu} \\
&\approx \nu^{-L} |\boldsymbol{A}|^L\, E_1,
\end{aligned}$$

where $E_1 = \prod_{j,k:m_{jk=1}} \int_{\boldsymbol{V}} f_1\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau}) + (\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu})\}\; f_2(\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu})\, \mathrm{d}\boldsymbol{\mu}$ can be simplified as

$$
\begin{aligned}
E_1 &= \prod_{j,k:m_{jk}=1} \int_{\boldsymbol{V}} f_1\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau}) + (\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu})\}\ f_2(\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu})\, \mathrm{d}\boldsymbol{\mu} \\
&= \int_V f_1(\boldsymbol{z} + \boldsymbol{u})\, f_2(\boldsymbol{u})\, \mathrm{d}\mu \quad \Big[\boldsymbol{z} = (\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau}) \,\text{and}\, \boldsymbol{u} = (\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau} - \boldsymbol{\mu})\Big] \\
&= g(\boldsymbol{z}) \quad \Big[\text{Using convolution formula}\Big] \\
&= g(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau}) \\
&= g(\boldsymbol{\epsilon}_{1j} - \boldsymbol{\epsilon}_{2k}).
\end{aligned}
$$

Assuming $f_1 \sim N_d(\mathbf{0}, \sigma^2 I_d)$ and $f_2 \sim N_d(\mathbf{0}, \sigma^2 I_d)$ lead to $\boldsymbol{x}_j \sim N_d(\boldsymbol{\mu}_{\psi_j}, \sigma^2 I_d)$ and $\boldsymbol{A}\,\boldsymbol{y}_k + \boldsymbol{\tau} \sim N_d(\boldsymbol{\mu}_{\eta_k}, \sigma^2 I_d)$. After doing some algebraic simplification the density of $(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})$, $g(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})$, can be written as

$$
g(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau}) = \frac{1}{(\sigma\sqrt{2})^d}\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\},
$$

where ϕ denotes the standard normal density in $\mathbb{R}^d$. Now combining l_1, l_2 and l_3 the complete data likelihood, $\pi(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma)$, is

$$
\pi(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma) = \nu^{-(m+n)+L}\, |\boldsymbol{A}|^n \prod_{j,k:m_{jk}=1} \frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d} \tag{1.23}
$$

1.2.2 Prior distributions for Unknown Parameters

We have already seen $\pi(\boldsymbol{M}) \propto \left(\frac{\rho}{\lambda\nu}\right)^L$, prior probability of $\boldsymbol{M}$, under the settings in which true $\boldsymbol{\mu}$ form a homogeneous Poisson process with rate λ over the region $\boldsymbol{V}$ of volume v. Here larger values of ρ/λ give a stronger prior preference to larger numbers of matched points. Substituting the prior distributions $\pi(\boldsymbol{M})$,

$\pi(\boldsymbol{A})$, $\pi(\boldsymbol{\tau})$ and $\pi(\sigma)$ for unknown parameters $\boldsymbol{M}$, $\boldsymbol{A}$, $\boldsymbol{\tau}$ and σ in equation (1.23), Green and Mardia (2006) derived the following posterior distribution of unknown parameters $\boldsymbol{M}$, $\boldsymbol{A}$, $\boldsymbol{\tau}$ and σ

$$\begin{aligned}
\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma; \boldsymbol{X}, \boldsymbol{Y}) &\propto \nu^{-(m+n)+L} \, |\boldsymbol{A}|^n \, \pi(\boldsymbol{A}) \, \pi(\boldsymbol{\tau}) \, \pi(\sigma) \prod_{j,k:m_{jk}=1} \frac{\rho \, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda \, \nu(\sigma\sqrt{2})^d} \\
&= \nu^{-(m+n)} \, |\boldsymbol{A}|^n \, \pi(\boldsymbol{A}) \, \pi(\boldsymbol{\tau}) \, \pi(\sigma) \prod_{j,k:m_{jk}=1} \frac{\rho \, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \\
&\propto |\boldsymbol{A}|^n \, \pi(\boldsymbol{A}) \, \pi(\boldsymbol{\tau}) \, \pi(\sigma) \prod_{j,k:m_{jk}=1} \frac{\rho \, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma)^d}
\end{aligned} \tag{1.24}$$

Green and Mardia (2006) considered a normal prior for the translation parameter vector $\boldsymbol{\tau}$ i.e. $\boldsymbol{\tau} \sim N_d(\boldsymbol{\mu}_{\boldsymbol{\tau}}, \sigma^2_{\boldsymbol{\tau}} \, I_d)$, where $\boldsymbol{\mu}_{\tau}$ is a mean vector and $\sigma^2_{\boldsymbol{\tau}} \, I_d$ a covariance matrix, with I_d the $d \times d$ identity matrix while the noise parameter σ, they assumed $\sigma^{-2} \sim \Gamma(\alpha, \beta)$, where $\pi(\sigma^{-2}) \propto \sigma^{-2(\alpha-1)} \exp(-\frac{\beta}{\sigma^2})$.

The matrix Fisher distribution was chosen as a prior distribution for rotation parameter matrix $\boldsymbol{A}$, where $\pi(\boldsymbol{A}) \propto \exp\left\{\text{tr}(\boldsymbol{F}_0{}^T \boldsymbol{A})\right\}$ and the parameter $\boldsymbol{F}_0$ is a $d \times d$ matrix. Choosing $\boldsymbol{F}_0$ as the $d \times d$ matrix of zeroes means giving a uniform prior on $\boldsymbol{A}$. Their implementation was only limited to a rigid body transformation (see subsection 1.2.4 for detail) although their settings can be extended for similarity transformation as well which was shown by Mardia et. al. (2013).

1.2.3 MCMC Updates for all Unknown Variables

Green and Mardia (2006) implemented a Markov chain Monte Carlo sampler to draw samples from the posterior distribution defined in (1.26). They used Gibbs updates for $\boldsymbol{\tau}$ and σ^2, Metropolis-Hastings updates for matching matrix $\boldsymbol{M}$, and Metropolis sampler for the matrix Fisher distribution for updating $\boldsymbol{A}$ in a 3-D

case. To make inference, firstly they derived the full conditional distributions for variables $\boldsymbol{\tau}$, σ^2 and $\boldsymbol{A}$ from (1.24) and then describe a sampling algorithm for each full conditional distribution.

The full conditional distribution of $\boldsymbol{\tau}$ given $\boldsymbol{M}$, $\boldsymbol{A}$ and σ^2, denoted as $\pi(\boldsymbol{\tau} \mid \boldsymbol{M}, \boldsymbol{A}, \sigma, \boldsymbol{X}, \boldsymbol{Y})$, is proportional to

$$\begin{aligned}\pi(\boldsymbol{\tau} \mid \boldsymbol{M}, \boldsymbol{A}, \sigma, \boldsymbol{X}, \boldsymbol{Y}) &\propto \pi(\boldsymbol{\tau}) \prod_{j,k:m_{jk}=1} \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\} \\ &= \pi(\boldsymbol{\tau}) \times E_2, \end{aligned} \tag{1.25}$$

where $E_2 = \prod_{j,k:m_{jk}=1} \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}$. Simplifying E_2 we have

$$\begin{aligned} E_2 &= \prod_{j,k:m_{jk}=1} \exp\Big\{ -\frac{1}{4\sigma^2} \times \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}} \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\} \Big\} \\ &= \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \Big\{ \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}} \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\} \Big\} \Big] \\ &\propto \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \Big\{ -{\boldsymbol{x}_j}^{\mathrm{T}}\boldsymbol{\tau} + (\boldsymbol{A}\boldsymbol{y}_k)^{\mathrm{T}}\boldsymbol{\tau} - \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{x}_j + \boldsymbol{\tau}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{y}_k) + \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\tau} \Big\} \Big]. \end{aligned}$$

Similarly simplifying $\pi(\boldsymbol{\tau})$ gives

$$\begin{aligned} \pi(\boldsymbol{\tau}) &\propto \exp\Big\{ -\frac{1}{2} (\boldsymbol{\tau} - \boldsymbol{\mu}_{\boldsymbol{\tau}})^{\mathrm{T}} (\sigma_{\boldsymbol{\tau}}^2 I)^{-1} (\boldsymbol{\tau} - \boldsymbol{\mu}_{\boldsymbol{\tau}}) \Big\} \\ &= \exp\Big\{ -\frac{1}{2\,\sigma_{\boldsymbol{\tau}}^2} (\boldsymbol{\tau} - \boldsymbol{\mu}_{\boldsymbol{\tau}})^{\mathrm{T}} (\boldsymbol{\tau} - \boldsymbol{\mu}_{\boldsymbol{\tau}}) \Big\} \\ &\propto \exp\Big\{ -\frac{1}{2\,\sigma_{\boldsymbol{\tau}}^2} (\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\tau} - \boldsymbol{\tau}^{\mathrm{T}}\mu_{\boldsymbol{\tau}} - \mu_{\boldsymbol{\tau}}^{T}\boldsymbol{\tau}) \Big\}. \end{aligned}$$

Plugging the $\pi(\boldsymbol{\tau})$ and E_2 values in equation (1.27) yields the full conditional

distribution of τ, $\pi(\boldsymbol{\tau} \mid \boldsymbol{M}, \boldsymbol{A}, \sigma, \boldsymbol{X}, \boldsymbol{Y})$, which is proportional to

$$\pi(\boldsymbol{\tau} \mid \boldsymbol{M}, \boldsymbol{A}, \sigma, \boldsymbol{X}, \boldsymbol{Y}) \propto \exp\Big\{ -\frac{1}{2}\left(e_1\, \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\tau} - e_2\, \boldsymbol{\tau}^{\mathrm{T}} - e_3\, \boldsymbol{\tau}\right)\Big\},$$

where e_3 is the transpose of e_2 and all three have the following expressions respectively:

$$e_1 = \left(1/\sigma_{\boldsymbol{\tau}}^2 + L/2\,\sigma^2\right)$$

$$e_2 = \Big(\boldsymbol{\mu}_{\boldsymbol{\tau}}/\sigma_{\boldsymbol{\tau}}^2 + \sum_{j,k:m_{jk}=1} (\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k)/2\,\sigma^2\Big)$$

$$e_3 = \Big(\boldsymbol{\mu}_{\boldsymbol{\tau}}^{\mathrm{T}}/\sigma_{\boldsymbol{\tau}}^2 + \sum_{j,k:m_{jk}=1} (\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k)^{\mathrm{T}}/2\,\sigma^2\Big).$$

Simplifying further gives

$$\begin{aligned}\pi(\boldsymbol{\tau} \mid \boldsymbol{M}, \boldsymbol{A}, \sigma, \boldsymbol{X}, \boldsymbol{Y}) &\propto \exp\Big\{ -\frac{e_1}{2}\left(\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\tau} - \frac{e_2}{e_1}\,\boldsymbol{\tau}^{\mathrm{T}} - \frac{e_3}{e_1}\,\boldsymbol{\tau}\right)\Big\} \\ &\propto N_d\Big(\boldsymbol{\mu}^{*},\, {\sigma^2}^{*} I\Big),\end{aligned}$$

where

$$\boldsymbol{\mu}^{*} = \frac{e_2}{e_1} = \frac{\Big(\boldsymbol{\mu}_{\boldsymbol{\tau}}/\sigma_{\boldsymbol{\tau}}^2 + \sum_{j,k:m_{jk}=1} (\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k)/2\,\sigma^2\Big)}{\left(1/\sigma_{\boldsymbol{\tau}}^2 + \sum_{j,k:m_{jk}=1} 1/2\,\sigma^2\right)}$$

$${\sigma^2}^{*} = \frac{1}{e_1} = \frac{1}{\left(1/\sigma_{\boldsymbol{\tau}}^2 + \sum_{j,k:m_{jk}=1} 1/2\,\sigma^2\right)}\,.$$

The full conditional distribution of $\boldsymbol{\tau}$ follows a d dimensional Multivariate normal distribution with mean vector $\boldsymbol{\mu}^*$ and variance-covariance matrix $\sigma^{2*}I$. Green and Mardia (2006) draw samples for $\boldsymbol{\tau}$ from its d dimensional Multivariate normal distribution directly using Gibbs updates.

Similarly the full conditional distribution of σ^{-2} given $\boldsymbol{M}$, $\boldsymbol{\tau}$ and $\boldsymbol{A}$, denoted as $\pi(\sigma^{-2} \mid \boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \boldsymbol{X}, \boldsymbol{Y})$, is proportional to

$$\pi(\sigma^{-2} \mid \boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \boldsymbol{X}, \boldsymbol{Y}) \propto \pi(\sigma^{-2}) \prod_{j,k:m_{jk}=1} \left\{ \frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d} \right\}$$

$$= (\sigma^{-2})^{(\alpha-1)} \exp\{-\beta/\sigma^2\} \times E_3, \tag{1.26}$$

where $E_3 = \prod_{j,k:m_{jk}=1} \left\{ \frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d} \right\}$, simplifying E_3 further gives

$$E_3 \propto \exp\left\{ - \frac{\sum_{j,k:m_{jk}=1} \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}} \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}}{4\sigma^2} \right\} \left(\sigma^{-2}\right)^{\left(d\,L\right)/2}$$

$$= \left(\sigma^{-2}\right)^{\left(d\,L\right)/2} \exp\left\{ - \frac{\sum_{j,k:m_{jk}=1} \|(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\|^2}{4\sigma^2} \right\}.$$

Plugging the E_3 value in the equation (1.26) we have the full conditional distribution of σ^{-2}, which is proportional to

$$\pi(\sigma^{-2} \mid \boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \boldsymbol{X}, \boldsymbol{Y}) \propto \left(\sigma^{-2}\right)^{\left[\alpha+(dL)/2-1\right]}$$

$$\times \exp\left[-\left\{ \beta + \frac{\sum_{j,k:m_{jk}=1} \|(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\|^2}{4} \right\} \sigma^{-2} \right]$$

$$\propto \Gamma(\alpha^*, \beta^*),$$

where

$$\alpha^* = \alpha + \left(dL\right)/2$$

$$\beta^* = \left\{ \beta + \frac{\sum_{j,k:m_{jk}=1} \|(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\|^2}{4} \right\}.$$

The full conditional distribution of σ^{-2} also follows a gamma distribution, with shape parameter α^* and rate parameter β^*. Like $\boldsymbol{\tau}$, Green and Mardia (2006) draw samples for σ^{-2} from its full conditional distribution directly in Gibbs style fashion.

Green and Mardia (2006) considered $\boldsymbol{A}$ as an unknown parameter in their application, and application was limited to 2 and 3 dimension cases. They also considered $\boldsymbol{A}$ a rotation matrix in 2 and 3 dimension cases i.e. $|\boldsymbol{A}|$ =1. The full conditional distribution of $\boldsymbol{A}$ given $\boldsymbol{M}, \sigma^{-2}$ and $\boldsymbol{\tau}$, denoted as $\pi(\boldsymbol{A} \mid \boldsymbol{M}, \boldsymbol{\tau}, \sigma^{-2}, \boldsymbol{X}, \boldsymbol{Y})$, is proportional to

$$\pi(\boldsymbol{A} \mid \boldsymbol{M}, \boldsymbol{\tau}, \sigma^{-2}, \boldsymbol{X}, \boldsymbol{Y}) \propto \pi(\boldsymbol{A}) \prod_{j,k:m_{jk}=1} \left\{ \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\} \right\}$$

$$= \exp\left\{\mathrm{tr}(\boldsymbol{F}_0^{\mathrm{T}} \boldsymbol{A})\right\} \times E_4, \tag{1.27}$$

where $E_4 = \prod_{j,k:m_{jk}=1} \Big\{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}\Big\}$, and simplifying E_4 further gives

$$
\begin{aligned}
E_4 &= \prod_{j,k:m_{jk}=1} \exp\Big\{ -\frac{1}{4\sigma^2} \times \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}\Big\} \\
&= \prod_{j,k:m_{jk}=1} \exp\Big\{ -\frac{1}{4\sigma^2} \times \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}\Big\} \\
&= \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \mathrm{tr}\Big\{\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \tau)\}\Big\}\Big] \\
&\qquad\qquad \big[\text{as } \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\} \text{ is a scalar}\big] \\
&= \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \mathrm{tr}\Big\{\{(\boldsymbol{x}_j - \boldsymbol{\tau}) - \boldsymbol{A}\boldsymbol{y}_k\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{\tau}) - \boldsymbol{A}\boldsymbol{y}_k\}\Big\}\Big] \\
&\propto \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \mathrm{tr}\big\{ -(\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{y}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{x}_j - \boldsymbol{\tau}) + \boldsymbol{y}_k^{\mathrm{T}}\boldsymbol{y}_k\big\}\Big] \\
&\qquad\qquad \big[\text{as } \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = I \text{ and } (\boldsymbol{A}\,\boldsymbol{y}_k)^{\mathrm{T}} = \boldsymbol{y}_k^{\mathrm{T}}\,\boldsymbol{A}^{\mathrm{T}}\big] \\
&\propto \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \mathrm{tr}\big\{ -(\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{y}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{x}_j - \boldsymbol{\tau})\big\}\Big] \\
&= \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \big\{ -\mathrm{tr}\{(\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\,\boldsymbol{y}_k\} - \mathrm{tr}\{\boldsymbol{y}_k^{\mathrm{T}}\,\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{x}_j - \boldsymbol{\tau})\}\big\}\Big] \\
&= \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \big\{ -\mathrm{tr}\{\boldsymbol{y}_k\,(\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\} - \mathrm{tr}\{(\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\,\boldsymbol{y}_k\}\big\}\Big] \\
&\big[\text{as } \mathrm{tr}(\boldsymbol{A}+\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{A}) + \mathrm{tr}(\boldsymbol{B}),\ \mathrm{tr}(\boldsymbol{A}\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{B}\boldsymbol{A}) \text{ and } \mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}) = \mathrm{tr}(\boldsymbol{A})\big] \\
&= \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \big\{ -\mathrm{tr}\{\boldsymbol{y}_k\,(\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\} - \mathrm{tr}\{\boldsymbol{y}_k\,(\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\}\big\}\Big]
\end{aligned}
$$

$$E_4 = \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \big\{ -\operatorname{tr}\{\boldsymbol{y}_k (\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}} \boldsymbol{A}\} - \operatorname{tr}\{\boldsymbol{y}_k (\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}} \boldsymbol{A}\}\big\}\Big]$$

$$= \exp\Big[-\frac{1}{4\sigma^2} \sum_{j,k:m_{jk}=1} \big\{ -\operatorname{tr}\{2\, \boldsymbol{y}_k (\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}} \boldsymbol{A}\}\big\}\Big]$$

$$= \exp\Big[\operatorname{tr}\Big\{ \frac{1}{2\sigma^2} \sum_{j,k:m_{jk}=1} \boldsymbol{y}_k (\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}} \boldsymbol{A}\Big\}\Big]$$

Plugging the E_4 value in equation (1.27) we have the full conditional distribution of $\boldsymbol{A}$, which is proportional to

$$\pi(\boldsymbol{A} \mid \boldsymbol{M}, \boldsymbol{\tau}, \sigma\, \boldsymbol{X}, \boldsymbol{Y}) \propto \exp\{\operatorname{tr}(\boldsymbol{F}_0^{\mathrm{T}} \boldsymbol{A})\} \exp\Big[\operatorname{tr}\Big\{ \frac{1}{2\sigma^2} \sum_{j,k:m_{jk}=1} \boldsymbol{y}_k (\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}} \boldsymbol{A}\Big\}\Big]$$

$$= \exp\Big[\operatorname{tr}(\boldsymbol{F}_0^{\mathrm{T}} \boldsymbol{A}) + \operatorname{tr}\Big\{ \frac{1}{2\sigma^2} \sum_{j,k:m_{jk}=1} \boldsymbol{y}_k (\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}} \boldsymbol{A}\Big\}\Big]$$

$$= \exp\Big[\operatorname{tr}\Big\{ \boldsymbol{F}_0^{\mathrm{T}} + \frac{1}{2\sigma^2} \sum_{j,k:m_{jk}=1} \boldsymbol{y}_k (\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}}\Big\} \boldsymbol{A}\Big]$$

$$= \exp\Big[\operatorname{tr}\Big\{ \boldsymbol{F}^{\mathrm{T}} \boldsymbol{A}\Big\}\Big],$$

where

$$\boldsymbol{F} = \boldsymbol{F}_0 + \frac{1}{2\sigma^2} \sum_{j,k:m_{jk}=1} (\boldsymbol{x}_j - \boldsymbol{\tau})\, \boldsymbol{y}_k^{\mathrm{T}}. \tag{1.28}$$

The conditional distribution of $\boldsymbol{A}$ also follows a matrix Fisher distribution, with matrix parameter $\boldsymbol{F}$. For 2 and 3 dimensional cases, Green and Mardia (2006) used the following algorithms to update $\boldsymbol{A}$:

Updating A in 2-d Case. In 2 dimension case they parametrized the rotation matrix $\boldsymbol{A}$ by an angle θ as

$$\boldsymbol{A} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

and the *von Mises* distribution with respect to the Lebesgue measure on $(0, 2\pi)$ was considered for θ. In general , the *von Mises* distribution for θ, with concentration parameter κ and location parameter v, has the following form

$$\begin{aligned} \pi(\theta) &\propto \exp\{\kappa \cos(\theta - v)\} \\ &= \exp\{\kappa \cos v \cos \theta + \kappa \sin v \sin \theta\}. \end{aligned} \tag{1.29}$$

Indeed the form of $\pi(\theta)$ defined in equation (1.31) and form of matrix Fisher distribution defined in section 1.2.2, $\pi(\boldsymbol{A}) \propto \exp\{\mathrm{tr}(\boldsymbol{F}_0^{\mathrm{T}}\boldsymbol{A})\}$, are the same functional form when $\boldsymbol{F}_0$ has the following form

$$\boldsymbol{F}_0 = \kappa/2 \begin{pmatrix} \cos\nu & -\sin\nu \\ \sin\nu & \cos\nu \end{pmatrix}.$$

Thus, assuming *von Mises* as a prior distribution for θ, which covers uniform case when concentration parameter is zero, the full conditional distribution of θ also has the same *von Mises* form with updated parameters, say, κ^* and v^* i.e.

$$\begin{aligned} \pi(\theta) &\propto \exp\{\kappa^* \cos(\theta - v^*)\} \\ &= \exp\{\kappa^* \cos v^* \cos \theta + \kappa^* \sin v^* \sin \theta\}. \end{aligned} \tag{1.30}$$

The updated parameters κ^* and v^* can be determined by equating *von Mises* distribution with updated parameters defined in (1.30) and matrix Fisher distribution with updated parameter $\boldsymbol{F}$ (updated according to equation 1.28) which

has the following form

$$\pi(\boldsymbol{A}) \propto \exp\{(\kappa\, cos\, v + S_{11} + S_{22})\, cos\, \theta + (\kappa\, sin\, v - S_{12} + S_{21})\, \sin\, \theta\}, \quad (1.31)$$

where $S_{i\,j}$, i, j={1, 2}, are the element of 2×2 matrix $\boldsymbol{S} = 1/(2\sigma^2) \sum_{j,k:M_{jk}=1} (\boldsymbol{x}_j - \boldsymbol{\tau})\, \boldsymbol{y}_k^T$. Equating equations (1.32) and (1.33) gives

$$\kappa^* = \sqrt{a^2 + b^2} \quad (1.32)$$

$$v^* = \tan^{-1}(b/a), \quad (1.33)$$

where $a = (\kappa\, cos\, v + S_{11} + S_{22})$ and $b = (\kappa\, sin\, v - S_{12} + S_{21})$. An efficient rejection method developed by Best, D. J. and Fisher, N. I. (1997), was used to sample from the full conditional distribution of θ.

Updating A in 3-d Case. In 3 dimension case they parametrized the rotation matrix in terms of 3 Eulerian angles, namely, θ_{12}, θ_{13}, θ_{23} according to Raffenetti and Ruedenberg (1970) and Khatri and Mardia (1977) as

$$\boldsymbol{A} = \boldsymbol{A}_{12}(\theta_{12})\, \boldsymbol{A}_{13}(\theta_{13})\, \boldsymbol{A}_{23}(\theta_{23}), \quad (1.34)$$

where for $i < j$, i, j ={1, 2, 3}, $\boldsymbol{A}_{ij}(\theta_{ij})$ is the matrix with elements $m_{ii} = m_{jj} = \cos\theta_{ij}$, -$m_{ij} = m_{ji} = \sin\theta_{ij}$, $m_{rr} = 1$ for $r \neq i, j$ and other entries are equal to 0. For example, for $\boldsymbol{A}_{12}(\theta_{12})$ i= 1 and j = 2, so its all elements are; $m_{11} = m_{22} = \cos\theta_{12}$, -$m_{12} = m_{21} = \sin\theta_{12}$, $m_{33} = 1$ and $m_{13} = m_{23} = m_{31} = m_{32} = 0$. After substituting all $\boldsymbol{A}_{ij}(\theta_{ij})$ values in equation (1.34) the form of rotation $\boldsymbol{A}$ becomes

$$\boldsymbol{A} = \begin{pmatrix} \cos\,\theta_{12} & -\sin\,\theta_{12} & 0 \\ \sin\,\theta_{12} & \cos\,\theta_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\,\theta_{13} & 0 & -\sin\,\theta_{13} \\ 0 & 1 & 0 \\ \sin\,\theta_{13} & 0 & \cos\,\theta_{13} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\,\theta_{23} & -\sin\,\theta_{23} \\ 0 & \sin\,\theta_{23} & \cos\,\theta_{23} \end{pmatrix}.$$

The rotation $\boldsymbol{A}$ is treated as uniformly distributed on $\mathbb{SO}(3)$ as we don't have any specific prior information about $\boldsymbol{A}$. The uniform distribution on $\mathbb{SO}(3)$ is determined by defining a probability measure on it which is invariant under any group transformation. Under the above setting, the form of the uniform distribution on $\mathbb{SO}(3)$ is equal to 1 with respect to $\frac{1}{8\pi^2}\cos\theta_{13}\,\mathrm{d}\theta_{12}\,\mathrm{d}\theta_{13}\,\mathrm{d}\theta_{23}$ (see Khatri and Mardia (1977)). As $\boldsymbol{A}$ is parametrized in terms of Eulerian angles, the form of uniform distribution with respect to Lebesgue measure (with respect to $\mathrm{d}\theta_{12}\,\mathrm{d}\theta_{13}\,\mathrm{d}\theta_{23}$) is $\frac{1}{8\pi^2}\cos\theta_{13}$. Therefore, the joint full conditional density of the Euler angles is proportional to

$$\exp\{\mathrm{tr}(\boldsymbol{F}^T\boldsymbol{A})\}\ \cos\ \theta_{13}, \tag{1.35}$$

where $\theta_{12},\ \theta_{23} \in (-\pi, \pi)$ and $\theta_{13} \in (-\pi/2, \pi/2)$. After substituting the value of $\boldsymbol{A}$ in equation (1.37) the term $\mathrm{tr}(\boldsymbol{F}^T\boldsymbol{A})$ can be written variously as

$$\begin{aligned}
\mathrm{tr}(\boldsymbol{F}^T\boldsymbol{A}) &= a_{12}\cos\theta_{12} + b_{12}\sin\theta_{12} + c_{12} \\
&= a_{13}\cos\theta_{13} + b_{13}\sin\theta_{13} + c_{13} \\
&= a_{23}\cos\theta_{23} + b_{23}\sin\theta_{23} + c_{23},
\end{aligned}$$

where

$$\begin{aligned}
a_{12} &= \big(F_{22} - F_{13}\sin\theta_{13}\big)\cos\theta_{23} + \big(-F_{23} - F_{12}\sin\theta_{13}\big)\sin\theta_{23} + F_{11}\cos\theta_{13} \\
b_{12} &= \big(-F_{12} - F_{23}\sin\theta_{13}\big)\cos\theta_{23} + \big(F_{13} - F_{22}\sin\theta_{13}\big)\sin\theta_{23} + F_{21}\cos\theta_{13} \\
a_{13} &= F_{11}\cos\theta_{12} + F_{21}\sin\theta_{12} + F_{32}\sin\theta_{23} + F_{33}\cos\theta_{23} \\
b_{13} &= F_{31} - \big(F_{12}\sin\theta_{23} + F_{13}\cos\theta_{23}\big)\cos\theta_{12} - \big(F_{22}\sin\theta_{23} + F_{23}\cos\theta_{23}\big)\sin\theta_{23} \\
a_{23} &= \big(F_{22} - F_{13}\sin\theta_{13}\big)\cos\theta_{12} + \big(-F_{12} - F_{23}\sin\theta_{13}\big)\sin\theta_{12} + F_{33}\cos\theta_{13} \\
b_{23} &= \big(-F_{23} - F_{12}\sin\theta_{13}\big)\cos\theta_{12} + \big(F_{13} - F_{22}\sin\theta_{13}\big)\sin\theta_{12} + F_{32}\cos\theta_{13},
\end{aligned}$$

and the quantities c_{12}, c_{13} and c_{23} are all constants and can be combined into the normalising constants. Now each of the Euler angles can be updated in turn by conditioning on other two angles and other variables from equation (1.37). Thus, the full conditionals for θ_{12} and θ_{23} have the following *von Mises* form

$$\pi(\theta_{12}|\theta_{23},\theta_{13}) \propto \exp\ \big(a_{12}\, cos\, \theta_{12} + b_{12}\ \sin\ \theta_{12}\big) \tag{1.36}$$

$$\pi(\theta_{23}|\theta_{12},\theta_{13}) \propto \exp\ \big(a_{23}\, cos\, \theta_{23} + b_{12}\ \sin\ \theta_{23}\big), \tag{1.37}$$

and Gibbs update can be made for θ_{12} and θ_{23} using Best and Fisher algorithm. On the other hand, the full conditional distribution for θ_{13} is proportional to

$$\pi(\theta_{13}|\theta_{12},\theta_{23}) \propto \exp\ \big(a_{13}\, cos\, \theta_{13} + b_{13}\ \sin\ \theta_{13}\big)\, cos\, \theta_{13}. \tag{1.38}$$

Green and Mardia (2006) used a random walk Metropolis update for this variable, with a perturbation uniformly distributed on [-a, a], where $a \in \mathbb{R}$ and they chose a = 0.1.

Apart from Green and Mardia (2006) method for updating rotation A in 3-d case, we will use three existing methods for updating $\boldsymbol{A}$ in 3-d case proposed by Kent and Ganeiber (2012), Habeck's (2009) and Byrne and Girolami (2013) respectively in this thesis in the results and discussion chapter.

Updating Matching Matrix $\boldsymbol{M}$ To update the matching matrix $\boldsymbol{M}$, Green and Mardia (2006) used Metropolis-Hastings moves that only propose changes to a few entries. Under their setting the number of current matches can only increase or decrease by 1 or stay the same. The detail algorithm for updating $\boldsymbol{M}$ according to them is defined as follows:

(1) Select one point randomly from all $(m+n)$ data points $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n$. Suppose without loss of generality selected point from $\mathbf{X}$ and, say, $\boldsymbol{x}_j, j \in \{1, 2, \ldots, m\}$ is selected. The following steps are analogous if a point came from $\mathbf{Y}$ because of the symmetry of the model.

(2) Check either selected $\boldsymbol{x}_j$ in step 1 is matched or unmatched. Suppose this point is matched with $\boldsymbol{y}_k, k \in \{1, 2, \ldots, n\}$. Now with probability p^* they proposed deleting this match and with probability $(1 - p^*)$ proposed switching this match with one randomly selected unmatched $\boldsymbol{y}_{k'}$ if there is any. After proposing a move (i.e. deletion or switching) determine their acceptability on the basis of Metropolis Hastings acceptance probability. Accepting deletion decreases L (current match points) by one while L remains the same under switching.

(3) If $\boldsymbol{x}_j$ is unmatched in step (2) then they proposed adding this $\boldsymbol{x}_j$ with one randomly selected unmatched $\boldsymbol{y}_k$ if there is any. Based on Metropolis Hastings acceptance probability this proposal is accepted or not. The number of current matches L is increased by 1 if move is accepted.

Using equation (1.24) the acceptance probabilities for the above three moves can be derived easily as in each step the proposed new matching matrix say $\boldsymbol{M}'$ is only slightly different from current matching matrix $\boldsymbol{M}$, so there are only few factors in the ratio

$$\mathcal{R} = \frac{\pi(\boldsymbol{M}', \boldsymbol{\tau}, \sigma, \boldsymbol{A}; \boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{\tau}, \sigma, \boldsymbol{A}; \boldsymbol{X}, \boldsymbol{Y})}. \tag{1.39}$$

We call this ratio, $\mathcal{R}$, the relative probability of $\boldsymbol{M}'$ compare to $\boldsymbol{M}$ and definition of this relative probability will be used in Chapter 2. In case of proposal of adding a match (j, k), $\mathcal{R}$ and proposal ratio, $\mathcal{PR} = T(\boldsymbol{M}|\boldsymbol{M}')/T(\boldsymbol{M}'|\boldsymbol{M})$, are

$$\mathcal{R} = \frac{\rho\,\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d}$$

$$\mathcal{PR} = \frac{p^*}{{n_u}^{-1}},$$

where n_u is the number of unmatched $\boldsymbol{X}$ points in M. Therefore, the acceptance probability for adding a match (j,k) is

$$\min\left[1,\ \frac{\rho\,\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}\,p^* n_u}{\lambda(\sigma\sqrt{2})^d}\right]. \tag{1.40}$$

Similarly, in the case of proposing to delete a match (j,k), $\mathcal{R}$ and $\mathcal{PR}$, are

$$\mathcal{R} = \frac{\lambda(\sigma\sqrt{2})^d}{\rho\,\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}$$

$$\mathcal{PR} = \frac{({n'}_u)^{-1}}{p^*},$$

where $n'_u = n_u + 1$ is the number of unmatched $\boldsymbol{X}$ points in $\boldsymbol{M}'$. Therefore, the acceptance probability for deleting a match (j,k) is

$$\min\left[1,\ \frac{\lambda(\sigma\sqrt{2})^d}{\rho\,\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}\,p^* n'_u}\right]. \tag{1.41}$$

Finally for a proposal of switching (j,k) point to (j,k') point $\mathcal{R}$ and $\mathcal{PR}$, are

$$\mathcal{R} = \frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_{k'} - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}$$

$$\mathcal{PR} = \frac{(1-p^*)}{(1-p^*)} = 1,$$

Therefore, the acceptance probability for switching the match of $\boldsymbol{x}_j$ from $\boldsymbol{y}_k$ to $\boldsymbol{y}_{k'}$ is

$$\min\left[1, \frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_{k'} - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_{k} - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}\right]. \tag{1.42}$$

In a single update this sampler has a very modest effect on $\boldsymbol{M}'$ (small moves because of single pair change), resulting in a slow mixing into the parameter space. To speed up the mixing, they considered several updates to $\boldsymbol{M}$ along with just single update to the other parameters per sweep. In Chapter 2, we will introduce a new sampler for $\boldsymbol{M}$ proposed by Forbes and Lauritzen (2014) which considers bigger moves in a single update, and hence improve the mixing rate.

Taylor, Mardia and Kent (2003) Model

Given two configurations $\boldsymbol{X}(m \times d)$ and $\boldsymbol{Y}(n \times d)$, Taylor, Mardia and Kent (2003) developed a model to match $\boldsymbol{X}$ and $\boldsymbol{Y}$, where $\boldsymbol{X}$ comes from a suitable distribution with parameter depending on $\boldsymbol{Y}$. Under their settings their model consists the three main components which are discussed below:

Transformation Groups. As parameters of the distribution of $\boldsymbol{X}$ depend on $\boldsymbol{Y}$, so a group of transformations, say, $\mathcal{F}$ is applied on (a subset of landmarks of) $\boldsymbol{Y}$ to bring it close to (a subset of landmarks of) $\boldsymbol{X}$. Some common choices of $\mathcal{F}$ are:

(1) The identity group $\mathcal{F}_0$ where $f(\boldsymbol{Y}) = \boldsymbol{Y}$.

(2) The translation group $\mathcal{F}_{\text{tr}}$ where $f(\boldsymbol{Y}) = \boldsymbol{Y} + \boldsymbol{b}$, for some $\boldsymbol{b} \in \mathbb{R}^d$.

(3) The group of rigid body transformations $\mathcal{F}_{\text{rigid}}$ where $f(\boldsymbol{Y}) = \boldsymbol{A}\boldsymbol{Y} + \boldsymbol{b}$, where $\boldsymbol{A}$ is $d \times d$ rotation matrix.

(4) The group of similarity transformations $\mathcal{F}_{\text{sim}}$ where $f(\boldsymbol{Y}) = \boldsymbol{A}\boldsymbol{Y} + \boldsymbol{b}$, where $\boldsymbol{A}$ is c times a rotation matrix and $c > 0$ is a scale factor.

(5) The group of affine transformations $\mathcal{F}_{\text{aff}}$ where $f(\boldsymbol{Y}) = \boldsymbol{A}\boldsymbol{Y} + \boldsymbol{b}$, where $\boldsymbol{A}$ is $d \times d$ non-singular matrix.

Labelling. All statistical models for the matching problem work according to a rule by which points in $\boldsymbol{X}$ are matched with points in $\boldsymbol{Y}$, and labelling is such a rule which tells which elements of $\boldsymbol{X}$ correspond to which elements of $\boldsymbol{Y}$. Mathematically, labelling can be seen as a mapping $\gamma : \{1, 2, \ldots, m\} \rightarrow \{1, 2, \ldots, n, (n+1)\}$ with $\gamma(j) = k$, which indicates the corresponding landmark, $\boldsymbol{y}_k$, in $\boldsymbol{Y}$ for each landmark of $\boldsymbol{x}_j$ in $\boldsymbol{X}$. When correspondence for any landmark in $\boldsymbol{X}$ is not found then it matches with state (n+1), which is called "coffin bin". Taylor, Mardia and Kent (2003) considered two different labelling approches namely injective and free cases when they derived their statistical model for matching configurations $\boldsymbol{X}$ and $\boldsymbol{Y}$:

Injective Case. Under this case each point in $\boldsymbol{X}$ is only allowed to match at most one point in $\boldsymbol{Y}$. Mathematically, for any two different j, say, j_1 and j_2 in $\boldsymbol{X}$ and $\gamma(j_1)$, $\gamma(j_2) \neq (n+1)$, then $\gamma(j_1) \neq \gamma(j_2)$. As $\gamma(j_1) = k$ imples $\gamma(j_2) \neq k$ $\forall\, j_2 \neq j_1$ and $k \neq (n+1)$, so there is a dependencies between mapping γ and the observations x_j, $j = 1, 2, \cdots, m$.

Free Case. In case of free labelling, each point in $\boldsymbol{X}$ is allowed to match more than one point in $\boldsymbol{Y}$. Under this framework , mapping γ and the observations $\boldsymbol{x}_j$, $j = 1, 2, \cdots, m$ are independent as two different $\boldsymbol{x}_j$ points can be matched with one $\boldsymbol{y}_k, k = 1, 2, \cdots, n$, point.

A distribution of $\gamma(j)$ over both injective and free labelling cases is required when a statistical model is fitted for matching problem, and they chose two differenent distributions for $\gamma(j)$ for the free and injective cases.

When the labelling $\gamma(j)$ contains a coffin bin, state $(n+1)$, and N_0 landmarks in $\boldsymbol{X}$ goes to coffin bin, then N_0 follows a binomial distribution with parameter $\theta_{(n+1)}$ i.e.

$$N_0 \sim \text{Bin}(\theta_{(n+1)}),$$

where $\theta_{(n+1)} = Pr\left(\gamma(j) = (n+1)\right)$ is the probability that $\boldsymbol{x}_j$, $j = 1, 2, \cdots, m$, matches with coffin bin and assumed to be the same for each j and independent over j. Under free labelling case, Taylor, Mardia and Kent (2003) considered the following distribution for mapping $\gamma(j)$

$$Pr\left(\gamma(j) = k\right) = \theta_k, \quad \text{with} \sum_{k=1}^{(n+1)} \theta_k = 1, \tag{1.43}$$

and the above form is same for all j and independent over j which facilitates the fitting of statistical models.

On the other hand, $\gamma(j)$ is not independent over j in injective labelling case. When N_0 landmarks of $\boldsymbol{X}$ are sent to the coffin bin then $(m - N_0)$ landmarks of $\boldsymbol{X}$ and $(m - N_0)$ landmarks of $\boldsymbol{Y}$ match together as $m \leq n$ assumed here, and they can be chosen from m and n landmarks in $\binom{m}{m-N_0}$ and $\binom{n}{m-N_0}$ ways respectively. These two sets of indices match to each other in $(m-N_0)!$ different ways. Therefore they considered discrete uniform distribution for labelling $\boldsymbol{\gamma}$ over $\binom{m}{m-N_0} \times \binom{n}{m-N_0} \times (m - N_0)!$ elements.

The distribution of $\gamma(j)$ considered above both for the free and injective cases contains the coffin bin, and the case of without coffin bin can be accommodated under this setting by considering $\theta_{(n+1)} = 0$ and hence $N_0 = 0$.

Error Distribution. It is necessary to model how close the corresponding element of $\boldsymbol{X}$ and $\boldsymbol{Y}$ will be to one another under a given transformation f on $\boldsymbol{Y}$ and a labelling γ. Taylor, Mardia and Kent (2003) considered the following form for $\boldsymbol{x}_j$ given f and γ

$$g_k(\boldsymbol{x}_j) = \begin{cases} N_d(\boldsymbol{A}\boldsymbol{y}_k + \boldsymbol{\tau}, \sigma^2 I_d), & \text{for } \gamma(j) = k, \text{ and } k \in \{1, 2, \ldots, n\} \\ \dfrac{1}{|\Omega|}, & \text{for } \gamma(j) = k, \text{ and } k = \{(n+1)\}, \end{cases}$$

which has the following equivalent form:

$$g_k(x_j) = \begin{cases} \dfrac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/\sigma\}}{\sigma^{\,d}}, & \text{for } \gamma(j) = k, \text{ and } k \in \{1, 2, \ldots, n\} \\ \dfrac{1}{|\Omega|}, & \text{for } \gamma(j) = k, \text{ and } k = \{(n+1)\}, \end{cases}$$

where ϕ denotes the standard normal density in $\mathbb{R}^d$ and $j = 1, 2, \ldots, m$. Here $g_k(\boldsymbol{x}_j)$ denotes the conditional pdf of $\boldsymbol{x}_j$ given f and $\gamma(j)$. Ω denotes some region in $\mathbb{R}^d$ containing all the $\boldsymbol{x}_j$ values, which plays the role of a parameter of a broad nonlocalized distribution of landmarks associated with the coffin bin.

Link between Green and Mardia and Taylor et al. Models. Green and Mardia (2006) model is developed under the assumption that both the configurations have Gaussian perturbations from an underlying Poisson process while Taylor, Mardia and Kent (2003) model assumed configuration $\boldsymbol{X}$ is perturbed from configuration $\boldsymbol{Y}$ in a one sided fashion. However, the form of the likelihood of Green and Mardia (2006) model is actually of the same form as the likelihood of Taylor, Mardia and Kent (2003) model but the variance parameter is doubled.

Although the form of the likelihoods under two models is the same form, they are conceptually different. Green and Mardia (2006) model is considered as symmetrical in the sense that $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$ are both generated by the same hidden point location of $\boldsymbol{\mu}$ (common space), and are considered as a matched pair where $\boldsymbol{X}$ is in the true space $\boldsymbol{\mu}$ and $\boldsymbol{Y}$ is transformed (using geometric transformation) to bring into the $\boldsymbol{\mu}$ space. On the other hand, Taylor, Mardia and Kent (2003) model is considered as asymmetrical in the sense that one of them ($\boldsymbol{Y}$) is taken as the population and the remaining one ($\boldsymbol{X}$) is trated as a random sample from

this population after an unknown transformation. Therefore, the matching probabilities between $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$ will not be same if the positions of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are treated on the other way around.

Model for the Injective Labelling Case. Under the injective labelling case the labelling $\boldsymbol{\gamma}$ has a discrete set of parameters which can be represented as $m \times (n+1)$ matching matrix $\boldsymbol{M} = [m_{jk}]_{m\times(n+1)}$. If $\gamma(j) = k$ and $k = 1, 2, \cdots, n, (n+1)$, then $m_{jk} = 1$ otherwise $m_{jk} = 0$. Thus the matching matrix $\boldsymbol{M}$ satisfies the following constraints

$$m_{jk} \in \{0, 1\}, \quad \sum_{k=1}^{(n+1)} m_{jk} = 1 \ \ \forall j \quad \text{and} \quad \sum_{j=1}^{m} m_{jk} \leq 1 \ \ \forall k.$$

When the transformation f is known, common in simple case, along with known background parameters σ^2 and $|\Omega|$, the only task is to estimate the labelling $\gamma(j)$. Taylor, Mardia and Kent (2003) developed the following likelihood for $\gamma(j)$

$$L(\gamma) = \prod_{j=1}^{m} \prod_{k=1}^{(n+1)} \left[g_k(\boldsymbol{x}_j)\right]^{m_{jk}}$$

$$l(\gamma) = \sum_{j=1}^{m} \sum_{k=1}^{(n+1)} m_{jk} \log \left[g_k(\boldsymbol{x}_j)\right], \tag{1.44}$$

where $l(\gamma) = \log L(\gamma)$. Using a well known integer programming method the maximum likelihood estimates of γ can be determined from the likelihood defined in (1.44). This simple strategy is not available in more complicated cases where transformation f is not known. However, methodology used to derive the likelihood (1.44) can still be used as a building block for methodology based on free model which can handle more complicated case.

Model for the Free Labelling Case. In the free labelling case the distribution

of labelling γ, defined in (1.45), is a continuous distribution, and the labelling γ and observations $\boldsymbol{x}_j$ are independent over j. Thus the joint likelihood derived by Taylor, Mardia and Kent (2003) for free labelling case is

$$L(\boldsymbol{X}, \gamma) = \prod_{j=1}^{m} \prod_{k=1}^{(n+1)} \left[\theta_k\right]^{I\left[\gamma(j)=k\right]} \left[g_k(\boldsymbol{x}_j)\right]^{I\left[\gamma(j)=k\right]}$$

$$l(\boldsymbol{X}, \gamma) = \sum_{j=1}^{m} \sum_{k=1}^{(n+1)} I\left[\gamma(j)=k\right] \Big(\log\left[\theta_k\right] + \log\left[g_k(\boldsymbol{x}_j)\right] \Big),$$

where $l(\boldsymbol{X}, \gamma) = \log L(\boldsymbol{X}, \gamma)$ and $I\left[\gamma(j) = k\right]$ is an indicator variable which takes value 1 when $\gamma(j) = k$ and $k = 1, 2, \cdots, n, (n+1)$ otherwise $I\left[\gamma(j) = k\right]$ takes 0. For notational convenience we use $\gamma_{jk} = I\left[\gamma(j) = k\right]$, so the above joint likelihood can be rewritten as

$$l(\boldsymbol{X}, \gamma) = \sum_{j=1}^{m} \sum_{k=1}^{(n+1)} \gamma_{jk} \Big(\log\left[\theta_k\right] + \log\left[g_k(\boldsymbol{x}_j)\right] \Big), \tag{1.45}$$

where $\gamma_{jk} \in \{0, 1\}$ like m_{jk} in the injective labelling case, and the prior probabilities of mixing proportions $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_{(n+1)})$, $\boldsymbol{A}$, $\boldsymbol{\tau}$, σ^2 and Ω are the parameters of the above model. Among all parameters, they considered σ^2 as a fixed tuning parameter as estimating it along with other parameters can be numerically unstable, while $|\Omega|$ is considered the size of the rectangular region which contains all the data.

Inference for the Free Model. Taylor, Mardia and Kent (2003) used the EM algorithm to carry out the inference for the above model. The mapping information $\gamma(j)$ is not available i.e. unobserved. They considered averaging $\gamma(j)$ over all k, $k = 1, 2, \cdots, n, (n+1)$, which yielded the following mixture model for each $\boldsymbol{x}_j$, $j = 1, 2, \cdots, m$, with density

$$g(x_j) = \sum_{k=1}^{n+1} \theta_k \, g_k(\boldsymbol{x}_j), \tag{1.46}$$

where θ_k are the mixing proportions of the mixture density $g(\boldsymbol{x}_j)$. As the mapping information $\gamma(j)$ is missing, so in the E step unobserved mappings $\gamma_{jk} = I\left[\gamma(j) = k\right]$ are replaced by their expected values under the current parameter estimates and the data, and expectation are

$$\begin{aligned}
\theta_{jk}^{(e)} &= E\big[I\left[\gamma(j) = k\right] = 1\big] \\
&= E\big[\gamma_{jk} = 1\big] \\
&= \frac{\theta_k \, g_k(x_j)}{g(x_j)},
\end{aligned}$$

here $\theta_{jk}^{(e)}$ are the probabilities of matching between $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$. Finally in the M step, the following expected likelihood is optimized over $\boldsymbol{\theta}$, $\boldsymbol{A}$ and $\boldsymbol{\tau}$, to get their estimates:

$$\sum_{j=1}^{m} \sum_{k=1}^{(n+1)} \theta_{jk}^{(e)} \Big(\log\left[\theta_k\right] + \log\left[g_k(\boldsymbol{x}_j)\right] \Big).$$

These two steps need to be repeated until convergence is achieved. Under the above settings, the probabilities of matching between $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, $\theta_{jk}^{(e)}$, is not directly used in the model as an unknown but these fixed quantities are used to calculate an updated estimates of mixing proportion $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_{(n+1)})$. In Chapter 3, we have proposed a fully Bayesian model which has the similarity with Taylor, Mardia and Kent (2003) but this matching probabilities $\boldsymbol{\theta}_j = (\theta_{j1}, \theta_{j2}, \cdots, \theta_{j(n+1)})$, $j = 1, 2, \cdots, m$, are directly modeled as an unknown. By introducing a Dirichlet prior for each $\boldsymbol{\theta}_j$, sample can be obtained from each full conditional distribution of $\boldsymbol{\theta}_j$ using Gibbs sampling.

1.3 Introduction to Differential Geometry

Using the Metropolis-Hastings based sampler to simulate from any complicated target distribution instead of using the Gibbs sampler (provided that direction generation are available) sometime produces good proposals and mixing compared to the samples obtained by direct generation from this target density under Bayesian settings, which is discussed in the MCMC section. In the context of our problem, we will need to simulate from the high dimensional posterior distribution in Chapter 4, which involves 4 unknowns $(\boldsymbol{\tau}, \sigma, \boldsymbol{A}, \boldsymbol{\theta})$. Although direct generation is available for all the full conditional distributions but motivated by the previous example we will use Metropolis-Hastings based sampler to simulate from the full conditional distributions of $\boldsymbol{A}$ and $\boldsymbol{\theta}$ respectively (MH within Gibbs) in order to get better proposal and good mixing. In addition, $\boldsymbol{A}$ and $\boldsymbol{\theta}$ can also be sampled together from their joint full conditional distribution using Metropolis-Hastings based sampler.

The rotation $\boldsymbol{A}$ has a natural geometric structure which is on $\mathbb{SO}(3)$. Similarly, the matching probabilities $\boldsymbol{\theta}_j, j = 1, 2, \cdots, m$, also have a natural geometric structure which is on an n dimensional probability simplex. Implementing Metropolis-Hastings algorithm to simulate from these full conditional distributions which are defined on a manifold requires knowledge of differential geometry. Therefore, in this section we introduce some important ideas from differential geometry. We have used the text books Amari and Nagaoka (2000) and Schutz, Bernard F (1980) to prepare the following overview of the differential geometry required.

Neighborhood. Suppose $\boldsymbol{q}$ is a point in $\mathbb{R}^d$. Then a set of points whose distance is less than $r\,(> 0)$ from point $\boldsymbol{q}$ is called the neighborhood of radius r of point $\boldsymbol{q}$, and it is denoted by $N_r(\boldsymbol{q})$. Graph A of Figure 1.6 illustrates the idea

of neighborhood of point $\boldsymbol{q}$ in $\mathbb{R}^2$, where interior disc bounded by the circle of radius r forms the neighborhood of $\boldsymbol{q}$ but point on the circle is not included in the $N_r(\boldsymbol{q})$.

Open Set. Let U be a set in $\mathbb{R}^d$ and if every point $\boldsymbol{q}$ in U has a neighborhood entirely contained in U, then U is called an open set. For example, all points $\boldsymbol{q} \in \mathbb{R}^1$ for which $a < \boldsymbol{q} < b$ for some a, b is simply an open set in $\mathbb{R}^1$.

Mapping. A mapping $\varphi : M \to S$ is a rule by which for each element $\boldsymbol{p}$ in space M, a unique element $\boldsymbol{q} = \varphi(\boldsymbol{p})$ is found in space S (shown in graph B of Figure 1.6). For example, an ordinary real-valued function defined on $\mathbb{R}^1$ is simple example of mapping, where φ takes a point p in $\mathbb{R}^1$ and yields a value $q = \varphi(p)$ in $\mathbb{R}^1$ also. From this example, it is clear that both the spaces M and S need not be distinct. However, in general they could be completely different from each other.

Under a mapping for each $\boldsymbol{p}$ in M there is a unique $\boldsymbol{q} = \varphi(\boldsymbol{p})$ in S, but not necessarily a unique $\boldsymbol{p}$ for every $\varphi(\boldsymbol{p})$ which is shown in graph C of Figure 1.6. In this Figure, it is clear that both p_0 and p_1 are mapped into the same value $q = \varphi(p_0) = \varphi(p_1)$ in S and this type of mapping is known as many-to-one mapping. Under a many-to-one mapping, the inverse image for a single value, say, q in S is not a single point in M, so there is no map φ^{-1} from S to M as under any map there should be an unique image. So, in general, $\varphi^{-1}(q)$ is not treated as the image of q under a map φ^{-1} but simply a set. On the hand hand, when for each $\varphi(\boldsymbol{p})$ there is unique image point in M, then the mapping is called one-to-one mapping.

A mapping $\varphi : M \to S$ is called M into S if it is defined for all points of M, and if every point of S, φ^{-1} exits, not necessarily an unique value, then it is called M

onto N. A mapping which is both one-to-one and onto is called bijection. When there are two mappings $\varphi : M \to S$ and $\psi : S \to W$, then another mapping $\psi \circ \varphi : M \to W$ is defined using the composition of φ and ψ which maps from M to W.

Coordinate System. Let $\varphi : S \to \mathbb{R}^d$ denote a coordinate system for a manifold S (for manifold definition see the last part of this discussion). Then for each point $\boldsymbol{p}$ in S, φ maps $\boldsymbol{p}$ to in $\mathbb{R}^d$ (shown in graph A of Figure 1.7). Mathematically it can be written as

$$\varphi(\boldsymbol{p}) = [q_1(\boldsymbol{p}),\, q_2(\boldsymbol{p}),\, \cdots, q_d(\boldsymbol{p})] \tag{1.47}$$

$$= [q_1,\, q_2,\, \cdots, q_d] \tag{1.48}$$

$$= \boldsymbol{q}, \tag{1.49}$$

where the point $\boldsymbol{q} = [q_1,\, q_2,\, \cdots, q_d]$ are called the coordinates of point $\boldsymbol{p}$. From equations (1.49) and (1.50), it is obvious that each $q_i, 1,\, 2,\, \cdots, d$, works as a function which maps a point $\boldsymbol{p}$ to its i^{th} coordinate i.e. $q_i : \boldsymbol{p} \to \mathbb{R}^1$, and called them as coordinate functions. Usually the above coordinate system can be written as $\varphi = [q_1,\, q_2,\, \cdots, q_d] = [q_i]$, and we will also follow this convention in this thesis.

Local and Global Coordinate Systems. When every point of manifold S can be mapped to $\mathbb{R}^d$ under one coordinate system then $\varphi = [q_i]$ is called a global coordinate system. In other words, when φ has S as its domain then it is called global coordinate system. However, in general, global coordinate system is not available for all manifolds. For example, the surface of a sphere and torus do not have global coordinate system. Manifolds that don't have global coordinate system only have local coordinate systems, and using these local coordinate systems all points in S can be mapped to $\mathbb{R}^d$ in the following way:

(1) Consider an open subset U of S which has a coordinate system φ such that $\varphi : U \to \mathbb{R}^d$.

(2) Select a point not contained in U but in another subset V which has a another coordinate system ψ such that $\psi : V \to \mathbb{R}^d$.

(3) Continue the process of choosing new subset until the whole S is covered, and hence every point in S is contained in at least in one subset which has a coordinate system.

How a point in S can be mapped in $\mathbb{R}^d$ under local and global coordinate systems is shown by graph A and C of Figure 1.7 respectively.

Coordinate Transformation. Suppose a point $\boldsymbol{p}$ in S can be mapped to $\mathbb{R}^d$ under two different coordinate system $\varphi = [q_i]$ and $\psi = [y_j]$ respectively. The coordinates $[y_j]$ can be obtained from coordinates $[q_i]$ using the following transformation

$$\psi \circ \varphi^{-1} : [q_1, q_2, \cdots, q_d] \to [y_1, y_2, \cdots, y_d], \tag{1.50}$$

and this transformation is known as coordinate transformation. Similarly, the coordinates $[q_i]$ can be obtained from coordinates $[y_j]$ using the transformation $\varphi \circ \psi^{-1} : [y_1, y_2, \cdots, y_d] \to [q_1, q_2, \cdots, q_d]$. Differential geometry uses differential operators to analyze the objects geometry with respect to different coordinates systems (functions), and it would be problematic if these operators are not invariant under these coordinate systems. Therefore, the necessity of smooth transformation between one coordinate to another coordinate is confirmed in differential geometry. The idea of coordinate transformation is displayed in graph B and D of Figure 1.7 under both global and local coordinate systems.

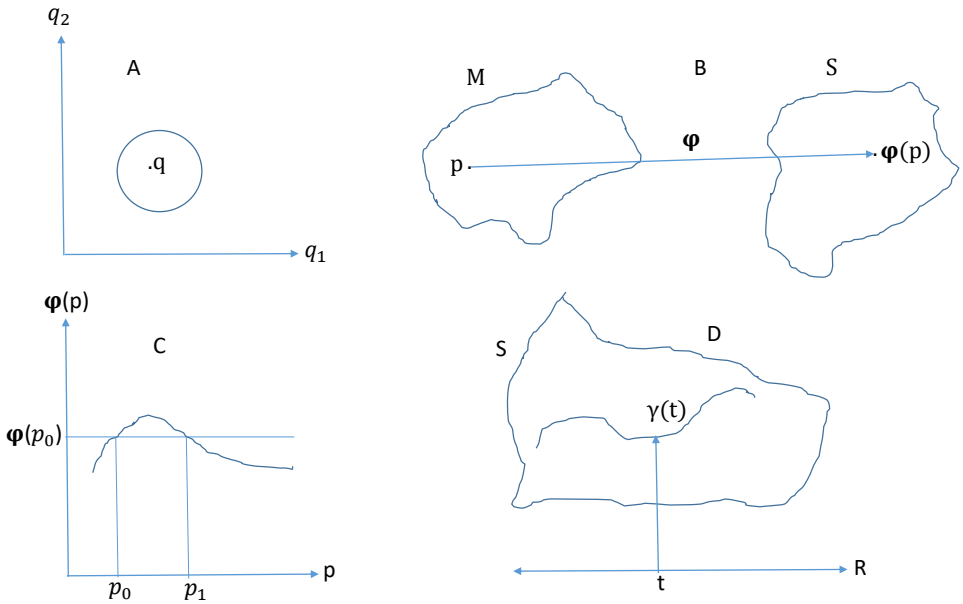

Figure 1.6: Pictorial representation of $N_r(\boldsymbol{q})$, mapping and curve of manifold; $N_r(\boldsymbol{q})$, mapping, many to one mapping and definition of curve in manifold are shown by the graphs A, B, C and D of this Figure respectively.

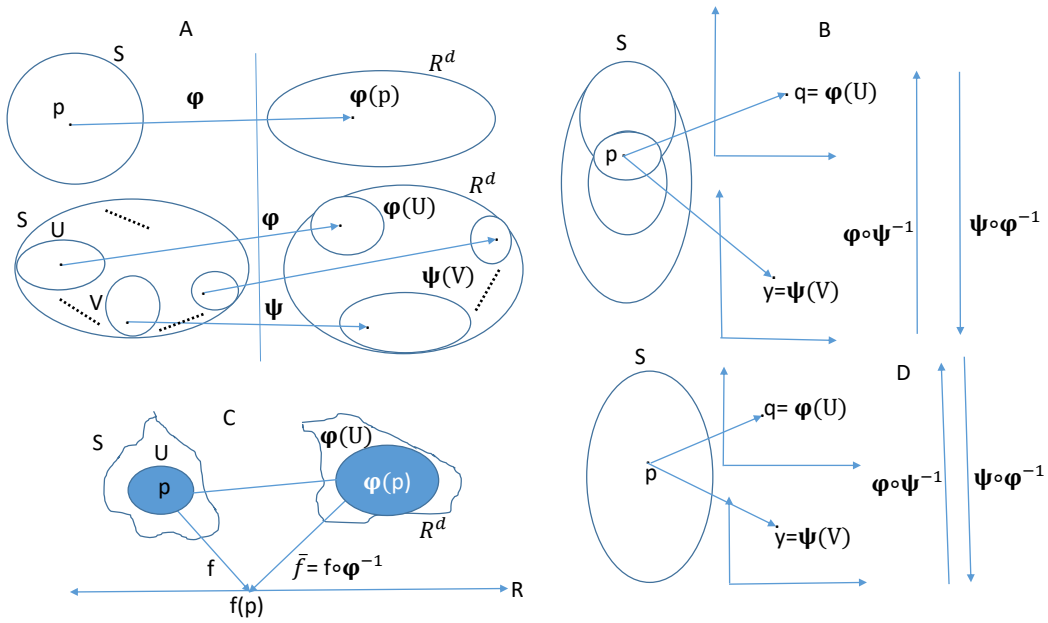

Figure 1.7: Pictorial representation of coordinate system, coordinate transformation and function on a manifold: Graph A shows the coordinate system from S to $\mathbb{R}^d$ both for global and local coordinate system; coordinate transformations between two coordinate systems under both global and local coordinate systems are explained by graphs D and B respectively; graph C explains how a function f defined on S works.

Manifold. To define a manifold a global coordinate system is considered here. However, this assumption does not affect the generality of the analysis as only local theory of the manifold is discussed in the later part of this thesis. Suppose S is a set, and for this set S a set of coordinates systems $\mathcal{A}$ exists. Then S is said to be a C^∞ manifold or simply manifold if the following two conditions hold:

(1) One-to-one mapping from S to some open subset of $\mathbb{R}^d$ is available for each $\varphi \in \mathcal{A}$.

(2) For any two $\varphi, \psi \in \mathcal{A}$ both $\varphi \circ \psi^{-1}$ and $\psi \circ \varphi^{-1}$ are defined and infinitely many times differentiable. This condition is also called C^∞ diffeomorphism.

Equation (1.50) and conditions (1) and (2) of manifold definition combined imply the existence of partial derivative of the functions $y_i = y_i(q_1, q_2, \cdots, q_d)$ and $q_i = q_i(y_1, y_2, \cdots, y_d)$ with respect to their variable arguments respectively as many times as needed.

Suppose U is a subset of manifold S, and for any coordinate system φ in S, $\varphi(U)$ is open subset in $\mathbb{R}^d$; then U is called an open subset of S. Condition 2 of manifold definition makes this property invariant to the choice of coordinate system φ. Now by restricting the domain of φ (whole S) to U, a coordinate system $\varphi : U \to \mathbb{R}^d$ can be defined and thus obtained a coordinate system for U $(\varphi|U)$. Hence U becomes a manifold whose dimension is the same as that of S.

Function on the Manifold. Consider $f : S \to \mathbb{R}$ be a function on a manifold S, which yields a real number $f(\boldsymbol{p})$ for a point $\boldsymbol{p}$ in S. Under a coordinate system $\varphi = [q_i]$, f can also be rewritten as a function of the coordinates i.e. $f(\boldsymbol{p}) = \bar{f}(q_1, q_2, \cdots, q_d)$ where $\bar{f} = f \circ \varphi^{-1}$. Here $\bar{f}$ is also a real valued function whose domain is $\varphi(S)$, a open subset in $\mathbb{R}^d$. How the function f works on manifold S and under a coordinate system φ is illustrated by graph C of Figure 1.7. The partial derivative $\frac{\partial \bar{f}}{\partial q_i}$ works as a function on $\varphi(S)$ when partial derivative of $\bar{f}$ is defined for all points in $\varphi(S)$. By transforming the domain back to S the partial derivatives of f with respect to q_i can be defined as

$$\frac{\partial f}{\partial q_i} \stackrel{\text{def}}{=} \frac{\partial \bar{f}}{\partial q_i} \circ \varphi : S \to \mathbb{R}, \tag{1.51}$$

where $\left(\frac{\partial f}{\partial q_i}\right)_{\boldsymbol{p}}$ denotes the value of this function at point $\boldsymbol{p}$. f is called a C^∞ function on S when $\bar{f}$ is differentiable an unbounded number of times with respect to its arguments, and it is invariant to the choice of coordinate system φ. Under two coordinate systems $[q_i]$ and $[y_j]$, both are C^∞ functions, the partial derivatives $\frac{\partial q_i}{\partial y_j}$ and $\frac{\partial y_j}{\partial q_i}$ exist and satisfy

$$\sum_{j=1}^{d} \frac{\partial q_i}{\partial y_j} \frac{\partial y_j}{\partial q_k} = \sum_{j=1}^{d} \frac{\partial y_i}{\partial q_j} \frac{\partial q_j}{\partial y_k} = \delta_k^i, \tag{1.52}$$

where δ_k^i is 1 if $k = i$, and 0 otherwise. In addition, for any C^∞ function f the following two conditions hold:

$$\frac{\partial f}{\partial y_j} = \sum_{j=1}^{d} \frac{\partial q_i}{\partial y_j} \frac{\partial f}{\partial q_i} \quad \text{and} \quad \frac{\partial f}{\partial q_i} = \sum_{j=1}^{d} \frac{\partial y_j}{\partial q_i} \frac{\partial f}{\partial y_j}. \tag{1.53}$$

Curve on the Manifold. For some $I \subset \mathbb{R}$, suppose a one-to-one function $\gamma : I \to S$ maps from I to manifold S. Then by defining $\gamma_i(t) \stackrel{\text{def}}{=} q_i(\gamma(t))$, the point $\gamma(t)$, $t \in I$, can be expressed using coordinates as $\tilde{\gamma}(t) = [\gamma_1(t), \gamma_2(t), \cdots, \gamma_d(t)]$. Now γ is called a C^∞ curve on S if $\tilde{\gamma}(t)$ is a C^∞ for $t \in I$, and this is independent of coordinate system $[q_i]$. This idea is graphically shown in Figure 1.6 (graph D).

Tangent Vectors and Tangent Spaces. Suppose γ is a curve on the manifold S, and $\gamma(t) = \boldsymbol{p}$ be a particular point in S for $t \in I$. Now if S is simply an open subset of $\mathbb{R}^d$ or embedded in some higher dimensional space $\mathbb{R}^l$ $(l > d)$, then the derivative of γ is simply known as tangent vector, and defined as

$$\frac{d\gamma}{dt} = \dot{\gamma} = \lim_{h \to 0} \frac{\gamma(t+h) - \gamma(t)}{h}. \tag{1.54}$$

In general, the definition of tangent vector defined in (1.54) is not meaningful when the surface is not flat (derivative is not defined). In this case, applying a C^∞ function f on $\gamma(t)$ makes it flat, and hence defines valid derivative, $\frac{d}{dt} f(\gamma(t))$. The derivative $\frac{d}{dt} f(\gamma(t))$ can be expressed in terms of coordinates as

$$\begin{aligned}\frac{d}{dt}f(\gamma(t)) &= \frac{d}{dt}\bar{f}(\tilde{\gamma}(t)) \\ &= \sum_{i=1}^{d}\left(\frac{\partial \bar{f}}{\partial q_i}\right)_{\tilde{\gamma}(t)}\frac{d\gamma_i(t)}{dt} \\ &= \sum_{i=1}^{d}\left(\frac{\partial f}{\partial q_i}\right)_{\gamma(t)}\frac{d\gamma_i(t)}{dt}. \end{aligned} \tag{1.55}$$

Equation (1.55) is known as directional derivative of function f along the curve γ, and this directional derivative can be expressed as tangent vector using an operator which maps f to $\frac{d}{dt}f(\gamma(t))$ and simply define $\left(\frac{d\gamma}{dt}\right)_{\boldsymbol{p}} = \dot{\gamma}(t)$. Hence the equation (1.56) can be written as

$$\left(\frac{d\gamma}{dt}\right)_{\boldsymbol{p}} = \dot{\gamma}(t) = \sum_{i=1}^{d}\dot{\gamma}_i(t)\left(\frac{\partial}{\partial q_i}\right)_{\boldsymbol{p}}, \tag{1.56}$$

where $\dot{\gamma}_i(t) = \frac{d\gamma_i(t)}{dt}$ and $\left(\frac{\partial}{\partial q_i}\right)_{\boldsymbol{p}}$ is an operator which maps : $f \to \left(\frac{\partial f}{\partial q_i}\right)_{\boldsymbol{p}}$. This definition of tangent vector as an operator can be seen as a generalization of the tangent vector defined in (1.54). The operator $\left(\frac{\partial}{\partial q_i}\right)_{\boldsymbol{p}}$ is called the tangent vector at point $\boldsymbol{p}$ of the i^{th} coordinate curve as partial derivative is simply a directional derivative along a coordinate axis. Suppose $\mathcal{T}_{\boldsymbol{p}}(S)$ is the set which contains all the curves which pass through the point $\boldsymbol{p}$, then equation (1.56) can be rewritten in terms of $\mathcal{T}_{\boldsymbol{p}}(S)$ as

$$\mathcal{T}_{\boldsymbol{p}}(S) = \sum_{i=1}^{d} c_i \left(\frac{\partial}{\partial q_i}\right)_{\boldsymbol{p}}, \tag{1.57}$$

where $[c_1, c_2, \cdots, c_d] \in \mathbb{R}^d$ and is a linear space formed by the natural basis $\left(\frac{\partial}{\partial q_i}\right)_{\boldsymbol{p}}$ of the coordinate system $[q_i]$ (as $\left(\frac{\partial}{\partial q_i}\right)_{\boldsymbol{p}}$ are linearly independent). Then this $\mathcal{T}_{\boldsymbol{p}}(S)$ and its elements are called tangent space and tangent vectors of manifold S at point $\boldsymbol{p}$.

Vector Fields. Suppose a mapping, $X : \boldsymbol{p} \to X_{\boldsymbol{p}}$, maps each point $\boldsymbol{p}$ from manifold S to a tangent space $\mathcal{T}_{\boldsymbol{p}}(S)$, then this mapping is called a vector field. For example, $\frac{\partial}{\partial q_i} : \boldsymbol{p} \to \left(\frac{\partial}{\partial q_i}\right)_{\boldsymbol{p}}$, $i = 1, 2, \cdots, d$, define d vector fields under a coordinate system $[q_i]$. In general, for a vector field X, for each point $\boldsymbol{p}$, $X_{\boldsymbol{p}}$ can be uniquely expressed as

$$X_{\boldsymbol{p}} = \sum_{i=1}^{d} X_{\boldsymbol{p}\, i} \left(\frac{\partial}{\partial q_i}\right)_{\boldsymbol{p}}, \tag{1.58}$$

where $X_{\boldsymbol{p}\,1}, X_{\boldsymbol{p}\,2}, \cdots, X_{\boldsymbol{p}\,d}$ are the d real numbers obtained according to the function $X_i : \boldsymbol{p} \to X_{\boldsymbol{p}\,i}$. These d functions $X_1, X_2, \cdots, X_d$ are called the components of X with respect to a coordinate system $[q_i]$. If $\sum_{i=1}^{d} X_i \left(\frac{\partial}{\partial q_i}\right)$ and $\sum_{j=1}^{d} \tilde{X}_j \left(\frac{\partial}{\partial y_j}\right)$ be the two components expressions of X under a two coordinate systems $[q_i]$ and $[y_j]$, then the following two conditions hold:

$$\tilde{X}_j = \sum_{j=1}^{d} X_i \frac{\partial y_j}{\partial q_i} \quad \text{and} \quad X_i = \sum_{j=1}^{d} \tilde{X}_j \frac{\partial q_i}{\partial y_j}. \tag{1.59}$$

The components are said to be C^∞ with respect to any other when they are C^∞ with respect to some coordinate system.

Submanifolds. Suppose S and M be two manifolds with dimension d and m respectively ($d > m$), where M is a subset of S. Let $[q_i]$ and $[y_j]$ be the two coordinate systems associated with S and M respectively. Then M is said to be a submanifold of S if the following three conditions hold:

(1) $q_i|M$(restricting the domain of q_i to M), $q_i : M \to \mathbb{R}$, is a C^∞ function on M.

(2) Suppose $B_j^i \overset{\text{def}}{=} \left(\frac{\partial q_i|M}{\partial y_j}\right)_{\boldsymbol{p}}$ and $B_j = [B_j^1, B_j^2, \cdots, B_j^d] \in \mathbb{R}^d$. Then $B_1, B_2, \cdots, B_m$ are linearly independent for each point $\boldsymbol{p}$ in M.

(3) For any open subset W of M, there exist another open subset U of S, such that $W = M \cap U$.

The choice of coordinate systems $[q_i]$ and $[y_j]$ does not affect the above three conditions i.e they are independent of choice of coordinate systems. The curve $\gamma : t \to \gamma(t)$ in M is also considered a curve in S if M is a submanifold of S. Hence the tangent vector $\left(\frac{d\gamma}{dt}\right)_{\boldsymbol{p}}$ of curve γ at point $\boldsymbol{p}$ belongs to both $\mathcal{T}_{\boldsymbol{p}}(M)$ and $\mathcal{T}_{\boldsymbol{p}}(S)$. Using equation (1.55), $\left(\frac{d\gamma}{dt}\right)_{\boldsymbol{p}}$ of curve γ under the two coordinate systems $[q_i]$ and $[y_j]$ can be written as

$$\sum_i \dot{\gamma}_i(t) \left(\frac{\partial}{\partial q_i}\right)_{\boldsymbol{p}} \in \mathcal{T}_{\boldsymbol{p}}(S) \tag{1.60}$$

and

$$\sum_j \dot{\gamma}_j(t) \left(\frac{\partial}{\partial y_j}\right)_{\boldsymbol{p}} \in \mathcal{T}_{\boldsymbol{p}}(M), \tag{1.61}$$

where $\gamma_i(t) \stackrel{\text{def}}{=} q_i(\gamma(t))$ and $\gamma_j(t) \stackrel{\text{def}}{=} y_j(\gamma(t))$ and the following condition holds (using equation 1.58)

$$\left(\frac{d\gamma_i}{dt}\right)_{\boldsymbol{p}} = \sum_{j=1}^{d} \frac{\partial q_i}{\partial y_j} \left(\frac{d\gamma_j}{dt}\right)_{\boldsymbol{p}} \tag{1.62}$$

Riemannian Manifold. A Riemannian manifold (M) is a differentiable manifold in which an inner product for the tangent space at each point $\boldsymbol{p}$ in manifold M is defined via a metric tensor. A metric tensor is a real valued function which takes two tangent vectors, say $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, from the tangent space at some point $\boldsymbol{p}$ as its arguments and yields a real number. Mathematically it can be expressed as

$$G(\boldsymbol{p}) : \mathcal{T}_{\boldsymbol{p}}(M) \times \mathcal{T}_{\boldsymbol{p}}(M) \to \mathbb{R}, \tag{1.63}$$

where $G(\boldsymbol{p})$ is a metric tensor at point $\boldsymbol{p}$ on the manifold M. To calculate the lengths of individual vectors and the angles between two vectors, metric tensor function defined in equation (1.62) can be used. For example, the square length

of a tangent vector $\boldsymbol{v}_1 \in \mathcal{T}_{\boldsymbol{p}}(M)$ is determined as

$$||\boldsymbol{v}_1||^2_{G(\boldsymbol{p})} = \langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle_{\boldsymbol{p}} = {\boldsymbol{v}_1}^{\mathrm{T}} G(\boldsymbol{p})\, \boldsymbol{v}_1, \tag{1.64}$$

where ${\boldsymbol{v}_1}^{\mathrm{T}}$ is the transpose of vector $\boldsymbol{v}_1$. Under a coordinate system $[q_i]$, the components g_{ij}, i, $j = 1, 2, \cdots, d$, of metric tensor G are determined by

$$g_{ij} = \left\langle \frac{\partial}{\partial q_i}, \frac{\partial}{\partial q_j} \right\rangle \tag{1.65}$$

which works as a C^∞ function by mapping each point $\boldsymbol{p}$ in M to $g_{ij}(\boldsymbol{p}) = \left(\left\langle \left(\frac{\partial}{\partial q_i} \right)_{\boldsymbol{p}}, \left(\frac{\partial}{\partial q_j} \right)_{\boldsymbol{p}} \right\rangle \right)_{\boldsymbol{p}}$.

Embedding. In this section we will discuss the concept of embedding as this is key to understanding the sampling procedures of rotation $\boldsymbol{A}$ and matching probabilities $\boldsymbol{\theta}$ from their full conditional distributions, respectively, (which are defined on the sphere and the simplex respectively) using the Metropolis-Hastings based sampler in Chapter 4.

A smooth map $f{:}\mathcal{M} \to \mathcal{N}$ between manifolds $\mathcal{M}$ of dimension m and $\mathcal{N}$ of dimension n is an embedding if it is diffeomorphism onto its range, and the range, f($\mathcal{M}$), is a submanifold of $\mathcal{N}$. In the following explicit embedding expressions both for sphere and simplex are provided:

Embedding of Sphere. The explicit embedding expression of an 2 dimensional sphere, $\mathbb{S}^2$, of radius r in 3 dimensional Euclidean space is well known which can

be written as

$$x_1 = r \sin \alpha_1 \sin \alpha_2,$$
$$x_2 = r \sin \alpha_1 \cos \alpha_2,$$
$$x_3 = r \cos \alpha_1,$$

where $-\pi/2 \leq \alpha_1 \leq \pi/2$ and $0 \leq \alpha_2 \leq 2\pi$ be the angular coordinate system, and $\sum_{i=1}^{3} x_i^2 = r^2$. By extending this expression we can obtain the explicit embedding expression of an $(n-1)$ dimensional sphere, $\mathbb{S}^{(n-1)}$, of radius r in n dimensional Euclidean space

$$\begin{aligned}
x_1 &= r \sin \alpha_1 \sin \alpha_2 \sin \alpha_3 \cdot \cdots \cdot \sin \alpha_{n-2} \sin \alpha_{n-1}, \\
x_2 &= r \sin \alpha_1 \sin \alpha_2 \sin \alpha_3 \cdot \cdots \cdot \sin \alpha_{n-2} \cos \alpha_{n-1}, \\
x_3 &= r \sin \alpha_1 \sin \alpha_2 \sin \alpha_3 \cdot \cdots \cdot \sin \alpha_{n-3} \cos \alpha_{n-2}, \\
&\vdots \\
x_{n-1} &= r \sin \alpha_1 \cos \alpha_2, \\
x_n &= r \cos \alpha_1,
\end{aligned}$$

where $\boldsymbol{\alpha} \in (0, 2\pi) \times (0, \pi)^{n-2}$ and $\sum_{i=1}^{n} x_i^2 = r^2$.

Embedding of Simplex. A n dimensional probability simplex is a $(n+1)$ dimensional Euclidean surface in which a distribution of the $(n+1)$ events lies. Each component of any random point in a simplex is non-negative and all $(n+1)$ components sum to 1. Mathematically we can express an n dimensional probability simplex, $\triangle^n$, as

$$\triangle^n = \left\{ \boldsymbol{\theta} \in \mathbb{R}^{n+1} : \theta_k \geq 0, \sum_{k=1}^{n+1} \theta_k = 1 \right\},$$

which is a n dimensional manifold embedded in $\mathbb{R}^{n+1}$.

Geodesics, Great Circle and Great Circle Distance. In Euclidean space,

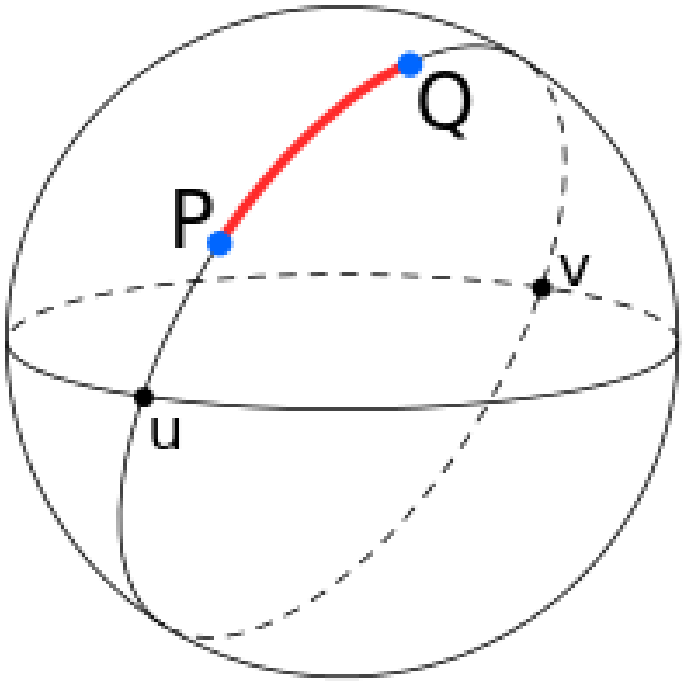

Figure 1.8: Illustration of a great circle and great circle distance. The distance between two points on a sphere, P and Q is the great-circle distance marked in red, whilst u and v are the the two antipodal points.

the distance between two points is the length of a straight line between them while in non euclidean geometry, geodesics are the equivalent of straight line. Geodesics on the sphere are the great circles, circles on the sphere whose centers and radii coincide with the center and radius of the sphere. The shortest distance between two distinct non-antipodal points (not directly opposite each other, P and Q in Figure 1.8) along the surface is on the unique great circle which includes the two points. The two points separate the great circle into two arcs, and the length of the shorter arc is known as great circle distance, marked by red colour in Figure 1.8. The relevant formulas of geodesic under different manifolds considered in this thesis are presented in Chapter 4 (section 4.2).

1.4 Description of the Data Sets

Firstly, we discuss the protein data in 3-dimensions, which was also used by Green and Mardia (2006), to illustrate their Bayesian model in bioinformatics for the unlabelled shape problem. A sequence of twenty different types of amino acids, organic compounds involving amine ($-\texttt{NH}_2$) and carboxyl ($-\texttt{COOH}$) functional groups along with a side chain ($-\texttt{R}$) are connected together by an amide bond called a "peptide bond" to make a protein. The function of these proteins depends on their structures, a 3d configuration of atoms determined by the sequence of their building blocks (amino acids). Most importantly, some key atoms like α-carbon atoms in each amino acid, and some other secondary structure features such as α-helices and β-sheets, play an important role in determining the structure of a protein. Predicting protein structure is considered one of the grand challenges in modern day science.

This particular protein data set contains a pair of proteins which are available as PDB codes `1a27` and `1cyd` respectively in PDB data bank. Biologically, the first protein `1a27` is known as Human Type I 17-Beta Hydroxysteroid Dehydrogenase and related to the process of oestrogen synthesis while the second one is a Mouse lung carbonyl reductase which works for metabolism of carbonyl compounds. This particular pair of proteins contain 63 (`1a27`) and 40 (`1cyd`) points respectively which are the active sites, regions where two molecules fit together to interact, for each protein. The corresponding coordinates of these points represent the centres of gravity of the amino acids that make up the common ligand nicotinamide adenine dinucleotide phosphate (NADP) binding sites of two protein which finally acts the basis of finding matched points.

From chemical properties of the active sites point of view the size and shape of the proteins (properties invariant to rigid transformation) are of interest, so finding geometric transformation (translation and rotation) is the main task to align these proteins. In this application, correspondence between landmarks in

two active sites of each protein also needs to be estimated as points are marked as unlabelled.

On the other hand, the Gel data in 2 dimensions contains two electrophoretic gels, and each gel consists of the locations of centres of 35 proteins in each. This data set was also used by Green and Mardia (2006) and available on the web at `http://www.stats.bris.ac.uk/mapjg/Align`. The objective here is to match these two gels based on the locations of the centres of 35 proteins on each gel which are unlabelled; that is the correspondence between the pairs of proteins, one protein from each gel, is unknown to us. Affine transformation was considered to align these two gels, where the linear part of this transformation was estimated based on 10 locations of the centres of proteins (one protein from each gel) which were identified and labelled by experts (see Horgan et al. (1992)). The estimate of this linear part is

$$\boldsymbol{A} = \begin{bmatrix} 0.973 & 0.0394 \\ -0.0231 & 0.9040 \end{bmatrix}. \tag{1.66}$$

Therefore, only translation τ and unknown matching labels between pairs of proteins need to be estimated to align these gels automatically.

Description of the Simulated Data Set

We have used an artificial data set in 3-dimensions for the purpose of conducting a simulation study in Chapter 5. To create this artificial data set we have chosen one protein (PDB code `1UBQ`) from the PDB data bank which contains ninty eight landmarks and treat this protein as our first configuration, say $\boldsymbol{Y}$. We then have created our second configuration $\boldsymbol{X}$ using $\boldsymbol{x}_i = \boldsymbol{A}\boldsymbol{y}_i + \boldsymbol{\tau} + \boldsymbol{\epsilon}_i$, $i = 1, 2, \cdots, 98$, where $\boldsymbol{A} = \boldsymbol{A}_{12}(0.5235)\,\boldsymbol{A}_{13}(1.047)\,\boldsymbol{A}_{23}(1.396)$, $\boldsymbol{\tau} = (25, 17, 8)^T$ and $\boldsymbol{\epsilon}_i \sim \mathcal{N}_3(\boldsymbol{0},\, 0.01^2 I_3)$. Here all the Eulerian angles are expressed in radians. In

this setting, it is expected that all the methods considered in this thesis will determine $L = 98$ match points with identity matching matrix $\boldsymbol{M}$ along with the above mentioned $\boldsymbol{A}$ and $\boldsymbol{\tau}$. To introduce unmatched landmarks between these two configurations we have added five noise terms generated from $\mathcal{N}_3(\mathbf{0}, 3^2 I_3)$ in the last five ponits of the second configuration repectively. Thefore, the last five points in each configuration are perturbed by independent $\mathcal{N}_3(\mathbf{0}, 3^2 I_3)$ errors. Finally, it is expected that this artificial data set will have ninty three matched points, the first ninty three of both configurations as we had earlier, and five unmatched points (the last five points).

1.5 Statistical Software

In this PhD thesis, C++ is used to write the codes as this this work is heavily reliant on statistical simulations where R is not feasible to use.

1.6 Contribution of this thesis

This thesis has three contribution chapters which are presented in Chapter 2, 3 and 4, followed by results and discussion while Chapter 6 discusses the conclusion and future work. The summaries of all the contribution chapters are given below:

Chapter 2 Motivated by the problem of Green and Mardia (2006) matching sampler which has low acceptance rate of M Forbes and Lauritzen (2014) have developed a new sampler to update a matching matrix $\boldsymbol{M}$ in finger print matching problem. Based on their analysis where both the simulated and real data sets are used, they have shown that their sampler is more efficient than Green and Mardia (2006) matching sampler as far as convergence time and acceptance rate of $\boldsymbol{M}$ are concerned. Motivated by this we have decided to use Forbes and Lauritzen (2014) matching sampler instead of Green and Mardia (2006) match-

ing sampler to update $\boldsymbol{M}$ under Green and Mardia (2006) framework to match two unlabelled configurations in size-and-shape space. Chapter 2 presents the detail algorithm of Forbes and Lauritzen (2014) matching sampler in the context of Green and Mardia (2006) framework along with a toy example which explains how this algorithm works. From our analysis, conducted in Chapter 5, where both the simulated and real data sets are used, we have observed that Forbes and Lauritzen (2014) matching sampler performs better than Green and Mardia (2006) matching sampler as far as convergence time and acceptance rate of $\boldsymbol{M}$ are concerned. However, Forbes and Lauritzen (2014) matching sampler needs slightly higher time compared to Green and Mardia (2006) matching sampler to execute.

Chapter 3 Although using the Forbes and Lauritzen (2014) matching sampler under Green and Mardia (2006) framework brought some improvement of the MCMC sampler but there is a much scope for improving the performance of the MCMC sampler further, which motivated us to consider a change to the model. In Chapter 3, we have proposed a fully Bayesian model for the alignment of two unlabelled configurations in size-and-shape space by generalizing Taylor, Mardia and Kent (2003) model (free labelling case) where matching probabilities are modelled directly as a random variables. Under the setting of our proposed model we don't need to resort Metropolis Hasting algorithm to sample matching matrix $\boldsymbol{M}$ as the full conditional distributions of matching probabilities (all are multinomial distributions) are much easier to simulate using Gibbs sampling. Treating matching probabilities as a random variables in our proposed model makes the space of $\boldsymbol{M}$ continuous but it increases the dimension of the parameter space of our proposed model compared to both the Green and Mardia (2006) and Taylor, Mardia and Kent (2003) models.

Chapter 4 In Chapter 4, we have proposed two types of inference techniques

for matching probabilities $\boldsymbol{\theta}$ and rotation $\boldsymbol{A}$ under our proposed model: (1) Gibbs sampling for all the full conditional distributions of matching probabilities $\boldsymbol{\theta}$ while rotation $\boldsymbol{A}$ is sampled using one of the methods proposed by Kent and Ganeiber (2012) (rejection sampling), Habeck's (2009) (MCMC) and Byrne and Girolami (2013) (MCMC on manifold) respectively; (2) Joint update of M (matching probabilities $\boldsymbol{\theta}$) and A using geodesic Monte Carlo algorithm from their joint full conditional distribution, which is defined on product manifold $\mathbb{SO}(3) \times \prod_{i=1}^{\mathrm{m}} \triangle_i{}^n$. In addition, we also determined rules of thumb for tuning parameters of geodesic Monte Carlo algorithm for generating samples from the matrix Fisher distribution ($\mathbb{SO}(3)$) and simplex ($\triangle$) separately by conducting a simulation study, which play the role of building blocks of tuning geodesic Monte Carlo on a product manifold. From our analysis, conducted in Chapter 5 where both the simulated and real data sets are used, we have observed that proposed MCMC sampler under our proposed model outperformed than MCMC sampler of Green and Mardia (2006) model and its variant where Forbes and Lauritzen (2014) matching sampler is used instead of Green and Mardia (2006) matching sampler.

Chapter 2

Efficient Sampler for Matching matrix M

From the simulation study conducted in Chapter 5, we have seen that MCMC sampler for Green and Mardia (2006) model can be slow to converge, and one of the main reasons of this slow convergence is that sampling matching matrix inefficiently using current Metropolis-Hastings sampler, especially where the space of possible matches is large. In this Chapter, we introduce a new sampler proposed by Forbes and Lauritzen (2014) for sampling matching matrix, we call it modified sampler, which is more efficient than Green and Mardia (2006) matching sampler shown in fingerprint analysis problem (Forbes and Lauritzen (2014)), under Green and Mardia (2006) framework. From our simulation simulation study, we have seen that the performance of MCMC sampler with Forbes and Lauritzen (2014) matching sampler is better compared to that of performance of MCMC sampler with Green and Mardia (2006) matching sampler as far as convergence time and acceptance rate of matching matrix $\boldsymbol{M}$ are concerned.

The Metropolis-Hastings sampler introduced by Green and Mardia (2006) uses proposals to sample matching matrix $\boldsymbol{M}$ which create, break or switch a single pair randomly chosen at each iteration. Hence the proposed modification of $\boldsymbol{M}$ is small, local changes, low acceptance rate and the sampler can be slow to mix (converge) in large problem. In addition, the sampler could be trapped in

local modes. Instead of creating or breaking a single pair there is in fact much more information available to us from the data which can be used to form a new pair or new block. Forbes and Lauritzen (2014) developed a sampler for fingerprint matching where not only a single entry but also a block change is allowed. The proposals consider the relative probabilities of matches, defined in equation (1.39), hence higher probability of acceptance of proposal than uniformly chosen points. This can allow large moves with a high probability of being accepted, which would hopefully improve the performance of the sampler. We therefore replace Green and Mardia (2006) matching by Forbes and Lauritzen (2014) matching sampler in the current MCMC sampler for which algorithm is designed by exploiting ideas from model for fingerprint matching in Forbes and Lauritzen (2014). This results in a more efficient sampler that appears to converge faster with high acceptance rate than the sampler in Green and Mardia (2006).

2.1 Why Consider a Relative Probabilistic Approach?

Our motivation for looking at efficient sampling for each full conditional distribution, especially for matching matrix $\boldsymbol{M}$, began with investigating efficient Bayesian inference over our posterior distribution that is used to solve the unlabelled shape problem.

Randomly breaking or creating a pair is already employed in many MCMC algorithms to update $\boldsymbol{M}$. For some instances, especially for small configurations, this approach may quickly converge to stationary distribution as exploring such a small $\boldsymbol{M}$ space is not affected too much because of random creation or break a pair. However, for larger configurations, such an approach explores such a large $\boldsymbol{M}$ space very slowly because of random creation or break and single entry change. On the other hand, breaking or creating a pair based on relative probability therefore gives us a much better idea of correct block matching which

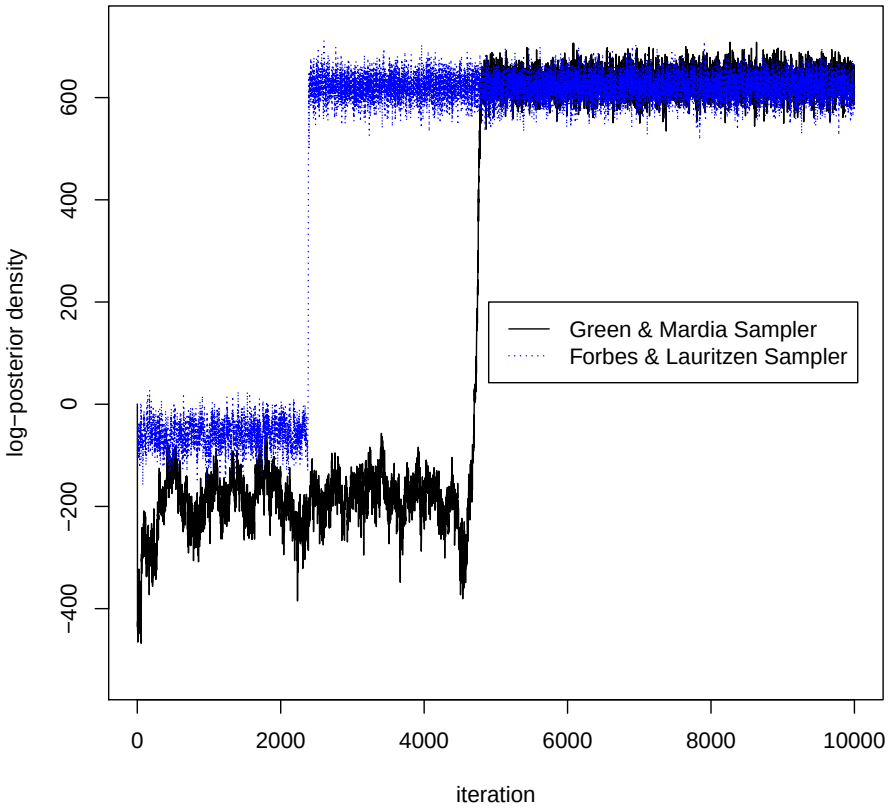

Figure 2.1: Comparison of two trace plots of unnormalized log posterior density of equation (1.24) for the matching simulated data where the two different approaches are used to update $\boldsymbol{M}$. This plot considers every 500^{th} simulated value to plot.

brings good exploration of $\boldsymbol{M}$ space, resulting in quick convergence. This can often be seen clearly from the trace plot of log posterior distribution (not normalized) density, as shown in Figure 2.1. From the trace plots of Figure 2.1, we can see that MCMC sampler with Forbes and Lauritzen (2014) matching sampler marked in blue requires less number of iterations to converge to the unnormalized log posterior value compared to the MCMC sampler with Green and Mardia (2006) matching sampler marked in black. A discussion regrading convergence diagnostic will be given in section 5.1.

2.2 Modified Sampler for $\boldsymbol{M}$

In this section, we will discuss the algorithm of our modified sampler for $\boldsymbol{M}$ which exploits the idea from the sampler developed by Forbes and Lauritzen (2014) for a finger print matching problem.

Given an initial matching matrix, say, M, the modified sampler starts working by introducing an auxiliary random variable, say, β. With the help of this auxiliary variable β, the modified sampler proposal steps begin by selecting a point uniformly either from configuration $\boldsymbol{X}$ or $\boldsymbol{Y}$. The algorithm is the same if β takes a value uniformly from either $\boldsymbol{X}$ or $\boldsymbol{Y}$. Without loss of generality, suppose β takes a value uniformly from $\boldsymbol{Y}$, say, $\boldsymbol{y}_\beta$, $\beta \in \{1, 2, \ldots, n\}$. There are two possibilities: either $\boldsymbol{y}_\beta$ is currently matched with any other $\boldsymbol{x}_j$, $j \in \{1, 2, \ldots, m\}$, or not, in that there is no such $\boldsymbol{x}_j$.

If $\boldsymbol{y}_\beta$ is currently unmatched the modified sampler considers all possible $(m{+}1)$ cases as a proposal. That is, all matches between $\boldsymbol{x}_j$, $j \in \{1, 2, \ldots, m\}$, and $\boldsymbol{y}_\beta$ simultaneously, as well as the current state i.e. $\boldsymbol{y}_\beta$ is unmatched. In order to select the next state, the modified sampler computes the relative probabilities for all possible proposal cases with respect to initial M. Finally, the next state is chosen by sampling one proposal case among all $(m{+}1)$ possible proposal cases based on relative probabilities with respect to initial M. This is different to Green and Mardia (2006), who choose one randomly selected $\boldsymbol{X}$ point to potentially match $\boldsymbol{y}_\beta$. By selecting the next state based on relative probabilities, modified sampler provides us extra information like which case has a higher probability to be the next state. If the $\boldsymbol{x}_j$ chosen to match $\boldsymbol{y}_\beta$ is already matched with any other $\boldsymbol{Y}$ point then this match is removed.

Similarly, the modified sampler also considers all possible $(m{+}1)$ cases simultaneously as a proposal if $\boldsymbol{y}_\beta$ is currently matched with any of the $\boldsymbol{X}$ point, say,

$\boldsymbol{x}_\gamma$, $\gamma \in \{1, 2, \ldots, m\}$. Apart from this current matched pair, $(\boldsymbol{x}_\gamma, \boldsymbol{y}_\beta)$, it also considers another $(m-1)$ matched pairs between $\boldsymbol{x}_j$, $j \in \{1, 2, \ldots, m\} \setminus \{\gamma\}$ with $\boldsymbol{y}_\beta$, and no match case simultaneously as the possible $(m+1)$ proposal states. Finally, it uses their relative probabilities computed with respect to initial $\boldsymbol{M}$ to sample one proposal case as a next state. Before considering a match between $\boldsymbol{x}_j$, $j \in \{1, 2, \ldots, m\} \setminus \{\gamma\}$ with $\boldsymbol{y}_\beta$, delete the current match pair, $(\boldsymbol{x}_\gamma, \boldsymbol{y}_\beta)$, and a match between $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, $k \in \{1, 2, \ldots, n\}$, and $k \neq \beta$ if there is any.

The relative probabilities for the aforementioned five possibilities can be easily derived under the posterior distribution derived in equation (1.24). Since in each case the proposed new matching matrix $\boldsymbol{M}'$ is only slightly different from current matching matrix $\boldsymbol{M}$, so the ratio $\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y}) / \pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})$ will have only few factors . In the following we will derive the general expressions for relative probabilities for each case with explanation by considering a toy example.

Suppose we have the following current matching matrix $\boldsymbol{M} = (m_{jk})$, $j \in \{1, 2, 3, 4\}$ and $k \in \{1, 2, 3, 4, 5\}$, where $m_{jk} = 1$ means $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$ are matched.

$$\boldsymbol{M} = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Case 1: $\boldsymbol{y}_\beta$ is currently unmatched with all $\boldsymbol{X}$ points and in proposal $\boldsymbol{M}'$ this $\boldsymbol{y}_\beta$ is considered as a matched pair with $\boldsymbol{x}_j$, which is currently unmatched with any points of $\boldsymbol{Y}$.

Illustration: For example, suppose β value is 3 i.e. $\boldsymbol{y}_3$ ($\boldsymbol{y}_\beta$) is chosen, which is

currently unmatched with any $\boldsymbol{X}$ points. Suppose the modified sampler selects $\boldsymbol{x}_3\,(\boldsymbol{x}_j)$, which has no corresponding matching point currently with $\boldsymbol{Y}$, for a match with $\boldsymbol{y}_3$ as a proposal $\boldsymbol{M}'$. Then $\boldsymbol{M}'$ differs from $\boldsymbol{M}$ by only a pair $(\boldsymbol{x}_3, \boldsymbol{y}_3)$ which is marked blue in $\boldsymbol{M}'$.

$$\boldsymbol{M} = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \Bigg| \; \boldsymbol{M}' = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Therefore, only $(\boldsymbol{x}_3, \boldsymbol{y}_3)$ pair is involved in the calculation of relative probability of $\boldsymbol{M}'$ with respect to $\boldsymbol{M}$ after canceling out the similar terms (pairs), and its expression is derived as follows

$$\begin{aligned} \frac{\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})} &= \frac{\displaystyle\prod_{j,k:m'_{jk}=1} \left[\frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \right]}{\displaystyle\prod_{j,k:m_{jk}=1} \left[\frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \right]} \\ &= \frac{\rho\, \phi\{(\boldsymbol{x}_3 - \boldsymbol{A}\boldsymbol{y}_3 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d}. \end{aligned}$$

The general form of the above expression is

$$\frac{\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})} = \frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_\beta - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d}, \tag{2.1}$$

where the matched $(\boldsymbol{x}_j, \boldsymbol{y}_\beta)$ pair creates a difference between $\boldsymbol{M}'$ and $\boldsymbol{M}$.

Case 2: $\boldsymbol{y}_\beta$ is currently unmatched with any $\boldsymbol{X}$ points and in proposal $\boldsymbol{M}'$ this $\boldsymbol{y}_\beta$ is considered as a matched pair with $\boldsymbol{x}_j$, which is currently matched with other $\boldsymbol{Y}$ point except $\boldsymbol{y}_\beta$, say, $\boldsymbol{y}_b$.

Illustration: For example, suppose β value is 3 i.e. $\boldsymbol{y}_3\,(\boldsymbol{y}_\beta)$ is chosen, which is currently unmatched with all $\boldsymbol{X}$ points. Suppose the modified sampler selects $\boldsymbol{x}_1\,(\boldsymbol{x}_j)$, which is currently matched with $\boldsymbol{y}_2\,(\boldsymbol{y}_b)$, for a match with $\boldsymbol{y}_3$ as a proposal $\boldsymbol{M}'$. In this case, modified sampler deletes the current matched pair of $\boldsymbol{x}_1$ with $\boldsymbol{y}_2$ before adding a match with $\boldsymbol{y}_3$, which are marked as red and blue respectively in $\boldsymbol{M}'$.

$$\boldsymbol{M} = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad \Big| \quad \boldsymbol{M}' = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Under this situation, the calculation of relative probability of $\boldsymbol{M}'$ with respect to $\boldsymbol{M}$ takes into account only $(\boldsymbol{x}_1, \boldsymbol{y}_2)$ and $(\boldsymbol{x}_1, \boldsymbol{y}_3)$ pairs as rest of the pairs cancel each other out, and its expression is derived as follows

$$\frac{\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma|\boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma|\boldsymbol{X}, \boldsymbol{Y})} = \frac{\prod_{j,k:m'_{jk}=1} \left[\frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \right]}{\prod_{j,k:m_{jk}=1} \left[\frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \right]}$$

$$= \frac{\frac{\rho\, \phi\{(\boldsymbol{x}_1 - \boldsymbol{A}y_3 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d}}{\frac{\rho\, \phi\{(\boldsymbol{x}_1 - \boldsymbol{A}\boldsymbol{y}_2 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d}}$$

$$= \frac{\frac{\phi\{(\boldsymbol{x}_1 - \boldsymbol{A}\boldsymbol{y}_3 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}{\frac{\phi\{(\boldsymbol{x}_1 - \boldsymbol{A}\boldsymbol{y}_2 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}.$$

The general form of the above expression is

$$\frac{\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma|\boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma|\boldsymbol{X}, \boldsymbol{Y})} = \frac{\frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_\beta - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}{\frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_b - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}, \tag{2.2}$$

where the new matched $(\boldsymbol{x}_j, \boldsymbol{y}_\beta)$ and deleted matched $(\boldsymbol{x}_j, \boldsymbol{y}_b)$ pairs create a difference between $\boldsymbol{M}'$ and $\boldsymbol{M}$.

Case 3: $\boldsymbol{y}_\beta$ is currently matched with $\boldsymbol{x}_\gamma$ and in proposal $\boldsymbol{M}'$ this $\boldsymbol{y}_\beta$ is considered as a matched pair with other $\boldsymbol{X}$ point say $\boldsymbol{x}_j$ instead of $\boldsymbol{x}_\gamma$, which is currently unmatched with all other $\boldsymbol{Y}$ points.

Illustration: For example, suppose β value is 4 i.e. $\boldsymbol{y}_4\,(\,\boldsymbol{y}_\beta)$ is chosen and is currently matched with $\boldsymbol{x}_2\,(\boldsymbol{x}_\gamma)$. Suppose the modified sampler selects $\boldsymbol{x}_3\,(\boldsymbol{x}_j)$, which has no corresponding matching point currently with $\boldsymbol{Y}$, for a match with $\boldsymbol{y}_4$ as a proposal $\boldsymbol{M}'$. In this case, modified sampler deletes the current matched

pair of $\boldsymbol{x}_2$ with $\boldsymbol{y}_4$ before adding a match between $\boldsymbol{x}_3$ and $\boldsymbol{y}_4$, which are marked as red and blue respectively in $\boldsymbol{M}'$.

$$\boldsymbol{M} = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \Bigg| \boldsymbol{M}' = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Under this situation, the calculation of relative probability of $\boldsymbol{M}'$ with respect to $\boldsymbol{M}$ takes into account only $(\boldsymbol{x}_2, \boldsymbol{y}_4)$ and $(\boldsymbol{x}_3, \boldsymbol{y}_4)$ pairs as rest of the pairs cancel each other out, and its expression is derived as follows

$$\begin{aligned} \frac{\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})} &= \frac{\displaystyle\prod_{j,k:m'_{jk}=1} \left[\frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \right]}{\displaystyle\prod_{j,k:m_{jk}=1} \left[\frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \right]} \\ &= \frac{\dfrac{\rho\, \phi\{(\boldsymbol{x}_3 - \boldsymbol{A}\boldsymbol{y}_4 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d}}{\dfrac{\rho\, \phi\{(\boldsymbol{x}_2 - \boldsymbol{A}\boldsymbol{y}_4 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d}} \\ &= \frac{\dfrac{\phi\{(\boldsymbol{x}_3 - \boldsymbol{A}\boldsymbol{y}_4 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}{\dfrac{\phi\{(\boldsymbol{x}_2 - \boldsymbol{A}\boldsymbol{y}_4 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}. \end{aligned}$$

The general form of the above expression is

$$\frac{\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})} = \frac{\frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_\beta - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}{\frac{\phi\{(\boldsymbol{x}_\gamma - \boldsymbol{A}\boldsymbol{y}_\beta - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}, \tag{2.3}$$

where the new matched $(\boldsymbol{x}_j, \boldsymbol{y}_\beta)$ and deleted matched $(\boldsymbol{x}_\gamma, \boldsymbol{y}_\beta)$ pairs create a difference between $\boldsymbol{M}'$ and $\boldsymbol{M}$.

Case 4: $\boldsymbol{y}_\beta$ is currently matched with $\boldsymbol{x}_\gamma$ and in proposal $\boldsymbol{M}'$ this $\boldsymbol{y}_\beta$ is considered as a matched pair with other $\boldsymbol{X}$ point say $\boldsymbol{x}_j$ instead of $\boldsymbol{x}_\gamma$, which is currently matched with a different $\boldsymbol{Y}$ point, say, $\boldsymbol{y}_b$.

Illustration: For example, suppose β value is 4 i.e. $\boldsymbol{y}_4\,(\boldsymbol{y}_\beta)$ is chosen uniformly which is currently matched with $\boldsymbol{x}_2\,(\boldsymbol{x}_\gamma)$. Suppose modified sampler selects $\boldsymbol{x}_1\,(\boldsymbol{x}_j)$, which is currently matched with $\boldsymbol{y}_2\,(\boldsymbol{y}_b)$, for a match with $\boldsymbol{y}_4$ as a proposal $\boldsymbol{M}'$. In this case, the modified sampler deletes the current matched pairs of $\boldsymbol{x}_2$ with $\boldsymbol{y}_4$ and $\boldsymbol{x}_1$ with $\boldsymbol{y}_2$ before adding a match between $\boldsymbol{x}_1$ and $\boldsymbol{y}_4$. Deletion and addition are marked as red and blue respectively in $\boldsymbol{M}'$.

$$\boldsymbol{M} = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad \Bigg| \quad \boldsymbol{M}' = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Under this situation, the calculation of relative probability of $\boldsymbol{M}'$ with respect to $\boldsymbol{M}$ takes into account only $(\boldsymbol{x}_2, \boldsymbol{y}_4)$, $(\boldsymbol{x}_1, \boldsymbol{y}_2)$ and $(\boldsymbol{x}_1, \boldsymbol{y}_4)$ pairs as rest of the pairs cancel each other out, and its expression is derived as follows

$$\frac{\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma|\boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma|\boldsymbol{X}, \boldsymbol{Y})} = \frac{\prod_{j,k:m'_{jk}=1} \left[\frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \right]}{\prod_{j,k:m_{jk}=1} \left[\frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \right]}$$

$$= \frac{\frac{\rho\, \phi\{(\boldsymbol{x}_1 - \boldsymbol{A}\boldsymbol{y}_4 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d}}{\frac{\rho\, \phi\{(\boldsymbol{x}_2 - \boldsymbol{A}\boldsymbol{y}_4 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \times \frac{\rho\, \phi\{(\boldsymbol{x}_1 - \boldsymbol{A}\boldsymbol{y}_2 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d}}$$

$$= \frac{\frac{\lambda\, \phi\{(\boldsymbol{x}_1 - \boldsymbol{A}\boldsymbol{y}_4 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}{\frac{\rho\, \phi\{(\boldsymbol{x}_2 - \boldsymbol{A}\boldsymbol{y}_4 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d} \times \frac{\phi\{(\boldsymbol{x}_1 - \boldsymbol{A}\boldsymbol{y}_2 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}$$

The general form of the above expression is

$$\frac{\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma|\boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma|\boldsymbol{X}, \boldsymbol{Y})} = \frac{\frac{\lambda\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}y_\beta - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}{\frac{\rho\, \phi\{(\boldsymbol{x}_\gamma - \boldsymbol{A}\boldsymbol{y}_\beta - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d} \times \frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_b - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}, \tag{2.4}$$

where the new matched $(\boldsymbol{x}_j,\, \boldsymbol{y}_\beta)$ and deleted matched $(\boldsymbol{x}_\gamma,\, \boldsymbol{y}_\beta)$ and $(\boldsymbol{x}_j,\, \boldsymbol{y}_b)$ pairs create a difference between $\boldsymbol{M}'$ and $\boldsymbol{M}$.

Case 5: $\boldsymbol{y}_\beta$ is currently matched with $\boldsymbol{x}_\gamma$ and in proposal $\boldsymbol{M}'$ this $\boldsymbol{y}_\beta$ is considered unmatched with any $\boldsymbol{X}$ points

Illustration: For example, suppose β value is 1 i.e. $\boldsymbol{y}_1\,(\,\boldsymbol{y}_\beta)$ is chosen, which is currently matched with $\boldsymbol{x}_4\,(\boldsymbol{x}_\gamma)$. Suppose the modified sampler chooses this $\boldsymbol{y}_1$ as unmatched with all $\boldsymbol{X}$ points as a proposal $\boldsymbol{M}'$. In this case, modified

sampler simply deletes the current matched pair of $\boldsymbol{x}_4$ with $\boldsymbol{y}_1$, and deletion is marked as red in $\boldsymbol{M}'$.

$$\boldsymbol{M} = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \Bigg| \boldsymbol{M}' = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Therefore, only $(\boldsymbol{x}_4, \boldsymbol{y}_1)$ pair is involved in the calculation of relative probability of $\boldsymbol{M}'$ with respect to $\boldsymbol{M}$ after canceling out the similar terms (pairs), and its expression is derived as follows

$$\begin{aligned} \frac{\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})} &= \frac{\displaystyle\prod_{j,k:m'_{jk}=1} \left[\frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \right]}{\displaystyle\prod_{j,k:m_{jk}=1} \left[\frac{\rho\, \phi\{(\boldsymbol{x}_j - \boldsymbol{A}y_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{\lambda(\sigma\sqrt{2})^d} \right]} \\ &= \frac{\lambda}{\dfrac{\rho\, \phi\{(\boldsymbol{x}_4 - \boldsymbol{A}\boldsymbol{y}_1 - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}. \end{aligned}$$

The general form of the above expression is

$$\frac{\pi(\boldsymbol{M}', \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})}{\pi(\boldsymbol{M}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma | \boldsymbol{X}, \boldsymbol{Y})} = \frac{\lambda}{\dfrac{\rho\, \phi\{(\boldsymbol{x}_\gamma - \boldsymbol{A}\boldsymbol{y}_\beta - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}}. \tag{2.5}$$

where the deleted matched $(\boldsymbol{x}_\gamma, \boldsymbol{y}_\beta)$ pair creates a difference between $\boldsymbol{M}'$ and $\boldsymbol{M}$.

Case 6: Apart from the aforementioned five cases, there is another case where $\boldsymbol{M}$ and $\boldsymbol{M}'$ are the same. In this case the relative probability of $\boldsymbol{M}'$ with respect to $\boldsymbol{M}$ is 1 as everything is cancel out in the ratio.

2.2.1 Illustrative Example: Updating M Using Modified Sampler

We return to the simple toy example considered above of deriving the modified sampler algorithm. In this section, we will narrate all necessary steps of modified sampler to update $\boldsymbol{M}$ for one iteration in the context of our toy example.

Consider $\boldsymbol{M}$ is our initial matching matrix and $\boldsymbol{y}_3$ is selected uniformly from all $\boldsymbol{Y}$ points by auxiliary random variable β. As $\boldsymbol{y}_3$ is currently unmatched with all $\boldsymbol{X}$ points, the modified sampler considers this $\boldsymbol{y}_3$ has a match with these four $\boldsymbol{X}$ points simultaneously as possible proposals. In addition to these four possible proposals it also considers initial $\boldsymbol{M}$ as its fifth proposal, i.e. $\boldsymbol{y}_3$ remains unmatched.

After deciding these five possible proposals, modified sampler calculates their relative probabilities with respect to $\boldsymbol{M}$. In this problem, Equation (2.2) is used to calculate the relative probabilities for ($\boldsymbol{x}_1$, $\boldsymbol{y}_3$), ($\boldsymbol{x}_2$, $\boldsymbol{y}_3$) and ($\boldsymbol{x}_4$, $\boldsymbol{y}_3$) matched pairs while Equation (2.1) is used for ($\boldsymbol{x}_3$, $\boldsymbol{y}_3$) matched pair. In addition, relative probability of $\boldsymbol{M}$ as a proposal with respect to $\boldsymbol{M}$ is 1 as proposal has no change compare to initial $\boldsymbol{M}$. Finally modified sampler samples one case based on these relative probabilities, and suppose the case of adding a match between $\boldsymbol{x}_1$ and $\boldsymbol{y}_3$ is chosen. Then set $\boldsymbol{M} = \boldsymbol{M}'$ as an update of $\boldsymbol{M}$, where

$$\boldsymbol{M}' = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

For a currently matched $\boldsymbol{Y}$ point, say $\boldsymbol{y}_1$, which is selected uniformly from all $\boldsymbol{Y}$ points by auxiliary random variable β, the modified sampler considers another 3 matched pairs between $\boldsymbol{y}_1$ and $\boldsymbol{x}_1$, $\boldsymbol{y}_1$ and $\boldsymbol{x}_2$, and $\boldsymbol{y}_1$ and $\boldsymbol{x}_3$ simultaneously as possible proposals apart from this current match pair $(\boldsymbol{x}_4, \boldsymbol{y}_1)$. In addition to these four possible proposals it also considers $\boldsymbol{y}_1$ has no match point with any of the $\boldsymbol{X}$ points i.e. $\boldsymbol{y}_1$ is unmatched as its fifth possible proposal.

After deciding these five possible proposals, the modified sampler calculates their relative probabilities with respect to $\boldsymbol{M}$. In this problem, Equation (2.4) is used to calculate the relative probabilities for $(\boldsymbol{x}_1, \boldsymbol{y}_1)$ and $(\boldsymbol{x}_2, \boldsymbol{y}_1)$ matched pairs while Equation (2.3) is used for $(\boldsymbol{x}_3, \boldsymbol{y}_3)$ matched pair. Furthermore, Equation (2.5) is used to calculate the relative probability for its fifth proposal while relative probability of $\boldsymbol{M}$ as a proposal, stays same as initial matching, with respect to $\boldsymbol{M}$ is 1 as proposal has no change compare to initial $\boldsymbol{M}$. Finally modified sampler samples one case based on these relative probabilities, and suppose the case of adding a match between $\boldsymbol{x}_1$ and $\boldsymbol{y}_1$ is chosen. Then set $\boldsymbol{M} = \boldsymbol{M}'$ as an update of $\boldsymbol{M}$, where

$$\boldsymbol{M}' = \begin{array}{c} \\ \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{array} \begin{array}{c} \begin{array}{ccccc} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Using Forbes and Lauritzen (2014) sampler in the current MCMC for Green and Mardia (2006) model, we have achieved some improvements as far as convergence time and acceptance rate are concerned (shown in Chapter 5), but still there is a scope to improve the overall performance of sampler of Green and Mardia (2006), which motivates a proposed modification to the model, described in Chapter 3.

If the discrete parameter space of the matching matrix in Green and Mardia (2006) model can be reparametrized as a continuous space, by expressing matching between $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$ in terms of probabilities, and sample these probabilities in Bayesian way like the other parameters by introducing prior for probabilities vectors, then MCMC sampler plausibly can move around this space easily and hence could bring quick convergence without depending initial starting values. Using this idea, we will develop a fully Bayesian model in the next Chapter.

Algorithm 3: Forbes and Lauritzen (2014) Matching Sampler

Data: Current matching matrix $\boldsymbol{M}$.

Result: Proposed matching matrix $\boldsymbol{M}'$.

begin

1: Sample β value uniformly over $\boldsymbol{Y}$ points

2: Determine if $\boldsymbol{y}_\beta$ is currently matched with any $\boldsymbol{X}$ point or not

3: **if** *unmatched* **then**

3a: Determine different $\boldsymbol{M}$ by considering $\boldsymbol{y}_\beta$ has a match with any $\boldsymbol{X}$ points simultaneously and consider all cases including initial $\boldsymbol{M}$ are the possible proposals

3b: Calculate the relative probabilities of these proposals with respect to initial $\boldsymbol{M}$

3c: Sample one proposal $\boldsymbol{M}'$ based on these relative probabilities and set $\boldsymbol{M} = \boldsymbol{M}'$ as an update

else

3a: Consider this $\boldsymbol{y}_\beta$ has a match with any $\boldsymbol{X}$ points plus has no match point $\boldsymbol{X}$ simultaneously and consider all these cases are possible proposals

3b: Calculate the relative probabilities of these proposals with respect to initial $\boldsymbol{M}$

3c: Sample one proposal $\boldsymbol{M}'$ based on these relative probabilities and set $\boldsymbol{M} = \boldsymbol{M}'$ as an update

Chapter 3

Proposed Bayesian Model for Unlabeled Size and Shape

In the previous Chapter, we showed some improvement of MCMC sampler for Green and Mardia (2006) model can be gained using Forbes and Lauritzen (2014) matching sampler instead of using Green and Mardia (2006) Metropolis - Hasting based matching sampler. Here we consider a change to the model : we propose a fully Bayesian model by modifying the Taylor, Mardia and Kent (2003) model where matching probabilities are modelled directly as a random variable, which makes the parameter space continuous. The motivation behind this idea is that MCMC sampler should be able to move around more easily in this continuous space compared to discrete space and hence improve mixing and convergence.

3.1 The Proposed Model

Consider a pair of configurations of points in d dimensions, say, $\boldsymbol{X}$ and $\boldsymbol{Y}$, where $\boldsymbol{X}$ consists of m points and $\boldsymbol{Y}$ of n points. The configurations $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be represented by $m \times d$ and $n \times d$ matrices, respectively, where the rows of $\boldsymbol{X}$ are $\boldsymbol{x}_j \in \mathbb{R}^d$, $j = 1, 2, \ldots, m$, and the rows of $\boldsymbol{Y}$ are $\boldsymbol{y}_k \in \mathbb{R}^d$, $k = 1, 2, \ldots, n$. Like the Taylor, Mardia and Kent (2003) regression based model, we also consider that $\boldsymbol{X}$ comes from a suitable distribution with parameters depending on $\boldsymbol{Y}$.

Mathematically, it takes the following form :

$$g_k(x_j) = \begin{cases} \dfrac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}y_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}, & \text{for } \gamma(j) = k, \text{ and } k \in \{1, 2, \ldots, n\} \\ \dfrac{1}{|\Omega|}, & \text{for } \gamma(j) = k, \text{ and } k = \{(n+1)\}, \end{cases}$$

for $j = 1, 2, \ldots, m$. Here we consider $\sigma = \sigma\sqrt{2}$ to keep the mathematical expression simillar like we had earlier for Green and Mardia (2006) model. The mapping $\gamma : \{1, 2, \ldots, m\} \to \{1, 2, \ldots, n, (n+1)\}$, with $\gamma(j) = k$ denotes the matching between $(\boldsymbol{x}_j, \boldsymbol{y}_k)$ pair, and state (n+1) represents a "coffin bin" for those landmarks $\boldsymbol{x}_j$ which have no corresponding match point in $\boldsymbol{Y}$. Therefore, $I[\gamma(j) = k]$ is an indicator variable which takes value 1 when $\gamma(j) = k$ and $k = 1, 2, \cdots, n, (n+1)$ otherwise $I[\gamma(j) = k]$ takes 0. For notational convenience, we rename $I[\gamma(j) = k] = \gamma_{jk}$, which will be used throughout this chapter. Here Ω denotes some region in $\mathbb{R}^d$ containing all the $\boldsymbol{x}_j$ values, which plays the role of a parameter of a broad nonlocalized distribution of landmarks associated with the coffin bin.

To develop a likelihood function we considered free mapping approach, as our aim is to parameterize the matching matrix as a continuous state space (in terms of matching probabilities). Under injective mapping approach, it is not possible as each point in $\boldsymbol{X}$ is only allowed to match at most one point in $\boldsymbol{Y}$. In free mapping, any $\boldsymbol{x}_j$ can be matched with any of $\boldsymbol{y}_k$ i.e. this framework allows two different $\boldsymbol{x}_j$ points to be matched with one $\boldsymbol{y}_k$ point. Under free mapping, we assume the independence of all $\gamma(j)$ over j i.e. any j can be matched with any k which facilitates the fitting of statistical model, with $Pr\big[I[\gamma(j) = k] = 1\big] = Pr[\gamma_{jk} = 1] = \theta_{jk}$, say, with $\sum_{k=1}^{(n+1)} \theta_{jk} = 1$, $\forall j$, i.e. the matrix $\boldsymbol{\theta} = [\theta_{jk}]_{m\times(n+1)}$, which is one of the parameters of interest about which to draw inference. The matrix $\boldsymbol{\theta}$ has m number of rows where its j^{th} row is a vector of length $n+1$, denoted by $\boldsymbol{\theta}_j = (\theta_{j1}, \theta_{j2}, \cdots, \theta_{j(n+1)})$. In this case all mappings $\gamma(j)$ and the observations $\boldsymbol{x}_j$ are independent over j, so the joint likelihood of observed data $\boldsymbol{x}_j$ and unob-

served (unknown) mappings $\gamma(j)$, given the parameters, is

$$\pi(\boldsymbol{x}_j, \gamma(j) \,|\boldsymbol{A}, \boldsymbol{\tau}, \sigma, \boldsymbol{\theta}_j, \Omega) = \pi(\gamma(j) \,|\, \boldsymbol{\theta}_j) \times \pi(\boldsymbol{x}_j \,|\, \gamma(j), \boldsymbol{A}, \boldsymbol{\tau}, \sigma, \Omega, \boldsymbol{\theta}_j), \text{ where}$$

$$\gamma(j) \sim \text{Multinomial}_{(n+1)}(\boldsymbol{\theta}_j), \text{ with } \pi(\gamma(j)\,;\, \boldsymbol{\theta}_j) \propto \prod_{k=1}^{(n+1)} \left[\theta_{jk}\right]^{\gamma_{jk}}, \text{ and}$$

$$\pi(\boldsymbol{x}_j \,|\, \gamma(j), \boldsymbol{A}, \boldsymbol{\tau}, \sigma, \Omega) \propto \prod_{k=1}^{(n+1)} \left[g_k(\boldsymbol{x}_j)\right]^{\gamma_{jk}}.$$

Therefore, the complete data likelihood is

$$\prod_{j=1}^{m} \left[\pi(\boldsymbol{x}_j, \gamma(j) \,|\boldsymbol{A}, \boldsymbol{\tau}, \sigma, \boldsymbol{\theta}_j, \Omega)\right] \propto \prod_{j=1}^{m} \prod_{k=1}^{(n+1)} \left[\theta_{jk}\right]^{\gamma_{jk}} \left[g_k(\boldsymbol{x}_j)\right]^{\gamma_{jk}} \tag{3.1}$$

To develop the above likelihod dependence between points of a configuration is not considered like the Green and Mardia (2006) hidden point model where independence Poisson assumption is considered. They justified their assumption by refering the findings of an unpublished work by K. V. Mardia, V. Nyirongo and D. R. Westhead where dependence assumption was considered. Green and Mardia (2006) mentioned in their paper that the ability of their method to detect matched pairs is little affected by the K. V. Mardia, V. Nyirongo and D. R. Westhead method, which required a substantially modified formulation and implementation technique because of considering dependence assumption. For the same reason, we have ignored the dependence between points of a configuration here to develop the likelihood 3.1.

Considering priors for the parameters $\boldsymbol{A}$, $\boldsymbol{\tau}$ and σ which are of the same form as in Green and Mardia (2006) while `Dirichlet` distribution is chosen as a prior

distribution for each $\boldsymbol{\theta}_j$, mathematically $\boldsymbol{\theta}_j \sim \texttt{Dir}_{n+1}\,(\boldsymbol{\eta}_k)$, we derive the posterior distribution of unknowns defined in (3.2). We use a uniform prior on $\boldsymbol{\theta}_j$, which is the special case where $\boldsymbol{\eta}_k$ is vector of length $(n+1)$ with all elements equal to 1. As $\boldsymbol{\theta}_j$ are independent over j, therefore prior distribution for $\boldsymbol{\theta}$ can be obtained by multiplying m independent $\texttt{Dir}_{n+1}\,(\boldsymbol{\eta}_k)$ distributions

$$\begin{aligned}
\pi(\boldsymbol{\theta}) &\propto p(\boldsymbol{\theta}_1) \times p(\boldsymbol{\theta}_2) \ldots \times p(\boldsymbol{\theta}_m) \\
&= \prod_{k=1}^{(n+1)} \left[\theta_{1k}\right]^{\eta_{1k}-1} \times \prod_{k=1}^{(n+1)} \left[\theta_{2k}\right]^{\eta_{2k}-1} \times \ldots \times \prod_{k=1}^{(n+1)} \left[\theta_{mk}\right]^{\eta_{mk}-1} \\
\pi(\boldsymbol{\theta}) &\propto \prod_{j=1}^{m} \prod_{k=1}^{(n+1)} \left[\theta_{mk}\right]^{\eta_{mk}-1} .
\end{aligned}$$

For the translation parameter vector $\boldsymbol{\tau}$, we use a normal prior i.e. $\boldsymbol{\tau} \sim N_d\,(\mu_{\boldsymbol{\tau}},\, \sigma^2_{\boldsymbol{\tau}}\, I_d)$, where $\mu_{\boldsymbol{\tau}}$ is a mean vector and $\sigma^2_{\boldsymbol{\tau}}\, I_d$ a covariance matrix, with I_d the $d \times d$ identity matrix. For the noise parameter σ, we have $\sigma^{-2} \sim \Gamma(\alpha, \beta)$, where $\pi(\sigma^{-2}) \propto \sigma^{-2(\alpha-1)} \exp\left(-\frac{\beta}{\sigma^2}\right)$.

We consider the matrix Fisher distribution as a prior distribution for rotation parameter matrix $\boldsymbol{A}$, where $\pi(\boldsymbol{A}) \propto \exp\left\{\text{tr}(\boldsymbol{F}_0{}^T \boldsymbol{A})\right\}$ and the parameter $\boldsymbol{F}_0$ is a $d \times d$ matrix. We use a uniform prior on $\boldsymbol{A}$, which is the special case where $\boldsymbol{F}_0$ is the $d \times d$ matrix of zeroes. Combining these prior distributions with (3.1), we have the following posterior distribution for the unknown mapping γ and unknown parameters $\boldsymbol{A}$, $\boldsymbol{\tau}$, σ and $\boldsymbol{\theta}$ given the data $\boldsymbol{X}$ and $\boldsymbol{Y}$

$$\pi(\gamma(j)\,, \boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma\,|\boldsymbol{X}, \boldsymbol{Y})\big] \propto \prod_{j=1}^{m} \prod_{k=1}^{(n+1)} \left[\theta_{jk}\right]^{\gamma_{jk}} \left[g_k(\boldsymbol{x}_j)\right]^{\gamma_{jk}} \left[\theta_{jk}\right]^{\eta_{jk}-1} \pi(\sigma^{-2})\, \pi(\boldsymbol{\tau})\, \pi(\boldsymbol{A}) \tag{3.2}$$

Our proposed model has the similarity with the Taylor, Mardia and Kent (2003)

model: uses regression approach to align configurations $\boldsymbol{X}$ and $\boldsymbol{Y}$; assumes free mapping assumption. Our proposed model is different from the Taylor, Mardia and Kent (2003) model in the sense that we directly modelled the probability of matching between $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$ as an unknown parameter while the Taylor, Mardia and Kent (2003) model consists of mixing proportions in their model, and matching probabilities were used in the E-step to replace the unknown mapping.

3.2 MCMC updates for all Unknowns

We perform inference by generating samples from the posterior distribution (3.2) using MCMC. To make inference, firstly we derive the full conditional distributions for each variable from (3.2) and then give a detailed sampling algorithm for each full conditional distribution.

From (3.2), as $\pi(\sigma^{-2})$, $\pi(\boldsymbol{\tau})$ and $\pi(\boldsymbol{A})$ are all independent of $\gamma(j)$, therefore the full conditional distribution of mapping $\gamma(j)$, $\pi(\gamma(j) \mid \boldsymbol{\theta}\boldsymbol{A}, \boldsymbol{\tau}, \sigma\,\boldsymbol{X}, \boldsymbol{Y})$, is

$$\begin{aligned}
\pi(\gamma(j) \mid \boldsymbol{\theta}, \boldsymbol{A}, \boldsymbol{\tau}, \sigma\,\boldsymbol{X}, \boldsymbol{Y}) &\propto \prod_{j=1}^{m} \prod_{k=1}^{(n+1)} \left[\theta_{jk}\right]^{\gamma_{jk}} \left[g_k(\boldsymbol{x}_j)\right]^{\gamma_{jk}} \\
&= \prod_{j=1}^{m} \prod_{k=1}^{(n+1)} \left[\theta_{jk}\, g_k(\boldsymbol{x}_j)\right]^{\gamma_{jk}} \\
&= \left(\prod_{k=1}^{(n+1)} \left[\theta_{1k} \times g_k(\boldsymbol{x}_1)\right]^{\gamma_{1k}}\right) \times \ldots \times \left(\prod_{k=1}^{(n+1)} \left[\theta_{mk} \times g_k(\boldsymbol{x}_m)\right]^{\gamma_{mk}}\right).
\end{aligned}$$

The full conditional distribution of $\gamma(j)$ is a product of m independent Multinomial distributions with updated unnormalized parameters $\left[\theta_{jk} \times g_k(\boldsymbol{x}_j)\right]$, $j = 1, 2, \ldots, m$, respectively. Therefore, drawing samples simultaneously from these m independent Multinomial distributions eventually yields the samples for the

full conditional distribution of $\gamma(j)$. To draw samples from each of these Multinomial distributions, we don't need their normalized version as most of the statistical packages like $\mathcal{C}++$ make them normalized by taking the relative weight of each parameter components before generating samples. However, we transform each of the above unnormalized Multinomial distributions to their corresponding normalized version to ensure identifiability. An unnormalized Multinomial distribution can be transformed to its normalized form just by dividing its parameter components by their total.

To illustarte the identifiability issue, consider that we have two unnormalized Multinomial distributions with parameter vectors $\boldsymbol{\alpha}$ = (4, 8, 12) and $\boldsymbol{\alpha^\star}$ = (1, 2, 3) respectively. From here we can easily show that one Multinomial distribution is some constant time the other one which is the case of nonidentifiable i.e. for two different unnormalized parameter vectors eventually the form of the two distributions is the same. On the other hand, there is no identifiability issue when Multinomial distribution is normalized. To prove this, we consider one of the Multinomial distributions from the right hand side of the full conditional distribution of $\gamma(j)$, say $\gamma(1)$ which has the following normalised form:

$$\pi(\gamma(1); \alpha_1, \alpha_2, \cdots, \alpha_k) \propto \prod_{k=1}^{(n+1)} {\alpha_k}^{\gamma_{1k}},$$

where $\alpha_k = \theta_{1k}\, g_k(\boldsymbol{x}_1) / \sum_{k=1}^{(n+1)} \theta_{1k}\, g_k(\boldsymbol{x}_1)$ and $\sum_{k=1}^{(n+1)} \alpha_k$ =1. Now for the two parameter vectors $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_k)$ and $\boldsymbol{\alpha^\star} = (\alpha_1^\star, \alpha_2^\star, \cdots, \alpha_k^\alpha)$ we can tell

$\pi(\gamma(1);\boldsymbol{\alpha}) = \pi(\gamma(1);\boldsymbol{\alpha}^{\star})$ only if

$$\pi(\gamma(1);\boldsymbol{\alpha}) - \pi(\gamma(1);\boldsymbol{\alpha}^{\star}) = 0$$

$$\implies \prod_{k=1}^{(n+1)} {\alpha_k}^{\gamma_{1k}} - \prod_{k=1}^{(n+1)} \alpha_k^{\star\gamma_{1k}} = 0$$

$$\implies \sum_{k=1}^{(n+1)} \gamma_{1k}\, log\Big(\frac{\alpha_k}{\alpha_k^{\star}}\Big) = 0$$

The left hand of the above expression is equal to zero for all γ_{1k} only when all its coefficients are equal to zero, which is only possible when $|\alpha_k| = |\alpha_k^{\star}|$. So we conclude that the multinomial model with normalized parameter vector is identifiable, and hence the full conditional distribution of $\gamma(j)$ is also identifiable.

Similarly from (3.2), as $\pi(\sigma^{-2})$, $\pi(\boldsymbol{\tau})$ and $\pi(\boldsymbol{A})$ do not involve any $\boldsymbol{\theta}$, therefore the full conditional distribution of $\boldsymbol{\theta}$, denoted as $\pi(\boldsymbol{\theta} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\tau}, \sigma\, \boldsymbol{X}, \boldsymbol{Y})$, is

$$\pi(\boldsymbol{\theta} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\tau}, \sigma\, \boldsymbol{X}, \boldsymbol{Y}) \propto \prod_{j=1}^{m} \prod_{k=1}^{(n+1)} \big[\theta_{jk}\big]^{\gamma_{jk}} \big[\theta_{jk}\big]^{\eta_{jk}-1}$$

$$= \prod_{j=1}^{m} \prod_{k=1}^{(n+1)} \big[\theta_{jk}\big]^{\left(\gamma_{jk}+\eta_{jk}-1\right)}$$

$$= \left(\prod_{k=1}^{(n+1)} \big[\theta_{1k}\big]^{\gamma_{1k}+\eta_{1k}-1}\right) \times \ldots \times \left(\prod_{k=1}^{(n+1)} \big[\theta_{mk}\big]^{\gamma_{mk}+\eta_{mk}-1}\right).$$

This full conditional distribution is a product of m independent Dirichlet distributions with updated parameters $\big(\gamma_{jk} + \eta_{jk}\big)$, $j = 1, 2, \ldots, m$, respectively. Therefore, we can draw samples for these parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_m)$ simultaneously from their corresponding Dirichlet distributions in Gibbs style fashion.

Similarly from (3.2), as $\boldsymbol{\tau}$ is only involved in $\pi(\boldsymbol{\tau})$ and $g_k(\boldsymbol{x}_j)$, $j = 1, 2, \ldots, m$

and $k = 1, 2, \ldots, n$, therefore the full conditional distribution of $\boldsymbol{\tau}$, denoted as $\pi(\boldsymbol{\tau} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\theta}, \sigma\, \boldsymbol{X}, \boldsymbol{Y})$, is

$$\begin{aligned}
\pi(\boldsymbol{\tau} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\theta}, \sigma\, \boldsymbol{X}, \boldsymbol{Y}) &\propto \pi(\boldsymbol{\tau}) \times \left[\prod_{j=1}^{m} \prod_{k=1}^{n} \left[g_k(\boldsymbol{x}_j) \right]^{\gamma_{jk}} \right] \\
&= \pi(\boldsymbol{\tau}) \times \prod_{j=1}^{m} \prod_{k=1}^{n} \left\{ \frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d} \right\}^{\gamma_{jk}} \\
&\propto \pi(\boldsymbol{\tau}) \times \prod_{j=1}^{m} \prod_{k=1}^{n} \left\{ \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\} \right\}^{\gamma_{jk}} \\
&= \pi(\boldsymbol{\tau}) \times \mathrm{E}_5,
\end{aligned}$$

where $\mathrm{E}_5 = \prod_{j=1}^{m} \prod_{k=1}^{n} \left\{ \phi\{(\boldsymbol{x}_j - \boldsymbol{A}y_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\} \right\}^{\gamma_{jk}}$. Simplifying E_5 we have

$$\begin{aligned}
\mathrm{E}_5 &= \prod_{j=1}^{m} \prod_{k=1}^{n} \left[\exp\left\{ -\frac{1}{4\sigma^2} \times \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}} \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\} \right\} \right]^{\gamma_{jk}} \\
&= \prod_{j=1}^{m} \prod_{k=1}^{n} \left[\exp\left\{ -\frac{1}{4\sigma^2} \times \gamma_{jk} \times \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}} \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\} \right\} \right] \\
&= \exp\left[-\frac{1}{4\sigma^2} \sum_{j=1}^{m} \sum_{k=1}^{n} \left\{ \gamma_{jk} \times \{(\boldsymbol{x}_j - \boldsymbol{A}y_k - \boldsymbol{\tau})\}^{\mathrm{T}} \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\} \right\} \right] \\
&\propto \exp\left[-\frac{1}{4\sigma^2} \sum_{j=1}^{m} \sum_{k=1}^{n} \gamma_{jk} \left\{ -\boldsymbol{x}_j{}^{\mathrm{T}}\boldsymbol{\tau} + (\boldsymbol{A}\boldsymbol{y}_k)^{\mathrm{T}}\boldsymbol{\tau} - \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{x}_j + \boldsymbol{\tau}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{y}_k) + \boldsymbol{\tau}^{\mathrm{T}}\tau \right\} \right] \\
&= \exp\left[-\frac{1}{4\sigma^2} \sum_{j=1}^{m} \sum_{k=1}^{n} \left\{ -\gamma_{jk}\boldsymbol{x}_j{}^{\mathrm{T}}\boldsymbol{\tau} + \gamma_{jk}(\boldsymbol{A}\boldsymbol{y}_k)^{\mathrm{T}}\boldsymbol{\tau} - \gamma_{jk}\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{x}_j + \gamma_{jk}\boldsymbol{\tau}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{y}_k) \right.\right. \\
&\quad \left.\left. + \gamma_{jk}\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\tau} \right\} \right].
\end{aligned}$$

Similarly from $\pi(\boldsymbol{\tau})$ we have

$$\pi(\boldsymbol{\tau}) \propto \exp\Big\{ -\frac{1}{2}\,(\boldsymbol{\tau}-\mu_{\boldsymbol{\tau}})^{\mathrm{T}}\boldsymbol{\tau}^{\mathrm{T}}(\sigma_{\boldsymbol{\tau}}^{2}I)^{-1}(\boldsymbol{\tau}-\mu_{\boldsymbol{\tau}})\Big\}$$

$$= \exp\Big\{ -\frac{1}{2\,\sigma_{\boldsymbol{\tau}}^{2}}\,(\boldsymbol{\tau}-\mu_{\boldsymbol{\tau}})^{\mathrm{T}}(\boldsymbol{\tau}-\mu_{\boldsymbol{\tau}})\Big\}$$

$$\propto \exp\Big\{ -\frac{1}{2\,\sigma_{\boldsymbol{\tau}}^{2}}\,(\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\tau}-\boldsymbol{\tau}^{\mathrm{T}}\mu_{\boldsymbol{\tau}}-\mu_{\boldsymbol{\tau}}^{T}\boldsymbol{\tau})\Big\}$$

plugging the $\pi(\boldsymbol{\tau})$ and E_5 values in the above expression we have the full conditional distribution of $\boldsymbol{\tau}$, $\pi(\boldsymbol{\tau} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\theta}, \sigma\,\boldsymbol{X}, \boldsymbol{Y})$, which is

$$\pi(\boldsymbol{\tau} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\theta}, \sigma\,\boldsymbol{X}, \boldsymbol{Y}) \propto \exp\Big\{ -\frac{1}{2}\,\big(e_1\,\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\tau} - e_2\,\boldsymbol{\tau}^{\mathrm{T}} - e_3\,\boldsymbol{\tau}\big)\Big\},$$

where e_3 is the transpose of e_2 and all three have the following expressions respectively:

$$e_1 = \big(1/\sigma_{\boldsymbol{\tau}}^{2} + \sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}/2\,\sigma^{2}\big)$$

$$e_2 = \big(\mu_{\boldsymbol{\tau}}/\sigma_{\boldsymbol{\tau}}^{2} + \sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k)/2\,\sigma^{2}\big)$$

$$e_3 = \big(\mu_{\boldsymbol{\tau}}^{\mathrm{T}}/\sigma_{\boldsymbol{\tau}}^{2} + \sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\big(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k\big)^{\mathrm{T}}/2\,\sigma^{2}\big).$$

Simplifying further gives

$$\pi(\boldsymbol{\tau} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\theta}, \sigma\,\boldsymbol{X}, \boldsymbol{Y}) \propto \exp\Big\{ -\frac{e_1}{2}\,\big(\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\tau} - \frac{e_2}{e_1}\,\boldsymbol{\tau}^{\mathrm{T}} - \frac{e_3}{e_1}\,\boldsymbol{\tau}\big)\Big\}$$

$$\propto\ N_d\Big(\boldsymbol{\mu}^{*},\ \sigma^{2^{*}}I\Big),$$

where

$$\boldsymbol{\mu}^* = \frac{e_2}{e_1} = \frac{\left(\mu_{\boldsymbol{\tau}}/\sigma^2_{\boldsymbol{\tau}} + \sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k)/2\,\sigma^2\right)}{(1/\sigma^2_{\boldsymbol{\tau}} + \sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}/2\,\sigma^2)}$$

$$\sigma^{2^*} = \frac{1}{e_1} = \frac{1}{(1/\sigma^2_{\boldsymbol{\tau}} + \sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}/2\,\sigma^2)}\,.$$

The full conditional distribution of $\boldsymbol{\tau}$ follows a d dimensional Multivariate normal distribution with mean vector $\boldsymbol{\mu}^*$ and variance-covariance matrix $\sigma^{2^*}I$. Green and Mardia (2006) draw samples for $\boldsymbol{\tau}$ from its d dimensional Multivariate normal distribution directly using Gibbs updates.

Similarly from (3.2), as σ^{-2} is only involved in $\pi(\sigma^{-2})$ and $g_k(\boldsymbol{x}_j)$, $j = 1, 2, \ldots, m$ and $k = 1, 2, \ldots, n$, therefore the full conditional distribution of σ^{-2}, denoted as $\pi(\sigma^{-2} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\theta}, \boldsymbol{\tau}\,\boldsymbol{X}, \boldsymbol{Y})$, is

$$\begin{aligned}
\pi(\sigma^{-2} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\theta}, \boldsymbol{\tau}\,\boldsymbol{X}, \boldsymbol{Y}) &\propto p(\sigma^{-2}) \times \left[\prod_{j=1}^{m}\prod_{k=1}^{n}\left[g_k(\boldsymbol{x}_j)\right]^{\gamma_{jk}}\right] \\
&= \pi(\sigma^{-2}) \times \prod_{j=1}^{m}\prod_{k=1}^{n}\left\{\frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}\right\}^{\gamma_{jk}} \\
&= (\sigma^{-2})^{(\alpha-1)}\exp\{-\beta/\sigma^2\} \times \mathrm{E}_6\,,
\end{aligned}$$

where $\mathrm{E}_6 = \prod_{j=1}^{m}\prod_{k=1}^{n}\left\{\frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d}\right\}^{\gamma_{jk}}$. Simplifying E_6 we have

$$\mathrm{E}_6 \propto \exp\left\{ -\frac{\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}}{4\sigma^2} \right\}$$

$$\times \left(\sigma^{-2}\right)^{\left(d \times \sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\right)/2}$$

$$= \left(\sigma^{-2}\right)^{\left(d \times \sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\right)/2} \times \exp\left\{ -\frac{\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\|(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\|^2}{4\sigma^2} \right\}.$$

Plugging the E_6 value in the above expression we have the full conditional distribution of σ^{-2}, which is proportional to

$$\pi(\sigma^{-2} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\theta}, \boldsymbol{\tau}\, \boldsymbol{X}, \boldsymbol{Y}) \propto \left(\sigma^{-2}\right)^{\left[\alpha + \left(d\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\right)/2 - 1\right]}$$

$$\times \exp\left[-\left\{ \beta + \frac{\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\|(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\|^2}{4} \right\}\sigma^{-2} \right]$$

$$\propto \Gamma(\alpha^*,\, \beta^*), \text{ where}$$

$$\alpha^* = \left[\alpha + \left(d\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\right)/2\right],$$

$$\text{and } \beta^* = \left\{ \beta + \frac{\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\|(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\|^2}{4} \right\}.$$

The full conditional distribution of σ^{-2} also follows also a gamma distribution, with shape parameter α^* and rate parameter β^*. Therefore, we can draw samples

for σ^{-2} from this gamma distribution directly in Gibbs style fashion.

Similarly from (3.2), as $\boldsymbol{A}$ is only involved in $\pi(\boldsymbol{A})$ and $g_k(\boldsymbol{x}_j)$, $j = 1, 2, \ldots, m$ and $k = 1, 2, \ldots, n$, therefore the full conditional distribution of $\boldsymbol{A}$, denoted as $\pi(\boldsymbol{A} \mid \gamma(j), \sigma^{-2}, \boldsymbol{\theta}, \boldsymbol{\tau}\, \boldsymbol{X}, \boldsymbol{Y})$, is

$$\begin{aligned}
\pi(\boldsymbol{A} \mid \gamma(j), \boldsymbol{\tau}, \boldsymbol{\theta}, \sigma\, \boldsymbol{X}, \boldsymbol{Y}) &\propto \pi(\boldsymbol{A}) \times \left[\prod_{j=1}^{m} \prod_{k=1}^{n} \left[g_k(x_j) \right]^{\gamma_{jk}} \right] \\
&= \pi(\boldsymbol{A}) \times \prod_{j=1}^{m} \prod_{k=1}^{n} \left\{ \frac{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}}{(\sigma\sqrt{2})^d} \right\}^{\gamma_{jk}} \\
&\propto \pi(\boldsymbol{A}) \times \prod_{j=1}^{m} \prod_{k=1}^{n} \left\{ \phi\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\} \right\}^{\gamma_{jk}} \\
&= \exp\{\mathrm{tr}(\boldsymbol{F}_0^{\mathrm{T}} \boldsymbol{A})\} \times \mathrm{E}_7,
\end{aligned}$$

where $\mathrm{E}_7 = \prod_{j=1}^{m}\prod_{k=1}^{n}\Big\{\phi\{(\boldsymbol{x}_j - \boldsymbol{A}y_k - \boldsymbol{\tau})/(\sigma\sqrt{2})\}\Big\}^{\gamma_{jk}}$. Simplifying E_7 we have

$$\begin{aligned}
\mathrm{E}_7 &= \prod_{j=1}^{m}\prod_{k=1}^{n}\left[\exp\Big\{-\frac{1}{4\sigma^2}\times\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}\Big\}\right]^{\gamma_{jk}} \\
&= \prod_{j=1}^{m}\prod_{k=1}^{n}\left[\exp\Big\{-\frac{1}{4\sigma^2}\times\gamma_{jk}\times\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}\Big\}\right] \\
&= \exp\Big[-\frac{1}{4\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,\mathrm{tr}\Big\{\{(\boldsymbol{x}_j - \boldsymbol{A}y_k - \boldsymbol{\tau})\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}\Big\}\Big] \\
&\qquad\qquad \big[\text{as } \{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{\tau})\} \text{ is a scalar}\big] \\
&= \exp\Big[-\frac{1}{4\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,\mathrm{tr}\Big\{\{(\boldsymbol{x}_j - \boldsymbol{\tau}) - \boldsymbol{A}\boldsymbol{y}_k\}^{\mathrm{T}}\{(\boldsymbol{x}_j - \boldsymbol{\tau}) - \boldsymbol{A}\boldsymbol{y}_k\}\Big\}\Big] \\
&\propto \exp\Big[-\frac{1}{4\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,\mathrm{tr}\big\{-(\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{y}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{x}_j - \boldsymbol{\tau}) + \boldsymbol{y}_k^{\mathrm{T}}\boldsymbol{y}_k\big\}\Big] \\
&\qquad\qquad \big[\text{as } \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = I \text{ and } (\boldsymbol{A}\,\boldsymbol{y}_k)^{\mathrm{T}} = \boldsymbol{y}_k^{\mathrm{T}}\,\boldsymbol{A}^{\mathrm{T}}\big] \\
&\propto \exp\Big[-\frac{1}{4\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,\mathrm{tr}\big\{-(\boldsymbol{x}_j - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y}_k - \boldsymbol{y}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{x}_j - \boldsymbol{\tau})\big\}\Big]
\end{aligned}$$

$$\mathrm{E}_7 \propto \exp\Big[-\frac{1}{4\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\big\{ -\mathrm{tr}\{(\boldsymbol{x}_j-\boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\,\boldsymbol{y}_k\} - \mathrm{tr}\{\boldsymbol{y}_k^{\mathrm{T}}\,\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{x}_j-\boldsymbol{\tau})\}\big\}\Big]$$

$$= \exp\Big[-\frac{1}{4\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\big\{ -\mathrm{tr}\{\boldsymbol{y}_k\,(\boldsymbol{x}_j-\boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\} - \mathrm{tr}\{(\boldsymbol{x}_j-\boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\,\boldsymbol{y}_k\}\big\}\Big]$$

$$\big[\text{as } \mathrm{tr}(A+B) = \mathrm{tr}(A)+\mathrm{tr}(B),\ \mathrm{tr}(AB) = \mathrm{tr}(BA) \text{ and } \mathrm{tr}(A^{\mathrm{T}}) = \mathrm{tr}(A)\big]$$

$$= \exp\Big[-\frac{1}{4\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\big\{ -\mathrm{tr}\{\boldsymbol{y}_k\,(\boldsymbol{x}_j-\boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\} - \mathrm{tr}\{\boldsymbol{y}_k\,(\boldsymbol{x}_j-\boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\}\big\}\Big]$$

$$= \exp\Big[-\frac{1}{4\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\big\{ -\mathrm{tr}\{2\,\boldsymbol{y}_k\,(\boldsymbol{x}_j-\boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\}\big\}\Big]$$

$$= \exp\Big[\mathrm{tr}\Big\{\frac{1}{2\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,\boldsymbol{y}_k\,(\boldsymbol{x}_j-\boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\Big\}\Big]$$

Plugging the E_7 value in the above expression we have the full conditional distribution of A, which is proportional to

$$\pi(\boldsymbol{A} \mid \gamma(j), \boldsymbol{\tau}, \boldsymbol{\theta}, \sigma\,\boldsymbol{X}, \boldsymbol{Y}) \propto \exp\{\mathrm{tr}(\boldsymbol{F}_0^{\mathrm{T}}\boldsymbol{A})\}\exp\Big[\mathrm{tr}\Big\{\frac{1}{2\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,\boldsymbol{y}_k\,(\boldsymbol{x}_j-\boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\Big\}\Big]$$

$$= \exp\Big[\mathrm{tr}(\boldsymbol{F}_0^{\mathrm{T}}\boldsymbol{A}) + \mathrm{tr}\Big\{\frac{1}{2\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,\boldsymbol{y}_k\,(\boldsymbol{x}_j-\boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{A}\Big\}\Big]$$

$$= \exp\Big[\mathrm{tr}\Big\{\boldsymbol{F}_0^{\mathrm{T}} + \frac{1}{2\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,\boldsymbol{y}_k\,(\boldsymbol{x}_j-\boldsymbol{\tau})^{\mathrm{T}}\Big\}\boldsymbol{A}\Big]$$

$$= \exp\Big[\mathrm{tr}\Big\{\boldsymbol{F}^{\mathrm{T}}\boldsymbol{A}\Big\}\Big] \text{, where}$$

$$\boldsymbol{F} = \boldsymbol{F}_0 + \frac{1}{2\sigma^2}\sum_{j=1}^{m}\sum_{k=1}^{n}\gamma_{jk}\,(\boldsymbol{x}_j-\boldsymbol{\tau})\,\boldsymbol{y}_k^{\mathrm{T}}.$$

Full conditional distribution of $\boldsymbol{A}$ also follows a matrix Fisher distribution, with matrix parameter $\boldsymbol{F}$. Apart from using Green and Mardia (2006) method (MCMC) to generate sample from the matrix Fisher distribution in 3-d case, we use another three existing methods proposed by Kent and Ganeiber (2012) (rejection sampling), Habeck's (2009) (MCMC) and Byrne and Girolami (2013) (MCMC on manifold) respectively in this thesis.

In this chapter, we have derived a fully Bayesian model in which the matching probabilities are modeled directly. We have also shown how each of the full conditional distribution is of a form which can be sampled from, and hence we can implement a Gibbs sampler to simulate from the joint distribution of all unknowns. Modeling the matching probabilities directly makes the space of matching more continuous, and hence plausibly easier to move around in. We explore the performance of sampler under this model in Chapter 5.

Overall performance of the MCMC sampler both for Green and Mardia (2006) and our proposed Bayesian models not only depends on efficient sampling of matching matrix but also an efficient sampling from the full conditional distribution of rotation $\boldsymbol{A}$. In the next chapter, we will discuss a method to sample matching probabilities $\boldsymbol{\theta}$ and $\boldsymbol{A}$ together (not in Gibbs style) using geodesic Monte Carlo, with the aim to achieve good proposal for $\boldsymbol{\theta}$ and $\boldsymbol{A}$ and less correlated sample.

In this chapter we have discussed an alternative method for unlabelled size and shape analysis by modifying the Taylor, Mardia and Kent (2003) model, where matching probabilities are modelled directly as a random variable. To estimate the parameters, $(\boldsymbol{\tau}, \boldsymbol{A}, \boldsymbol{\theta}, \sigma)$, of this alternative model using MCMC we will use two approaches; (i) $\boldsymbol{\tau}$, $\boldsymbol{\theta}$ and σ are updated using Gibbs sampler while $\boldsymbol{A}$ is updated using one of the methods proposed by Kent and Ganeiber (2012) (re-

jection sampling), Habeck's (2009) (MCMC) and Byrne and Girolami (2013) (MCMC on manifold) respectively; (ii) $\boldsymbol{\tau}$ and σ are updated using the Gibbs sampler while joint update of $\boldsymbol{\theta}$ and $\boldsymbol{A}$ is made using geodesic Monte Carlo algorithm from their joint full conditional distribution, which is defined on product manifold $\mathbb{SO}(3) \times \prod_{i=1}^{\mathrm{m}} \triangle_i{}^n$.

Chapter 4

Joint Update of Rotation and Matching Probabilities

Effective simulation methods for probability distributions defined on manifolds have drawn attention to the statistical community, as many advanced statistical models have been proposed based on these distributions in recent years. For example, the matrix Fisher distribution for random 3×3 rotation matrices, which is on $\mathbb{SO}(3)$, was used by Green and Mardia (2006) in their Bayesian model to align two unlabelled configurations of points in $\mathbb{R}^3$. As an another example we have seen in the previous chapter that the space of matching probabilities $\boldsymbol{\theta}$ in our proposed Bayesian model is the product of m independent Dirichlet distributions, each of dimension $(n+1)$ (which is in fact product of m simplex ($\triangle$), each of dimension n). In this chapter, we describe how Monte Carlo methods can be extended to manifolds, and illustrate them in the particular cases of interest to us.

Byrne and Girolami (2013) developed the geodesic Monte Carlo algorithm (differential geometric basis MCMC) to simulate from probability distributions defined on a manifold embedded in Euclidean space. This can be extended to sample from more than one probability distribution simultaneously based on the product of each constituent manifold. In this Chapter, we develop an algorithm for sampling independently both rotation and matching probabilities together (i.e, not

in Gibbs style) based on product manifold, $\mathbb{SO}(3) \times \prod_{i=1}^{m} \triangle_i{}^n$, where the geodesic Monte Carlo approach for simulating matrix Fisher and simplex separately play the role of building blocks. The motivation behind this idea is that proposed geodesic Monte Carlo sampler should be able to propose good proposal both for rotation and matching probabilities (less correlated samples) and hence improve mixing and convergence.

The presentation of this chapter is organized in several sections. In Section 4.1 we introduce the Hamiltonian Monte Carlo (HMC) approach in detail from a statistical view point, with an illustrative example. Section 4.2 shows how HMC is defined on different manifolds such as Riemannian and embedded manifolds, also called geodesic Monte Carlo, although here we are interested only on the latter. We choose the von Mises distribution to explain the geodesic Monte Carlo algorithm on an embedded manifold.

In the penultimate section we discuss the geodesic Monte Carlo technique to simulate from matrix Fisher and simplex separately, the building blocks in our proposed method, with an example for each respectively. Although Byrne and Girolami's algorithm can generate samples from the matrix Fisher distribution for any concentration parameter $\boldsymbol{F}$, consideration of different concentration levels of $\boldsymbol{F}$ requires extensive tuning, and it is one of the purposes of this chapter to address this problem. A similar type of tuning adjustment is necessary to simulate samples from the simplex, and hence it is also covered in this chapter as well. We have conducted a simulation study which is presented in this chapter to address this issue.

Finally, our proposed algorithm for independent update of rotation and matching probabilities together based on product manifold, $\mathbb{SO}(3) \times \prod_{i=1}^{\mathrm{m}} \triangle_i{}^n$, is discussed with an example.

4.1 Hamiltonian Monte Carlo (HMC)

Most of the Markov chain Monte Carlo algorithms based on random proposals explore a high dimensional target stationary distribution inefficiently, resulting in slow convergence. To achieve faster convergence, Markov chain needs to explore its space efficiently i.e. move in the right direction, and this is exactly what HMC does by effectively reducing the amount of random walk with the help of first order geometric information of target density (Neal (2011)).

4.1.1 Hamiltonian Dynamics

Before introducing HMC algorithm, firstly we need to know its building block called Hamiltonian dynamics which originated from the physics literature. These dynamics describe an object's movement throughout a system mathematically. According to the Hamiltonian dynamics, an object's motion depends on its d dimensional location vector $\boldsymbol{q}$ and momentum vector (defined as object's mass times its velocity) $\boldsymbol{p}$ at some time t in a particular system. The object has an associated potential energy $U(\boldsymbol{q})$ and an associated kinetic energy $K(\boldsymbol{p})$ for each location and momentum vector respectively, and the constant total energy of the system is the sum of these two energies, known as the Hamiltonian function $H(\boldsymbol{q}, \boldsymbol{p})$:

$$H(\boldsymbol{q}, \boldsymbol{p}) = U(\boldsymbol{q}) + K(\boldsymbol{p})$$

The partial derivatives of the Hamiltonian function known as Hamilton's equations describe how $\boldsymbol{q}$ and $\boldsymbol{p}$ change over time t, and are defined as follows:

$$\frac{dq_i}{dt} = \frac{\partial \mathrm{H}}{\partial p_i} = \frac{\partial K(\boldsymbol{p})}{\partial p_i}$$
$$\frac{dp_i}{dt} = -\frac{\partial \mathrm{H}}{\partial q_i} = -\frac{\partial U(\boldsymbol{q})}{\partial q_i}$$

for $i = 1, 2, \ldots, d$. Given an initial position vector $\boldsymbol{q}_0$ and momentum vector $\boldsymbol{p}_0$ at time t_0, new position and momentum vectors at time $t = (t_0 + s)$ can be determined by solving the above equations for a duration s.

To simulate Hamiltonian dynamics numerically on a computer, Hamilton's equations need to be approximated by discretizing time, and it can be done by splitting the time duration s up into a series of smaller step size ϵ. To get a better approximation it is desirable to make ϵ small enough. Euler's, modified Euler's and Leapfrog methods are commonly available methods in the literature for discretizing time but Leapfrog method offers better results than the other methods, particularly for use in MCMC (Neal, 2011). The detail algorithm of the Leapfrog method is given below:

To simulate dynamics over $(W \times \epsilon)$ units of time, the above algorithm needs to

Algorithm 4: Algorithm of Leapfrog Method

Data: $\boldsymbol{q}$ and $\boldsymbol{p}$ values at time t.

Result: $\boldsymbol{q}$ and $\boldsymbol{p}$ values at time $(t + \epsilon)$.

begin

1: Update $\boldsymbol{p}$ over $\epsilon/2$ time step using
$$p_i(t + \epsilon/2) = p_i(t) - (\epsilon/2)\left[\frac{\partial U}{\partial q_i(t)}\right]$$

2: Update $\boldsymbol{q}$ over ϵ time step using $q_i(t + \epsilon) = q_i(t) + \epsilon\left[\frac{\partial K}{\partial p_i(t + \epsilon/2)}\right]$

3: Update $\boldsymbol{p}$ for remaining half step using
$$p_i(t + \epsilon) = p_i(t + \epsilon/2) - \epsilon/2\left[\frac{\partial U}{\partial q_i(t + \epsilon)}\right]$$

run for W steps.

4.1.2 Constructing MCMC using Hamiltonian Dynamics

To construct a non physical Markov chain Monte Carlo method using the idea of Hamiltonian Dynamics, a Hamiltonian function $H(\boldsymbol{q}, \boldsymbol{p})$ can be viewed in terms of the probability distribution $\pi(\boldsymbol{q})$ from which samples need to be drawn.

This can be done by relating $\pi(\boldsymbol{q})$ to a potential energy function $U(\boldsymbol{q})$ via the concept of canonical distribution adopted from statistical mechanics. A canonical distribution corresponding to a given energy function $\mathrm{E}(\lambda)$ over a set of variables λ is defined as follows:

$$\pi(\lambda) \propto \exp\big\{ - \mathrm{E}(\lambda)\big\}$$

In the context of Bayesian statistics, we are interested in a posterior distribution of model parameters, and hence the position vector $\boldsymbol{q}$ plays the role of model parameters. The posterior distribution of model parameters is defined as follows:

$$\pi(\boldsymbol{q}|D) = \mathrm{L}(\boldsymbol{q} \mid D) \times \pi(\boldsymbol{q})$$

where $\pi(\boldsymbol{q})$ is the prior distribution of $\boldsymbol{q}$, and $\mathrm{L}(\boldsymbol{q} \mid D)$ denotes the likelihood given data D. As the Hamiltonian function $H(\boldsymbol{q}, \boldsymbol{p})$ defined above represents the energy function of two variables $\boldsymbol{q}$ and $\boldsymbol{p}$, so joint canonical distribution $(\boldsymbol{q}, \boldsymbol{p})$ is defined as follows:

$$\begin{aligned}
\pi(\boldsymbol{q}, \boldsymbol{p}) &\propto \exp\big\{ - H(\boldsymbol{q}, \boldsymbol{p})\big\} \\
&= \exp\big\{ - U(\boldsymbol{q})\big\} \times \exp\big\{ - K(\boldsymbol{p})\big\} \\
&= \pi(\boldsymbol{q}) \times \pi(\boldsymbol{p}),
\end{aligned}$$

where $\pi(\boldsymbol{q}) = \exp\big\{ - U(\boldsymbol{q})\big\}$ and $\pi(\boldsymbol{p}) = \exp\big\{ - K(\boldsymbol{p})\big\}$ are the canonical distributions for $\boldsymbol{q}$ and $\boldsymbol{p}$ respectively and these two are independent of each other as their joint canonical distribution is factorized. This means that sampling from $\pi(\boldsymbol{q})$ is equivalent to sampling from $\pi(\boldsymbol{q}, \boldsymbol{p})$ if the contribution of $\pi(\boldsymbol{p})$ is ignored at the end. Using $\pi(\boldsymbol{q}, \boldsymbol{p})$ instead of using only $\pi(\boldsymbol{q})$ allows us to use Hamiltonian Dynamics to sample from $\pi(\boldsymbol{q})$, and without introducing $\boldsymbol{p}$ this scope is unavailable. Constructing MCMC using Hamiltonian Dynamics in this way can be seen as an example of introducing auxiliary variable to facilitate the Markov Chain path. For practical implementation, the canonical distribution of $\boldsymbol{p}$ can be chosen independently as $\pi(\boldsymbol{p})$ and $\pi(\boldsymbol{q})$ are independent, and the standard normal distribution is the common choice for $\pi(\boldsymbol{p})$.

4.1.3 Hamiltonian Monte Carlo Algorithm

Hamiltonian Monte Carlo works in a two step procedure. In the first step, HMC uses Hamiltonian dynamics to generate proposals for the Markov chain rather than simply using any probability distribution to explore the target distribution $\pi(\boldsymbol{q}) = \exp\{-U(\boldsymbol{q})\}$ efficiently. From a given initial state $(\boldsymbol{q}_0, \boldsymbol{p}_0)$, new proposal state $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ is obtained by simulating Hamiltonian dynamics for a short time period s using the Leapfrog method. This proposed new state $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ is accepted as the next state of the Markov chain with probability $\frac{\pi(\boldsymbol{q}^*, \boldsymbol{p}^*)}{\pi(\boldsymbol{q}_0, \boldsymbol{p}_0)}$. If $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ is rejected as a next state of the Markov Chain then the next state is the same as the current state $\pi(\boldsymbol{q}_0, \boldsymbol{p}_0)$.

4.1.4 Illustrative Example

Suppose we wish to draw samples from a probability density function (pdf) $\pi(\boldsymbol{q})$ using HMC. In addition to $\pi(\boldsymbol{q})$ be the pdf on $\mathcal{R}^d$, we also consider that the partial derivatives of $\log \pi(\boldsymbol{q})$ exist. In order to simulate from $\pi(\boldsymbol{q})$, firstly we need to define the Hamiltonian function using the canonical distributions $\pi(\boldsymbol{q})$ and $\pi(\boldsymbol{p})$, where $\pi(\boldsymbol{p})$ is chosen independently and is $\mathcal{MVN}(\mathbf{0}, \boldsymbol{\Sigma})$. Now we can define the following potential and kinetic energy functions for $\pi(\boldsymbol{q})$ and $\pi(\boldsymbol{p})$ respectively:

$$\begin{aligned} U(\boldsymbol{q}) &= -\log \pi(\boldsymbol{q}) \\ &= -\log\Big[\mathrm{L}(\boldsymbol{q} \mid D) \times \mathrm{P}(\boldsymbol{q})\Big] \end{aligned}$$

and

$$\begin{aligned} K(\boldsymbol{p}) &= -\log \pi(\boldsymbol{p}) \\ &= \frac{1}{2}\log\{|2\pi\Sigma|\} + \frac{1}{2}\boldsymbol{p}^T\Sigma^{-1}\boldsymbol{p}. \end{aligned}$$

Therefore, the corresponding Hamiltonian function is

$$H(\boldsymbol{q}, \boldsymbol{p}) = -\log \pi(\boldsymbol{q}) + \frac{1}{2}\log\left\{\, |2\pi\Sigma| \,\right\} + \frac{1}{2}\boldsymbol{p}^T\Sigma^{-1}\boldsymbol{p}.$$

The partial derivatives of the above Hamiltonian function are:

$$\frac{d\boldsymbol{q}}{dt} = \dot{\boldsymbol{q}} = \frac{\partial \mathrm{H}}{\partial \boldsymbol{p}} = \Sigma^{-1}\boldsymbol{p}, \tag{4.1}$$

$$\frac{d\boldsymbol{p}}{dt} = -\frac{\partial \mathrm{H}}{\partial \boldsymbol{q}} = -\nabla_{\boldsymbol{q}} \log \pi(\boldsymbol{q}). \tag{4.2}$$

Now we can calculate $\pi\big(\boldsymbol{q}_0, \boldsymbol{p}_0\big) = \exp\big\{ -H(\boldsymbol{q}_0, \boldsymbol{p}_0)\big\}$ at initial state $\big(\boldsymbol{q}_0, \boldsymbol{p}_0\big)$ before running the Leapfrog algorithm for duration s to obtain a proposed state $\big(\boldsymbol{q}^*, \boldsymbol{p}^*\big)$. After having a new state $\big(\boldsymbol{q}^*, \boldsymbol{p}^*\big)$, calculate $\pi\big(\boldsymbol{q}^*, \boldsymbol{p}^*\big) = \exp\big\{ -H(\boldsymbol{q}^*, \boldsymbol{p}^*)\big\}$, and make the decision either to accept the proposed state or not, based on the following Metropolis acceptance probability and a random number u from $\mathrm{Unif}\,(0, 1)$:

$$\mathrm{P}_{\mathrm{accep}} = \frac{\exp\big\{ -H(\boldsymbol{q}^*, \boldsymbol{p}^*)\big\}}{\exp\big\{ -H(\boldsymbol{q}_0, \boldsymbol{p}_0)\big\}}$$

If u is less than $\mathrm{P}_{\mathrm{accep}}$, then we accept the proposed state as our new state. We can summarize the Hamiltonian Monte Carlo algorithm briefly as follows:

4.2 HMC on Riemannian and embedded manifolds

HMC that we defined in section 4.1 can be extended in a more general form on a Riemannian manifold, which allows HMC method to explore local geometric structure of target density efficiently. In order to define HMC on the Riemannian manifold we need to define $\pi(\boldsymbol{q})$ and $\pi(\boldsymbol{p})$ by taking into account the effect of

Algorithm 5: Algorithm of Hamiltonian Monte Carlo

Data: $\boldsymbol{q}_0$ and $\boldsymbol{p}_0$ values at time t.

Result: $\boldsymbol{q}$ and $\boldsymbol{p}$ values at time $(t + s)$.

begin

1: Sample initial momentum variables $\boldsymbol{p}_0$ from $\pi(\boldsymbol{p})$, independent of the values of position variables $\boldsymbol{q}_0$

2: Propose new state $\left(\boldsymbol{q}^*, \boldsymbol{p}^*\right)$ by simulating Hamiltonian dynamics for W steps using Leapfrog method with a step size ϵ ($\mathrm{s} = W \times \epsilon$)

3: Calculate $\mathrm{P_{accep}} = \exp\{-\mathrm{U}(\boldsymbol{q}^*) + \mathrm{U}(\boldsymbol{q}_0) - \mathrm{K}(\boldsymbol{p}^*) + \mathrm{K}(\boldsymbol{p}_0)\}$

4: **begin**

4a: Start accept reject steps

4b: Draw $U \sim \mathrm{Uniform}\big[0, 1\big]$

4c: **if** $U < P_{accep}$ **then**

$\boldsymbol{q}^{\mathrm{new}} \leftarrow \boldsymbol{q}^{\star}$

else

$\boldsymbol{q}^{\mathrm{new}} \leftarrow \boldsymbol{q}_0$

4d: Stop accept reject steps

the geometry of the Riemannian manifold.

From equation 4.1, we see that momentum variable $\boldsymbol{p} = \Sigma\,(\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t}) = \Sigma\,\dot{\boldsymbol{q}}$, where Σ and $\dot{\boldsymbol{q}}$ stand for some metric tensor and velocity (tangent vector) respectively. The position specific metric tensor $G(\boldsymbol{q})$ plays the role of Σ to define $\boldsymbol{p}$ which is considered as a family of probability distributions $\{\pi(.|G(\boldsymbol{q})) : G(\boldsymbol{q}) \in \Theta\}$. This family of probability distributions can be viewed as a Riemannian manifold, and according to equation (1.64) the squared distance of tangent vector $\dot{\boldsymbol{q}}$ in terms of $\boldsymbol{p}$ can be written in the following way:

$$\begin{aligned} \|\dot{\boldsymbol{q}}\|^2_{G(\boldsymbol{q})} &= \dot{\boldsymbol{q}}^{\mathrm{T}} G(\boldsymbol{q}) \dot{\boldsymbol{q}} \\ &= \big\{G(\boldsymbol{q})\,\dot{\boldsymbol{q}}\big\}^{\mathrm{T}} G(\boldsymbol{q})^{-1} \big\{G(\boldsymbol{q})\,\dot{\boldsymbol{q}}\big\} \\ &= \boldsymbol{p}^{\mathrm{T}} G(\boldsymbol{q})^{-1} \boldsymbol{p} \end{aligned} \tag{4.3}$$

Using a position specific inverse metric tensor we can define the kinetic energy term for the Hamiltonian function (Chang and Calin, 2005). We need to add a normalizing constant to equation (4.3) if we want to see it as an expression of the negative log of some Gaussian distribution, as we did in the original HMC, which is known as Kinetic energy. Finally, according to Girolami and Calderhead (2011), we can define HMC on the Riemannian manifold to simulate from $\pi(\boldsymbol{q}) \sim \mathcal{MVN}(\mathbf{0}, \boldsymbol{\Sigma})$ as :

$$H(\boldsymbol{q}, \boldsymbol{p}) = -\log \pi(\boldsymbol{q}) + \frac{1}{2} \log\big\{\,|2\pi G(\boldsymbol{q})|\,\big\} + \frac{1}{2}\,\boldsymbol{p}^{\mathrm{T}} G(\boldsymbol{q})^{-1} \boldsymbol{p}, \tag{4.4}$$

where $\pi(\boldsymbol{q})$ is the target density with respect to the Lebesgue measure in some coordinate system $[q_i]$, $i = 1, 2, \ldots, d$. So their joint canonical distribution with respect to the Lebesgue measure is

$$\begin{aligned}\pi(\boldsymbol{q},\boldsymbol{p}) &\propto \exp\big\{-H(\boldsymbol{q},\boldsymbol{p})\big\}\\ &= \pi(\boldsymbol{q}) \times \pi(\boldsymbol{p} \mid \boldsymbol{q}),\end{aligned} \tag{4.5}$$

where $\pi(\boldsymbol{q}) = \exp\big\{\log \pi(\boldsymbol{q})\big\}$ and $\pi(\boldsymbol{p} \mid \boldsymbol{q}) = \mathcal{MVN}(\boldsymbol{p} \mid \mathbf{0}, G(\boldsymbol{q}))$. As the joint canonical distribution $\pi(\boldsymbol{q},\boldsymbol{p})$ is not factorized in equation (4.5), the Hamiltonian isn't separable as in the previous case for original HMC. Therefore the standard leapfrog method is not applicable to approximate the Hamiltonian dynamics in this case.

By combining the first two terms of equation (4.4) into the negative log of the target density with respect to the Hausdorff measure of the manifold, the Hamiltonian function can be written as follows (Byrne and Girolami, 2013) :

$$H(\boldsymbol{q},\boldsymbol{p}) = -\log \pi_{\mathcal{H}}(\boldsymbol{q}) + \frac{1}{2}\boldsymbol{p}^{\mathrm{T}} G(\boldsymbol{q})^{-1}\boldsymbol{p}. \tag{4.6}$$

Now the Hamiltonian equations for this Hamiltonian function are

$$\frac{dq_i}{dt} = \frac{\partial \mathrm{H}}{\partial p_i} = \Big[G(\boldsymbol{q})^{-1}\boldsymbol{p}\Big]_i, \tag{4.7}$$

$$\frac{dp_i}{dt} = -\frac{\partial \mathrm{H}}{\partial q_i} = \nabla_{q_i}\Big[\log \pi_{\mathcal{H}}(\boldsymbol{q}) - \frac{1}{2}\boldsymbol{p}^{T} G(\boldsymbol{q})^{-1}\boldsymbol{p}\Big]. \tag{4.8}$$

Approximating these Hamiltonian equations by discretizing time is not straight forward anymore as in original HMC because of non separable Hamiltonian function. Generalized Leapfrog method proposed by Girolami and Calderhead (2011) can be used to approximate these dynamics which requires some numerical techniques as some of its steps are defined implicitly. In addition to the burden of using numerical techniques it also needs to calculate both the inverse and derivative of the metric tensor which is highly computationally expensive for high

dimensional case. Finally, a global coordinate system assumption is required by this method which may be violated for manifolds for which none exists, such as the sphere.

Another approach named geodesic integrator, developed by Byrne and Girolami (2013), can be used to approximate the above dynamics. We will describe here in detail how the geodesic integrator works as it is used in Geodesic HMC later, which is our main interest. The main idea of constructing the geodesic integrator is to consider each term of equation (4.6) as a separate Hamiltonian function and alternate simulating between the exact solutions. By considering each term of equation (4.6) as a separate Hamiltonian function we have the following two different Hamiltonian functions denoted by $H^{[1]}(\boldsymbol{q}, \boldsymbol{p})$ and $H^{[2]}(\boldsymbol{q}, \boldsymbol{p})$ respectively:

$$H^{[1]}(\boldsymbol{q}, \boldsymbol{p}) = -\log \pi_{\mathcal{H}}(\boldsymbol{q}), \tag{4.9}$$

$$H^{[2]}(\boldsymbol{q}, \boldsymbol{p}) = \frac{1}{2}\, \boldsymbol{p}^{\mathrm{T}} G(\boldsymbol{q})^{-1} \boldsymbol{p}. \tag{4.10}$$

The Hamiltonian equations for equation (4.9), which consist of only a potential term, are given as

$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = \frac{\partial \mathrm{H}^{[1]}}{\partial p_i} = 0 \tag{4.11}$$

$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial \mathrm{H}^{[1]}}{\partial q_i} = \nabla_{q_i}\Big[\log \pi_{\mathcal{H}}(\boldsymbol{q})\Big]. \tag{4.12}$$

We can write the above two equations in vector form in the following way:

$$\dot{\boldsymbol{q}} = \boldsymbol{0} \tag{4.13}$$

$$\dot{\boldsymbol{p}} = \nabla_{\boldsymbol{q}}\Big[\log \pi_{\mathcal{H}}(\boldsymbol{q})\Big]. \tag{4.14}$$

Solving the differential equations (4.13) and (4.14) with initial condition ($\boldsymbol{q}(0)$, $\boldsymbol{p}(0)$) gives us the following exact solution ($\boldsymbol{q}$(t), $\boldsymbol{p}$(t))

$$\int \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = 0$$

$$\boldsymbol{q}(t) = \mathrm{c}_1 \qquad \big[\mathrm{c}_1 \text{ is a integration constant}\big]$$

Using initial condition we have $\mathrm{c}_1 = \boldsymbol{q}(0)$, so

$$\boldsymbol{q}(t) = \boldsymbol{q}(0).$$

Similarly we have

$$\int \mathrm{d}\boldsymbol{p} = \int \nabla_{\boldsymbol{q}} \log \pi_{\mathcal{H}}(\boldsymbol{q})\, \mathrm{d}t$$

$$\boldsymbol{p}(t) = \mathrm{t}\, \nabla_{\boldsymbol{q}} \log \pi_{\mathcal{H}}(\boldsymbol{q}) + \mathrm{c}_2$$

The value of integration constant $\mathrm{c}_2 = \boldsymbol{p}(0)$, at $\mathrm{t} = 0$, so we have

$$\boldsymbol{p}(t) = \boldsymbol{p}(0) + \mathrm{t}\, \nabla_{\boldsymbol{q}} \log \pi_{\mathcal{H}}(\boldsymbol{q}).$$

Finally we can write the exact solution of $H^{[1]}$ as

$$\big(\boldsymbol{q}(t),\, \boldsymbol{p}(t)\big) = \big(\boldsymbol{q}(0),\, \boldsymbol{p}(0) + \mathrm{t}\, \nabla_{\boldsymbol{q}} \log \pi_{\mathcal{H}}(\boldsymbol{q})\big). \tag{4.15}$$

The Hamiltonian, defined in equation (4.10), which involves only a kinetic term,

has an intuitive meaning in a physical point of view. Before solving Hamiltonian equations for this Hamiltonian we need to be familiarized with Newton's first law. In a flat surface, a particle moving in a straight line will continue to move in a straight line subject to no external forces according to Newton's first law. Such a motion is described by the Hamiltonian defined as $H = \frac{1}{2}\,\boldsymbol{p}^{\mathrm{T}}\,\Sigma^{-1}\,\boldsymbol{p}$ where Σ denotes some metric tensor i.e. a Hamiltonian solely defined in terms of kinetic energy. This idea is exactly the same for a moving particle on a curved surface but needs to incorporate the metric tensor G(q) to define kinetic energy. Under this condition geodesic flow becomes a Hamiltonian flow (Abraham and Marsden, 1978). Hamiltonian equations for (4.10) can then be written as

$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = \frac{\partial \mathrm{H}^{[2]}}{\partial p_i} = \Big[\, G(\boldsymbol{q})^{-1}\boldsymbol{p}\Big]_i, \tag{4.16}$$

$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial \mathrm{H}^{[2]}}{\partial q_i} = -\frac{1}{2}\,\boldsymbol{p}^{\mathrm{T}}\,G(\boldsymbol{q})^{-1}\,\frac{G(\boldsymbol{q})}{\partial q_i}\,G(\boldsymbol{q})^{-1}\boldsymbol{p}. \tag{4.17}$$

Solution of these Hamiltonian equations is known as geodesic flow as this Hamiltonian consists of only kinetic energy, and when exact geodesic flow is known to us we can construct the geodesic integrator by alternating simulation from the dynamics $H^{[1]}$ and $H^{[2]}$ for some time step ϵ. Starting at state $(\boldsymbol{q}, \boldsymbol{p})$ the geodesic integrator for a single step works as follows:

(i) Firstly update state $(\boldsymbol{q}, \boldsymbol{p})$ for a time step $\epsilon/2$ according to the solution to $H^{[1]}$ which gives just a linear update of momentum variable $\boldsymbol{p}$ by setting

$$\boldsymbol{p} \leftarrow \boldsymbol{p} + \epsilon/2\,\nabla_{\boldsymbol{q}} \log \pi_{\mathcal{H}}(\boldsymbol{q}).$$

(ii) Then according to geodesic flow under $H^{[2]}$, do another update for a period of ϵ starting at $(\boldsymbol{q}, \boldsymbol{p})$ obtained from step (i).

(iii) Finally update once again according to $H^{[1]}$ for a period of $\epsilon/2$. The whole procedure should be repeated for W times to generate a new state $(\boldsymbol{q}, \boldsymbol{p})$.

Byrne and Girolami (2013) also defined the above HMC algorithm (on the Riemannian manifold) in terms of embedding coordinates by applying an isometric embedding transformation. The advantages of applying this transformation are; (i) we don't need to compute the metric tensor and its inverse (ii) we don't require the assumption that a global coordinate system exists.

Under an isometric embedding, $\xi : \mathcal{M} \to \mathbb{R}^n$, suppose we have a new path $x(t) = \xi\big(q(t)\big)$. Then according to equation (1.62) in Chapter 1, we can write

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = \sum_{j=1}^{d} \frac{\partial x_i}{\partial q_j} \frac{\mathrm{d}q_j(t)}{\mathrm{d}t} \tag{4.18}$$

$$\dot{x}_i(t) = \sum_{j=1}^{d} \frac{\partial x_i}{\partial q_j} \dot{q}_j(t) \tag{4.19}$$

Using equation (4.19) we can write down the square distance of $\dot{\boldsymbol{x}}$ in terms of square distance of $\dot{\boldsymbol{q}}$

$$\dot{\boldsymbol{x}}^{\mathrm{T}} \dot{\boldsymbol{x}} = \dot{\boldsymbol{q}}^{\mathrm{T}} G(\boldsymbol{q}) \dot{\boldsymbol{q}}, \tag{4.20}$$

where $G(\boldsymbol{q}) = \big[G_{ij}\big]_{\mathrm{d\,x\,d}}$ with $G_{ij} = \sum_{l=1}^{n} \frac{\partial x_l}{\partial q_i} \frac{\partial x_l}{\partial q_j}$. Therefore the embedded phase space $(\boldsymbol{x}, \boldsymbol{v})$ can be obtained from the phase space $(\boldsymbol{q}, \boldsymbol{p})$, where $\boldsymbol{p} = G(\boldsymbol{q})\, \dot{\boldsymbol{q}}$, in the following way

$$\begin{aligned} \boldsymbol{v} &= \dot{\boldsymbol{x}} \\ &= \boldsymbol{D}\, \dot{\boldsymbol{q}} \qquad \left[\boldsymbol{D} = [D_{ij}]_{\mathrm{n\,x\,d}} \text{ with } D_{ij} = \frac{\partial x_i}{\partial q_j}\right] \\ &= \boldsymbol{D}\, [G(\boldsymbol{q})]^{-1} \boldsymbol{p} \end{aligned} \tag{4.21}$$

The matrix product $\boldsymbol{D}^{\mathrm{T}} \boldsymbol{D}$ is equal to the metric tensor $G(\boldsymbol{q})$ which is shown in

the following

$$\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} = \begin{pmatrix} \frac{\partial x_1}{\partial q_1} & \frac{\partial x_2}{\partial q_1} & \cdots & \frac{\partial x_n}{\partial q_1} \\ \frac{\partial x_1}{\partial q_2} & \frac{\partial x_2}{\partial q_2} & \cdots & \frac{\partial x_n}{\partial q_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial x_1}{\partial q_d} & \frac{\partial x_2}{\partial q_d} & \cdots & \frac{\partial x_n}{\partial q_d} \end{pmatrix} \begin{pmatrix} \frac{\partial x_1}{\partial q_1} & \frac{\partial x_1}{\partial q_2} & \cdots & \frac{\partial x_1}{\partial q_d} \\ \frac{\partial x_2}{\partial q_1} & \frac{\partial x_2}{\partial q_2} & \cdots & \frac{\partial x_2}{\partial q_d} \\ \vdots & \vdots & & \vdots \\ \frac{\partial x_n}{\partial q_1} & \frac{\partial x_n}{\partial q_2} & \cdots & \frac{\partial x_n}{\partial q_d} \end{pmatrix}$$

$$\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} = \begin{pmatrix} \sum_{l=1}^{n} \frac{\partial x_l}{\partial q_1}\frac{\partial x_l}{\partial q_1} & \sum_{l=1}^{n} \frac{\partial x_l}{\partial q_1}\frac{\partial x_l}{\partial q_2} & \cdots & \sum_{l=1}^{n} \frac{\partial x_l}{\partial q_1}\frac{\partial x_l}{\partial q_d} \\ \sum_{l=1}^{n} \frac{\partial x_l}{\partial q_2}\frac{\partial x_l}{\partial q_1} & \sum_{l=1}^{n} \frac{\partial x_l}{\partial q_2}\frac{\partial x_l}{\partial q_2} & \cdots & \sum_{l=1}^{n} \frac{\partial x_l}{\partial q_2}\frac{\partial x_l}{\partial q_d} \\ \vdots & \vdots & & \vdots \\ \sum_{l=1}^{n} \frac{\partial x_l}{\partial q_d}\frac{\partial x_l}{\partial q_1} & \sum_{l=1}^{n} \frac{\partial x_l}{\partial q_d}\frac{\partial x_l}{\partial q_2} & \cdots & \sum_{l=1}^{n} \frac{\partial x_l}{\partial q_d}\frac{\partial x_l}{\partial q_d}. \end{pmatrix}$$

$$\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} = G(\boldsymbol{q}) \tag{4.22}$$

Replaccing $G(\boldsymbol{q})$ by $\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}$ in equation (4.21) we can write $\boldsymbol{v}$ in terms of $\boldsymbol{D}$ and $\boldsymbol{p}$

$$\boldsymbol{v} = \boldsymbol{D}\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}\boldsymbol{p} \tag{4.23}$$

Again as $\frac{1}{2}\,\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}$ is equal to the second term of equation (4.6), shown below, we can write down equation (4.6) in terms of embedding coordinates by plugging $\frac{1}{2}\,\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}$ instead of $\frac{1}{2}\,\boldsymbol{p}^{T}G(\boldsymbol{q})^{-1}\boldsymbol{p}$;

$$
\begin{aligned}
\frac{1}{2}\,\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} &= \frac{1}{2}\Big[\boldsymbol{D}\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}\boldsymbol{p}\Big]^{\mathrm{T}}\Big[\boldsymbol{D}\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}\boldsymbol{p}\Big] \\
&= \frac{1}{2}\Big[\boldsymbol{p}^{\mathrm{T}}\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}\boldsymbol{D}^{\mathrm{T}}\Big]\Big[\boldsymbol{D}\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}\boldsymbol{p}\Big] \\
&= \frac{1}{2}\,\boldsymbol{p}^{\mathrm{T}}\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}]\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}\boldsymbol{p} \\
&= \frac{1}{2}\,\boldsymbol{p}^{\mathrm{T}}\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}\boldsymbol{p} \\
&= \frac{1}{2}\,\boldsymbol{p}^{\mathrm{T}}\left[G(\boldsymbol{q})\right]^{-1}\boldsymbol{p} \qquad \big[\text{using equation (4.22)}\big]
\end{aligned}
$$

and

$$
H(\boldsymbol{x},\boldsymbol{v}) = -\log \pi_{\mathcal{H}}(\boldsymbol{x}) + \frac{1}{2}\,\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}. \tag{4.24}
$$

The target density in equation (4.24) is now written as $\pi_{\mathcal{H}}(\boldsymbol{x})$ instead of $\pi_{\mathcal{H}}(\boldsymbol{q})$ because under isometric embedding the surface measure of sphere $\mathcal{H}^d$ is invariant (as length is preserved under isometric embedding). The solution to $H^{[1]}$ in (4.15) can be rewritten in the context of equation (4.24) i.e. in terms of embedding coordinates. Like $\boldsymbol{q}(t)$, $\boldsymbol{x}(t)$ remains constant i.e. $\boldsymbol{x}(t) = \boldsymbol{x}(0)$ while a linear update is made for velocity $\boldsymbol{v}(t)$. From equation (4.19), we have the change of variables of operator $\nabla_{\boldsymbol{q}} = \boldsymbol{D}^{\mathrm{T}}\nabla_{\boldsymbol{x}}$, then taking into account this effect along with equation (4.23) we can update $\boldsymbol{v}(t)$ linearly as follows:

$$
\boldsymbol{v}(t) = \boldsymbol{v}(0) + \mathrm{t}\ \underbrace{\boldsymbol{D}\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}\boldsymbol{D}^{\mathrm{T}}}\ \nabla_{\boldsymbol{x}} \log \pi_{\mathcal{H}}(\boldsymbol{x}). \tag{4.25}
$$

The underlined linear quantity $\boldsymbol{D}\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}\boldsymbol{D}^{\mathrm{T}}$ in equation (4.25) is known as the 'hat matrix' in ordinary linear regression (OLS) literature, which is used for an orthogonal projection of arbitrary vector onto the column space of $\boldsymbol{D}$ (the

span of the columns of $\boldsymbol{D}$). In the context of our problem, we can also consider this an orthogonal projection onto the tangent space to the embedded manifold.

Although the OLS method is available to compute this projection matrix, this method can be prone to numerical instability at the boundary of the coordinate system, plus there is a burden of computational cost, especially in high dimensional case. Fortunately, we can completely avoid this OLS method to compute this orthogonal projection as the closed form of orthogonal projection is calculated for all the manifolds (Spheres) that are considered in this project, using their known explicit form of orthonormal basis, $\boldsymbol{N}$, of normal to the tangent space. The dimension of orthonormal basis $\boldsymbol{N}$ depends on what type of manifold it is. Then the orthogonal projection, $\perp_{\mathcal{T}_x}$, onto the tangent space to the embedded manifold at $\boldsymbol{x}$ is given by the simple formula

$$\perp_{\mathcal{T}_x} = \boldsymbol{I} - \boldsymbol{N}(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x})^{\mathrm{T}}. \tag{4.26}$$

Therefore we can rewrite equation (4.25) using $\perp_{\mathcal{T}_x}$.

$$\boldsymbol{v}(t) = \boldsymbol{v}(0) + \mathrm{t}\left(\boldsymbol{I} - \boldsymbol{N}(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x})^{\mathrm{T}}\right)\nabla_{\boldsymbol{x}} \log \pi_{\mathcal{H}}(\boldsymbol{x}). \tag{4.27}$$

From (4.23), we have $\boldsymbol{v} \sim \mathcal{MVN}\big(\boldsymbol{0},\, \boldsymbol{D}\left[\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\right]^{-1}\boldsymbol{D}^{\mathrm{T}}\big) = \mathcal{MVN}\big(\boldsymbol{0},\, \boldsymbol{I} - \boldsymbol{N}(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x})^{\mathrm{T}}\big)$ as $\boldsymbol{P} \sim \mathcal{MVN}\big(\boldsymbol{0},\, \boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}\big)$. Therefore, we can draw initial velocity $\boldsymbol{v}_0$ from $\mathcal{MVN}\big(\boldsymbol{0},\, \boldsymbol{I} - \boldsymbol{N}(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x})^{\mathrm{T}}\big)$, and we don't need any Cholesky decomposition as $\big(\boldsymbol{I} - \boldsymbol{N}(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x})^{\mathrm{T}}\big)$ is an idempotent matrix. All we need is to draw a sample $\boldsymbol{z}$ from $\mathcal{MVN}\big(\boldsymbol{0},\, \boldsymbol{I}\big)$ and project $\boldsymbol{v}_0 = \big(\boldsymbol{I} - \boldsymbol{N}(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x})^{\mathrm{T}}\big)\boldsymbol{z}$ to get the necessary sample. The above discussion regarding geodesic Monte Carlo is presented as a general algorithm in Algorithm 4. To implement the geodesic Monte Carlo on an embedded manifold $\mathcal{M} \subseteq \mathbb{R}^n$, the following three ingredients are needed to compute at each $\boldsymbol{x} \in \mathcal{M}$:

- the target log density $\log \pi_{\mathcal{H}}(\boldsymbol{x})$, and its gradients;
- the orthogonal projection from $\mathbb{R}^{\mathrm{n}}$ onto the tangent space $\mathcal{T}_{\boldsymbol{x}}\,\mathcal{M}$;
- the geodesic flow associated with any initial velocity $\boldsymbol{v} \in \mathcal{T}_{\boldsymbol{x}}\,\mathcal{M}$.

In this thesis, we consider only sphere (manifold) for which the form of orthonormal basis $\boldsymbol{N}(\boldsymbol{x})$ is $\boldsymbol{x}$ itself at $\boldsymbol{x}$, so the form of equation (4.26) becomes

$$\perp_{\mathcal{T}_x} = \boldsymbol{I} - \boldsymbol{x}\,\boldsymbol{x}^{\mathrm{T}}. \tag{4.28}$$

A standard result in differential geometry is that the geodesics of the n sphere are rotation about the origin, known as great circles. The geodesics flows are then (Byrne and Girolami (2013))

$$\begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{v}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}(0)\cos(\alpha t) + \alpha^{-1}\boldsymbol{v}(0)\sin(\alpha t) \\ -\alpha \boldsymbol{x}(0)\sin(\alpha t) + \boldsymbol{v}(0)\cos(\alpha t) \end{bmatrix}, \tag{4.29}$$

where $\boldsymbol{x}(0) \in \mathbb{S}^n$ denotes the initial position, $\boldsymbol{v}(0)$ denotes the initial velocity in the tangent space and $\alpha = \|\boldsymbol{v}(0)\|$ is the constant angular velocity.

4.3 Illustrative Example: Implementing Geodesic Monte Carlo

In this section we show how the geodesic Monte Carlo is implemented on an embedded manifold, and for this we consider the following *von Mises* distribution defined on the unit circle with respect to Lebesgue measure by

$$\pi(\theta) \propto \exp\{\kappa \cos(\theta - \mu)\},$$

Algorithm 6: Geodesic Monte Carlo on Embedded Manifold

Data: Initial value $\boldsymbol{x}$ where $\boldsymbol{x} \sim \pi_{\mathcal{H}}(\boldsymbol{x})$ defined on manifold $\mathbb{M}$.

Result: Simulated value from the target embedded density $\pi_{\mathcal{H}}(\boldsymbol{x})$.

begin

1: $\boldsymbol{v} \sim \mathcal{MVN}\big(\boldsymbol{0}, \boldsymbol{I}\big)$

2: $\boldsymbol{v} \leftarrow \boldsymbol{v} - \boldsymbol{N}(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{v}$

3: $h \leftarrow -\log \pi_{\mathcal{H}}(\boldsymbol{x}) + \frac{1}{2}\,\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}$

4: $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$

5: **for** *j = 1, 2, ⋯, W* **do**

5a: $\boldsymbol{v} \leftarrow \boldsymbol{v} + \frac{\epsilon}{2}\,\nabla_{\boldsymbol{x}} \log \pi_{\mathcal{H}}(\boldsymbol{x})$

5b: $\boldsymbol{v} \leftarrow \boldsymbol{v} - \boldsymbol{N}(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{v}$

5c: Update $(\boldsymbol{x}, \boldsymbol{v})$ along the geodesic flow for a ϵ time by $(\boldsymbol{x}^*, \boldsymbol{v})$

5d: Set $\boldsymbol{x} = \boldsymbol{x}^*$

5e: $\boldsymbol{v} \leftarrow \boldsymbol{v} + \frac{\epsilon}{2}\,\nabla_{\boldsymbol{x}} \log \pi_{\mathcal{H}}(\boldsymbol{x})$

5f: $\boldsymbol{v} \leftarrow \boldsymbol{v} - \boldsymbol{N}(\boldsymbol{x})\boldsymbol{N}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{v}$

6: $h^* \leftarrow -\log \pi_{\mathcal{H}}(\boldsymbol{x}) + \frac{1}{2}\,\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}$

7: **begin**

7a: Start accept reject steps

7b: Draw $U \sim \text{Uniform}\big[0, 1\big]$

7c: **if** $U < \exp(h^* - h)$ **then**

$\boldsymbol{x}^{\text{new}} \leftarrow \boldsymbol{x}^{\star}$

else

$\boldsymbol{x}^{\text{new}} \leftarrow \boldsymbol{x}_0$

7d: Stop accept reject steps

where $\theta \in [0, 2\pi)$, $\kappa \geq 0$ and $\|\mu\| = 1$. Under an embedding transformation, $\boldsymbol{x} = (\cos\theta, \sin\theta)^{\mathrm{T}}$, the above density with respect to one dimensional Hausdorff measure becomes

$$\begin{aligned}
\pi_{\mathcal{H}}(\boldsymbol{x}) &\propto \exp\{\kappa\cos\theta\cos\mu + \kappa\sin\theta\sin\mu\} \\
&= \exp\{(\kappa\cos\mu, \kappa\sin\mu)\,(\cos\theta, \sin\theta)^{\mathrm{T}}\} \\
&= \exp\{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}\}, \qquad \left[\text{where } \boldsymbol{c} = (\kappa\cos\mu, \kappa\sin\mu)^{\mathrm{T}}\right]
\end{aligned}$$

since the metric-based volume element, $\sqrt{|G(\boldsymbol{\theta})|}$, is unity as follows: Using equation (4.22), we can write

$$G(\theta) = \boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} = \begin{pmatrix} \dfrac{\partial x_1}{\partial\theta} & \dfrac{\partial x_2}{\partial\theta} \end{pmatrix} \begin{pmatrix} \dfrac{\partial x_1}{\partial\theta} \\ \dfrac{\partial x_2}{\partial\theta} \end{pmatrix}$$

$$G(\theta) = \boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} = \begin{pmatrix} -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix} = \sin^2\theta + \cos^2\theta = 1.$$

Now we need to calculate $\log\pi_{\mathcal{H}}(\boldsymbol{x})$ and its gradient which are $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ and $\boldsymbol{c}$ respectively. Then according to equation (4.28), orthogonal projection onto the tangent space to the embedded manifold at $\boldsymbol{x}$ can be calculated as

$$\perp_{\mathcal{T}_x} = \boldsymbol{I} - \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}.$$

Finally we need to have geodesic flow in hand before running Algorithm 4 which can be calculated according to equation (4.29) as

$$\left[\boldsymbol{x}(t), \boldsymbol{v}(t)\right] = \left[\boldsymbol{x}(0)\cos(\alpha\,t) + \frac{\boldsymbol{v}(0)\sin(\alpha\,t)}{\alpha},\ -\alpha\,\boldsymbol{x}(0)\sin(\alpha\,t) + \boldsymbol{v}(0)\cos(\alpha\,t)\right]$$

4.4 Simulating A and $\boldsymbol{\theta}$ using Geodesic approach

Before discussing the sampling procedure for $\boldsymbol{A}$ and $\boldsymbol{\theta}$ using geodesic approach, firstly we briefly discuss some of the common distributions which arise in Directional Statistics. Consider $\mathbb{SO}(3) = \left\{\boldsymbol{A} \in \mathbb{R}^{3x3} : \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \mathrm{I}_{3\times 3}, |\boldsymbol{A}| = 1\right\}$ the space of all 3×3 rotation matrices. A parametric distribution on this space is matrix Fisher distribution, denotes as mF, which has the following density function

$$\begin{aligned}\pi(\boldsymbol{A};\boldsymbol{F}) &= \left\{{}_0F_1(\frac{d}{2};\frac{1}{4}\boldsymbol{F}^T\boldsymbol{F})\right\}^{-1} \exp\left\{\mathrm{tr}\,(\boldsymbol{F}^T\boldsymbol{A})\right\} \\ &= \mathrm{C}_{\mathrm{mF}} \exp\left\{\mathrm{tr}\,(\boldsymbol{F}^T\boldsymbol{A})\right\}, \end{aligned} \tag{4.30}$$

with respect to $[\mathrm{d}\boldsymbol{A}]$, geometric measure scaled to unity. Here C_{mF} is the parameter-dependent normalizing constant which involves the hypergeometric function ${}_0F_1$, and $\boldsymbol{F}$ denotes the 3×3 parameter matrix. The parameter matrix $\boldsymbol{F}$ can be expressed as the following singular value decomposition form to understand the concentration properties of this distribution

$$\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{V}^T, \tag{4.31}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are the rotation matrices and the elements of diagonal matrix $\boldsymbol{\Delta} = \mathrm{diag}\,(\delta_1, \delta_2, \delta_3)$ satisfy $\delta_1 \geq \delta_2 \geq |\delta_3|$. Under the above parameterization (4.30) can be written as

$$\begin{aligned}\pi(\boldsymbol{A};\boldsymbol{U},\boldsymbol{\Delta},\boldsymbol{V}) &\propto \exp\left\{\mathrm{tr}\{\boldsymbol{V}\boldsymbol{\Delta}^T\boldsymbol{U}^T\boldsymbol{A}\}\right\} \\ &= \exp\left\{\mathrm{tr}\{\boldsymbol{\Delta}^T\boldsymbol{U}^T\boldsymbol{A}\,\boldsymbol{V}\}\right\} \\ &= \exp\left\{\mathrm{tr}\{\boldsymbol{\Delta}^T\boldsymbol{A}^\star\}\right\} \end{aligned} \tag{4.32}$$

where $\boldsymbol{A}^{\star} = \boldsymbol{U}^{T}\boldsymbol{A}\,\boldsymbol{V} \in \mathbb{SO}(3) \sim \mathrm{mF}(\boldsymbol{\Delta})$ as uniform measure is invariant under the rotation transformation. If $\boldsymbol{U}$ and $\boldsymbol{V}$ both are chosen as 3×3 identity matrices then $\pi(\boldsymbol{A}; \boldsymbol{F})$ becomes $\pi(\boldsymbol{A}; \boldsymbol{\Delta})$.

A 3×3 rotation matrix $\boldsymbol{A}$ can be obtained from an unsigned vector $\pm\boldsymbol{x}$ which lies on unit sphere $\mathbb{S}^3$ embedded in $\mathbb{R}^4$ by applying the following quadratic mapping, $\boldsymbol{A} = \mu(\pm\boldsymbol{x})$, (Wood (1993), Mardia and Jupp (2000))

$$\boldsymbol{A} = \begin{bmatrix} x_1^2 + x_2^2 - x_3^2 - x_4^2 & -2(x_1x_4 - x_2x_3) & 2(x_1x_3 + x_2x_4) \\ 2(x_1x_4 + x_2x_3) & x_1^2 + x_3^2 - x_2^2 - x_4^2 & -2(x_1x_2 - x_3x_4) \\ -2(x_1x_3 - x_2x_4) & 2(x_1x_2 + x_3x_4) & x_1^2 + x_4^2 - x_2^2 - x_3^2 \end{bmatrix} \tag{4.33}$$

To model such an unsigned direction $(\pm\boldsymbol{x})$ it is helpful to use the Bingham distribution (Bingham; 1974; Mardia and Jupp; 2000) which has the following density function with respect to the uniform measure on $\mathbb{S}^{\mathrm{q}-1}$

$$\pi(\boldsymbol{x}; \boldsymbol{B}) = \frac{\exp(-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{x})}{\mathrm{c}(\boldsymbol{B})}, \qquad \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1 \,\text{and}\, \boldsymbol{x} \in \mathbb{R}^{\mathrm{q}}, \tag{4.34}$$

where $\boldsymbol{B}$ is a $q \times q$ parameter matrix and $c(\boldsymbol{B})$ is a parameter-dependent normalizing constant. If the Bingham parameter $\boldsymbol{B}$ is expressed as singular value decomposition form $\boldsymbol{B} = \boldsymbol{\Gamma}\boldsymbol{\Lambda}\boldsymbol{\Gamma}^{\mathrm{T}}$, where $\boldsymbol{\Gamma}$ is orthogonal matrix and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots \lambda_q)$, then Kume and Walker (2006) showed that if $\boldsymbol{x}$ is drawn from $f(\boldsymbol{x}; \boldsymbol{\Lambda})$, the transformation $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Gamma}$ eventually produces a sample from $f(\boldsymbol{x}; \boldsymbol{B})$. So, without loss of generality, we consider $\boldsymbol{B} = \boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots \lambda_q)$. Therefore, the above density becomes

$$\pi(\boldsymbol{x}; \boldsymbol{\Lambda}) = \frac{\exp\left\{-\sum_{i=1}^{q} \lambda_i x_i^2\right\}}{\mathrm{c}(\boldsymbol{\Lambda})}, \qquad \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1 \,\text{and}\, \boldsymbol{x} \in \mathbb{R}^{\mathrm{q}}. \tag{4.35}$$

As $f(\boldsymbol{x}; \boldsymbol{\Lambda})$ and $\pi(\boldsymbol{x}; \boldsymbol{\Lambda} + l)$ give the same Bingham distribution for any $l \in \mathbb{R}$

because of constraint $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1$, we assume $\lambda_1 = 0 \leq \lambda_2 \leq \cdots \leq \lambda_q$ to ensure identifiability. In the context of our problem the value of q is 4. Furthermore, Prentice(1986) has shown that $\boldsymbol{A} = \mu(\pm\boldsymbol{x})$ follows a matrix Fisher distribution if and only if the unsigned unit vector $\pm\boldsymbol{x}$ in $\mathbb{R}^4$ follows a Bingham distribution, and the relationship between the parameterization of these two distributions is also established. For a given $\boldsymbol{\Delta} = \mathrm{diag}\,(\delta_1, \delta_2, \delta_3)$, there exist scalar $l \in \mathbb{R}$ and $\lambda_1 = 0 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4$ such that

$$\begin{aligned}\mathrm{tr}(\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{A}) &= -\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{\Lambda} + l)\boldsymbol{x} \\ &= -\Big[lx_1^2 + (l+\lambda_2)x_2^2 + (l+\lambda_3)x_3^2 + (l+\lambda_4)x_4^2\Big] \end{aligned} \tag{4.36}$$

If (4.36) holds then the matrix Fisher distribution with parameter matrix $\boldsymbol{\Delta}$ is equivalent to Bingham distribution with parameter matrix $\boldsymbol{\Lambda}$. Now using equation (4.33) the left hand of equation (4.36) can be expressed as

$$\begin{aligned}\mathrm{tr}(\boldsymbol{\Delta}^{\mathrm{T}}A) &= \delta_1(x_1^2 + x_2^2 - x_3^2 - x_4^2) + \delta_2(x_1^2 + x_3^2 - x_2^2 - x_4^2) + \delta_3(x_1^2 + x_4^2 - x_2^2 - x_3^2) \\ &= (\delta_1 + \delta_2 + \delta_3)x_1^2 + (\delta_1 - \delta_2 - \delta_3)x_2^2 + (-\delta_1 + \delta_2 - \delta_3)x_3^2 + (-\delta_1 - \delta_2 + \delta_3)x_4^2 \end{aligned} \tag{4.37}$$

Comparing equations (4.37) and (4.38) we can finally establish the relationship between the parameters of these distributions

$$\lambda_2 = 2(\delta_2 + \delta_3), \qquad \lambda_3 = 2(\delta_1 + \delta_3), \qquad \lambda_4 = 2(\delta_1 + \delta_2). \tag{4.38}$$

4.4.1 Simulation Techniques

To simulate samples from the matrix Fisher distribution for a given parameter matrix $\boldsymbol{F}$, currently there are some good methods available in the existing literature. For example, Kent and Ganeiber (2012) developed a method to generate

a Bingham sample in $\mathbb{R}^4$ using Angular Central Gaussian (ACG) distribution as an envelope within a rejection framework, and then transform this to a matrix Fisher sample according to the relation between matrix Fisher and Bingham distributions (discussed in the previous section). In the context of our problem, we need to simulate from the matrix Fisher distribution, the full conditional distribution of $\boldsymbol{A}$, which depends not only on $\boldsymbol{F}$ but also on $\boldsymbol{\theta}$, $\boldsymbol{\tau}$ and σ. In other words sampling rotation correctly is influenced by other variables as well, especially by matching probabilities $\boldsymbol{\theta}$. This is why we are interested to develop a new approach which takes into account this issue to simulate rotation instead of using existing method directly. We can tackle this issue by developing a sampling scheme which allows us to sample both rotation ($\boldsymbol{A}$) and matching probabilities ($\boldsymbol{\theta}$) together (not in Gibbs style), and geodesic Monte Carlo within product manifold framework gives us such opportunity (see the results of the simulation study).

We have seen in Chapter 3 that the full conditional distributions both for rotation ($\boldsymbol{A}$) and matching probabilities ($\boldsymbol{\theta}$) are defined on manifolds namely $\mathbb{SO}(3)$ and $\prod_{i=1}^{m} \triangle_i{}^n$ respectively, which allows us to define geodesic Monte Carlo for sampling these two together under the product of their constituent manifold (product manifold). Sampling rotation and matching probability separately based on $\mathbb{SO}(3)$ and $\prod_{i=1}^{m} \triangle_i{}^n$ respectively using geodesic Monte Carlo are the building blocks for defining geodesic Monte Carlo on their product manifold. Therefore, in the following two sections we discuss geodesic Monte Carlo technique to simulate from matrix Fisher distribution and simplex separately before implementing geodesic technique to simulate from their joint density based on the product manifold.

4.4.2 Simulation from matrix Fisher distribution using Geodesic Monte Carlo Approach

To simulate from the matrix Fisher distribution using geodesic Monte Carlo, we use here Kent and Ganeiber (2012) method but in a slightly modified way.

From the literature review, we have already seen that there is an another distribution defined on $\mathbb{S}^{(q-1)}$ proposed by Paine, Preston, Tsagris and Wood (2017) called the elliptically symmetric angular Gaussian distribution (ESAG). This ESAG distribution is a subfamily of the angular Gaussian distribution closely analogous to the Kent subfamily of the general Fisher–Bingham distribution. Like the Kent distribution it also offers modelling rotational asymmetry about the mean direction because of its ellipse-like contours. Morover, it has some additional advantages compared to the Kent model. For the ESAG distribution there exists a very simple and fast simulation technique which does not require rejection methods like the Fisher–Bingham family. In addition to this, the likelihood of this distribution can be computed quickly and exactly as the density is free of awkward normalising constants. Because of these advantages modelling rotation using the ESAG is beneficial for computationally intensive statistical methods. However, we are currently not using this distribution to modeling rotation but we have a plan to use this distribution in future and compare its performance over the other methods currently used in this thesis.

Kent and Ganeiber's Modified Approach

We already have discussed the link between the matrix Fisher distribution on $\mathbb{SO}(3)$ and the Bingham distribution on $\mathbb{S}^3$ embedded in $\mathbb{R}^4$ in section 4.1. Recall that Kent and Ganeiber (2012) used this relationship to sample from the matrix Fisher distribution, using Bingham samples generated via ACG as an envelope distribution within a rejection framework. To simulate from the matrix Fisher distribution using geodesic Monte Carlo approach we use the same idea of Kent and Ganeiber (2012), but Bingham samples are generated using geodesic Monte Carlo approach instead of rejection sampling. Therefore, the task of simulating from the matrix Fisher distribution using the geodesic Monte Carlo approach simply boils down to simulating from Bingham samples using geodesic Monte

Carlo approach. The idea is that, in some situations, this could produce good proposals when used in conjunction with updates to other parameters, rather than directly sampling from the full conditional in a Gibbs sampler (see the results of the simulation study in section 5.1).

As the Bingham distribution defined in (4.35) is on $\mathbb{S}^3$ embedded in $\mathbb{R}^4$, we can consider it our target embedded density from which we want to draw samples and it can be written as

$$\pi_{\mathcal{H}}(\boldsymbol{x}) \propto \exp\big\{ -\boldsymbol{x}^{\mathrm{T}} \boldsymbol{\Lambda}\, \boldsymbol{x} \big\}, \qquad \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1 \,\text{and}\, \boldsymbol{x} \in \mathbb{R}^4. \tag{4.39}$$

Before running Algorithm 4 to draw a Bingham sample, we need to have in our hand $\log \pi_{\mathcal{H}}(\boldsymbol{x})$ and its gradient which are $-\boldsymbol{x}^{\mathrm{T}} \boldsymbol{\Lambda}\, \boldsymbol{x}$ and $-2\, \boldsymbol{\Lambda}\, \boldsymbol{x}$ respectively. In addition, we also need to have an orthogonal projection onto the tangent space to the embedded manifold at $\boldsymbol{x}$ and geodesic flow which can be calculated by equations (4.28) and (4.29) respectively. Finally, we can summarize the whole simulation procedure of simulating a matrix Fisher sample as follows:

1. For a general $\boldsymbol{F}$, decompose $\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{V}^T$ using equation (4.31);
2. Calculate $\boldsymbol{\Lambda} = \mathrm{diag}\,(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ according to equation (4.38);
3. Run Algorithm 6 to draw Bingham sample, $\boldsymbol{x}$, for parameter $\boldsymbol{\Lambda}$;
4. Calculate the corresponding rotation matrix $\boldsymbol{A} = \mu(\boldsymbol{x})$ using (4.33);
5. Calculate $\boldsymbol{U}\boldsymbol{A}\boldsymbol{V}^T$ which returns a matrix Fisher sample for parameter $\boldsymbol{F}$.

Illustrative Example

Suppose we want to draw a sample from the matrix Fisher distribution for the following parameter matrix

$$\boldsymbol{F} = \begin{pmatrix} 1.1 & 0.2 & 4.1 \\ 3.1 & 0.5 & 0.7 \\ 8.1 & 1.1 & 2.5 \end{pmatrix}.$$

In order to draw a sample first express $\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{V}^T$ in the following singular value decomposition form

$$\boldsymbol{F} = \begin{pmatrix} -0.27 & 0.95 & -0.06 \\ -0.33 & -0.15 & -0.92 \\ -0.90 & -0.23 & 0.36 \end{pmatrix} \begin{pmatrix} 9.44 & 0 & 0 \\ 0 & 3.50 & 0 \\ 0 & 0 & 0.07 \end{pmatrix} \begin{pmatrix} -0.91 & -0.38 & -0.13 \\ -0.12 & -0.04 & 0.99 \\ -0.38 & 0.92 & -0.01 \end{pmatrix}.$$

In above $\det(\boldsymbol{U}) = +1$ and $\det(\boldsymbol{V}) = -1$, so we flip the sign of the last column of $\boldsymbol{V}$ matrix to make it a rotation matrix. In addition we also need to multiply $\delta_3 = 0.07$ by (-1) as $\det(\boldsymbol{F}) < 0$. We don't need this type of adjustment when $\det(\boldsymbol{F}) > 0$. Finally the above form can be written as

$$\boldsymbol{F} = \begin{pmatrix} -0.27 & 0.95 & -0.06 \\ -0.33 & -0.15 & -0.92 \\ -0.90 & -0.23 & 0.36 \end{pmatrix} \begin{pmatrix} 9.44 & 0 & 0 \\ 0 & 3.50 & 0 \\ 0 & 0 & -0.07 \end{pmatrix} \begin{pmatrix} -0.91 & -0.38 & 0.13 \\ -0.12 & -0.04 & -0.99 \\ 0.38 & -0.92 & 0.01 \end{pmatrix}.$$

Using the diagonal element of $\boldsymbol{\Delta} = \text{diag}\,(9.44, 3.50, 0.07)$, calculate the diagonal elements of the Bingham parameter $\boldsymbol{\Lambda} = \text{diag}\,(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ where $\lambda_1 = 0$, $\lambda_2 = 6.86$, $\lambda_3 = 18.74$ and $\lambda_3 = 25.88$.

Suppose we have $\boldsymbol{x} = (-0.514, 0.799, 0.214, 0.222)$ after running Algorithm 6 which is the Bingham sample corresponding to $\boldsymbol{\Lambda}$. Now transform this Bingham sample $\boldsymbol{x}$, using equation (4.33), to give the following rotation matrix

$$\mu(\boldsymbol{x}) = \begin{pmatrix} 0.808 & 0.572 & 0.134 \\ 0.114 & -0.377 & 0.918 \\ 0.576 & -0.727 & -0.371 \end{pmatrix}.$$

Finally, calculate the quantity $\boldsymbol{U}\mu(\boldsymbol{x})\boldsymbol{V}^T$ which gives the following matrix Fisher sample corresponding to $\boldsymbol{F}$

$$\begin{pmatrix} 0.530 & -0.725 & 0.439 \\ 0.743 & 0.149 & -0.651 \\ 0.406 & 0.672 & 0.618 \end{pmatrix}.$$

4.4.3 Simulation from Simplex using Geodesic Monte Carlo Approach

Before discussing the simulation technique from the simplex using geodesic Monte Carlo, firstly we briefly discuss the simplex. A n dimensional probability simplex is a $(n+1)$ dimensional euclidean surface which a distribution over the $(n+1)$ events lies on. Each component of any random point in a simplex is non-negative, and all $(n+1)$ components sum to be 1. Mathematically we can express a n dimensional probability simplex, $\triangle^n$, as

$$\triangle^n = \left\{ \boldsymbol{\theta} \in \mathbb{R}^{n+1} : \theta_k \geq 0, \sum_{k=1}^{n+1} \theta_k = 1 \right\}, \tag{4.40}$$

which is a n dimensional manifold embedded in $\mathbb{R}^{n+1}$. For example, a Dirichlet distribution with respect to Lebesgue measure, considered a natural prior distribution for $(n+1)$ dimensional multinomial data, defined on n dimensional

simplex is:

$$\pi(\boldsymbol{\theta};\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\prod_{k=1}^{(n+1)}\theta_k^{\alpha_k-1} \qquad 0 \leq \theta_k \leq 1 \quad \text{and} \quad \sum_{k=1}^{n+1}\theta_k = 1, \tag{4.41}$$

where $\alpha_k \in \mathbb{R}^+$ and B is the Beta function. Many real world phenomena, especially analyzing compositional data, can be modeled by using Dirichlet distribution because of its constrained support. However, this constrained support not only makes it widely applicable in many situations but also hinders practical implementations, especially those applying MCMC techniques (Betancourt, 2013). For example, a sample from the Dirichlet distribution proposed by a Markov chain can fall outside of the simplex with probability 1 if the constraint is not taken into account by a Markov chain. As a result, the chain will not be able to move beyond this sample point. Taking into account this constraint means parameterizing the simplex directly which helps the Markov chain to explore the full support of Dirichlet distribution but real difficulties start when the boundary of the simplex is considered. For example, integration paths ($\mathrm{W} \times \epsilon$) of geodesic Monte Carlo on a simplex directly may violate the positivity constraints, which needs reflection of this path to satisfy constraints, and frequent reflections requires additional computational cost.

Byrne et. al (2013) proposed a square root transformation from n dimensional simplex $\triangle^n$ to the positive orthant of the sphere $\mathbb{S}^n$ to simplify the structure of simplex, which is more amenable as far as efficient MCMC is concerned. As MCMC proposals can fall outside the positive orthant of the sphere, so distribution defined only on positive orthant needs to be extended to the whole sphere to tackle this issue. Extending this distribution to the whole sphere can be done by reflecting the axes which makes the original distribution $2^{(n+1)}$ times larger, but we can ignore this quantity as we only need to know the distribution up to proportionality. The main advantage of this transformation is that the surface

is now smooth and without boundaries, so proposal outside the positive orthant is not the issue anymore. This motivates us to use this transformation to sample from Dirichlet distribution (simplex) instead of parameterizing the simplex directly. The element-wise square root transformation $y_k = \sqrt{\theta_k}$ with the following form of partial derivatives

$$\frac{\partial y_l}{\partial \theta_k} = \begin{cases} \frac{1}{2}\,\theta_k^{-1/2} & \text{if } k = l < (n+1), \\ 0 & \text{if } k \neq l < (n+1), \\ \frac{1}{2}\big(1 - \sum_{k=1}^{n} \theta_k\big)^{-1/2} & \text{if } k = l = (n+1), \end{cases}$$

yields the n dimensional Jacobian

$$\frac{1}{2^n} \prod_{k=1}^{(n+1)} \theta_k^{-1/2}. \tag{4.42}$$

Now under square root map the Dirichlet distribution defined in (4.41) can be written as

$$\begin{aligned} \pi_{\mathcal{H}}(\boldsymbol{y}) &= \frac{\prod_{k=1}^{(n+1)} (y_k^2)^{\alpha_k - 1}}{\mathrm{B}(\boldsymbol{\alpha})} \times \frac{2^n}{\prod_{k=1}^{(n+1)} (y_k^2)^{-1/2}} \\ &= \frac{\prod_{k=1}^{(n+1)} (y_k)^{2\alpha_k - 2}}{\mathrm{B}(\boldsymbol{\alpha})} \times \frac{2^n}{\prod_{k=1}^{(n+1)} (y_k)^{-1}} \\ &= \frac{2^n}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{k=1}^{(n+1)} (y_k)^{2\alpha_k - 1}. \end{aligned} \tag{4.43}$$

Distribution defined in (4.43) becomes uniform when α_k =1/2, whereas α_k =1 gives the uniform distribution for (4.41). Taking the log of both sides of equation (4.43) gives us $\log \pi_{\mathcal{H}}(\boldsymbol{y})$ which is

$$\log \pi_{\mathcal{H}}(\boldsymbol{y}) = \sum_{k=1}^{(n+1)} (2\alpha_k - 1) \log y_k + \text{constant}. \tag{4.44}$$

Finally, from equation (4.44), the $(n+1)$ dimensional vector $\nabla_{\boldsymbol{y}} \log \pi_{\mathcal{H}}(\boldsymbol{y})$ can be written as

$$\nabla_{\boldsymbol{y}} \log \pi_{\mathcal{H}}(\boldsymbol{y}) = \left\{ \frac{(2\alpha_1 - 1)}{y_1}, \frac{(2\alpha_2 - 1)}{y_2}, \cdots, \frac{(2\alpha_{(n+1)} - 1)}{y_{(n+1)}} \right\}. \tag{4.45}$$

Now we have everything in our hand to run Algorithm 4 for $\pi_{\mathcal{H}}(\boldsymbol{y})$ defined in (4.43) to generate a sample from simplex, $\triangle^n$, defined in (4.41), which is illustrated in the following example.

Illustrative Example

Suppose we want to draw a sample from a Dirichlet distribution with parameter $\boldsymbol{\alpha} = (5, 7, 11)$, which is on unit 2-simplex $\triangle^2$. In other words, we want to draw a sample from the following probability distribution

$$\pi(\boldsymbol{\theta}; \boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{k=1}^{3} \theta_k^{\alpha_k - 1} \qquad 0 \leq \theta_k \leq 1 \ \text{ and } \ \sum_{k=1}^{3} \theta_k = 1, \tag{4.46}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3) = (5, 7, 11)$. Consider we have an initial Dirichlet sample $\boldsymbol{\theta} = (1/3, 1/3, 1/3)$ on the simplex $\triangle^2$. In general, we can choose this initial value of $\boldsymbol{\theta}$ by drawing a sample of size 3 from uniform distribution and then normalize it.

According to the discussion in section 4.4.3, instead of using directly this initial $\boldsymbol{\theta} = (1/3, 1/3, 1/3)$ value on simplex $\triangle^2$ we use $\boldsymbol{y} = \sqrt{\boldsymbol{\theta}} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$ which is on the sphere $\mathbb{S}^2$, to propose a new $\boldsymbol{\theta}$ value via a new $\boldsymbol{y}$. Now using equation (4.43), the corresponding distribution of equation (4.46) due to the square root transformation is

$$\pi_{\mathcal{H}}(\boldsymbol{y}) = \frac{2^2}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{k=1}^{3} (y_k)^{2\alpha_k - 1}. \tag{4.47}$$

Suppose running Algorithm 6 with initial $\boldsymbol{y} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$ to generate a sample from $\pi_{\mathcal{H}}(\boldsymbol{y})$ defined in (4.47) gives us a new $\boldsymbol{y} = (0.498, 0.582, 0.641)$ value. Transforming back this $\boldsymbol{y} = (0.498, 0.582, 0.641)$ to $\boldsymbol{\theta} = (0.249, 0.340, 0.411)$, using $\boldsymbol{y}^2 = \boldsymbol{\theta}$, gives us the required sample from equation (4.46). On the other hand, if the Algorithm 4 returns the value which is same as initial $\boldsymbol{y}$ then $\boldsymbol{\theta}$ new will be the same as its initial value. The whole procedure needs to be repeated r number of times to obtain a sample of size r.

4.5 Geodesic Monte Carlo on Product Manifold

We have already seen in sections 4.4.2 and 4.4.3 how geodesic Monte Carlo is implemented to generate samples separately from matrix Fisher and Dirichlet distributions which are defined on $\mathbb{SO}(3)$ and $\triangle^n$ respectively. In this section, we will discuss a geodesic Monte Carlo algorithm to generate samples jointly from a matrix Fisher and m Dirichlet distributions of dimension $(n+1)$ (product of a matrix Fisher and m Dirichlet distributions) defined on $\mathbb{SO}(3) \times \underbrace{\triangle_1^n \times \triangle_2^n \cdots \times \triangle_m^n}$ using product manifold.

Let's briefly discuss here product manifolds before explaining the algorithm of geodesic Monte Carlo on a product manifold. A compound topological space composed by a set of constituent manifolds (topological space) is called a product manifold. In other words, suppose if we have s manifolds $\mathbb{M}_1, \mathbb{M}_2, \cdots, \mathbb{M}_s$, then their Cartesian product $\mathbb{M}_1 \times \mathbb{M}_2 \times \cdots \times \mathbb{M}_s$ is also a manifold, called product manifold, as product manifold topology is equivalent to the constituent manifold topology. Mathematically, we can define the product manifold as

$$\begin{aligned}\mathbb{M} &= \mathbb{M}_1 \times \mathbb{M}_2 \times \cdots \times \mathbb{M}_s \\ &= \big\{(x_1, x_2, \cdots, x_s) : x_1 \in \mathbb{M}_1, x_2 \in \mathbb{M}_2, \cdots, x_s \in \mathbb{M}_s\big\},\end{aligned}$$

where $\times$ represents the Cartesian product and x_s is an element of the constituent manifold $\mathbb{M}_s$. Product manifolds play a vital role in statistics as many statistical problems can be defined on product manifold. For example, Singh et al., (2002) used the product of two von Mises distributions defined on $\mathbb{S} \times \mathbb{S}$ to model molecular angles. As an another example, we have seen in the previous chapter that the joint density of $\boldsymbol{A}$ and $\boldsymbol{\theta}$ is the product of a matrix Fisher and m independent Dirichlet distributions which are defined on the product manifold $\mathbb{SO}(3) \times \underbrace{\triangle_1^n \times \triangle_2^n \cdots \times \triangle_m^n}$.

The product of geodesics in $\mathbb{M}_1, \mathbb{M}_2, \cdots, \mathbb{M}_s$ form the geodesic in product manifold (Lui, Y. M. (1998) and Byrne et al., (2013)), and are of the form $(\boldsymbol{\delta_1}, \boldsymbol{\delta_2}, \cdots, \boldsymbol{\delta_s}) = \boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is the geodesic on a product manifold and $\boldsymbol{\delta}_s$ is the geodesic on the constituent manifold $\mathbb{M}_s$. Similarly, $(\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_s) = \boldsymbol{v}$ represents the tangent vector where $\boldsymbol{v}$ is the tangent vector to a product manifold and $\boldsymbol{v}_s$ is the tangent to the constituent manifold $\mathbb{M}_s$. Therefore, the form of orthogonal projection onto the tangent space for an arbitrary vector $(\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_s) = \boldsymbol{u}$ is $\left\{\left(\boldsymbol{I} - \boldsymbol{N}_1 \boldsymbol{N}_1{}^{\mathrm{T}}\right)\boldsymbol{u}_1, \left(\boldsymbol{I} - \boldsymbol{N}_2 \boldsymbol{N}_2{}^{\mathrm{T}}\right)\boldsymbol{u}_2, \cdots \left(\boldsymbol{I} - \boldsymbol{N}_s \boldsymbol{N}_s{}^{\mathrm{T}}\right)\boldsymbol{u}_s\right\} = \left(\boldsymbol{I} - \boldsymbol{N}\boldsymbol{N}^{\mathrm{T}}\right)\boldsymbol{u}$ where $\boldsymbol{N}_s$ is the orthogonal basis of constituent manifold $\mathbb{M}_s$ and $\boldsymbol{N}$ is the orthogonal basis of product manifold $\mathbb{M}$.

Given s embedded densities $\pi^1_{\mathcal{H}}(\boldsymbol{x}_1), \pi^2_{\mathcal{H}}(\boldsymbol{x}_2), \cdots, \pi^s_{\mathcal{H}}(\boldsymbol{x}_s)$ defined on s constituent manifolds $\mathbb{M}_1, \mathbb{M}_2, \cdots, \mathbb{M}_s$ respectively, embedded density $\pi_{\mathcal{H}}(\boldsymbol{x})$ on product manifold $\mathbb{M}$ is simply the product of these s embedded densities, and can be written as

$$\pi_{\mathcal{H}}(\boldsymbol{x}) = \pi^1_{\mathcal{H}}(\boldsymbol{x}_1) \times \pi^2_{\mathcal{H}}(\boldsymbol{x}_2) \times \cdots \times \pi^s_{\mathcal{H}}(\boldsymbol{x}_s), \tag{4.48}$$

where $\pi^s_{\mathcal{H}}(\boldsymbol{x}_s)$ is the embedded density of the s^{th} constituent manifold. Taking the log of both sides of equation (4.48) gives us $\log \pi_{\mathcal{H}}(\boldsymbol{x})$ which is

$$\log \pi_{\mathcal{H}}(\boldsymbol{x}) = \log \pi^1_{\mathcal{H}}(\boldsymbol{x}_1) + \cdots + \log \pi^s_{\mathcal{H}}(\boldsymbol{x}_s), \tag{4.49}$$

and differentiation of equation (4.49) gives the s dimensional vector $\nabla_{\boldsymbol{x}} \log \pi_{\mathcal{H}}(\boldsymbol{x})$ which can be written as

$$\nabla_{\boldsymbol{x}} \log \pi_{\mathcal{H}}(\boldsymbol{x}) = \left\{ \nabla_{\boldsymbol{x}_1} \log \pi_{\mathcal{H}}(\boldsymbol{x}), \nabla_{\boldsymbol{x}_2} \log \pi_{\mathcal{H}}(\boldsymbol{x}), \cdots, \nabla_{\boldsymbol{x}_s} \log \pi_{\mathcal{H}}(\boldsymbol{x}) \right\}, \tag{4.50}$$

where $\nabla_{\boldsymbol{x}_s} \log \pi_{\mathcal{H}}(\boldsymbol{x})$ is the derivative of $\log \pi_{\mathcal{H}}(\boldsymbol{x})$ with respect to the element of the s^{th} constituent manifold. Orthogonal projection and geodesic update on a product manifold can be determined by calculating the orthogonal projection and geodesic update of each constituent manifold at the same time in parallel. Finally, to tune the whole algorithm, it is often useful to choose different ϵ values for each of the constituent manifolds, especially when elements in different manifolds are not on the same scale (Byrne et al., 2013). In the following, we summarize the detailed procedure of geodesic Monte Carlo implementation in Algorithm 7. In the light of our problem, the corresponding equation of equation (4.48) is

$$\pi_{\mathcal{H}}(\boldsymbol{A}, \boldsymbol{\theta}) \propto \exp\left\{ \operatorname{tr}\left(\boldsymbol{F}^T \boldsymbol{A}\right) \right\} \times \prod_{k=1}^{(n+1)} \left[\theta_{1k}\right]^{\alpha_{1k}-1} \times \ldots \times \prod_{k=1}^{(n+1)} \left[\theta_{mk}\right]^{\alpha_{mk}-1}, \tag{4.51}$$

where $\boldsymbol{A} \in \mathbb{SO}(3)$, $\boldsymbol{\theta}_i \in \triangle^n, i = 1, 2, \cdots, m$ and $\alpha_{ik} = (\gamma_{ik} + \eta_{ik})$. Now applying the relationship between $\mathbb{SO}(3)$ and $\mathbb{S}^3$ along with square root transformation from simplex $(\triangle^n)$ to sphere $\mathbb{S}^n$ on (4.51), we can rewrite the joint density $\pi_{\mathcal{H}}(\boldsymbol{A}, \boldsymbol{\theta})$ in terms of $\boldsymbol{x}$ and $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m)$ as follows, which is defined on $\mathbb{S}^3 \times \underbrace{\mathbb{S}^n_1 \times \mathbb{S}^n_2 \cdots \times \mathbb{S}^n_m}$

$$\pi_{\mathcal{H}}(\boldsymbol{x}, \boldsymbol{y}) \propto \exp\left\{ -\boldsymbol{x}^{\mathrm{T}} \Lambda \, \boldsymbol{x} \right\} \times \prod_{k=1}^{(n+1)} \left[y_{1k}\right]^{2\alpha_{1k}-1} \times \ldots \times \prod_{k=1}^{(n+1)} \left[y_{mk}\right]^{2\alpha_{mk}-1}, \tag{4.52}$$

Algorithm 7: Geodesic Monte Carlo on Product Manifold

Data: Initial value $\boldsymbol{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_s)$ where $\boldsymbol{x}_i \sim \pi^i_{\mathcal{H}}(\boldsymbol{x}_i)$ defined on manifold $\mathbb{M}_i$.

Result: Sample from $\pi_{\mathcal{H}}(\boldsymbol{x}) = \pi^1_{\mathcal{H}}(\boldsymbol{x}_1) \times \cdots \times \pi^s_{\mathcal{H}}(\boldsymbol{x}_s)$ defined on $\in \mathbb{M}_1 \times \cdots \times \mathbb{M}_s$.

begin

1 **for** *i = 1, 2, ⋯, s* **do**

1a: $\boldsymbol{v}_i \sim \mathcal{MVN}\big(\boldsymbol{0}, \boldsymbol{I}\big)$

1b: $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i - \boldsymbol{N}_i(\boldsymbol{x}_i)\boldsymbol{N}_i(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{v}_i$

2: $h \leftarrow -\log \pi_{\mathcal{H}}(\boldsymbol{x}) + \frac{1}{2}\sum_{i=1}^{s} \boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{v}_i$

3: $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$

4: **for** *i = 1, 2, ⋯, s* **do**

5: **for** *j = 1, 2, ⋯, W* **do**

5a: $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \frac{\epsilon_i}{2}\nabla_{\boldsymbol{x}_i} \log \pi_{\mathcal{H}}(\boldsymbol{x}_i)$

5b: $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i - \boldsymbol{N}_i(\boldsymbol{x}_i)\boldsymbol{N}_i(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{v}_i$

5c: Update $(\boldsymbol{x_i}, \boldsymbol{v})$ along the geodesic flow for a ϵ time by $(\boldsymbol{x}^*_i, \boldsymbol{v})$

5d: Set $\boldsymbol{x}_i = \boldsymbol{x}^*_{\boldsymbol{i}}$

5e: $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \frac{\epsilon_i}{2}\nabla_{\boldsymbol{x}_i} \log \pi_{\mathcal{H}}(\boldsymbol{x}_i)$

5f: $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i - \boldsymbol{N}_i(\boldsymbol{x}_i)\boldsymbol{N}_i(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{v}_i$

6: $h^* \leftarrow -\log \pi_{\mathcal{H}}(\boldsymbol{x}) + \frac{1}{2}\sum_{i=1}^{s} \boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{v}_i$

7: **begin**

7a: Start accept reject steps

7b: Draw $U \sim \mathrm{Uniform}\big[0, 1\big]$

7c: **if** $U < \exp(h^* - h)$ **then**

$\boldsymbol{x}^{\mathrm{new}} \leftarrow \boldsymbol{x}^{\star}$

else

$\boldsymbol{x}^{\mathrm{new}} \leftarrow \boldsymbol{x}_0$

7d: Stop accept reject steps

where $\boldsymbol{x}$ follows a Bingham distribution with parameter $\boldsymbol{\Lambda}$, corresponding to $\mathrm{mF}(\boldsymbol{\Delta})$, in $\mathbb{S}^3$, and distribution of $\boldsymbol{y}_{ik}, i = 1, 2, \cdots, m$, defined on $\mathbb{S}_i^n$. The motivation behind redefining joint density (4.51) to (4.52) is to efficiently sample $\boldsymbol{A}$ and $\boldsymbol{\theta}$ together (not in Gibbs style) from the joint density $\pi_{\mathcal{H}}(\boldsymbol{A}, \boldsymbol{\theta})$ using geodesic Monte Carlo approach. Taking log of both sides of equation (4.52) gives us $\log \pi_{\mathcal{H}}(\boldsymbol{x}, \boldsymbol{y})$ which is

$$\log \pi_{\mathcal{H}}(\boldsymbol{x}, \boldsymbol{y}) = -\boldsymbol{x}^{\mathrm{T}} \Lambda \, \boldsymbol{x} + \sum_{k=1}^{(n+1)} \big[2\alpha_{1k} - 1\big] \log y_{1k} + \cdots + \sum_{k=1}^{(n+1)} \big[2\alpha_{mk} - 1\big] \log y_{mk} \tag{4.53}$$

and differentiation of equation (4.53) w.r.t. $\boldsymbol{x}^* = (\boldsymbol{x}, \boldsymbol{y})$ gives the $s = (m + 1)$ dimensional vector $\nabla_{\boldsymbol{x}^*} \log \pi_{\mathcal{H}}(\boldsymbol{x}, \boldsymbol{y})$, which can be written as

$$\begin{aligned} \nabla_{\boldsymbol{x}^*} \log \pi_{\mathcal{H}}(\boldsymbol{x}^*) &= \Big\{ \nabla_{\boldsymbol{x}} \log \pi_{\mathcal{H}}(\boldsymbol{x}^*), \nabla_{\boldsymbol{y}_1} \log \pi_{\mathcal{H}}(\boldsymbol{x}^*), \cdots, \nabla_{\boldsymbol{y}_m} \log \pi_{\mathcal{H}}(\boldsymbol{x}^*) \Big\} \\ &= \Big\{ -2A\boldsymbol{x}, \frac{2\boldsymbol{\alpha}_{1k} - 1}{\boldsymbol{y}_{1k}}, \frac{2\boldsymbol{\alpha}_{2k} - 1}{\boldsymbol{y}_{2k}} \cdots, \frac{2\boldsymbol{\alpha}_{mk} - 1}{\boldsymbol{y}_{mk}} \Big\} \end{aligned} \tag{4.54}$$

Now applying the Algorithm 7 on (4.52) gives us a sample in terms of $\boldsymbol{x}$ and $\boldsymbol{y}$. Once we have a sample from the joint density defined in (4.52), we need to transform it back using the above relations to get a sample from the joint density defined in (4.51).

4.6 Tuning Geodesic Monte Carlo on a product manifold

Before discussing tuning the geodesic Monte Carlo algorithm for sampling from probability distribution defined on manifold or on a product manifold, firstly we

will discuss here in detail how to tune standard HMC when used for sampling from probability distribution defined on euclidean space. We will then use this idea when we tune the geodesic Monte Carlo algorithm.

4.6.1 Tuning HMC Algorithm

The efficiency of HMC algorithm depends on the correct choices of ϵ and W values for a particular problem, and the wrong choices of ϵ and W values can give completely misleading results. Therefore, choosing ϵ and W values correctly is very crucial to run a HMC algorithm successfully for a particular problem.

To illustrate how to choose ϵ and W values correctly for a particular problem, we consider here exactly the same example that was considered by Neal (2012). Suppose we wish to sample from the bi-variate Gaussian distribution $\boldsymbol{q} \sim \mathcal{MVN}(\mathbf{0}, \boldsymbol{\Sigma})$ using HMC, where elements of diagonal and off-diagonal of $\boldsymbol{\Sigma}$ are 1 and 0.95 respectively. To facilitate the sampling procedure, we consider an auxiliary variable $\boldsymbol{p}$ which is $\mathcal{MVN}(\mathbf{0}, \boldsymbol{I})$. Now we can write down the following Hamiltonian function using these two distributions $\pi(\boldsymbol{q})$ and $\pi(\boldsymbol{p})$ (see subsection 4.1.2 for how this is obtained):

$$\begin{aligned} H(\boldsymbol{q}, \boldsymbol{p}) &= U(\boldsymbol{q}) + K(\boldsymbol{p}) \\ &= \log \pi(\boldsymbol{q}) + \log \pi(\boldsymbol{p}) \\ &= \frac{1}{2}\, \boldsymbol{q}^T \Sigma^{-1} \boldsymbol{q} + \frac{1}{2}\, \boldsymbol{p}^T \boldsymbol{p}. \end{aligned}$$

To set ϵ and W values for a particular problem, Neal (2012) suggested the following guidelines:

- Choose ϵ and W in such a way that step length, $\epsilon \times W$, is sufficient to move to a distant point in the distribution.
- For chosen values of ϵ and W, calculate the Hamiltonian values H^0 at initial $(\boldsymbol{q}_0, \boldsymbol{p}_0)$ and $H^\star$ at every point after each Leapfrog step.

- Plot the Hamiltonian energy H^0 or the difference of two energies known as error, $(H^\star - H^0)$, and observe the pattern of this Hamiltonian energy or the error.

- The pattern of H^0 or $(H^\star - H^0)$ should be oscillatory (roughly periodic) if sample is simulated correctly from the target distribution with some error.

- The values of H^0 and $(H^\star - H^0)$ will be constant and zero respectively if sample is simulated from target distribution without error, which is the ideal situation.

- Higher values of H^0 and $(H^\star - H^0)$ indicate that larger ϵ value were used which needs careful attention. In that case, we need to reduce the ϵ value by keeping step length same as before, and hence expect to have lower H^0 or $(H^\star - H^0)$ value. Simulated values come from the target distribution with some error as long as H^0 or $(H^\star - H^0)$ value is bounded (periodic).

- If the value of H^0 or $(H^\star - H^0)$ grows unboundedly which indicates that we need to reset the values of ϵ and W.

For the above Hamiltonian function, Leapfrog integrator becomes Figure 4.1

Algorithm 8: Algorithm of Leapfrog Method

Data: $\boldsymbol{q}$ and $\boldsymbol{p}$ values at time t.

Result: $\boldsymbol{q}$ and $\boldsymbol{p}$ values at time $(t + \epsilon)$.

begin

1: Update $\boldsymbol{p}$ over $\epsilon/2$ time step using $\boldsymbol{p}(t + \epsilon/2) = \boldsymbol{p}(t) - (\epsilon/2)\left[\boldsymbol{\Sigma}^{-1}\boldsymbol{q}\right]$

2: Update $\boldsymbol{q}$ over ϵ time step using $\boldsymbol{q}(t + \epsilon) = \boldsymbol{q}(t) + \epsilon\left[\boldsymbol{p}\right]$

3: Update $\boldsymbol{p}$ for remaining half step using

$\boldsymbol{p}(t + \epsilon) = \boldsymbol{p}(t + \epsilon/2) - \epsilon/2\left[\boldsymbol{\Sigma}^{-1}\boldsymbol{q}\right]$

shows the plots of Hamiltonian energy and error for the above Hamiltonian function for different ϵ and W values. First row of this figure is plotted for $\epsilon = 0.25$

and $W = 25$ while $\epsilon = 0.125$ and $W = 50$ and $\epsilon = 0.45$ and $W = 25$ are considered for the last two rows respectively. From the first row, we have seen that maximum error is 0.45, and this error is reduced to 0.084 in the second row when ϵ is considered $\epsilon/2$ and W=50. Hamiltonian value (error) grows unboundedly when $\epsilon = 0.45$ regardless of number of Leapfrog steps. As long as step size ϵ is less than 0.45 Hamiltonian values (error) stays bounded, which guarantees samples obtained using this ϵ and W values will be the correct sample from our targeted bi-variate Gaussian distribution. This is the way one can tune the HMC algorithm, which can be extended for Hamiltonian defined on manifold.

4.6.2 Tuning Geodesic Monte Carlo Algorithm for sampling from matrix Fisher distribution

The idea of tuning geodesic Monte Carlo is the same as HMC tuning but geodesic integrator is used to discretize time here instead of Leapfrog integrator (discussed in section 4.2). In the context of our problem, we have used the geodesic Monte Carlo algorithm to sample from the joint full conditional distribution of rotation and matching probabilities defined on a product manifold, $\mathbb{SO}(3) \times \underbrace{\triangle_1^n \times \triangle_2^n \cdots \times \triangle_m^n}$, under our proposed model. Before discussing tuning geodesic Monte Carlo on a product manifold, we will discuss here tuning geodesic Monte Carlo algorithm defined on a single manifold. For example, in our case geodesic Monte Carlo algorithm can be used to obtain good proposals for rotation and matching probabilities, which is on $\mathbb{SO}(3)$, and matching probabilities θ_j, $j = 1, 2, \cdots, m$ which is on $\triangle$.

We have already explained in section (4.4) how a sample is drawn from the matrix Fisher distribution via Bingham distribution using geodesic Monte Carlo algorithm. Byrne and Girolami's algorithm can generate samples from the matrix Fisher distribution via Bingham distribution for any concentration parameter $\boldsymbol{F}$, but consideration of different concentration levels of $\boldsymbol{F}$ requires extensive tuning, and this situation occurs in our case as at each iteration the matrix Fisher

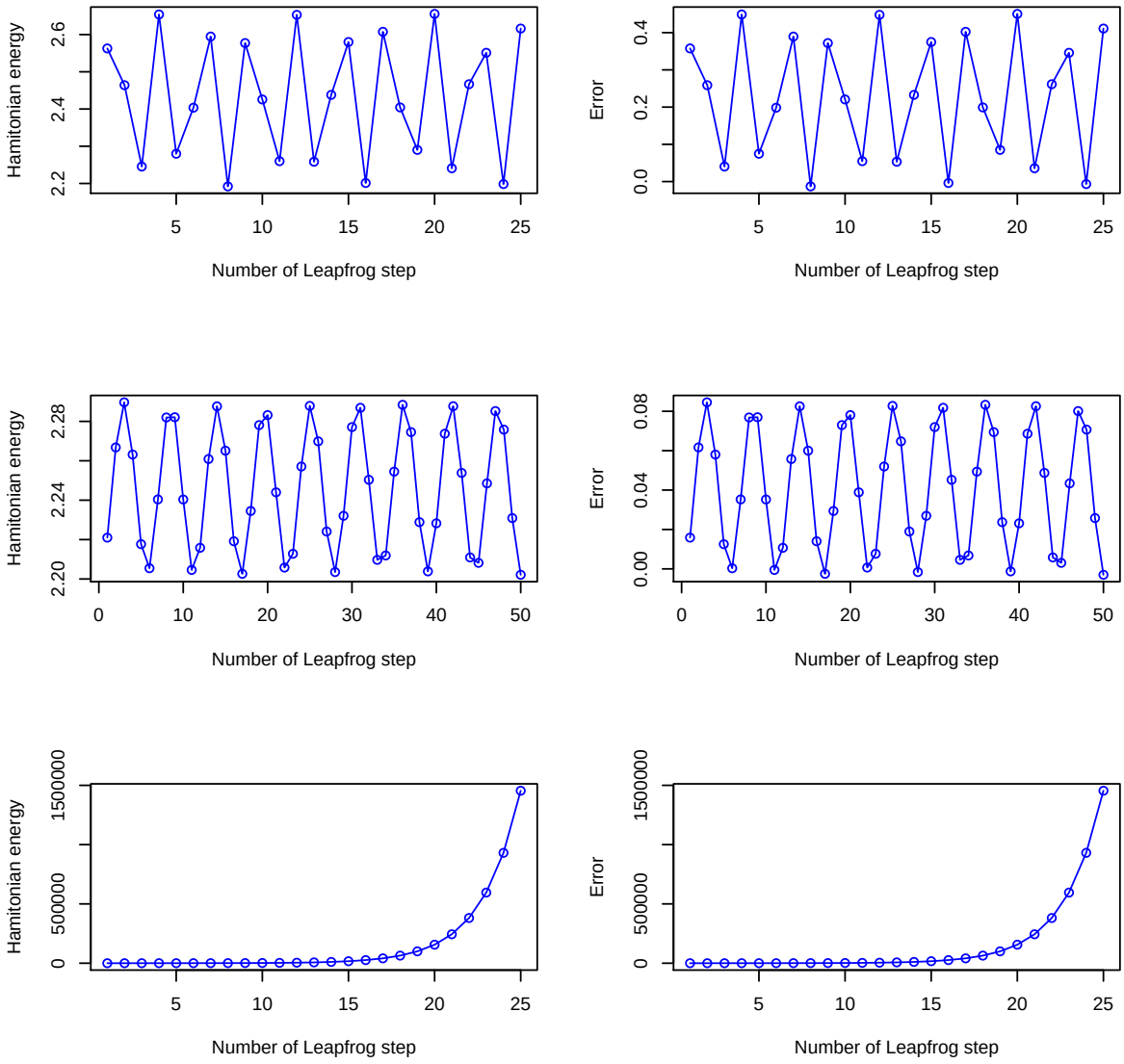

Figure 4.1: Illustration of choosing tuning parameters ϵ and W of HMC algorithm used for sampling from probability distribution defined on euclidean space.

parameter $\boldsymbol{F}$ is updated and takes a new value.

For a given matrix Fisher distribution, mF($\boldsymbol{F}$), with parameter matrix $\boldsymbol{F}$, we reparametrize $\boldsymbol{F}$ in terms of concentration parameter $\boldsymbol{\Delta}$, mF($\boldsymbol{\Delta}$), using equations (4.30), (4.31) and (4.32). Now using equations (4.33), (4.34) and (4.35), we have the corresponding Bingham distribution embedded in $\mathbb{R}^4$ of mF($\boldsymbol{\Delta}$). We will use this embedding density to sample from Bingham distribution, and then transform this sample to matrix Fisher sample. To tune geodesic Monte Carlo algorithm to sample from Bingham distribution, we followed the same guideline like HMC tuning but points 1 to 12 of Algorithm 4 are applied to produce H^0 or $(H^\star - H^0)$.

To have the idea about what type of matrix Fisher parameter $\boldsymbol{F}(\boldsymbol{\Delta})$ geodesic Monte Carlo will need to sample from the matrix Fisher distribution for every iteration, we have collected different updated $\boldsymbol{F}(\boldsymbol{\Delta})$ for every 50^{th} iteration from the output of MCMC sampler of Green and Mardia (2006) model for protein data. For each and every $\boldsymbol{F}(\boldsymbol{\Delta})$, we then write down the corresponding embedded Bingham distribution. Now based on simulation study, visualization of Hamiltonian energy plot or error corresponding to Bingham distribution, we have constructed the following table for ϵ and W for a set of $\boldsymbol{\Delta}$ =diag$(\delta_1, \delta_2, \delta_3)$ values. For any updated $\boldsymbol{F}(\boldsymbol{\Delta})$, using the following rules for setting ϵ and W guarantees the lack of infinitely growing Hamiltonian value or error. From the above table, we have seen that with the increasing of value of $\sum_{i=1}^{3} \delta_i$, the value of ϵ is decreasing. In addition, we have also investigated that weight proportion of $\delta_i, i = 1, 2, 3$, does not affect the above rules but only sum of δ_i.This tables tells how one should tune geodesic Monte Carlo algorithm (setting ϵ and W values) to simulate sample from matrix Fisher distribution with parameter $\boldsymbol{F}(\boldsymbol{\Delta})$ using geodesic Monte Carlo approach. For a given $\boldsymbol{\Delta}$, one just needs to determine in which row of the above table $\sum_{i=1}^{3} \delta_i$ belongs, and choose corresponding ϵ and W values of this row as their respective values. Suppose we wish to sample from a matrix Fisher distribution with parameter $\boldsymbol{F}$, mF($\boldsymbol{F}$), using geodesic Monte

Carlo algorithm where

$$\boldsymbol{F} = \begin{pmatrix} 8.5 & 7.8 & 4.3 \\ 1.1 & 3.9 & 6.4 \\ 4.1 & 6.0 & 4.8 \end{pmatrix},$$

and its singular value decomposition, $\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{V}^T$, has the following form:

$$\boldsymbol{F} = \begin{pmatrix} -0.74 & 0.54 & -0.38 \\ -0.39 & -0.82 & -0.40 \\ -0.53 & -0.14 & 0.83 \end{pmatrix} \begin{pmatrix} 16.16 & 0 & 0 \\ 0 & 4.80 & 0 \\ 0 & 0 & 0.56 \end{pmatrix} \begin{pmatrix} -0.55 & 0.65 & -0.51 \\ -0.65 & 0.03 & 0.75 \\ -0.51 & -0.75 & -0.40 \end{pmatrix}.$$

In above $\det(\boldsymbol{U}) = +1$ and $\det(\boldsymbol{V}) = -1$, so we flip the sign of the last column of $\boldsymbol{V}$ matrix to make it a rotation matrix. In addition we also need to multiply $\delta_3 = 0.56$ by (-1) as $\det(\boldsymbol{F}) < 0$.

$$\boldsymbol{F} = \begin{pmatrix} -0.74 & 0.54 & -0.38 \\ -0.39 & -0.82 & -0.40 \\ -0.53 & -0.14 & 0.83 \end{pmatrix} \begin{pmatrix} 16.16 & 0 & 0 \\ 0 & 4.80 & 0 \\ 0 & 0 & -0.56 \end{pmatrix} \begin{pmatrix} -0.55 & 0.65 & 0.51 \\ -0.65 & 0.03 & -0.75 \\ -0.51 & -0.75 & 0.40 \end{pmatrix}.$$

As $\sum_{i=1}^{3} \delta_i = 20.40 < 60$, so we use $\epsilon = 0.01$ and W=40 in geodesic Monte Carlo algorithm when sample is drawn from Bingham distribution (corresponding to mF($\boldsymbol{\Delta}$)). The top plot of Figure 4.2 shows the error plot of Hamiltonian function corresponding to embedded Bingham distribution for $\epsilon = 0.01$ and W=40 while second plot shows the log density of the above matrix Fisher distribution (Bingham samples are converted to matrix Fisher samples) for the simulated values. The horizontal line indicates the maximum value of this density, which is 20.4. For the same problem, if we use $\epsilon = 0.007$ and W=40, then sample will also be from the same distribution but less representative compared to the sample obtained under $\epsilon = 0.01$ and W=40, as error plot is higher than previous one.

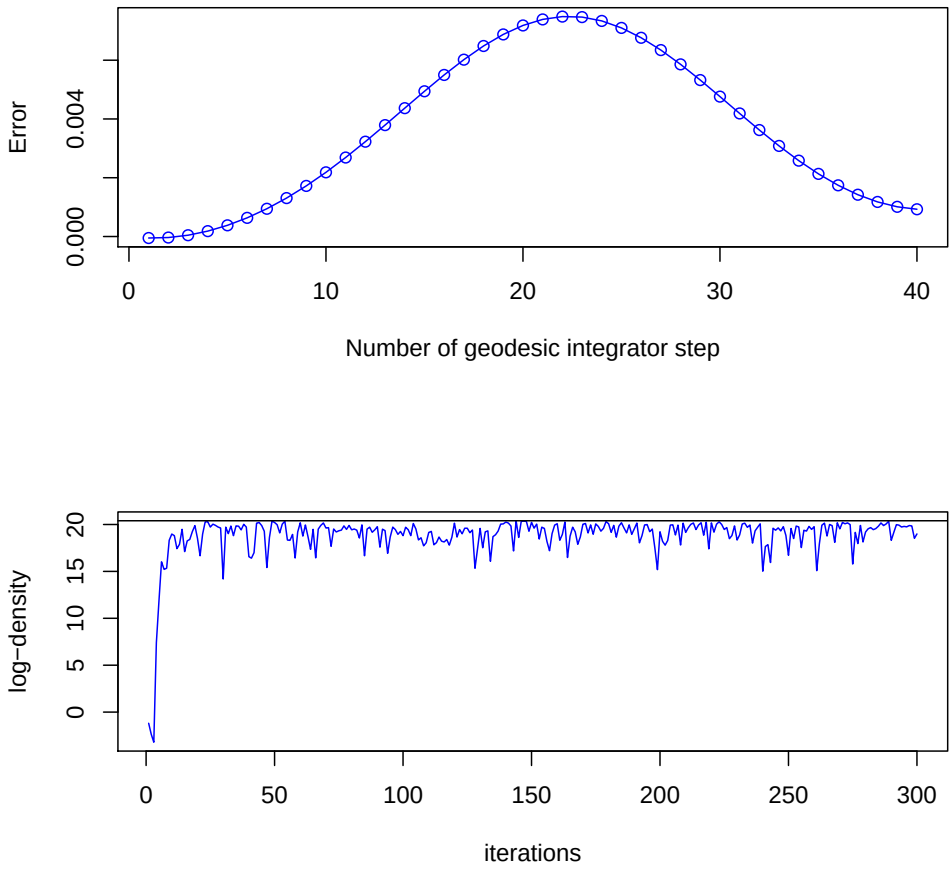

Figure 4.2: Illustration of choosing tuning parameters ϵ and W of geodesic Monte Carlo algorithm when used to generate sample from matrix Fisher distribution.

4.6.3 Tuning Geodesic Monte Carlo Algorithm for sampling from the simplex

The tuning idea is the same as tuning geodesic Monte Carlo algorithm for mF($\boldsymbol{\Delta}$). In the context of our problem, full conditional distribution of matching probabilities of $\boldsymbol{\theta}$ is the product of m independent $(n+1)$ dimensional Dirichlet distribution, which are in fact product of m simplex $\triangle$, each of dimension n. Recall that the equation defined in Chapter 3 is

$$\pi(\boldsymbol{\theta} \mid \gamma(j), \boldsymbol{A}, \boldsymbol{\tau}, \sigma\, \boldsymbol{X}, \boldsymbol{Y}) \propto \left(\prod_{k=1}^{(n+1)} \left[\theta_{1k}\right]^{\gamma_{1k}+\eta_{1k}-1} \right) \times \ldots \times \left(\prod_{k=1}^{(n+1)} \left[\theta_{mk}\right]^{\gamma_{mk}+\eta_{mk}-1} \right).$$

Now using equations (4.41), (4.42) and (4.43) (square root transformation from simplex to sphere), we have the following density of j^{th} simplex on the sphere (from $\boldsymbol{\theta}_j$ to $\boldsymbol{y}_j$)

$$\pi_{\mathcal{H}}(\boldsymbol{y}_j) = \prod_{k=1}^{(n+1)} \left[\boldsymbol{y}_{jk}\right]^{2(\gamma_{jk}+\eta_{jk})-1}, \tag{4.55}$$

which is embedded in $\mathbb{R}^{(n+1)}$. The parameter, $2 \times (\gamma_{jk} + \eta_{jk})$, of this embedded density takes either value $2 \times \frac{(n+2)}{(n+1)}$ or $\frac{2}{(n+1)}$ at every iteration in our MCMC sampler as $\gamma_{jk} \in \{0, 1\}$ and $\eta_{jk} = \frac{1}{n+1}$. Considering $\epsilon = .0001$ and $W = 40$ ensures lack of growth for the Hamiltonian value for the above embedded density defined in (4.55) in the context of our problem.

Now we discuss here how we have tuned the geodesic Monte Carlo algorithm when we sampled $\boldsymbol{A}$ and $\boldsymbol{\theta}$ together based on a product manifold, $\mathbb{SO}(3) \times \underbrace{\triangle_1^n \times \triangle_2^n \cdots \times \triangle_m^n}$. Byrne et al. (2013) suggested to choose different ϵ values for each constituent manifold when geodesic Monte Carlo on a product manifold is tuned, which can be helpful when variables have different scales of variation. In the context of our problem, rotation A and matching probabilities $\boldsymbol{\theta}$ have

different kind of variation, hence different ϵ values are chosen for $\mathbb{SO}(3)$ and $\triangle$. We have used Table 4.1 for $\mathbb{SO}(3)$ and $\epsilon = .0001$ for all $\triangle_i$ (as all simplex has same scale of variation).

Table 4.1: Setting ϵ and W based on simulation study

Condition	Values of ϵ and W
If $\sum_{i=1}^{3} \delta_i \leq 60$	$\epsilon = .01$ and W =40
If $\sum_{i=1}^{3} \delta_i > 60$ and $\sum_{i=1}^{3} \delta_i \leq 130$	$\epsilon = .007$ and W =40
If $\sum_{i=1}^{3} \delta_i > 130$ and $\sum_{i=1}^{3} \delta_i \leq 235$	$\epsilon = .005$ and W =40
If $\sum_{i=1}^{3} \delta_i > 235$ and $\sum_{i=1}^{3} \delta_i \leq 301$	$\epsilon = .004$ and W =40
If $\sum_{i=1}^{3} \delta_i > 301$ and $\sum_{i=1}^{3} \delta_i \leq 446$	$\epsilon = .0035$ and W =40
If $\sum_{i=1}^{3} \delta_i > 446$ and $\sum_{i=1}^{3} \delta_i \leq 582$	$\epsilon = .003$ and W =40
If $\sum_{i=1}^{3} \delta_i > 582$ and $\sum_{i=1}^{3} \delta_i \leq 800$	$\epsilon = .0025$ and W =40
If $\sum_{i=1}^{3} \delta_i > 800$ and $\sum_{i=1}^{3} \delta_i \leq 1300$	$\epsilon = .002$ and W =40
If $\sum_{i=1}^{3} \delta_i > 1300$ and $\sum_{i=1}^{3} \delta_i \leq 3000$	$\epsilon = .0007$ and W =40
If $\sum_{i=1}^{3} \delta_i > 3000$ and $\sum_{i=1}^{3} \delta_i \leq 4000$	$\epsilon = .0006$ and W =40
If $\sum_{i=1}^{3} \delta_i > 4000$ and $\sum_{i=1}^{3} \delta_i \leq 6000$	$\epsilon = .0005$ and W =40
If $\sum_{i=1}^{3} \delta_i > 6000$ and $\sum_{i=1}^{3} \delta_i \leq 8000$	$\epsilon = .0004$ and W =40
If $\sum_{i=1}^{3} \delta_i > 8000$ and $\sum_{i=1}^{3} \delta_i \leq 13000$	$\epsilon = .0003$ and W =40
If $\sum_{i=1}^{3} \delta_i > 13000$	$\epsilon = .0002$ and W =40

Chapter 5

Results and Discussions

In this chapter we will show the comparison of our proposed method for alternatively modelling matching matrix ($\mathcal{PMAMM}$) with the method of Green and Mardia (2006) ($\mathcal{GM}$) and its modified version, which we call $\mathcal{MVGM}$, for the unlabelled size and shape problem. The $\mathcal{MVGM}$ framework is similar to the $\mathcal{GM}$ framework but the sampling procedure for the rotation and matching matrices are different; Forbes and Lauritzen sampler is used for the matching matrix and Kent's rejection method or Habeck's method or Geodesic Monte Carlo approach is used for rotation. We use the notations $\mathcal{MGM}$ and $\mathcal{MGM}(.)$ throughout this chapter to represent the two cases under $\mathcal{MVGM}$; the former of which only updates the matching matrix but not rotation while the latter one in addition to updating matching matrix rotation is also updated. For example, $\mathcal{MGM}$(K) means the $\mathcal{MVGM}$ uses Forbes and Lauritzen sampler for matching matrix and Kent's method for rotation.

We use the same notational convention like $\mathcal{MVGM}$ for $\mathcal{PMAMM}$ as well. Like $\mathcal{MGM}$ and $\mathcal{MGM}(.)$ notation, we use the notation $\mathcal{PMAMM}^{\star}$ and $\mathcal{PMAMM}(.)$ for $\mathcal{PMAMM}$; $\mathcal{PMAMM}^{\star}$ means only matching matrix is updated in Gibbs fashion while in $\mathcal{PMAMM}(.)$ one of the rotation methods used in $\mathcal{MGM}(.)$ is used for sampling rotation along with the updates for the matching matrix. Furthermore, we use the notation $\mathcal{PMAMM}^{\star\star}$ to denote the

situation where both the rotation $\boldsymbol{A}$ and matching probabilities $\boldsymbol{\theta}$ are updated together using a geodesic Monte Carlo approach based on a product manifold.

To show these comparisons, we choose the same problems (gel data in 2 and protein data in 3 dimensions) from bioinformatics like Green and Mardia (2006) along with a simulated data set discussed in section 1.4 and apply $\mathcal{PMAMM}^{\star}$, $\mathcal{PMAMM}(.)$, $\mathcal{PMAMM}^{\star\star}$, $\mathcal{GM}$ and $\mathcal{MVGM}$ to these data sets.

We organize this this Chapter as follows: In section 5.1 we present the results of our simulation study, while analysis of gel and protein data sets are presented in section 5.2 and 5.3 respectively.

5.1 A Simulation Study

In this section we conduct a simulation study, using the artificial data set described in section 1.4, where we know what the true probabilities of matching along with the true values of translation and rotation parameters are and compare the MCMC algorithms of all methods considered in this thesis to see how they perform. For this artificial data set, we have ninty three matched and five unmatched pairs with matching matrix $\boldsymbol{M}$ whose first ninty three diagonal elements are one and last five diagonal elements are zero. The true values of rotation and translation parametrs are $\boldsymbol{A} = \boldsymbol{A}_{12}(0.5235)\,\boldsymbol{A}_{13}(1.047)\,\boldsymbol{A}_{23}(1.396)$, $\boldsymbol{\tau} = (25, 17, 8)^T$ respectively.

The hyperparameters $\boldsymbol{F}_0$, $\boldsymbol{\mu}_{\boldsymbol{\tau}}$, $\sigma_{\boldsymbol{\tau}}$, α and β are common in both the Green and Mardia (2006) and $\mathcal{PMAMM}$ models. In addition to these hyperparameters, the $\mathcal{PMAMM}$ model contains two more hyperparameters; $\boldsymbol{\eta}_k$ and $|\Omega|$ while the Green and Mardia (2006) model contains yet one more hyperparameter,

λ/ρ, by which the prior for the number of matched pairs L is parameterised. Choosing hyperparameters values is crucial as our analysis depends on their chosen settings. For default analysis, we have considered uniform prior on $\boldsymbol{A}$ by choosing $\boldsymbol{F}_0$ to the zero matrix, $\boldsymbol{\mu_\tau}$ and $\sigma_{\boldsymbol{\tau}}$ are the values of difference of two configurations centres and order of twice the distance between the centres of two configurations, which are (3.8541, -9.5701, 23.5492) and 50 respectively for this particular data set. By choosing $\boldsymbol{\eta}_k = (1, 1, \cdots, 1)$ we have also considered uniform prior on $\boldsymbol{\theta}_j$ while $|\Omega| = 21000$ is chosen accoding to the formula $max\Big[\prod_j (max_i\, x_{ij} - min_i\, x_{ij}), \prod_j (max_i\, y_{ij} - min_i\, y_{ij})\Big]$ suggested by Taylor, Mardia and Kent (2003).

Furthermore by choosing $\alpha = 1$, we have considered exponential prior distribution for σ^{-2} while various choices of β, which is the inverse of scale parameter of σ^{-2}, values are considered in this simulation study to check the sensitivity of this hyperparameter. As β is the inverse scale parameter of σ^{-2}, so with the increases of β values we expect that $\sigma^2 = \text{var}(\boldsymbol{\epsilon}_i)$ increases too. In this simulation study we have choosen three different values of β which are 18, 36 and 72 respectively. Finally, we discuss the setting off the value of hyperparameter λ/ρ. Green and Mardia (2006) outlined a guidline how to choose λ/ρ in their paper. They suggested λ/ρ equal to $(m - \bar{L})(n - \bar{L})/\bar{L}\nu$ which yields a mode of the prior distribution of L that is within 1 of $\bar{L}$. Here $\bar{L}$ is the prior for L. From their numerical investigation, it was also showed that the prior expectation and median are also both equal to $\bar{L}$ to the nearest integer. Thus knowledge about λ/ρ can be gained through directly $\bar{L}$ (prior gauess of L). It is possible to make a prior guess $\bar{L}$ as long as ν is known or a representative value is provided which altogether suggest a reasonable way of specifying λ/ρ. In this simulation study, we consider two different values of λ/ρ which are 0.00047 and 0.000047.

The Markov chain Monte Carlo samplers described in Chapters 1 and 3 were run

for 1000,000 sweeps with the settings of the above mentioned hyperparameters for this artificial data set under $\mathcal{GM}$, $\mathcal{MGM}$(K), $\mathcal{MGM}$(G), $\mathcal{PMAMM}$(K), $\mathcal{PMAMM}$(G) and $\mathcal{PMAMM}^{\star\star}$ frameworks to get the MCMC output. Performances of the samplers were evaluated based on how quickly (number of steps) the Markov chain converges to the posterior distribution, acceptance rate of proposed moves of $\boldsymbol{M}$, computing time, proposal quality and number of successful runs. We repeated our experiment 50 times with the same parameter settings but with different random starting points for $\boldsymbol{\tau}$ and $\boldsymbol{A}$, to investigate the multimodality of the posterior and compute average estimates of the above quantities.

In the following, we have discussed the results of our simulation study where $\beta = 36$ and $\lambda/\rho = 0.00047$ are considered. Simulations conducted under the settings of other combinations of β and λ/ρ values are not presented here as we have seen that the results obtained under the other settings have a very similar trend like the trend observed in the presented results. Table 5.1 presents the detail summary of our simulation study for aligning two artificial objects in size and shape space using $\mathcal{GM}$, $\mathcal{MGM}$(K), $\mathcal{MGM}$(G), $\mathcal{PMAMM}$(K), $\mathcal{PMAMM}$(G), and $\mathcal{PMAMM}^{\star\star}$ approaches. First column of Table 5.1 stands for the different approaches while the other columns stand for the mean convergence iterations, mean computing time, mean acceptance rate and percentage of successful runs respectively.

In a single iteration with a single update of $\boldsymbol{\tau}$, $\boldsymbol{A}$ and σ^2, we have considered two different number of update to $\boldsymbol{M}$ respectively in $\mathcal{GM}$ approach. Here we consider one and fifty update to $\boldsymbol{M}$ along with the single update to $\boldsymbol{\tau}$, $\boldsymbol{A}$ and σ^2 respectively. For example, considering fifty update to $\boldsymbol{M}$ in $\mathcal{GM}$ approach means $\boldsymbol{M}$ is updated fifty times along with the single update to $\boldsymbol{\tau}$, $\boldsymbol{A}$ and σ^2 in each iteration in a particular MCMC run. Simillarly we consider one $\boldsymbol{\theta}$ and fifty $\boldsymbol{\theta}$ out of ninty eight $\boldsymbol{\theta}$ to update in a single iteration along with the single

update to $\boldsymbol{\tau}$, $\boldsymbol{A}$, and σ^2 in $\mathcal{PMAMM}$(K), $\mathcal{PMAMM}$ G), and $\mathcal{PMAMM}^{\star\star}$ approaches respectively.

From the second column of Table 5.1, we can tell the mean number of iterations needs to be required to converge for each of the methods. Here we use a simple graphical technique for MCMC convergence assessment. For a particular MCMC algorithm, we run mutiple chains with different starting values, and overlying the generated values on a common graph of each parameter provided that the simulation process stabilies in some sense. When all the chains traversing the sample space in the same way provides evidence of convergence. In addition, we also plot the log posterior (up to normalizing constant) and monitor its value. By monitoring the log posterior value, we can determine either the chains converge to any other local solutions.

To illustrate this, we run 50 chains for each $\mathcal{GM}$ and $\mathcal{MGM}$(.) methods for 1000,000 sweeps to align these two artificial configurations. We then overlay the simulated values (MCMC output) on the same graph for each parameter ($\boldsymbol{\tau}$, $\boldsymbol{A}$ and L) whenever the chains are stabilized in some sense. From our inspection, we have seen that all the components of $\boldsymbol{\tau}$, Euler's angles, and number of match pairs L pass through in the same way which is a clear evidence of convergence. In this case, the components of $\boldsymbol{\tau}$, Euler's angles, and L pass through their true values when they are stabilized, and these are shown in Figure 5.2 and Figure 5.7 respectively. The same conclusion regrading the number of steps required to acheive convergence can be made based on the plot of unnormalized log posteror density which is shown in Figure 5.1. From these three figures we can see that $\mathcal{GM}$ (1 update) and $\mathcal{MGM}$(G) approaches require at least four hundred seventy eight thousand and two hundred fourty three thousand iterations as a burn in period for the safest inference procedure. We have followed this same trick to determine the convergence for $\mathcal{PMAMM}$(.) and $\mathcal{PMAMM}^{\star\star}$ approaches

which are shown in Figure 5.3 - Figure 5.7.

From the second column of Table 5.1, we can see that $\mathcal{MGM}$(G) approach needs on average 34,444 iterations to converge with 0.57 % mean acceptance rate of $\boldsymbol{M}$ while $\mathcal{GM}$ (one update) approach requires on avaerage 66,809 iterations with 0.077 % mean acceptance rate of $\boldsymbol{M}$. Considering 50 updates to $\boldsymbol{M}$ in $\mathcal{GM}$ gave quicker convergence compared to $\mathcal{MGM}$ (G) approach but acceptance rate of $\boldsymbol{M}$ reduced to between 0.0004 % and 0.23 % from between 0.08 % and 0.7 %. $\mathcal{MGM}$ (G) approach needs almost 3 times higher time compared to $\mathcal{GM}$ (1 update) approach to implement a run with 1000,000 sweeps while both the approaches require almost same amount of time when 50 updates to $\boldsymbol{M}$ is considered in $\mathcal{GM}$ approach. However, we can ignore this time comparison as it is not too large to be problematic, and no emphasis has been placed on coding as efficiently as possible.

Considering the $\mathcal{PMAMM}$ (G) approach for aligning and estimation technique yeilded a very simillar kind of result like the $\mathcal{GM}$ approach ($\mathcal{GM}$ is slightly better) as far as mean covergence iterations is concerned in both 1 update and 50 updates cases but $\mathcal{PMAMM}$ (G) requires almost 2.5 and 6 times higher computing time compared to $\mathcal{GM}$ approach for 1 update and 50 updates cases respectively to implement a run with 1000,000 sweeps. However, in the $\mathcal{PMAMM}$(G) approach there is no need to resort to Metropolis Hastings algorithm to sample matching matrix $\boldsymbol{M}$ as the full conditional distributions of matching probabilities (all are multinomial distributions) are much easier to simulate using Gibbs sampling, and hence there is no issue of acceptance rate of $\boldsymbol{M}$. Here we also ignore this time comparison as it is not too large to be problematic, and no emphasis has been placed on coding as efficiently as possible.

Moving to the $\mathcal{PMAMM}^{\star\star}$ approach from the $\mathcal{PMAMM}$(G) approach for

aligning and estimation technique where both the rotation $\boldsymbol{A}$ and the matching probabilities $\boldsymbol{\theta}$ are updated jointly using geodesic Monte Carlo method yielded quick convergence compared to all the methods discussed so far in both 1 update and 50 updates cases. More specifically, from Table 5.1, it is clear that $\mathcal{PMAMM}^{\star\star}$ requires almost 2.5 and 10 times less number of iterations compared to $\mathcal{GM}$ approach to converge for 1 update and 50 updates cases respectively. Furthermore, for the safest inference $\mathcal{PMAMM}^{\star\star}$ needs at least 125,000 and 4308 iterations while it is 478,000 and 88,000 for $\mathcal{GM}$ approach for 1 update and 50 updates cases respectively. However, $\mathcal{PMAMM}^{\star\star}$ requires almost 70 and 140 times higher computing time compared to $\mathcal{GM}$ approach for 1 update and 50 updates cases respectively.

From Table 5.1, we observed that $\mathcal{MGM}(.)$ produces quick convergence when $\boldsymbol{A}$ is updated using geodesic Monte Carlo approach compared to using rejection method. A simillar pattern can be seen in the $\mathcal{PMAMM}(.)$ approach as well although in 50 $\boldsymbol{\theta}$ updates case updating rotation using rejection method yields slightly quicker convergence. From the last column of Table 5.1, it is observed that the percentage of successful runs for $\mathcal{PMAMM}^{\star\star}$ are 28% and 40% whilst it is 22 % and 24 % in $\mathcal{GM}$ for 1 update and 50 updates cases respectively. The percentage of successful runs under $\mathcal{MGM}()$ and $\mathcal{GM}()$ are very simillar whilst highest percentage of successful can be seen in $\mathcal{PMAMM}(K)$ approach.

The quality of the MCMC outputs obtained under the above mentioned methods are assessed using the ACF plots which are shown in Figure 5.8. Here we consider the ACF of unnormalized log posterior values obtained under $\mathcal{GM}$, $\mathcal{MGM}(\mathrm{G})$, $\mathcal{PMAMM}(\mathrm{G})$, and $\mathcal{PMAMM}^{\star\star}$ methods. The reason for choosing the ACF of unnormalized log posterior values is that unnormalized log posterior is a function of unknown $\boldsymbol{\tau}$, $\boldsymbol{A}$, $\boldsymbol{M}$ and σ^2. From Figure 5.8, it is observed that lag diminishes very quickly for $\mathcal{MGM}(\mathrm{G})$, $\mathcal{PMAMM}(\mathrm{G})$, and $\mathcal{PMAMM}^{\star\star}$ methods

compared to $\mathcal{GM}$ method. More specifically, the lag coefficients are significant up to 2000 lag for $\mathcal{GM}$ while for $\mathcal{MGM}$(G), $\mathcal{PMAMM}$(G), and $\mathcal{PMAMM}^{\star\star}$ none of the lag coefficients are significant after 500 lag.

For the unknown parameters $\boldsymbol{\tau}$, $\boldsymbol{A}$, and $\boldsymbol{M}$ we obtained their point estimate $\hat{\boldsymbol{\tau}}_{\mathcal{GM}}$ $\hat{\boldsymbol{\tau}}_{\mathcal{MGM}(\mathcal{G})}$, $\hat{\boldsymbol{\tau}}_{\mathcal{PMAMM}^{\star\star}}$, $\hat{\boldsymbol{A}}_{\mathcal{GM}}$, $\hat{\boldsymbol{A}}_{\mathcal{MGM}(\mathcal{G})}$ and $\hat{\boldsymbol{A}}_{\mathcal{PMAMM}^{\star\star}}$ by maximizing the joint posterior i.e. their posterior modes. The posterior mode of $\boldsymbol{\tau}$ and $\boldsymbol{A}$ under $\mathcal{GM}$, $\mathcal{MGM}$ (G) and $\mathcal{PMAMM}^{\star\star}$ are $\hat{\boldsymbol{\tau}}_{\mathcal{GM}}$=$(24.97, 16.88, 7.80)^{\mathrm{T}}$, $\hat{\boldsymbol{\tau}}_{\mathcal{MGM}(G)}$ = $(24.92, 17.30, 8.05)^{\mathrm{T}}$ and $\hat{\boldsymbol{\tau}}_{\mathcal{PMAMM}^{\star\star}}$ = $(24.90, 17.01, 7.67)^{\mathrm{T}}$ while $\hat{\boldsymbol{A}}_{\mathcal{GM}}$ = $\boldsymbol{A}_{12}(0.510)\, \boldsymbol{A}_{13}(1.040)\, \boldsymbol{A}_{23}(1.408)$, $\hat{\boldsymbol{A}}_{\mathcal{MGM}(G)} = \boldsymbol{A}_{12}(0.554)\, \boldsymbol{A}_{13}(1.044)\, \boldsymbol{A}_{23}(1.378)$ and $\hat{\boldsymbol{A}}_{\mathcal{PMAMM}^{\star\star}} = \boldsymbol{A}_{12}(0.520)\, \boldsymbol{A}_{13}(1.033)\, \boldsymbol{A}_{23}(1.400)$ respectively.

For the unknown number of matched pairs, L, we have found that at $L = 93$ log posterior is maximized foll all the methods mentioned earlier which is shown in Figure 5.9. The continuity of stripe of three image plots are the evidence of the matching matrix $\boldsymbol{M}$ whose first 93 diagonal elements are 1 i.e. all the methods correctly identify the correct number of matched pairs. To calculate the matching probabilities of these pairs we take the posterior mean of all matching matrices $\boldsymbol{M}$ after considering the burn in period for a particular run, and the matching probabilities of these pairs under $\mathcal{GM}$, $\mathcal{MGM}$(G), and $\mathcal{PMAMM}^{\star\star}$ are presented in Tables 5.2 - Table 5.4 respectively.

Finally, we conclude from this simulation study that $\mathcal{PMAMM}^{\star\star}$ performs better than all the methods considered in this thesis but, it requires more computational time compared to all other methods.

Table 5.1: Comparison between $\mathcal{GM}$, $\mathcal{MGM}$(K), $\mathcal{MGM}$(G), $\mathcal{PMAMM}$(K), $\mathcal{PMAMM}$(G) and $\mathcal{PMAMM}^{\star\star}$ approaches based on MCMC output for artificial data analysis; the hyperparameters $\beta = 36$ and $\lambda/\rho = 0.00047$.

Sampler	Mean Convergence Iterations (min, max)	Mean Computing Time in seconds	Mean Acceptance Rate (min, max)	Successful Runs (out of 50 runs)
$\mathcal{GM}$(1 update)	66,809 (2400, 478,000)	56.62	0.00077 (.0008, .007)	22 %
$\mathcal{GM}$(50 update)	14,416 (4000, 88,000)	145.32	0.00036(.000004, .0023)	24%
$\mathcal{MGM}(K)$	69,272 (3000, 450,000)	161.01	0.0057 (.0043, .0081)	20 %
$\mathcal{MGM}(G)$	34,444 (6000, 243,000)	175.46	0.0059 (.0041, .0074)	19 %
$\mathcal{PMAMM}$(K) (1 θ update)	268, 450 (1000, 850, 000)	129	-	80 %
$\mathcal{PMAMM}$(K) (50 θ update)	13,281 (100, 37,750)	854	-	82 %
$\mathcal{PMAMM}$(G) (1 θ update)	70,395 (1000, 327,000)	141	-	48 %
$\mathcal{PMAMM}$ G) (50 θ update)	16,450 (100, 76,300)	862	-	82 %
$\mathcal{PMAMM}^{\star\star}$ (G) (1 θ update)	27,100 (1600, 125,000)	3896	-	28 %
$\mathcal{PMAMM}^{\star\star}$ (G) (50 θ update)	1418 (3, 4368)	20,373	-	40 %

Table 5.2: Posterior mode of M at L=93 for artificial data using the $\mathcal{GM}$ approach in terms of probability of matching landmarks $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, p_{jk}, and presented in rank order based on probabilities which are calculated considering a burn in period of 478000.

$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}
10	10	.99975	9	9	.99879	26	26	.99819	80	80	.99730	38	38	.99606	89	89	.99438
39	39	.99971	83	83	.99878	55	55	.99818	40	40	.99709	29	29	.99603	50	50	.99430
35	35	.99959	36	36	.99877	28	28	.99808	52	52	.99707	93	93	.99601	64	64	.99342
21	21	.99957	79	79	0.99837	73	73	.99791	20	20	.99686	82	82	.99598	49	49	.99330
91	91	.99950	25	25	.99860	67	67	.99785	33	33	.99684	70	70	.995851	12	12	.99286
5	5	.99943	81	81	.99853	77	77	.99785	27	27	.9968	31	31	.995840	85	85	.99239
18	18	.99939	44	44	.99844	22	22	.99780	13	13	.99674	66	66	.995713	24	24	.99151
53	53	.99925	15	15	.99840	6	6	.99780	30	30	.99673	32	32	.995684	34	34	.99059
37	37	.99917	63	63	.99840	51	51	.99778	4	4	.99669	72	72	.995675	-	-	-
87	87	.99913	42	42	.99840	43	43	.99776	60	60	.99668	78	78	.995625	-	-	-
61	61	.99911	57	57	.99840	17	17	.99769	14	14	.99658	41	41	.995380	-	-	-
62	62	.99910	8	8	.99835	92	92	.99759	65	65	.99657	46	46	.995335	-	-	-
19	19	.99907	3	3	.99833	7	7	.99757	56	56	.99656	84	84	.995275	-	-	-
59	59	.99894	75	75	.99831	58	58	.99743	71	71	.99639	2	2	.995187	-	-	-
48	48	.99891	74	74	.99827	88	88	.99738	11	11	.99627	16	16	.994904	-	-	-
23	23	.99889	90	90	.99826	1	1	.99735	76	76	.99615	47	47	.994671	-	-	-
69	69	.99881	54	54	.99824	86	86	.99734	68	68	.99614	45	45	.994636	-	-	-

Table 5.3: Posterior mode of M at L=93 for artificial data using the $\mathcal{MGM}(G)$ approach in terms of probability of matching landmarks $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, p_{jk}, and presented in rank order based on probabilities which are calculated considering a burn in period of 243000.

$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}
67	67	.99854	9	9	.99772	88	88	.99732	84	84	.99705	40	40	.99671	48	48	.99611
12	12	.99819	56	56	.99771	33	33	.99732	7	7	.99703	80	80	.99670	64	64	.9960
31	31	.99812	25	25	.99771	75	75	.99731	82	82	.99702	52	52	.99670	34	34	.99602
13	13	.99808	57	57	.99770	37	37	.99728	87	87	.99700	6	6	.99665	19	19	.99599
44	44	.99798	14	14	.997632	83	83	.99727	21	21	.99698	65	65	.99663	28	28	.99598
16	16	.99791	63	63	.99853	36	36	.99727	54	54	.99698	17	17	.99662	49	49	.99582
91	91	.99791	15	15	.99752	4	4	.99726	73	73	.99697	43	43	.99655	10	10	.99570
11	11	.99788	89	89	.99751	23	23	.99722	93	93	.99693	2	2	.99654	18	18	.99518
69	69	.99786	78	78	.99751	30	30	.99720	90	90	.99690	8	8	.99653	45	45	.99448
32	32	.99786	59	59	.99748	71	71	.99720	81	81	.99685	55	55	.99651	-	-	-
41	41	.99783	-	-	-	74	74	.9972	77	77	.99685	72	72	.99642	-	-	-
20	20	.99783	46	46	.99748	35	35	.99719	85	85	.99684	53	53	.99636	-	-	-
27	27	.99781	26	26	.99742	61	61	.99718	29	29	.99680	58	58	.99632	-	-	-
68	68	.99776	22	22	.99742	86	86	.99717	62	62	.99680	47	47	.99632	-	-	-
76	76	.99774	50	50	.99737	38	38	.99715	1	1	0.99677	92	92	.99626	-	-	-
42	42	.99774	39	39	.99736	70	70	.99710	66	66	.99675	79	79	.99623	-	-	-
60	60	0.99773	5	5	.99735	3	3	.99708	24	24	.99673	51	51	.99619	-	-	-

Table 5.4: Posterior mode of M at L=93 for artificial data using the $\mathcal{PMAMM}^{\star\star}$ approach in terms of probability of matching landmarks $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, p_{jk}, and presented in rank order based on probabilities which are calculated considering a burn in period of 125000.

$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}	$\boldsymbol{x}_j$	$\boldsymbol{y}_k$	p_{jk}
57	57	.99979	60	60	.99972	70	70	0.99971	8	8	0.99970	65	65	.99967	4	4	.99963
26	26	.99977	43	43	.99972	6	6	.99971	85	85	.99969	56	56	.99967	91	91	.99963
67	67	.99976	45	45	.99972	80	80	.99971	16	16	.99969	36	36	.99967	75	75	.99962
68	68	.999763	42	42	.99972	3	3	.99971	24	24	.99969	64	64	.99967	54	54	.99957
55	55	.99975	1	1	.99972	66	66	.99971	15	15	.99969	7	7	.99967	89	89	.99948
63	63	.99975	78	78	.99972	5	5	.99971	81	81	.99969	73	73	.99967	48	48	.99944
46	46	.99975	12	12	.99972	83	83	.99971	10	10	.99969	50	50	.99967	53	53	.99929
28	28	.99975	86	86	.99972	40	40	.99970	39	39	.99969	27	27	0.99966	49	49	0.99916
29	29	.99974	69	69	.99972	21	21	.99970	17	17	.99968	92	92	.99966	-	-	-
62	62	.99974	22	22	.99972	84	84	.99970	13	13	.99968	71	71	.99966	-	-	-
32	32	.99974	90	90	.99972	44	44	.99970	72	72	.99968	14	14	.99965	-	-	-
41	41	.99974	79	79	.99971	2	2	.99970	11	11	.99968	19	19	.99965	-	-	-
82	82	.99974	74	74	.99971	76	76	.99970	47	47	.99968	30	30	.99964	-	-	-
87	87	0.99974	77	77	.99971	88	88	.99970	93	93	.99968	52	52	.99964	-	-	-
20	20	.99973	51	51	.99971	18	18	.99970	61	61	.99968	58	58	.99964	-	-	-
31	31	.99973	35	35	.99971	37	37	.99970	59	59	.99968	9	9	.99964	-	-	-
23	23	.99973	34	34	.99971 5	33	33	.99970	25	25	.99968	38	38	.99963	-	-	-

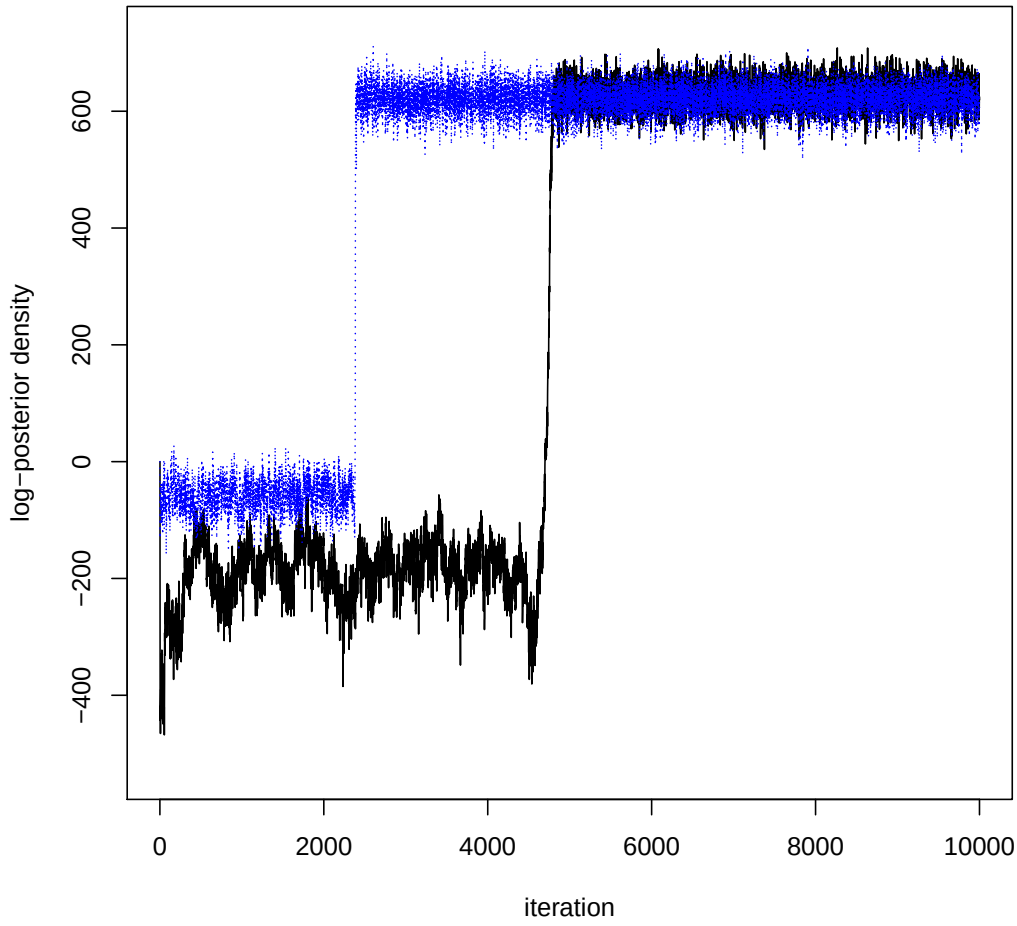

Figure 5.1: Output from the MCMC algorithm for matching artificial data using $\mathcal{GM}$ and $\mathcal{MGM}$(G) approaches. The above plot is the value of joint log-posterior (up to normalizing constant) versus iteration number for $\mathcal{GM}$(1 update) and $\mathcal{MGM}$(G) approaches where every 100^{th} simulated value is plotted; results obtained using $\mathcal{GM}$ approach is shown as solid black line while blue dotted line is used to show the results for $\mathcal{MGM}$(G) approach respectively.

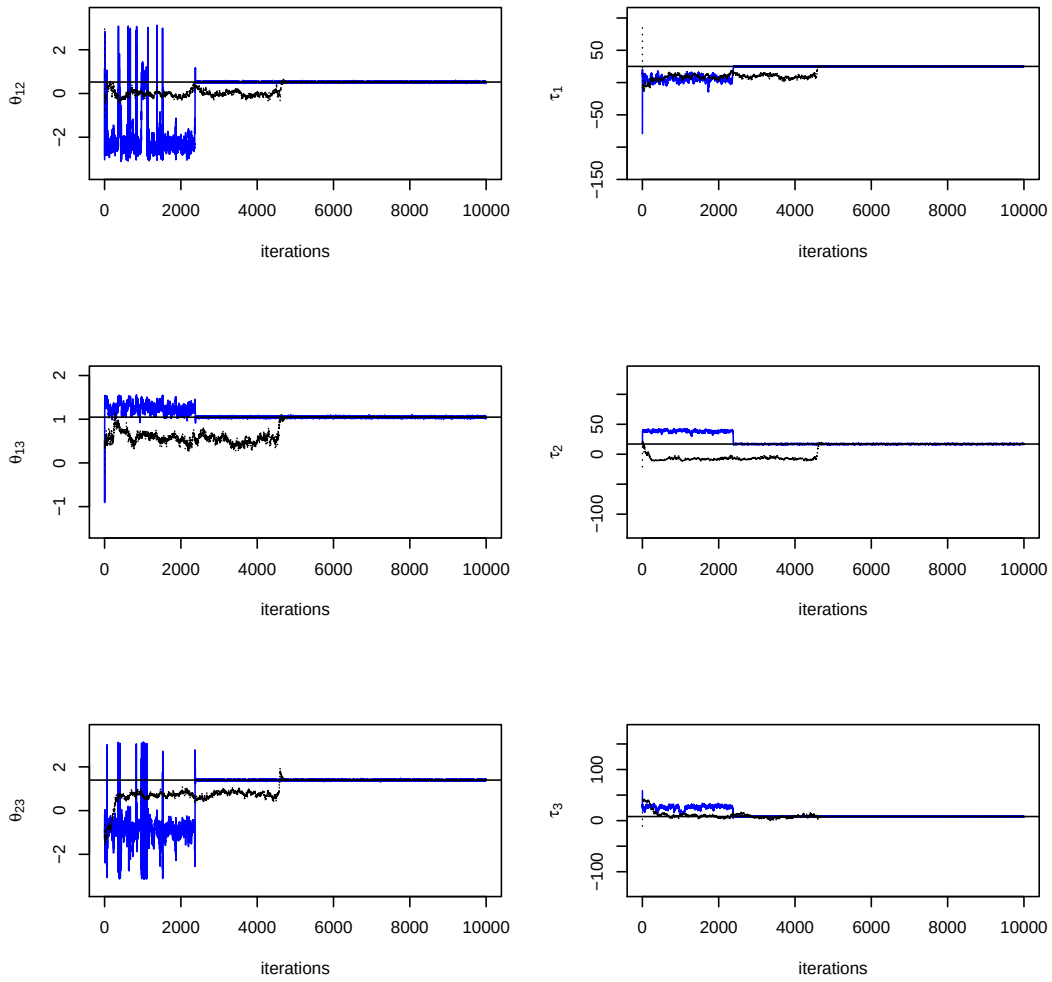

Figure 5.2: Output from the MCMC algorithm for matching artificial data using $\mathcal{GM}$ and $\mathcal{MGM}(\mathrm{G})$ approaches. First and second columns of Figure 5.2 display the trace plots of Eulerian angles and $\boldsymbol{\tau}$ respectively where every 100^{th} simulated value is plotted; results obtained using $\mathcal{GM}$ (1 update) approach are shown as solid black line while blue dotted lines is used to show the results for $\mathcal{MGM}(\mathrm{G})$ approaches respectively.

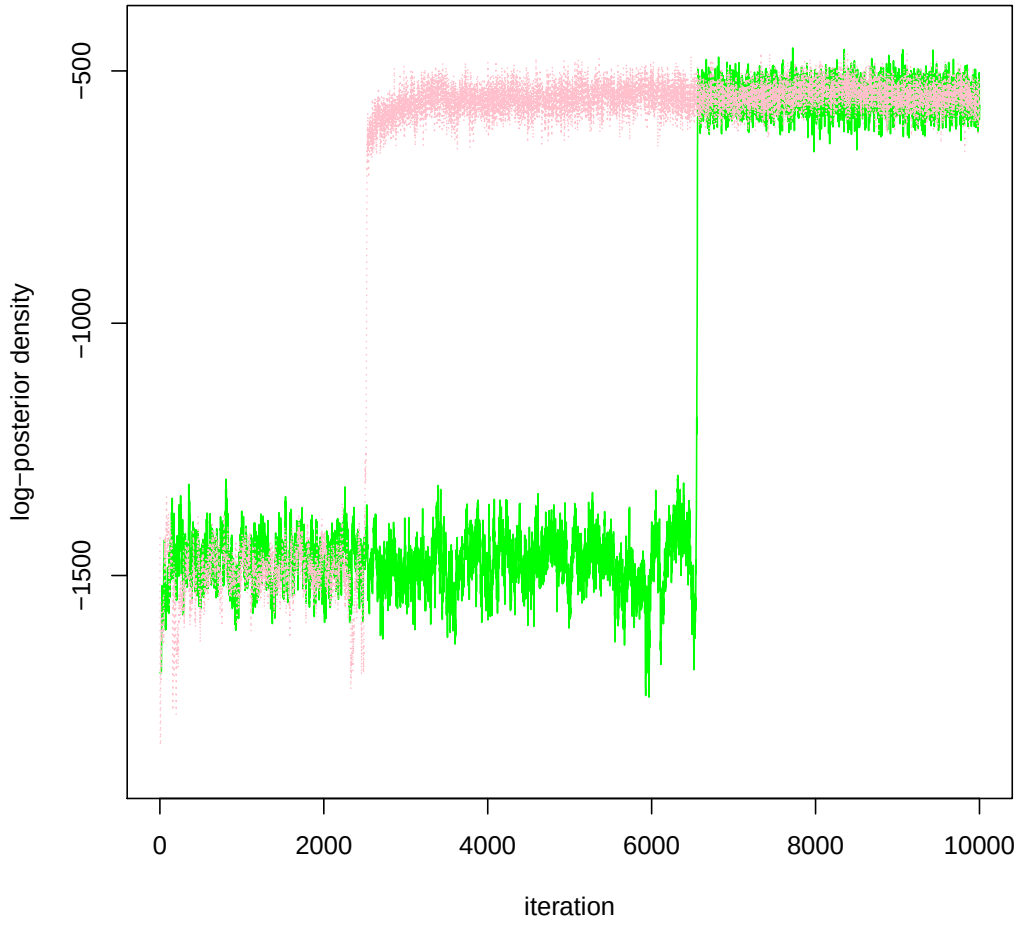

Figure 5.3: Output from the MCMC algorithm for matching artificial data using $\mathcal{PMAMM}$(G) (1 θ update) and $\mathcal{PMAMM}^{\star\star}$(1 θ update) approaches. The above plot is the value of joint log-posterior (up to normalizing constant) versus iteration number where every 50^{th} simulated value is plotted; results obtained using $\mathcal{PMAMM}$(G) approach is shown as solid green line while pink dotted line is used to show the results for $\mathcal{PMAMM}^{\star\star}$ approaches respectively. As the $\mathcal{PMAMM}$ is different compared to $\mathcal{GM}$ and $\mathcal{MGM}$, so $\mathcal{PMAMM}$ converges to different log-posterior value compared to $\mathcal{GM}$ and $\mathcal{MGM}$.

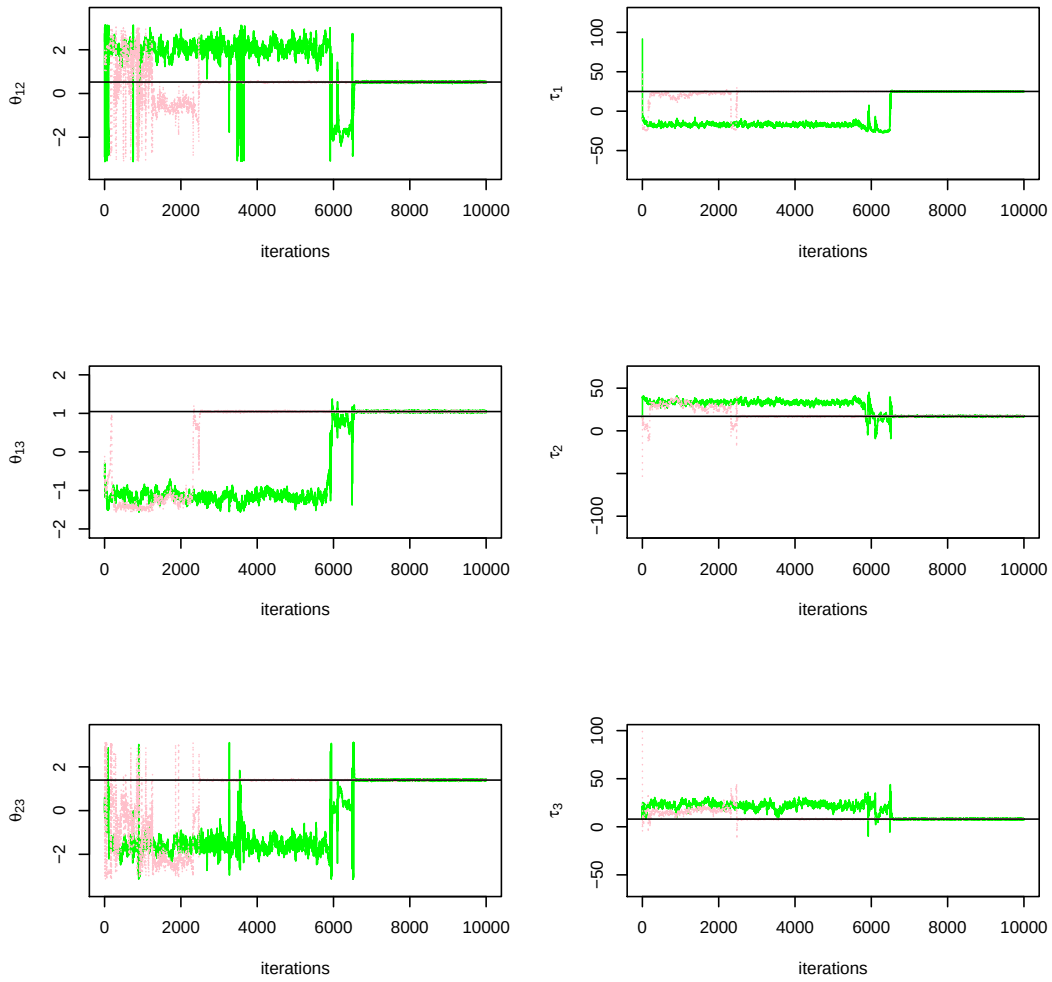

Figure 5.4: Output from the MCMC algorithm for matching artificial data using $\mathcal{PMAMM}$(G) (1 θ update) and $\mathcal{PMAMM}^{\star\star}$(1 θ update) approaches. First and second columns of Figure 5.4 display the trace plots of Eulerian angles and $\boldsymbol{\tau}$ respectively where every 50^{th} simulated value is plotted; results obtained using $\mathcal{PMAMM}$(G) approach is shown as solid green line while a pink dotted line is used to show the results for $\mathcal{PMAMM}^{\star\star}$ approaches respectively.

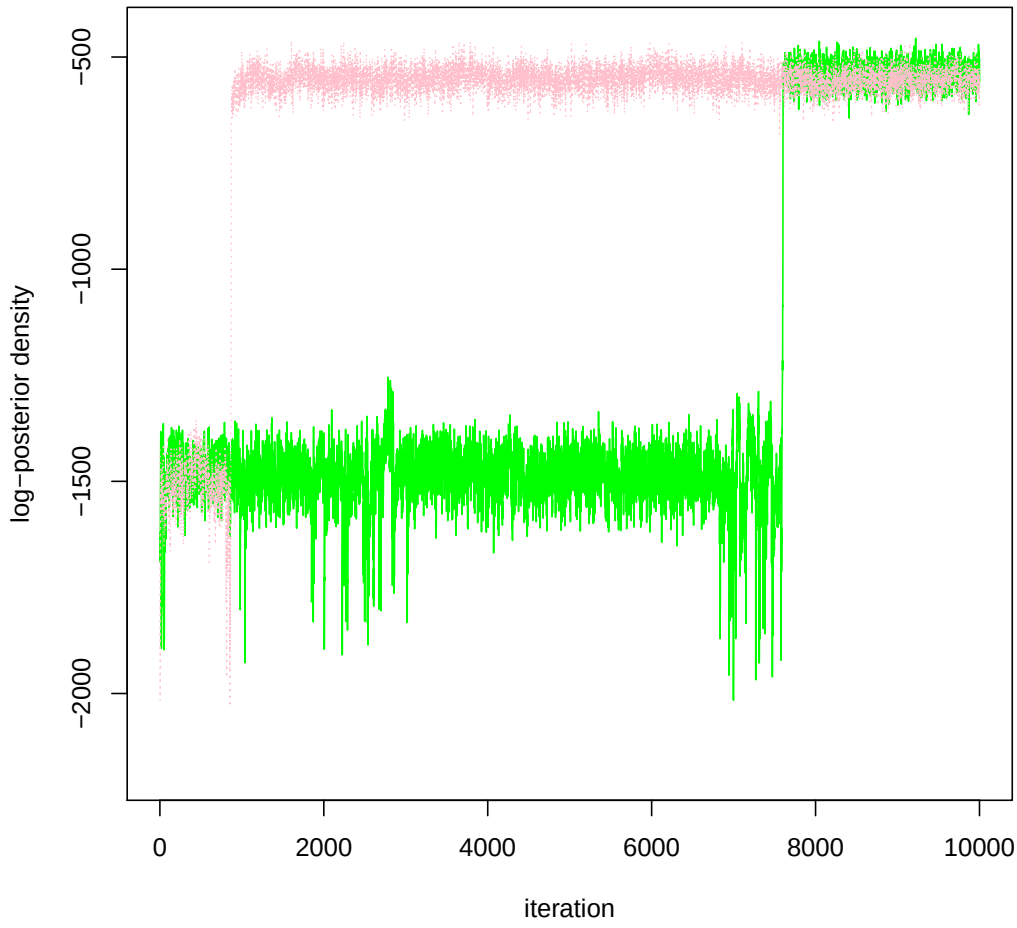

Figure 5.5: Output from the MCMC algorithm for matching artificial data using $\mathcal{PMAMM}$(K) (50 θ update) and $\mathcal{PMAMM}^{\star\star}$(50 θ update) approaches. The above plot is the value of joint log-posterior (up to normalizing constant) versus iteration number where every 5^{th} simulated value is plotted; results obtained using $\mathcal{PMAMM}$(K) approach is shown as solid green line while a pink dotted line is used to show the results for $\mathcal{PMAMM}^{\star\star}$ approaches respectively.

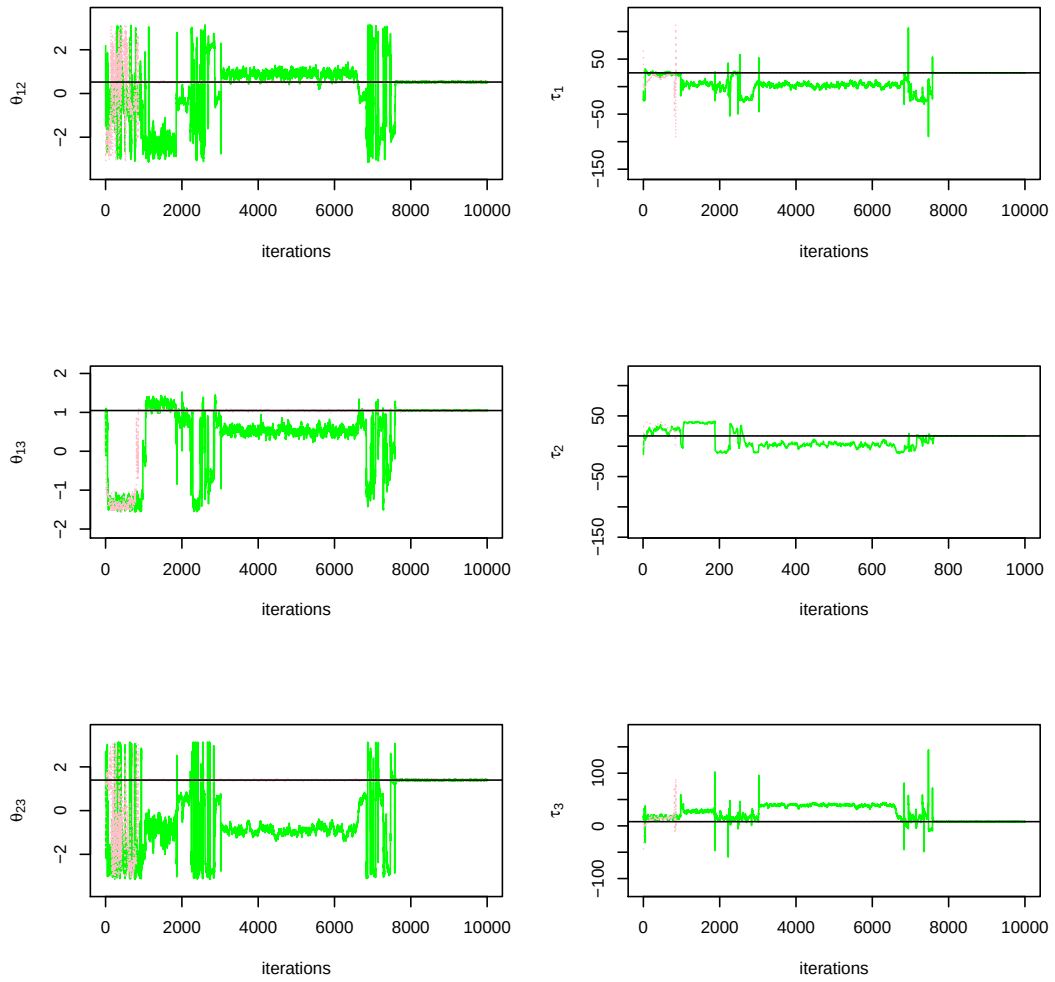

Figure 5.6: Output from the MCMC algorithm for matching artificial data using $\mathcal{PMAMM}$(K) (50 θ update) and $\mathcal{PMAMM}^{\star\star}$(50 θ update) approaches. First and second columns of Figure 5.6 display the trace plots of Eulerian angles and $\boldsymbol{\tau}$ respectively where every 5^{th} simulated value is plotted; results obtained using $\mathcal{PMAMM}$(K) approach is shown as solid green line while a pink dotted line is used to show the results for $\mathcal{PMAMM}^{\star\star}$ approaches respectively.

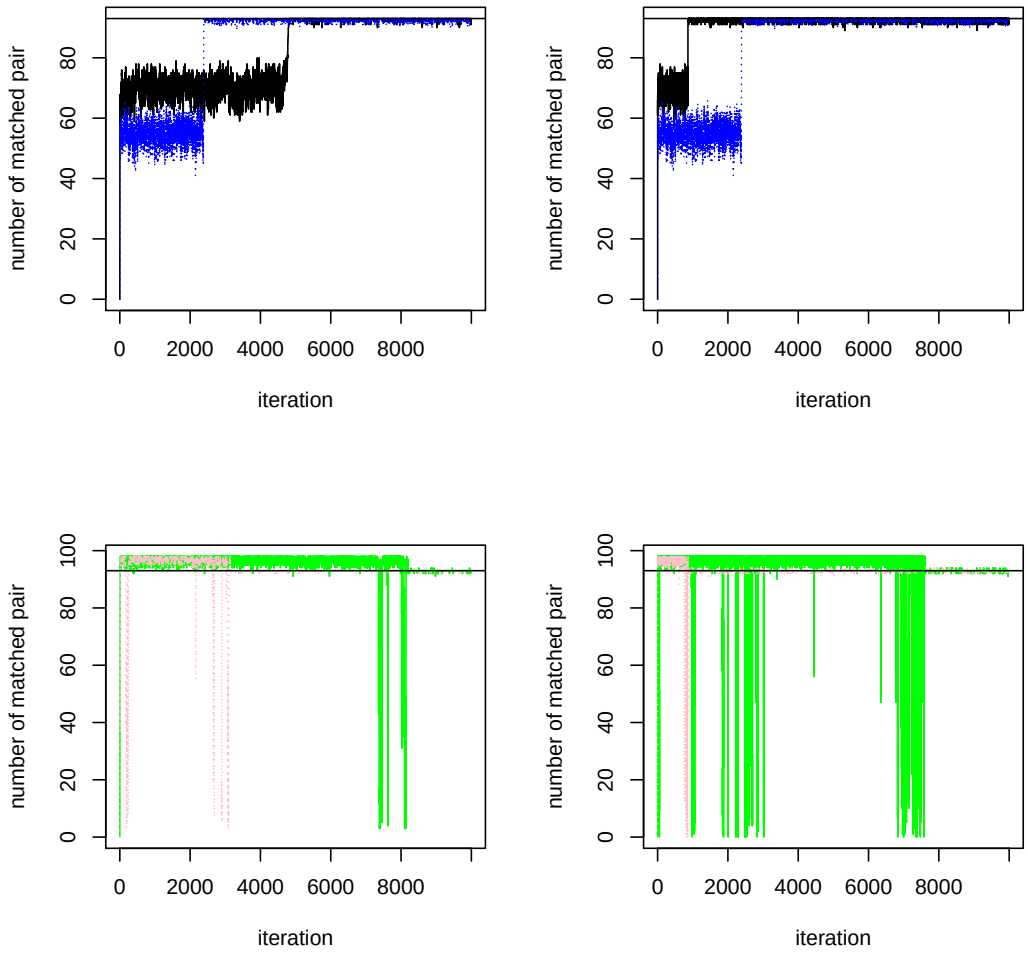

Figure 5.7: Output from the MCMC algorithm for matching artificial data using $\mathcal{GM}$, $\mathcal{MGM}$ and $\mathcal{PMAMM}$ approaches. The first row of this figure shows the trace plot of L for $\mathcal{GM}$ (1 update) and $\mathcal{MGM}$ (G) and $\mathcal{GM}$ (10 update) and $\mathcal{MGM}$(G) respectively where every 500^{th} simulated value is plotted; results obtained using $\mathcal{GM}$ approach is shown as solid black line while blue dotted line is used to show the results for $\mathcal{MGM}$. The trace plot of L obtained under $\mathcal{PMAMM}$(G)(1 θ update) (green color) and $\mathcal{PMAMM}^{\star\star}$(1 θ update) (pink color) and $\mathcal{PMAMM}$(K)(50 θ update) and $\mathcal{PMAMM}^{\star\star}$(50 θ update) is shown by the second row where every 50^{th} and 5^{th} simulated value are ploted respectively. The horizontal line in each plot is drawn at L=93.

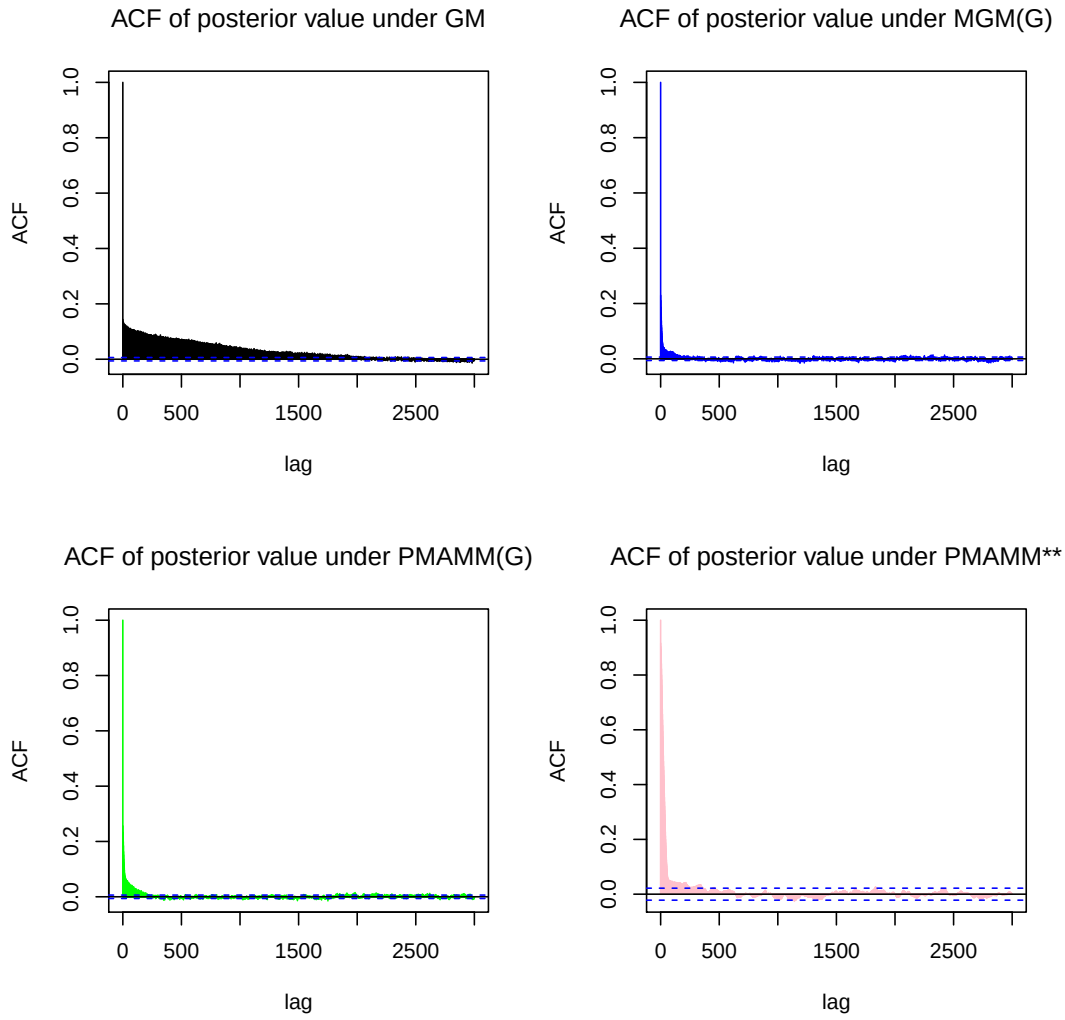

Figure 5.8: Output from the MCMC algorithm for matching artificial data using $\mathcal{GM}$, $\mathcal{MGM}(G)$, $\mathcal{PMAMM}(G)$ and $\mathcal{PMAMM}^{**}$ approaches. The above plot shows the ACF plots of first 100,000 values (after burn in period) of log-posterior (up to normalizing constant) for $\mathcal{GM}$, $\mathcal{MGM}(\mathrm{G})$, $\mathcal{PMAMM}(\mathrm{G})$ and $\mathcal{PMAMM}^{**}$ approaches; results obtained using $\mathcal{GM}$ approach is shown as solid black line while blue, green and pink lines are used to show the results for $\mathcal{MGM}(\mathrm{G})$, $\mathcal{PMAMM}(\mathrm{G})$ and $\mathcal{PMAMM}^{**}$ approaches respectively.

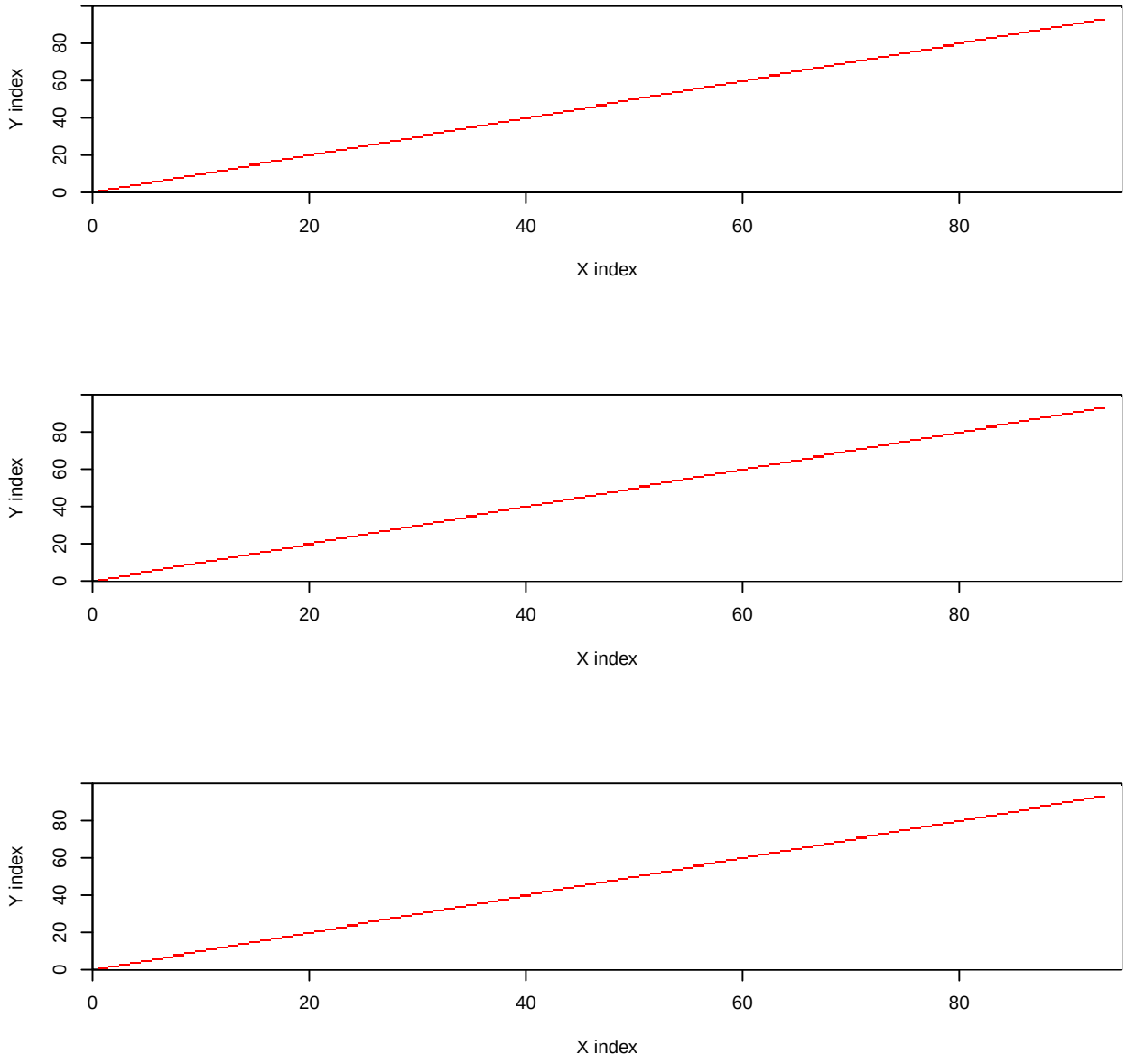

Figure 5.9: Output from the MCMC algorithm for matching artificial data using $\mathcal{GM}$, $\mathcal{MGM}$ and $\mathcal{PMAMM}^{\star\star}$ approaches. The top, middle and bottom plots of the above figure plot the matching matrices obtained under $\mathcal{GM}$, $\mathcal{MGM}$ and $\mathcal{PMAMM}^{\star\star}$ approaches respectively. The diagonal line, like a set of stairs, of each plot shows the matching pairs between $\boldsymbol{X}$ and $\boldsymbol{Y}$ indices; the continuity of stairs tells us that the first 93 points of $\boldsymbol{X}$ match with the first 93 points of $\boldsymbol{Y}$ respectively.

5.2 Analysis of Gel Data in 2 Dimension Case

In this section we analyze and discuss the output of gel data in 2 dimension case using the $\mathcal{GM}$, $\mathcal{MVGM}$ and $\mathcal{PMAMM}$ approaches. As we have already discussed in section 5.1 about the convergence assessment of a Markov chain and the number of updates to $\boldsymbol{M}$ and $\boldsymbol{\theta}$, so we will not repeat this explanation in this section and subsequent sections.

5.2.1 Experimental Results for $\mathcal{MVGM}$ and $\mathcal{GM}$ Approaches

We now compare the performances of the $\mathcal{MGM}$ and $\mathcal{GM}$ approaches for aligning two gels. For these two approaches, we run the Markov chain Monte Carlo sampler described in Chapter 1 for 100 000 sweeps with d=2 and $\boldsymbol{A}$ defined in equation (1.66) to get the MCMC output while prior and hyperprior settings are kept the same as in Green and Mardia (2006) except $\mu_{\boldsymbol{\tau}} = (-36.85, 23.14)^T$ instead of $\mu_{\boldsymbol{\tau}} = (0, 0)^T$, where the quantity $(-36.85, 23.14)^T$ is the value of difference of two configurations centers. The performance of these two approaches is evaluated based on how quickly the Markov chain converges to the posterior distribution, acceptance rate of proposed move of $\boldsymbol{M}$ and computing time due to these two samplers. We repeat our experiment 50 times with the same parameter settings but with different random starting points for $\boldsymbol{\tau}$ to get an average estimates of the above quantities.

Table 5.5: Comparison of $\mathcal{MGM}$ and $\mathcal{GM}$(1 update) approaches based on MCMC output averaged over 50 runs with 100 000 iterations each, mean acceptance rate is calculated after considering the burn in periods

Sampler	Mean Convergence Iterations (min, max)	Mean Computing Time in seconds	Mean Acceptance Rate (min, max)
$\mathcal{MGM}$	317 (50, 1000)	2.65	.0340 (.03205, .0356)
$\mathcal{GM}$	3854 (500, 7000)	0.96	.00154 (.00096, .0021)

Table 5.6: Comparison of $\mathcal{MGM}$ and $\mathcal{GM}$(10 update) matching samplers based on MCMC output averaged over 50 runs with 100 000 iterations each, mean acceptance rate is calculated after considering the burn in periods

Sampler	Mean Convergence Iterations (min, max)	Mean Computing Time in seconds	Mean Acceptance Rate (min, max)
$\mathcal{MGM}$	317 (50, 1000)	2.65	.0340 (.03205, .0356)
$\mathcal{GM}$	438 (200, 900)	1.95	.00014 (.00010, .00019)

Figure 5.10 and Table 5.5 are presented here to compare the performances between $\mathcal{MGM}$ and $\mathcal{GM}$ approaches where only one update to $\boldsymbol{M}$ is made along with just one of each other parameter update per sweep in $\mathcal{GM}$ approaches. The purpose of creating Figure 5.11 and Table 5.6 is the same as before but in this case $\mathcal{GM}$ approach considers ten (10) update to $\boldsymbol{M}$ along with just one of each other parameter update per sweep. The first column of Figure 5.10 shows the unnormalised log-posterior values and the traces of first and second coordinates of $\boldsymbol{\tau}$ respectively, obtained with these two samplers where every 500^{th} simulated value is considered to plot. From these plots (constructed using MCMC

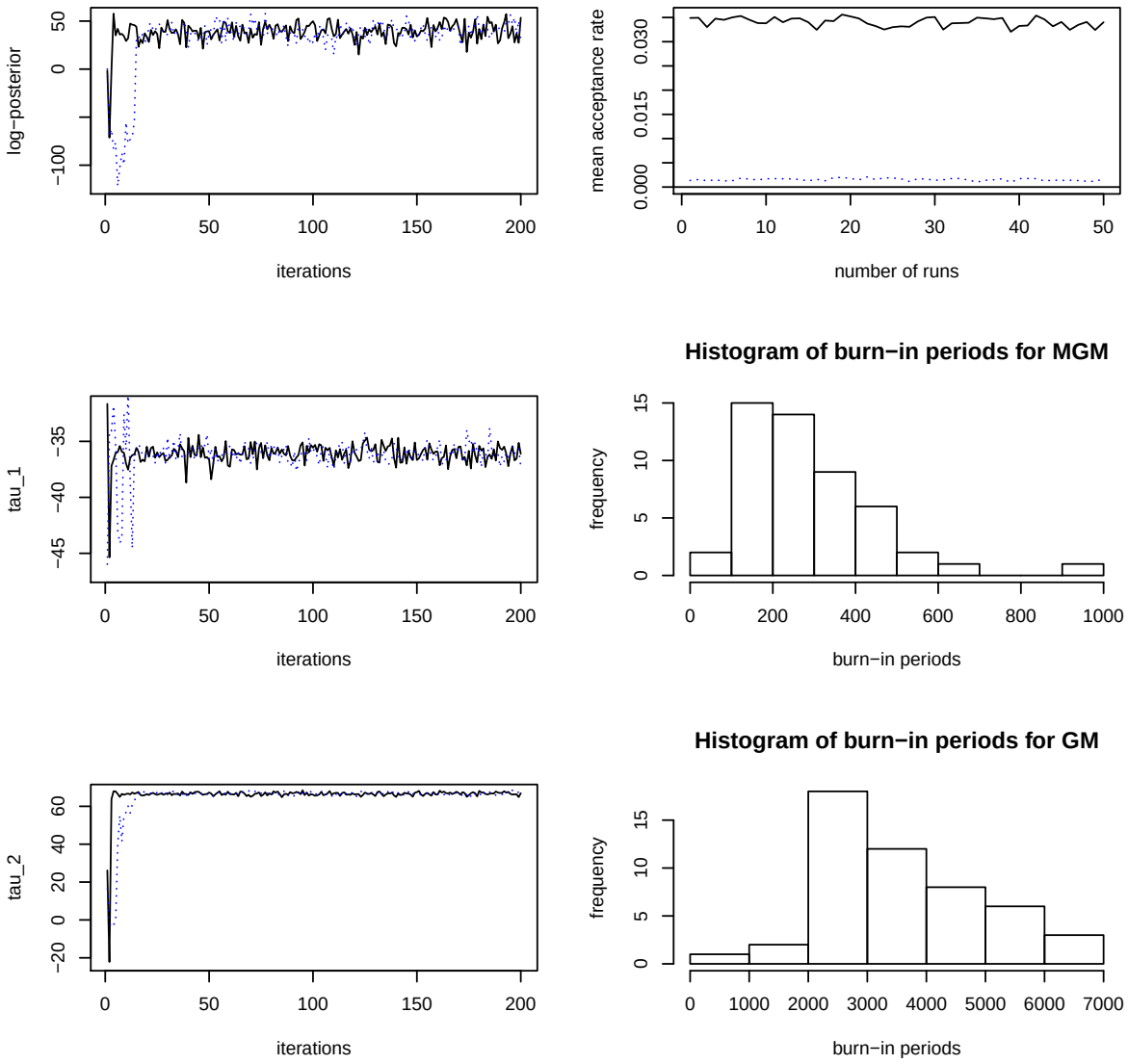

Figure 5.10: Output from the MCMC algorithm for matching gel data using $\mathcal{MGM}$ and $\mathcal{GM}$(1 update to $\boldsymbol{M}$) approaches: results obtained using $\mathcal{MGM}$ approach are shown as solid black lines while dotted blue line is used to show the results for $\mathcal{GM}$ approach. First column of Figure 5.10 displays the log-posterior (up to normalizing constant) and the trace plots of $\boldsymbol{\tau}$ respectively where every 500^{th} simulated value is plotted. Plot of acceptance rate based on MCMC output after burn in period for 50 runs and histograms of burn in periods are shown in column 2.

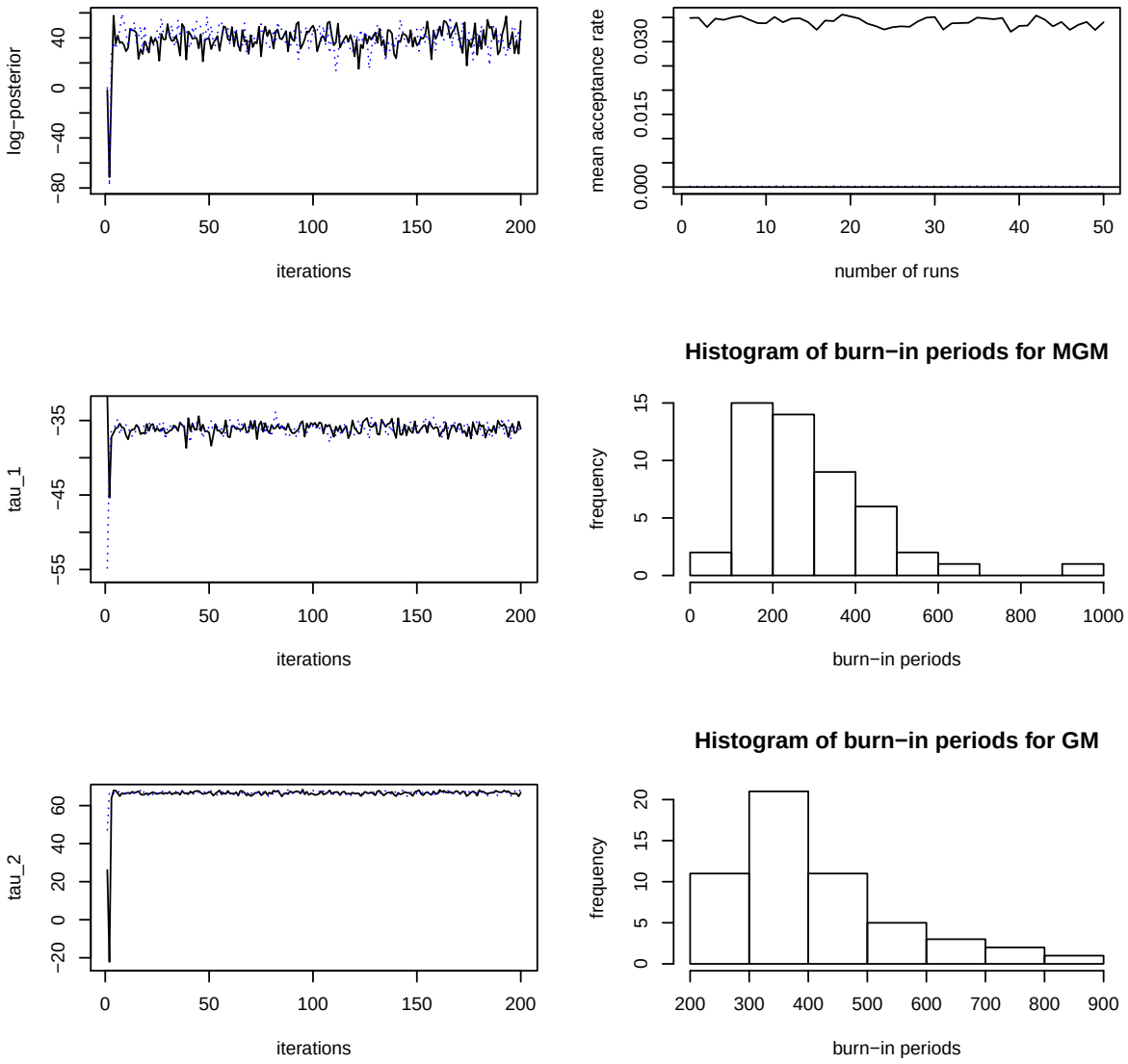

Figure 5.11: Output from the MCMC algorithm for matching gel data using $\mathcal{MGM}$ and $\mathcal{GM}$ (10 update to $\boldsymbol{M}$) approaches: results obtained using $\mathcal{MGM}$ approach are shown as solid black lines while dotted blue line is used to show the results for $\mathcal{GM}$ approach. The first column of Figure 5.11 displays the log-posterior (up to normalizing constant) and the trace plots of $\boldsymbol{\tau}$ respectively where every 500^{th} simulated value is plotted. Plot of acceptance rate based on MCMC output after burn in period for 50 runs and histograms of burn in periods are shown in column 2.

output for one particular run), we can see that $\mathcal{MGM}$ approach converges almost instantaneously and outperforms the $\mathcal{GM}$ approach. This pattern remains the same for all other remaining runs which is shown in the histograms of burn in periods presented in column 2 of Figure 5.10 and in Table 5.5. It is evident from these histograms and Table 5.5, constructed on the basis of all 50 runs, on average $\mathcal{MGM}$ approach requires fewer iterations (317), almost 12 times faster, compared to $\mathcal{GM}$ approach (3854) to converge while they need minimum at least 1000 and 7000 iterations as a burn in period respectively for the safest inference procedure. For convenient graphical display every 500^{th} simulated value or thinned value of MCMC output is considered to plot in all the Figures. The histograms in Figures 5.5 and 5.6 present the frequency distribution of burn in period of several runs of a particular MCMC method.

Beside the convergence time criteria of the samplers, mean computing time and mean acceptance rate of proposed $\boldsymbol{M}$ were also taken into account as further criteria to evaluate the performances of these two approaches. From Table 5.5, we can see that $\mathcal{MGM}$ approach takes on an average 2.65 seconds to complete a run with 100 000 sweeps on a 2.3 GHz PC which is almost 3 times higher than $\mathcal{GM}$ approach (0.96 seconds), and we believe this is because $\mathcal{MGM}$ needs to perform some additional calculations (relative probabilities for all points) compared to $\mathcal{GM}$ to select one pair instead of choosing randomly one pair like Green and Mardia matching sampler in $\mathcal{GM}$ approach. However, we can ignore this time comparison as it is not too large to be problematic, and no emphasis has been placed on coding as efficiently as possible.

Although implementing $\mathcal{MGM}$ is a bit expensive it has a much better acceptance rate of $\boldsymbol{M}$ compared to GM approach, which is shown by Table 5.5 and top right plot of Figure 5.10. Table 5.5 shows that mean acceptance rate of $\boldsymbol{M}$ for $\mathcal{MGM}$ and $\mathcal{GM}$ approaches are 3.4 % and 0.0154 % respectively, which is supported by

the top right plot of Figure 5.10 (acceptance rate versus runs). Therefore, we can say that $\mathcal{MGM}$ performance is superior compared to $\mathcal{GM}$(1 update) as far as convergence time and acceptance rate of $\boldsymbol{M}$ criteria are concerned.

When we consider 10 updates of $\boldsymbol{M}$ along with just one of each other parameter update per sweep in $\mathcal{GM}$ approach, it is plausible that the sampler will converge about 10 times faster than the case when 1 update to $\boldsymbol{M}$ along with just one of each other parameter update per sweep is considered. In addition, it is also expected that sampler (with 10 updates) will require higher computing time than the sampler (with 1 update) to complete a run as it needs to update $\boldsymbol{M}$ 10 times along with other parameters single update (if the total number of sweeps is fixed as before). From Figure 5.11 and Table 5.6, we can see that the $\mathcal{GM}$ approach with 10 updates to $\boldsymbol{M}$ needs on average slightly more iterations (by 120) and almost equal amount of time (1.95 s) as $\mathcal{MGM}$ approach to converge and complete a run respectively i.e. its efficiency is almost as same as the $\mathcal{MGM}$ approach as far as convergence and computing time are concerned. However, the $\mathcal{GM}$ approach with 10 updates still has a lower acceptance rate of .0014 % for $\boldsymbol{M}$ proposals.

For the unknown parameters $\boldsymbol{\tau}$ and $\boldsymbol{M}$, we have obtained their point estimate $\hat{\boldsymbol{\tau}}_{\mathcal{MGM}}$, $\hat{\boldsymbol{\tau}}_{\mathcal{GM}}$, $\hat{\boldsymbol{M}}_{\mathcal{MGM}}$ and $\hat{\boldsymbol{M}}_{\mathcal{GM}}$ by maximizing the overall joint posterior i.e. their posterior modes. The posterior mode of $\boldsymbol{\tau}$ is $\hat{\boldsymbol{\tau}}_{\mathcal{MGM}}$=(- 35.947, 67.043) and $\hat{\boldsymbol{\tau}}_{\mathcal{GM}}$ = (- 35.983, 67.263) compared to (- 36.08, 66.64) obtained by Dryden and Mardia (1998) while $\hat{\boldsymbol{M}}_{\mathcal{MGM}}$ and $\hat{\boldsymbol{M}}_{\mathcal{GM}}$ are displayed by Table 5.7 and Table 5.8 respectively. From Table 5.7 and 5.8, we see that both the approaches have reached their posterior maximum at the same point of $\boldsymbol{M}$ which has L=16 matched pairs but they have different rank order of these matched pairs.

Table 5.7: Posterior mode of $\boldsymbol{M}$ for gel data using the $\mathcal{MGM}$ approach expressed in terms of probability of matching landmarks $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, p_{jk}, and presented in rank order based on probabilities which are calculated after considering a burn in period of 1000 iterations

Rank	$\mathbf{x}_j$	$\mathbf{y}_k$	p_{jk}	*Rank*	$\mathbf{x}_j$	$\mathbf{y}_k$	p_{jk}
1	5	5	0.99606	9	19	19	0.99064
2	8	8	0.99575	10	10	10	0.98917
3	4	4	0.99371	11	6	6	0.98712
4	3	3	0.99364	12	31	30	0.98700
5	2	2	0.99361	13	32	31	0.98615
6	7	7	0.99315	14	1	1	0.98447
7	24	23	0.99192	15	9	9	0.98336
8	15	21	0.99136	16	26	32	0.78918

Table 5.8: Posterior mode of $\boldsymbol{M}$ for gel data using the $\mathcal{GM}$ approach expressed in terms of probability of matching landmarks $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, p_{jk}, and presented in rank order based on probabilities which are calculated after considering a burn in period of 7000 iterations

Rank	$\mathbf{x}_j$	$\mathbf{y}_k$	p_{jk}	*Rank*	$\mathbf{x}_j$	$\mathbf{y}_k$	p_{jk}
1	15	21	1	9	31	30	0.98943
2	7	7	1	10	24	23	0.98837
3	4	4	1	11	19	19	0.98401
4	2	2	1	12	6	6	0.97849
5	1	1	1	13	3	3	0.97799
6	32	31	0.99733	14	9	9	0.97671
7	8	8	0.99598	15	5	5	0.97376
8	10	10	0.99306	16	26	32	0.78819

5.2.2 Performance of $\mathcal{PMAMM}^{\star}$ over MVGM and GM Approaches

In this subsection we compare the performances of $\mathcal{PMAMM}^{\star}$ over $\mathcal{MVGM}$ and $\mathcal{GM}$ approaches for gel data. For $\mathcal{PMAMM}^{\star}$, we run the Markov chain Monte Carlo sampler for 100 000 sweeps as well with d=2 and $\boldsymbol{A}$ defined in equation (1.66) to get the MCMC output. The priors for $\boldsymbol{\tau}$ and σ are kept the same as in Green and Mardia (2006) except $\mu_{\boldsymbol{\tau}} = (-36.85, 23.14)^T$ instead of $\mu_{\boldsymbol{\tau}} = (0, 0)^T$, while we use a uniform prior for all $\boldsymbol{\theta}_j$ i.e. $\boldsymbol{\eta}_k$= $(1, 1, \ldots, \ldots, 1)$ for $j = (1, 2, \ldots, m)$. For this analysis, the coffin bin parameter is set equal to ten thousands i.e. $|\Omega|$=10000, and we also consider two cases in our proposed matching sampler; update one (1) and ten (10) $\boldsymbol{\theta}$ instead of updating all $\boldsymbol{\theta}$ per iteration which are selected randomly. We repeat our experiment 50 times with the same parameters settings but with different starting point for $\boldsymbol{\tau}$ to get average estimates of the above quantities.

We have already seen in subsection 5.2.1 that $\mathcal{MGM}$ and $\mathcal{GM}$ with 10 updates to $\boldsymbol{M}$ have the similar performances as far as convergence and computing time are concerned, although $\mathcal{MGM}$ has much better acceptance rate of $\boldsymbol{M}$ compared to $\mathcal{GM}$ irrespective of number of updates of $\boldsymbol{M}$. Now from Table 5.9, we can say that our $\mathcal{PMAMM}^{\star}$ with 10 updates of $\boldsymbol{\theta}$ is the fastest among all the cases which needs on average 103 iterations (at least 3 times faster than $\mathcal{MGM}$ approach) to converge although it needs almost 2 times more computing time (6.50 seconds) than $\mathcal{MGM}$ approach. However, this computing time is not so large to be problematic. The most important thing of our $\mathcal{PMAMM}^{\star}$ is that there is no issue about the acceptance rate of $\boldsymbol{M}$ as it doesn't require any Metropolis step to update $\boldsymbol{M}$.

The top plot of Figure 5.12 plots the log-posterior value (up to a normalizing

constant) for $\mathcal{PMAMM}^{\star}$ which converges at different value of log-posterior compare to $\mathcal{GM}$ and $\mathcal{MGM}$ approaches as $\mathcal{PMAMM}$ uses a different model. From the log-posterior plot, it is evident that $\mathcal{PMAMM}^{\star}$ with 1 and 10 updates needs a minimum of 1600 and 200 iterations as a burn in period respectively for the safest inference procedure while histograms in the same figure for $\mathcal{PMAMM}^{\star}$ with 1 and 10 updates show the frequency distribution of burn in period for 50 runs.

For the unknown parameters $\boldsymbol{\tau}$ and $\boldsymbol{M}$, we have obtained their point estimate $\hat{\boldsymbol{\tau}}_{\mathcal{PMAMM}^{\star}}$ and $\hat{\boldsymbol{M}}_{\mathcal{PMAMM}^{\star}}$ by maximizing the overall joint posterior i.e. their posterior modes. The posterior mode of $\boldsymbol{\tau}$ is $\hat{\boldsymbol{\tau}}_{\mathcal{PMAMM}^{\star}}=(-35.968, 66.767)^{\mathrm{T}}$ compared to $\hat{\boldsymbol{\tau}}_{\mathcal{GM}} = (-35.983, 67.263)^{\mathrm{T}}$ and $(-36.08, 66.64)^{\mathrm{T}}$ obtained by $\mathcal{GM}$ approach and Dryden and Mardia (1998) respectively, while $\hat{\boldsymbol{M}}_{\mathcal{PMAMM}^{\star}}$ is displayed by Table 5.10. From Table 5.10, we see that our $\mathcal{PMAMM}^{\star}$ have reached their posterior maximum at the same point of $\boldsymbol{M}$ like $\mathcal{GM}$ and $\mathcal{MGM}$ approaches, which have L=16 matched pairs.

Table 5.9: Performance of our $\mathcal{PMAMM}^{\star}$ based on MCMC outputs for gel data averaged over 50 runs with 100 000 iterations each

Sampler	Mean Convergence Iterations (min, max)	Mean Computing Time in seconds
$\mathcal{PMAMM}^{\star}$ (1 θ update)	937 (50, 1600)	2.50
$\mathcal{PMAMM}^{\star}$ (10 θ update)	103 (50, 200)	6.50
$\mathcal{GM}$ (1 update)	3854 (500, 7000)	0.96
$\mathcal{GM}$ (10 update)	438 (200,900)	1.95
$\mathcal{MGM}$	317 (50, 1000)	2.65

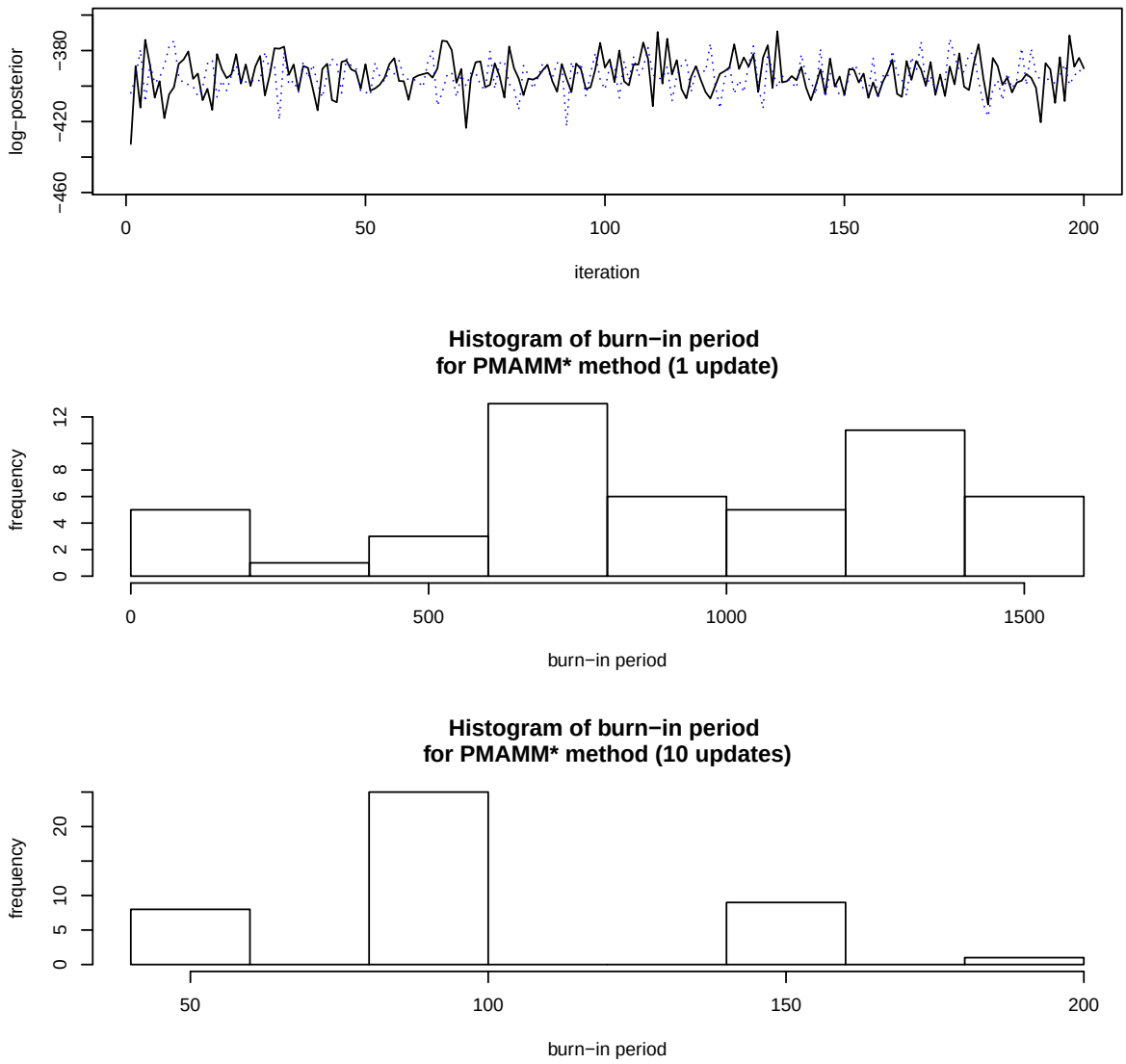

Figure 5.12: Output from the MCMC algorithm for matching gel data using our $\mathcal{PMAMM}^{\star}$: results obtained using $\mathcal{PMAMM}^{\star}$ (1 θ update) are shown as solid black lines while dotted blue line is used to show the results for $\mathcal{PMAMM}^{\star}$ approach (10 θ update). The top graph of Figure 5.12 displays the log-posterior values (up to a normalizing constant) where every 500^{th} simulated value is plotted and the histograms of burn-in period for $\mathcal{PMAMM}^{\star}$ using 50 runs.

Table 5.10: Posterior mode of $\boldsymbol{M}$ for gel data using the $\mathcal{PMAMM}^{\star}$ approach expressed in terms of probability of matching landmarks $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, p_{jk}, and presented in rank order based on probabilities which are calculated after considering a burn in period of 1600

Rank	$\mathbf{x}_j$	$\mathbf{y}_j$	p_{jk}	*Rank*	$\mathbf{x}_j$	$\mathbf{y}_j$	p_{jk}
1	2	2	0.99852	9	8	8	0.98996
2	7	7	0.99726	10	1	1	0.98978
3	3	3	0.99507	11	31	30	0.98907
4	15	21	0.99448	12	19	19	0.98890
5	4	4	0.99392	13	24	23	0.98878
6	32	31	0.99280	14	10	10	0.98438
7	6	6	0.99036	15	9	9	0.98389
8	5	5	0.99035	16	26	32	0.78763

5.3 Analysis of Protein Data in 3 Dimension Case

In this section we analyze and discuss the output of protein data in 3 dimension case using the $\mathcal{GM}$, $\mathcal{MVGM}$ and $\mathcal{PMAMM}$ approaches. Markov chain Monte Carlo needs greater effort to align two proteins in a protein data set than the problem of aligning two gels in gel data set where $\boldsymbol{A}$ is held fixed, as simultaneous inference for the rotation $\boldsymbol{A}$, matching matrix $\boldsymbol{M}$, $\boldsymbol{\tau}$ and σ are required to align proteins. In such a case, in addition to an efficient matching sampler is required an efficient sampler for simulating rotation matrix $\boldsymbol{A}$ to achieve convergence quickly. Furthermore, the more sophisticated Markov chain Monte Carlo may be required for severe multi-modal posterior distributions, for example when the full conditional posterior for $\boldsymbol{M}$ and $\boldsymbol{\tau}$ given $\boldsymbol{A}$ strongly depends on $\boldsymbol{A}$.

Habeck (2009) developed an algorithm based on MCMC to generate a three

dimensional rotation matrix, and its efficiency over Green and Mardia (2006) and random walk Metropolis approaches for generating three dimensional rotation matrix was shown in a protein structure superposition problem under a Procrustes setup (labelled shape). Latter in, Kent et al. (2013) developed an algorithm to generate a three dimensional rotation matrix based on a rejection scheme which produces independent realizations compared to correlated realizations produced by any methods based on MCMC algorithms. To the best of our knowledge there is no study so far regarding how these simulation techniques for three dimensional rotation perform in unlabelled shape problems where an efficient matching sampler has an impact on $\boldsymbol{A}$ or vice verse. Therefore, to align two proteins we use $\mathcal{GM}$ framework, $\mathcal{MGM}(.)$ framework, $\mathcal{PMAMM}(.)$ framework and $\mathcal{PMAMM}^{\star\star}$ framework.

5.3.1 Experimental Results for $\mathcal{MVGM}$ and $\mathcal{GM}$ Approaches

The Markov chain Monte Carlo sampler described in Chapter 1 was run for 1000,000 sweeps with d=3 under both $\mathcal{GM}$ and $\mathcal{MGM}(.)$ frameworks to get the MCMC output while prior and hyper prior settings both for $\mathcal{GM}$ and $\mathcal{MGM}(.)$ approaches are kept same as in Green and Mardia (2006), except $\mu_{\boldsymbol{\tau}} = (33.90, 3.39, 22.33)^T$ instead of $\mu_{\boldsymbol{\tau}} = (0, 0, 0)^T$, where the quantity $(33.90, 3.39, 22.33)^T$ is the value of difference of two configurations centers . Performances of $\mathcal{GM}$ and $\mathcal{MGM}(.)$ approaches were evaluated based on how quickly the Markov chain converges to the posterior distribution, acceptance rate of proposed moves of $\boldsymbol{M}$ and computing time. We repeated our experiment 100 times with the same parameter settings but with different random starting points for $\boldsymbol{\tau}$ and $\boldsymbol{A}$, to investigate the multi-modality of the posterior and compute average estimates of the above quantities.

Table 5.11: Comparison of $\mathcal{MGM}(.)$ and $\mathcal{GM}$ approaches based on MCMC outputs for protein data; mean acceptance rate is calculated after considering burn in

Sampler	Mean Convergence Iterations (min, max)	Mean Computing Time in seconds	Mean Acceptance Rate (min, max)	Successful Runs (out of 100 runs)
$\mathcal{GM}$(1 update)	146,079 (10,000, 550,000)	11.36	0.005645 (0.00534, 0.00605)	44 %
$\mathcal{GM}$(10 update)	43,153 (500, 260,000)	22.72	0.00421(0.0006378, 0.0113)	52 %
$\mathcal{MGM}$(K)	84,130 (10, 490,000)	43.92	0.01775 (0.0127, 0.0243)	46 %
$\mathcal{MGM}$(G)	126,696 (10, 575, 000)	57.20	0.017842 (0.0163, 0.0222)	46 %
$\mathcal{MGM}$(H)	134,630 (10, 540,000)	42.45	0.017846 (0.01698, 0.02692)	44 %

Results for aligning two proteins using $\mathcal{GM}$ and $\mathcal{MGM}(.)$ frameworks are presented in Table 5.11 and in Figure 5.13. From Table 5.11, we can see that on average the $\mathcal{MGM}$(K) framework takes fewer iterations (84,130) compared to $\mathcal{MGM}$(G), $\mathcal{MGM}$(H) and $\mathcal{GM}$ with 1 update approaches, but it is the second best as far as mean convergence iterations is concerned when $\mathcal{GM}$ approach with 10 update is considered, which required on average 43,153 iterations to converge. In addition to the mean convergence iterations, on average all $\mathcal{MGM}$ (.) instances have higher acceptance rate of $\boldsymbol{M}$ (around 1.78 %) compared to the acceptance rate of $\boldsymbol{M}$ for $\mathcal{GM}$ framework with 1 and 10 updates (around 0.421 % and 0.566 % respectively), while all $\mathcal{MGM}(.)$ approaches took on average almost two times longer in computing time than the $\mathcal{GM}$ approach with 10 updates (23 seconds).

Moreover, the last column of Table 5.11 shows the percentage of successful runs for each method, which are determined based on how many runs out of 100 runs the unnormalized log-posteriors converged. From the log-posterior plots of each method, we have observed that whenever the Markov chains converge they are converged to the same place (shown in Figure 5.13, main mode of the posterior) which justify the Markov Chains are not trapped in a local mode of the posterior and conclude that it is safe to draw inference from these results. $\mathcal{GM}$ with 10 update to $\boldsymbol{M}$ has the highest percentage of successful runs, 52 %, while in other cases it is around 45 %.

The top left plot of Figure 5.13 plots the three log-posteriors for $\mathcal{MGM}$(K) and $\mathcal{GM}$ with one and 10 updates respectively, where only one particular run for each approach which took maximum number of iterations to converge is considered. Here we considered only the $\mathcal{MGM}$(K) variant among all three variants of $\mathcal{MGM}(.)$ to plot as it performed better than the other two. From the log-posterior plot, it is evident that $\mathcal{MGM}$(K) and $\mathcal{GM}$ with 10 and 1 update needed

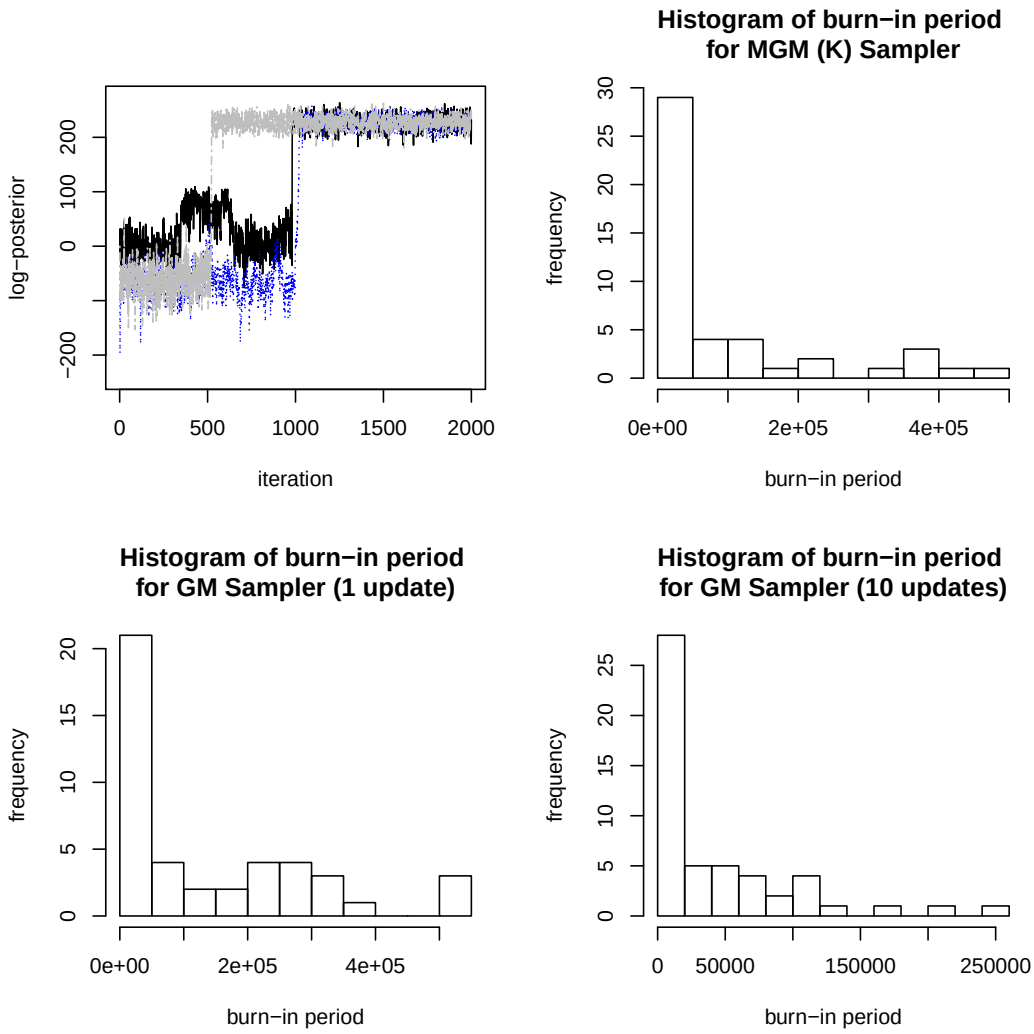

Figure 5.13: Output from the MCMC algorithm for matching protein data using the $\mathcal{GM}$ and $\mathcal{MGM}(.)$ approaches. The top left plot is the log-posteriors (up to normalizing constant) values versus iteration number for $\mathcal{GM}$ and $\mathcal{MGM}(.)$ approaches where every 500^{th} simulated value is plotted; results obtained using $\mathcal{MGM}(\text{K})$ approach are shown as solid black line while gray two dash and blue dotted lines are used to show the results for $\mathcal{GM}$ approach for 10 update and 1 update respectively. The top right plot shows the histogram of burn-in periods of $\mathcal{MGM}(\text{K})$ approach while the histograms of burn-in periods of $\mathcal{GM}$ approach are shown by the bottom row of Figure 5.13.

a minimum of at least 490,000, 260,000 and 550,000 iterations as burn in periods respectively, while histograms in the same Figure for $\mathcal{MGM}$(K) and $\mathcal{GM}$ with one and 10 update approaches show the distribution of convergence iterations for 46, 52 and 44 runs respectively.

For the unknown parameters $\boldsymbol{\tau}$, $\boldsymbol{A}$, and $\boldsymbol{M}$ we obtained their point estimate $\hat{\boldsymbol{\tau}}_{\mathcal{MGM}(\mathcal{K})}$, $\hat{\boldsymbol{\tau}}_{\mathcal{GM}}$, $\hat{\boldsymbol{A}}_{\mathcal{MGM}(\mathcal{K})}$, $\hat{\boldsymbol{A}}_{\mathcal{GM}}$, $\hat{\boldsymbol{M}}_{\mathcal{MGM}(\mathcal{K})}$ and $\hat{\boldsymbol{M}}_{\mathcal{GM}}$ by maximizing the overall joint posterior i.e. their posterior modes. The posterior mode of $\boldsymbol{\tau}$ and $\boldsymbol{A}$ under $\mathcal{MGM}$ (K) and $\mathcal{GM}$ are $\hat{\boldsymbol{\tau}}_{\mathcal{MGM}(\mathcal{K})}=(31.58, 8.89, 17.44)^{\mathrm{T}}$, $\hat{\boldsymbol{\tau}}_{\mathcal{GM}} = (31.72, 8.93, 17.21)^{\mathrm{T}}$:

$$\hat{\boldsymbol{A}}_{\mathcal{MGM}(\mathcal{K})} = \begin{bmatrix} 0.4318 & -0.8620 & 0.2653 \\ -0.7055 & -0.5061 & -0.4959 \\ 0.5618 & 0.02693 & -0.8268 \end{bmatrix}, \tag{5.1}$$

and

$$\hat{\boldsymbol{A}}_{\mathcal{GM}} = \begin{bmatrix} 0.4245 & -0.8776 & 0.2226 \\ -0.7151 & -0.4758 & -0.5120 \\ 0.5553 & 0.05816 & -0.8296 \end{bmatrix}, \tag{5.2}$$

while $\hat{\boldsymbol{M}}_{\mathcal{MGM}(\mathcal{K})}$ and $\hat{M}_{\mathcal{GM}}$ are displayed in Table 5.12 and Table 5.13 respectively. From Table 5.12 and 5.13, we see that both the approaches have reached their posterior maximum at the same point $\boldsymbol{M}$, which has L=36 matched pairs.

Table 5.12: Posterior mode of $\boldsymbol{M}$ for protein data using the $\mathcal{MGM}$(K) approach in terms of probability of matching landmarks $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, p_{jk}, and presented as in rank order based on probabilities which are calculated after considering a burn in period of 490000.

Rank	$\mathbf{x}_j$	$\mathbf{y}_k$	p_{jk}	*Rank*	$\mathbf{x}_j$	$\mathbf{y}_k$	p_{jk}
1	10	9	0.99936	19	16	15	0.99786
2	7	7	0.99905	20	15	14	0.99783
3	18	17	0.99904	21	27	29	0.99765
4	37	51	0.99903	22	9	8	0.99758
5	23	25	0.99882	23	33	47	0.99749
6	35	49	0.99877	24	20	19	0.99727
7	36	50	0.99869	25	2	2	0.99696
8	24	26	0.99862	26	3	3	0.99654
9	6	6	0.99862	27	38	52	0.99653
10	1	1	0.99855	28	22	23	0.99587
11	17	16	0.99847	29	29	34	0.99548
12	4	4	0.99846	30	30	37	0.99400
13	34	48	0.99824	31	31	41	0.99226
14	26	28	0.99823	32	11	10	0.98947
15	25	27	0.99802	33	28	33	0.98754
16	5	5	0.99798	34	14	13	0.98154
17	19	18	0.99795	35	12	11	0.88398
18	32	46	0.99794	36	13	12	0.87264

Table 5.13: Posterior mode of $\boldsymbol{M}$ for protein data using the $\mathcal{GM}$ approach with one update per sweep in terms of probability of matching landmarks $\boldsymbol{x}_j$ and $\boldsymbol{y}_k$, p_{jk}, and presented in rank order based on probabilities which are calculated after a burn in of 550000.

Rank	$\mathbf{x}_j$	$\mathbf{y}_k$	p_{jk}	*Rank*	$\mathbf{x}_j$	$\mathbf{y}_k$	p_{jk}
1	26	28	0.99964	19	20	19	0.99731
2	1	1	0.99947	20	9	8	0.997086
3	17	16	0.99928	21	27	29	0.99707
4	15	14	0.99925	22	24	26	0.99660
5	34	48	0.99920	23	6	6	0.99618
6	37	51	0.99920	24	29	34	0.99606
7	4	4	0.99900	25	7	7	0.995957
8	23	25	0.99874	26	22	23	0.99570
9	35	49	0.99859	27	33	47	0.99562
10	2	2	0.998427	28	3	3	0.99483
11	19	18	0.99841	29	31	41	0.99369
12	36	50	0.99841	30	38	52	0.99294
13	5	5	0.99830	31	30	37	0.99031
14	25	27	0.99786	32	28	33	0.98908
15	10	9	0.99780	33	11	10	0.98458
16	18	17	0.99754	34	14	13	0.97940
17	16	15	0.99751	35	12	11	0.88983
18	32	46	0.99744	36	13	12	0.88141

5.3.2 Performance of $\mathcal{PMAMM}(.)$ over MVGM and GM Approaches

For the analysis of protein data using $\mathcal{PMAMM}(.)$, we run the Markov chain Monte Carlo sampler for 1000 000 sweeps with d=3 to get the MCMC output. Prior for $\boldsymbol{\tau}$ and σ are kept same as in Green and Mardia (2006) except $\mu_{\boldsymbol{\tau}} = (29.0, 0, 26.30)^T$ instead of $\mu_{\boldsymbol{\tau}} = (0, 0, 0)^T$ while we use uniform prior for all θ_j i.e. $\boldsymbol{\eta}_j$= $(1, 1, \ldots, \ldots, 1)$ for $j = (1, 2, \ldots, m)$. For this analysis, the coffin bin parameter $|\Omega|$=10000, and we also consider two cases in our proposed matching sampler; update one and ten $\boldsymbol{\theta}$ instead of updating all $\boldsymbol{\theta}$ per iteration, which are selected randomly. We repeat our experiment 100 times with the same parameters settings but with different starting points for $\boldsymbol{\tau}$ and $\boldsymbol{A}$ to investigate the multi-modality of the posterior as well as compute estimates of the above quantities.

Table 5.14 and Figure 5.14 present the results for aligning two proteins using $\mathcal{PMAMM}(.)$, and hence we can see that all earlier results obtained under $\mathcal{GM}$ with 1 and 10 updates and all $\mathcal{MGM}(.)$ variants are now superseded by all $\mathcal{PMAMM}(.)$ variants, although all $\mathcal{PMAMM}(.)$ variants need more computing time compared to $\mathcal{GM}$ with one and ten updates and all $\mathcal{MGM}(.)$ variants respectively to implement one run. More specifically, $\mathcal{PMAMM}(.)$ with one and ten $\boldsymbol{\theta}$ updates needs on average a maximum of 3100 and 24,000 iterations compared to GM with one and ten updates and $\mathcal{MGM}(.)$ approaches which take on an average 146,079, 43,153 and 134,630 (maximum) iterations respectively to converge. In addition, all $\mathcal{PMAMM}(.)$ variants converge in all 100 runs to the same value of log-posterior (not trapped) i.e. percentage of successful runs of $\mathcal{PMAMM}(.)$ is 100 % while it is around 50 % for $\mathcal{GM}$ with one and ten updates and $\mathcal{MGM}(.)$ approaches. However, all $\mathcal{PMAMM}(.)$ variants need almost 7 times higher computing time compared to $\mathcal{GM}$ approaches irrespective of number of updates while $\mathcal{PMAMM}(.)$ with one $\boldsymbol{\theta}$ update requires slightly higher computing time compared to $\mathcal{MGM}()$ approaches.

Table 5.14: Comparison of $\mathcal{PMAMM}(.)$, $\mathcal{MGM}(.)$ and $\mathcal{GM}$ approaches based on MCMC outputs for protein data analysis; mean convergence iterations and time are calculated based on 100 runs each of size 1000,000

Sampler	Mean Convergence Iterations (min, max)	Mean Computing Time in seconds	Successful Runs (out of 100 runs)
$\mathcal{GM}$(1 update)	146,079 (10,000, 550,000)	11.36	44 %
$\mathcal{GM}$(10 update)	43,153 (500, 260,000)	22.72	52 %
$\mathcal{MGM}$(K)	84,130 (10, 490, 000)	43.92	46 %
$\mathcal{MGM}$(G)	126,696 (10, 575, 000)	57.20	46 %
$\mathcal{MGM}$(H)	134,630 (10, 540,000)	42.45	44 %
$\mathcal{PMAMM}$(K)(1 θ update)	22,678 (1000, 100,005)	53.50	100 %
$\mathcal{PMAMM}$(K)(10 θ update)	2281 (10, 9,505)	140.91	100 %
$\mathcal{PMAMM}$(H)(1 θ update)	23,770 (400, 122,010)	52.6	100 %
$\mathcal{PMAMM}$(H)(10 θ update)	2299 (10, 11,511)	139.92	100 %
$\mathcal{PMAMM}$(G)(1 θ update)	20,798 (700, 120,509)	68.73	100 %
$\mathcal{PMAMM}$(G)(10 θ update)	3076 (50, 16006)	155.30	100 %

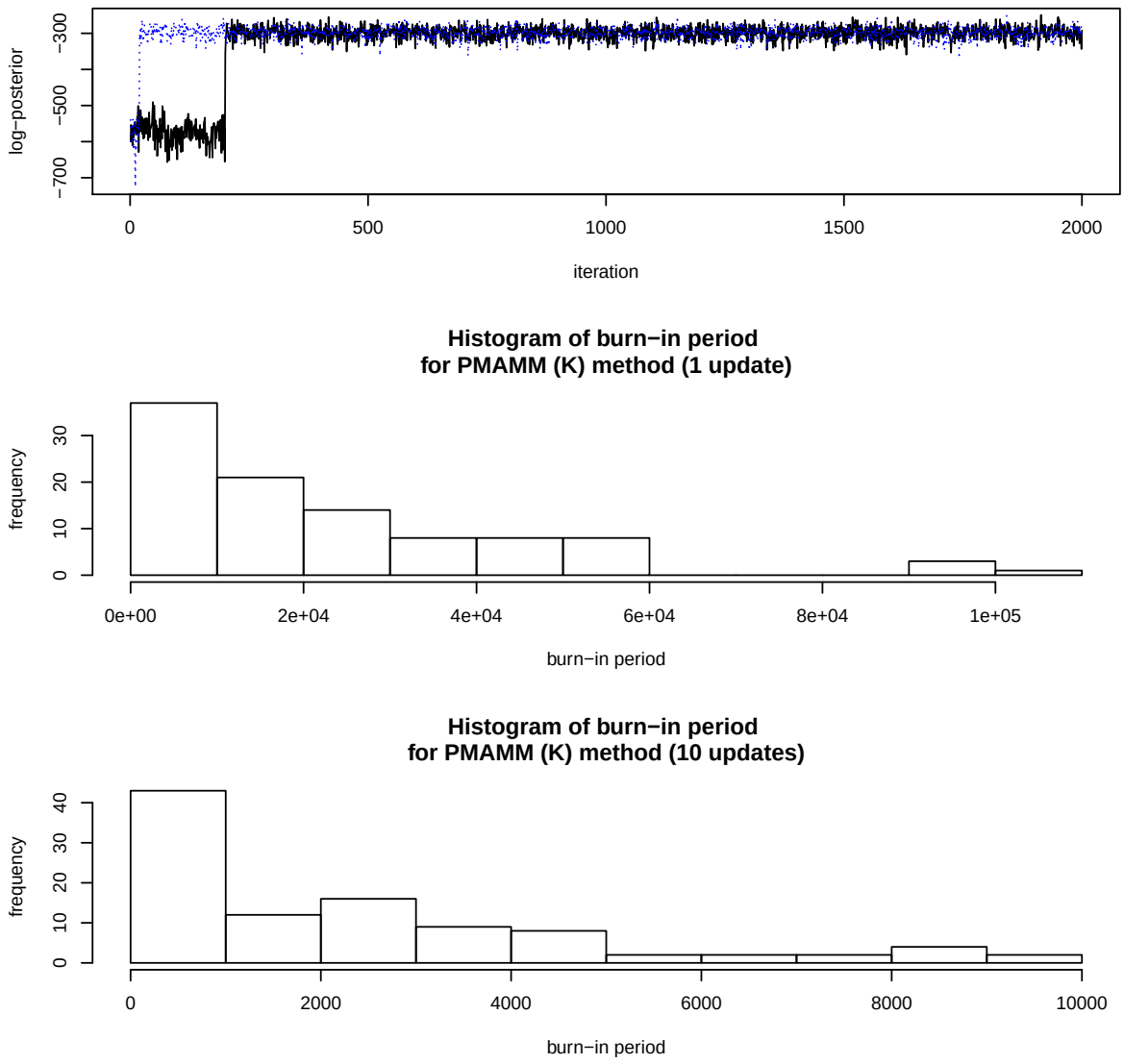

Figure 5.14: Output from the MCMC algorithm for matching protein data using $\mathcal{PMAMM}$(K): results obtained using $\mathcal{PMAMM}$(K) (1 θ update) are shown as solid black lines while dotted blue line is used to show the results for $\mathcal{PMAMM}$(K) approach (10 $\boldsymbol{\theta}$ updates). The top graph of Figure 5.14 displays the log-posterior values (up to a normalizing constant) where every 500^{th} simulated value is plotted and the histograms of burn-in periods for $\mathcal{PMAMM}$(K) using 100 runs each of size 1000,000.

The $\mathcal{PMAMM}$(K), $\mathcal{PMAMM}$(H) and $\mathcal{PMAMM}$(G) have shown very similar performance both for one and ten $\boldsymbol{\theta}$ update cases but $\mathcal{PMAMM}$(G) needs slightly more computing time (by 15 seconds) than the other two variants, as updating rotation using geodesic Monte Carlo approach requires computation of a matrix exponential in the proposal steps, as well as tuning. Although $\mathcal{PMAMM}$(K), $\mathcal{PMAMM}$(H) and $\mathcal{PMAMM}$(G) have very similar performance, we think $\mathcal{PMAMM}$(K) is the best among all 3 variants as it requires a smaller burn in period compared to other two variants to converge in both one and ten $\boldsymbol{\theta}$ update cases. Therefore, we consider all outputs obtained by $\mathcal{PMAMM}$(K) to plot Figure 5.14, which has a similar interpretation to Figure 5.13.

For the unknown parameters $\boldsymbol{\tau}$, $\boldsymbol{A}$ and $\boldsymbol{M}$ we obtained their point estimate $\hat{\boldsymbol{\tau}}_{\mathcal{PMAMM}(\mathcal{K})}$, $\hat{\boldsymbol{A}}_{\mathcal{PMAMM}(\mathcal{K})}$ and $\hat{\boldsymbol{M}}_{\mathcal{PMAMM}(\mathcal{K})}$ by maximizing the overall joint posterior i.e. their posterior modes. The posterior mode of $\boldsymbol{\tau}$ and $\boldsymbol{A}$ under $\mathcal{PMAMM}$(K) are $\hat{\boldsymbol{\tau}}_{\mathcal{PMAMM}(\mathcal{K})} = (31.29, 8.80, 17.28)^{\mathrm{T}}$ and

$$\hat{\boldsymbol{A}}_{\mathcal{PMAMM}(\mathcal{K})} = \begin{bmatrix} 0.4463 & -0.8370 & 0.3165 \\ -0.6982 & -0.5469 & -0.4618 \\ 0.5597 & -0.0148 & -0.8285 \end{bmatrix} \tag{5.3}$$

respectively, while $\hat{\boldsymbol{M}}_{\mathcal{PMAMM}(\mathcal{K})}$ is displayed by Table 5.15. From Table 5.15, we see that the $\mathcal{PMAMM}$(K) approach has reached posterior maximum at the same point of $\boldsymbol{M}$ as $\mathcal{GM}$ and $\mathcal{MGM}$(K) approaches, which have L=36 matched pairs.

Table 5.15: Posterior mode of $\boldsymbol{M}$ for protein data using the $\mathcal{PMAMM}$(K) approach in terms of probability of matching landmarks x_j and y_k, p_{jk}, and presented in rank order based on probabilities which are calculated considering a burn in period of 9505.

Rank	$\mathbf{x}_j$	$\mathbf{y}_k$	p_{jk}	*Rank*	$\mathbf{x}_j$	$\mathbf{y}_k$	p_{jk}
1	17	16	0.99805	19	36	50	0.99327
2	1	1	0.99794	20	30	37	0.99272
3	34	48	0.99769	21	27	29	0.992408
4	25	27	0.99735	22	33	47	0.99169
5	26	28	0.9973	23	33	47	0.99749
6	7	7	0.99689	24	15	14	0.99072
7	18	17	0.99677	25	38	52	0.99069
8	19	18	0.99646	26	14	13	0.98961
9	23	25	0.99625	27	31	41	0.98895
10	20	19	0.99621	28	6	6	0.98863
11	35	49	0.99599	29	4	4	0.98797
12	22	23	0.99556	30	11	10	0.98695
13	2	2	0.99540	31	9	8	0.98678
14	24	26	0.99540	32	28	33	0.98373
15	10	9	0.99497	33	3	3	0.97874
16	37	51	0.99496	34	32	46	0.96737
17	16	15	0.99466	35	12	11	0.91458
18	29	34	0.99376	36	13	12	0.60848

5.3.3 Performances of $\mathcal{PMAMM}$ under Product Manifold Inference Technique

In this section, we will present the results of $\mathcal{PMAMM}$ approach for protein data analysis where both rotation ($\boldsymbol{A}$) and matching probabilities ($\boldsymbol{\theta}$) are updated together using a geodesic Monte Carlo approach based on a product manifold, along with results of $\mathcal{PMAMM}(.)$ shown in the previous subsection, and we use the notation $PMAMM^{\star\star}$ to denote this.

From Table 5.16, we can see that the $\mathcal{PMAMM}^{\star\star}$ requires slightly more iterations compared to all $\mathcal{PMAMM}(.)$ variants to converge, irrespective of number of updates of $\boldsymbol{\theta}$. More specifically, $\mathcal{PMAMM}^{\star\star}$ needs on average more than 7000 and 1000 iterations compared to all variants of $\mathcal{PMAMM}(.)$ in 1 and 10 $\boldsymbol{\theta}$ update cases respectively. In addition, $\mathcal{PMAMM}^{\star\star}$ takes on average 16,464 seconds (272 minutes) to execute a run of size 1000,000 compared to 140 seconds in $\mathcal{PMAMM}(.)$ variants (10 $\boldsymbol{\theta}$ update case), which is a huge computational burden for $\mathcal{PMAMM}^{\star\star}$, which comes from the execution step of geodesic updates both for $\boldsymbol{A}$ and each ($\boldsymbol{\theta}$) which need computation of a matrix exponential, as well as from tuning.

For $\mathcal{PMAMM}^{\star\star}$, we haven't plotted the unnormalized log-posterior plot here as $\mathcal{PMAMM}^{\star\star}$ converges to the same posterior value as $\mathcal{PMAMM}$ (.), which is already shown in Figure 5.14. Furthermore, for $\mathcal{PMAMM}^{\star\star}$ we also have very similar estimated values of $\hat{\boldsymbol{\tau}}$, $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{M}}$ (same matched pairs) as $\mathcal{PMAMM}(.)$ which are also omitted here.

Finally, we can summarize the above results and discussions for both the protein and gel data analysis as follows: (1.a) For gel data $\mathcal{MGM}$ is not only on average almost 12 times faster than $\mathcal{GM}$ with one update to $\boldsymbol{M}$ but also has

better acceptance rate of $\boldsymbol{M}$, while both $\mathcal{MGM}$ and $\mathcal{GM}$ with ten update to $\boldsymbol{M}$ are equally fast but the latter has low acceptance rate compared to $\mathcal{MGM}$, (1.b) For the same data set $\mathcal{PMAMM}^{\star}$ with ten $\boldsymbol{\theta}$ updates is on average 3 and 4 times faster than $\mathcal{MGM}$ and $\mathcal{GM}$ with the same update to $\boldsymbol{M}$ respectively, although $\mathcal{PMAMM}^{\star}$ with single $\boldsymbol{\theta}$ update is on average 3 times slower than $\mathcal{MGM}$. In addition, in $\mathcal{PMAMM}^{\star}$ there is no issue of acceptance rate of $\boldsymbol{M}$, and we ignore here the computing time for all cases as these are not expensive to compute at all.

(2.a) Similarly for the protein data $\mathcal{MGM}$(K) is not only on average almost 1.7 times faster than $\mathcal{GM}$ with one update to $\boldsymbol{M}$, but also has better acceptance rate of $\boldsymbol{M}$, while $\mathcal{GM}$ with ten update to $\boldsymbol{M}$ is almost 2 times faster than $\mathcal{MGM}$ (K) but has low acceptance rate compared to $\mathcal{MGM}$ (K). (2.b) For the same data set $\mathcal{PMAMM}$(K) with one $\boldsymbol{\theta}$ update is on average 4 and 7 times faster than $\mathcal{MGM}$(K) and $\mathcal{GM}$ with the same update to $\boldsymbol{M}$ but $\mathcal{PMAMM}$(K) with ten $\boldsymbol{\theta}$ update is on average much faster than both $\mathcal{MGM}$(K) and $\mathcal{GM}$ with the same update to $\boldsymbol{M}$ (37 and 19 times faster respectively).

(3.a) $\mathcal{PMAMM}^{\star\star}$ is not only slightly slower than all $\mathcal{PMAMM}$(.) variants but brings huge computational cost irrespective of number of $\boldsymbol{\theta}$ update per iteration.

In conclusion, the best methods based on the above results are $\mathcal{PMAMM}^{\star}$ and $\mathcal{PMAMM}$(K) for the gel and protein data respectively.

Table 5.16: Comparison between $\mathcal{PMAMM}(.)$ and $\mathcal{PMAMM}^{\star\star}$ approaches based on MCMC output for protein data analysis; mean convergence iterations and time are calculated based on 100 runs each of size 1000,000.

Sampler	Mean Convergence Iterations (min, max)	Mean Computing Time in seconds	Successful Runs (out of 100 runs)
$\mathcal{PMAMM}$(K) (1 $\boldsymbol{\theta}$ update)	22,678 (1000, 100,005)	53.50	100 %
$\mathcal{PMAMM}$(K) (10 $\boldsymbol{\theta}$ update)	2281 (10, 9,505)	140.91	100 %
$\mathcal{PMAMM}$(H) (1 $\boldsymbol{\theta}$ update)	23,770 (400, 122,010)	52.6	100 %
$\mathcal{PMAMM}$(H) (10 $\boldsymbol{\theta}$ update)	2299 (10, 11,511)	139.92	100 %
$\mathcal{PMAMM}$(G) (1 $\boldsymbol{\theta}$ update)	20,798 (700, 120,509)	68.73	100 %
$\mathcal{PMAMM}$(G) (10 $\boldsymbol{\theta}$ update)	3076 (50, 16006)	155.30	100 %
$\mathcal{PMAMM}^{\star\star}$ (1 $\boldsymbol{\theta}$ update)	30,697 (1000, 125,000)	1822	100 %
$\mathcal{PMAMM}^{\star\star}$ (10 $\boldsymbol{\theta}$ update)	3878 (150, 19500)	16,464	100 %

Chapter 6

Conclusion

In this thesis we have presented a fully Bayesian model for the pairwise alignment of two unlabeled configurations in the size-and-shape space, along with implementation methods. The fully Bayesian model makes the parameter space more continuous by directly modeling the matching probabilities, which is a key conceptual difference between our model and other existing models.

We first reviewed Bayesian statistics in section 1 of Chapter 1 followed by a detailed review of two existing models for the pairwise alignment of two unlabelled configurations in the size-and-shape space. We highlighted there how knowledge of Bayesian statistics was required to understand the implementation methods of models developed for shape analysis. We also covered some ideas from differential geometry that have been used in Chapter 4.

From a review of existing models, we saw in Green and Mardia (2006) model that the proposed modifications of matching matrix $\boldsymbol{M}$ are small, local changes, with low acceptance rate under Metropolis-Hastings based MCMC sampler for sampling matching matrix $\boldsymbol{M}$, and thus the overall MCMC sampler can be slow to mix (converge) in large problems. In addition, the sampler could be trapped in local modes. We then saw in Chapter 2, how the use of a new sampler for $\boldsymbol{M}$ proposed by Forbes and Lauritzen (2014) within the Green and Mardia (2006) framework can improve the overall performance as far as convergence time and

acceptance rate of $\boldsymbol{M}$ are concerned (shown in Chapter 5). However, there is much scope for improving the performance of the MCMC sampler further.

On the other hand, the Taylor, Mardia and Kent (2003) regression model converged quickly provided that EM starts from good initial values. However, when good starting values are not available, MCMC is more effective than EM algorithm. In addition, results obtained using EM don't provide any confidence measure of the estimated values as only the maximum of the likelihood is obtained. Chapter 3 presented our proposed fully Bayesian model, which is developed by generalising the Taylor, Mardia and Kent (2003) model, where matching probabilities are modelled directly instead of using matching probabilities to replace the unknown mapping in the E step. We have also shown how these matching probabilities are sampled from their full conditional distributions using Gibbs sampling by introducing a conjugate prior, which dramatically improves the performance of the sampler (shown in Chapter 5).

The overall performance of the MCMC sampler not only depends on efficient sampling of matching matrix $\boldsymbol{M}$ but also an efficient sampling from the full conditional distribution of rotation $\boldsymbol{A}$. Thus an algorithm was developed in Chapter 4 to sample $\boldsymbol{M}$ (matching probabilities $\boldsymbol{\theta}$) and $\boldsymbol{A}$ together using geodesic Monte Carlo algorithm from their joint full conditional distribution, defined on product manifold, $\mathbb{SO}(3) \times \prod_{i=1}^{\mathrm{m}} \triangle_i{}^n$. In addition, we also determined rules of thumb for tuning parameters of geodesic Monte Carlo algorithm for generating samples from the matrix Fisher distribution ($\mathbb{SO}(3)$) and simplex ($\triangle$) separately by conducting a simulation study, which play the role of building blocks of tuning geodesic Monte Carlo on a product manifold.

The final chapter presented simulation results and discussion, for gel and protein data, obtained under different methods considered in this thesis.

6.1 Future Work and Extensions

From a literature review, we have seen that much work related to statistical shape analysis is designed for aligning two point configurations or multiple configurations under rigid body or similarity transformations. For example, the two models discussed in Chapter 1 are designed for aligning two point configurations under rigid body transformations. Later, Ruffieux and Green (2009) and Mardia, K. V., Fallaize, C. J., Barbar, S., Jackson, R. M. and Theobold, D. L. (2013) extended the Green and Mardia (2006) model to develop a fully Bayesian model for the alignment of multiple configurations (under rigid body transformation) and pairwise configurations (under similarity transformations) respectively.

Although their generalized models cover pairwise (multiple) configurations under similarity transformations (rigid transformation), they didn't concentrate on efficient sampling from the joint posterior. Therefore, incorporating the methodology of our proposed Bayesian model discussed in this thesis within their framework could be a natural extension. In addition, sampling matching probabilities ($\boldsymbol{\theta}$) and rotation $\boldsymbol{A}$ together using geodesic Monte Carlo based on product manifold under our proposed Bayesian model is computationally expensive even for $m = n = 98$ landmarks. Therefore, future research could focus on developing an alternative inference technique, which might turn out to be computationally more efficient than geodesic Monte Carlo, along with investigating how the performance of the proposed model scale up with the increases of number of landmarks.

Bibliography

[1] Abraham, R. and Marsden, J. E. (1978). **Foundations of Mechanics, (2nd ed.)**. *Benjamin/Cummings Publishing Company Incorporated Advanced Book Program, Reading, Massachussetts, ISBN 0-8053-0102-X*.

[2] Amari, S. and Nagaoka, H. (2000). **Methods of Information Geometry, Translations of Mathematical Monographs**. *Oxford University Press*. **191**.

[3] Best, D. J. and Fisher, N. I. (1979). **Efficient Simulation of the von Mises Distribution**. *Applied Statistics*. **28**(2), 152-157.

[4] Bingham, C. (1974). **An Antipodally Symmetric Distribution on the Sphere**. *The Annals of Statistics*. **2**(6), 1201-1225.

[5] Byrne, S. and Girolami, M. (2013). **Geodesic Monte Carlo on Embedded Manifolds**. *Scandinavian Journal of Statistics*. **40**(4), 825-845.

[6] Calin, O. and Chang, D. E (2005). **Geometric Mechanics on Riemannian Manifolds: Applications to Partial Differential Equations**. *Birkhauser*.

[7] Chui, H. and Rangarajan, A. (2003). **A New Point Matching Algorithm for Non-Rigid Registration**. *IEEE Computer Vision and Understanding*, **89**(2-3), 114–141.

[8] Cross, A. D. J. and Hancock, E. R. (1998). **Graph Matching with a Dual-Step EM Algorithm**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **20**(11), 1236-53.

[9] Dryden, I. L. and Mardia, K. V. (1998). **Statistical Shape Analysis**, *John Wiley and Sons, Chichester*.

[10] Eidhammer, T., Jonassen, T. and Taylor, W.R. (2004). **Protein Bioinformatics**. *Chichester: Wiley*.

[11] Forbes, P. G. M. and Lauritzen, S. (2013). **Fingerprint Analysis Using Bayesian Alignment**. *In Proceedings of the Leeds Annual Statistics Research Workshop*.

[12] Geman, S. and Geman, D. (1984). **Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images**. *IEEE Transaction on Pattern Analysis and Machine Intelligence*. **6**(6), 721–741.

[13] Geyer, C. J. (1991). **Markov chain Monte Carlo Maximum Likelihood**. *Interface Foundation of North America*.

[14] Glasbey, C. A. and Mardia, K. V. (2001). **A Penalized Approach to Image Warping (with discussion)**. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. **63**(3), 465-514.

[15] Green, P. J. and Mardia, K. V. (2006). **Bayesian Alignment Using Hierarchical Models, with Applications in Proteins Bioinformatics**. *Biometrika*. **93**(2), 235-254.

[16] Habeck, M. (2009). **Generation of Three-Dimensional Random Rotations in Fitting and Matching Problems**. *Computational Statistics*. **24**(4), 719–731.

[17] Hastings, W. (1970). **Monte Carlo Sampling Methods using Markov chains and Their Applications**. *Biometrika*. **57**(1), 97–109.

[18] Huelsenbeck, J. P. and Ronquist, F. (2001). **Bayesian Inference of Phylogenetic Trees**. *Bioinformatics*. **17**(8), 754–755.

[19] Jeffreys, H. (1946).**An Invariant Form for the Prior Probability in Estimation Problems**. *Proceedings of the Royal Society of London: Series A*. **186**(1007), 453–461.

[20] Kendall, D. G. (1977). **The Diffusion of Shape**. *Advances in Applied Probability*. **9**(3), 428-430.

[21] Kent, J. T., Asaad, M. G., and Mardia, K. V. (2013). **A New Method to Simulate the Bingham and Related distributions in Directional Data Analysis with Applications**. ArXiv Preprint. http://arxiv.org/abs/1310.8110.

[22] Khatri, C. G. and Mardia, K. V. (1977). **The Von Mises–Fisher Distribution in Orientation Statistics**. *Journal of the Royal Statistical Society. Series B (Methodological)*. **39**(1), 95–106.

[23] Kume, A. and Walker, S. G. (2006). **Sampling from Compositional and Directional Distributions**. *Statistics and Computing*. **16**(3), 261–265.

[24] Lin, L., Zhu, S. C., and Wang, Y. (2007). **Layered Graph Match with Graph Editing**. *IEEE Conference Computer Vision and Pattern Recognition (CVPR)*. 1-8.

[25] Liu, J. S. (2008). **Monte Carlo Strategies in Scientific Computing**. *Springer Series in Statistics, Springer, New York, ISBN 978-0-387-76369-9; 0-387-95230-6*.

[26] Lui, Y. M (1998). **Human Gesture Recognition on Product Manifolds**. *S. Escalera et al. (eds.), Gesture Recognition, The Springer Series on Challenges in Machine Learning*.

[27] Mardia, K. V., Fallaize, C. J., Barber, S., Jackson, R. M., and Theobald, D. L. (2013). **Bayesian Alignment of Similarity Shapes**. *The Annals of Applied Statistics*. **97**(2), 989–1009.

[28] Mardia, K. V. and Jupp, P. E. (2000). **Directional Statistics, Wiley, Chichester**.

[29] Mardia K. V., Taylor, C. C, and Westhead, D. R. (2003). **Structural Bioinformatics Revisited**. *In LASR 2003*.11–18.

[30] Mardia, K. V., Walder, A. N., Xu, C., Dowd, P. A., Fowell, R. J., Nyirongo, V. and Kent, J. T.(2007). **A Line Finding Assignment Problem and Rock Fracture Modelling**. *Bayesian Statistics and its Applications, Eds. Upadhaya, S.K., Singh, U. and Dey, D.K. Anamaya Publisher, New Delhi (2007)*. 319–330.

[31] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). **Equation of State Calculations by Fast Computing Machines**. *The Journal of Chemical Physics*. **21**(6), 1087–1092.

[32] Metropolis, N. and Ulam, S. (1949). **The Monte Carlo Method**. *Journal of the American Statistical Association*. **44**(247), 335-341.

[33] Michael, B. (2012). **Cruising the Simplex: Hamiltonian Monte Carlo and the Dirichlet distribution**. *AIP Conference Proceedings 1443, 157*.

[34] Michael J. P. (1986). **Orientation Statistics Without Parametric Assumptions**. *Journal of the Royal Statistical Society: Series B (Methodological)*. **48**(2), 214-222.

[35] Neal, R. M. (2011). **MCMC Using Hamiltonian Dynamics**. *In Handbook of Markov chain Monte Carlo*. **2**(11), 2.

[36] Paine, P. J., Preston, S. P., Tsagris, M. and Wood, A. T. A. (2017). **An Elliptically Symmetric Angular Gaussian Distribution**. *Statistics and Computing*. **28**(3), 689-697.

[37] Raffenetti, R. C. and Ruedenberg, K. (1970). **Parameterization of An Orthogonal Matrix in Terms of Generalized Eulerian Angles**. *International Journal of Quantum Chemistry*. 625–34.

[38] Rao, C. R (1945). **Information and the Accuracy Attainable in the Estimation of Several Parameters**. *Calcutta Mathematical Bulletin*. **37**, 81–91.

[39] Robert, C. P. and Casella, G. (2004). **Monte Carlo Statistical Methods**. *New York: Springer*

[40] Singh, H., Hnizdo, V. and Demchuk, E. (2002). **Probabilistic Model for Two Dependent Circular Variables**. *Biometrika*. **89**(3), 719–723.

[41] Taylor, C. C., Mardia, K. V. and Kent, J. T. (2003). **Matching Unlabelled Configurations Using the EM Algorithm**. *LASR Proceedings: Stochastic Geometry, Biological Structure and Images*. 19–21.

[42] Valdarnini, R. (2001). **Detection of Non-Random Patterns in Cosmological Gravitational Clustering**. *Astronomy and Astrophysics*. **366**(2), 376–386.

[43] Wood, A. T. A. (1993). **Estimation of the Concentration Parameters of the Fisher Matrix distribution on $\mathbb{SO}(3)$ and the Bingham distribution on $\mathbb{S}_q$, $q \geq 2$**. *Australian Journal of Statistics*. **35**(1), 69-79.