PHD

An application of MCMC methods for ionospheric tomography

Khorsheed, Eman

Award date:
2007

Awarding institution:
University of Bath

An Application of MCMC Methods for Ionospheric Tomography

submitted by

Eman Khorsheed

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mathematical Sciences

November 2007

Signature of Author ....

Eman Khorsheed

Summary

Tomography is a powerful reconstruction technique for producing cross-sectional images of an object. It depends on measurements recorded along many intersecting paths between transmitters and detectors through the object. Each path is called a projection and can provide information about the object from a specific direction. Tomographic methods have found widespread application in many fields such as medical imaging, oceanography and recently in mapping the ionosphere, the subject of the present study. In all fields, the task is to estimate the distribution of a physical quantity in the object from the data.

In many applications of tomography, due to the configuration of the projection acquisition system, the data set is incomplete, and hence the imaging problem is ill-posed, as is the case in our ionospheric tomography problem with ground-based satellite-to-receiver radio signal measurements.

In this thesis, we propose a new Bayesian non-linear model for the spatial distribution of electron density in the ionosphere that can overcome the ill-posedness problem by incorporating Gaussian Markov Random Fields distributions in the formalism of the posterior model. This model has two main advantages: first, its competence in accounting for the expected spatial correlation between neighbouring regions of the ionosphere; second, it can be used to reconstruct 2D or 3D high-level images of vertical electron-density profiles.

To simulate efficiently from the posterior distribution, we develop two MCMC algorithms: the mixed algorithm and the Principal Components MCMC algorithm. Both approaches adopt reparameterisation ideas in attempts to overcome the mixing problem of the standard MCMC. To demonstrate our approaches, we use data obtained from the Navy Navigation Satellite System to produce 2D images of the ionosphere.

Acknowledgements

This research was carried out under the supervision of Professor Christopher Jennison and Dr. Merrilee Hurn. I am extremely grateful to both of them for their invaluable guidance and encouragement during my study in University of Bath.

I would like to thank University of Bahrain for their scholarship. I would also like to thank Dr. Cathryn Mitchell and Dr. Paul Spencer from Department of Electronic & Electrical engineering at University of Bath for providing the package MIDAS and the data set for this thesis.

Thanks for Mark Willis and Adam from computer support, my colleagues and staff members within the school of Mathematical sciences at the University of Bath for being there when needed.

I would specially thank my parents and brothers for their continuous support, especially my mother whose tears did not stop while being away from her. I am grateful to my husband whose love and care surrounded me while doing my course. My special thanks to my children Fatima, Ahmed, Shaima and Hawra, for being the secret of my success in this life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis objective and outline

The aim of this thesis is to develop a new approach for tackling the imaging of the spatial distribution of electron concentration in the ionosphere over a wide geographical region. The solution is to be extracted from inverting a set of satellite-to-receiver transmission measurements using MCMC techniques in a Bayesian framework.

In Chapter 2, we will describe the Earth's ionosphere, its effects on radio signals and the associated measurements, known as the Total Electron Content (TEC). A brief description of some TEC data acquisition systems is also given. Additionally, we introduce the idea of tomography, in particular ionospheric tomography.

Traditional approaches to the ionospheric imaging problem aim to invert a linear model when a set of TEC measurements is given for the unknown pixel electron densities. However, a lack of near-horizontal ray paths imposes limitations on the direct inversion leading to an ill-posed reconstruction problem. To tackle this difficulty, model regularisation techniques are required. Some of the general techniques available in the literature are described in Chapter 3, and four conventional approaches in which some of these techniques are used to regularise the ionospheric problem are reviewed. Unfortunately, almost all these inversion methods rely heavily on the initial background ionospheres used to regularise the model. In addition, some of these approaches produce negative estimates of electron densities. We will introduce a new method for regularisation based on the inclusion of some smoothing parameters and which also uses the available *a priori* information including that the density cannot be negative. Moreover, our final results will be almost independent of the initial conditions as they will only be used to start the iterative algorithm.

The development of the new Bayesian model is based on the assumption that an ionospheric scene can be viewed as a set of vertical electron-density profiles. In this model, smoothing is to be considered between neighbouring sub-cross-sectional regions of the ionosphere rather than between adjacent pixels. The formulation of this Bayesian model and the reasons justifying our choice of modelling will be stated in Chapter 4.

In Chapter 5, we shall give a general description of Markov Chain Monte Carlo sampling methodology, and discuss some associated implementation issues. We then build a standard MCMC algorithm based on implementing the Metropolis algorithm and Gibbs sampler. The standard algorithm will be tested via an illustrative example based on a set of simulated data for a 5×3 grid of pixels. To improve the mixing of the Markov chain we will include regular reparameterisation steps in the standard MCMC approach, and therefore call the resulting sampler the mixed algorithm.

To avoid a lengthy burn-in period when applying the mixed algorithm with the real data set in Chapter 6, we develop an approach to obtain a good starting state for the Bayesian model parameters using estimates of pixel electron densities gathered from the deterministic algorithm, MIDAS. Initial estimates of the smoothing parameters will be obtained by Maximum Pseudo-Likelihood (MPL). Having derived the necessary initial values, we will then implement the mixed algorithm for the real data by following two approaches. In the first approach, we will fix the smoothing parameters all the way through the MCMC simulating process, whereas in the second approach these parameters will be allowed to vary by regularly recalculating their MPL estimates.

In Chapter 7, we will develop a more sophisticated MCMC sampler to improve mixing (and call this the Principal Components MCMC algorithm). Further, we will implement this algorithm for sampling a set of Gaussian and non-Gaussian distributions with highly correlated variables. These demonstrations are considered in order to test the performance of the algorithm and to comment on its limitations.

In Chapter 8, the Principal Components MCMC algorithm will be initially applied to a simulated data set. The purpose of this study is to test the Principal Components MCMC algorithm and compare its performance with that of the standard MCMC algorithm. The new sampler is then applied to the real TEC observations.

Finally, in Chapter 9, we state the principle conclusions of this thesis and discuss the potential contributions that our Bayesian model and the proposed Principal Components MCMC algorithm could make in future research in this field.

Chapter 2

Ionospheric data and Tomography

2.1 An introduction to the ionosphere

The ionosphere is the region of the atmosphere that is ionised by solar radiation. It approximately lies between 50 km and 1000 km above the Earth's surface. Photons of energy of solar radiation at extreme ultraviolet (EUV) and other X-rays are capable of releasing an electron from a neutral gas atom or molecule during a collision. This process is called *ionisation.* The rate of ionisation is proportional to the gas density and the intensity of solar emissions. Actually, the ionosphere is a plasma of ionised gases. This plasma consists of free negative electrons and positive ions. The free electrons and ions are attracted to each other by the electromagnetic force, but are too energetic to remain fixed together in an electrically neutral molecule. The process in which a free electron is "captured" by an ion is called *recombination.*

The ionosphere is not static. This is because electron concentrations vary with geographical location, eg, auroral zones, polar, mid latitudes and equatorial region, as well as with time of the day, season and amount of radiation received from the sun. Vertically, because the intensity of the solar radiation increases with height whereas the density of neutral gas decreases, 3 main layers of ionisation usually form. These layers are:

- The D region: Between 50 and 90 km;
- The E region: Between 90 and 140 km;
- The F region: 140 km and above.

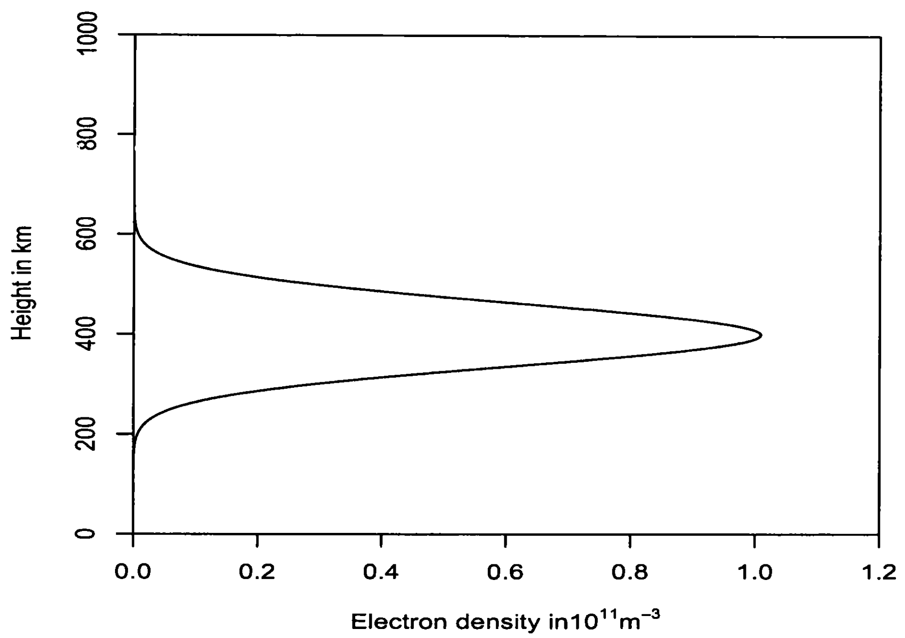


Figure 2-1: *A Vertical electron-density profile with peak height at 400 km above the Earth's surface.*

During daytime, the F region may be divided into two layers, namely, F_1 (140 - 200 km) and F_2 (200 km and above). Very often, F_1 is not well defined because it merges with F_2. The latter is the most variable region with the greatest electron density. At night, only the F_2 layer remains, but it then rises to higher altitudes and all other ionospheric regions become insignificant. This is because at this period there is no solar radiation and the recombination time of free electrons and ions is longer, due to the low gas density at these high altitudes. For a full description of the ionosphere, see Davies (1990) or Hargreaves (1992).

Within any vertical section of the ionosphere, the electron density forms approximately a profile, known as the ***Vertical electron-density profile*** or ***Chapman profile*** (Chapman, 1931). This profile is maximised at the F_2 layer producing a peak at an altitude in the range 250 - 500 km as shown in Figure 2-1. This peak is called the electron density peak. Each vertical electron-density profile is centred at the electron density peak height with variance known as the spread. Obviously this curve is scaled by the peak electron concentration.

Radio signals traversing the ionosphere are affected by the dispersive nature of ionisation present along the wave paths, with a stronger effect on low frequency signals. This effect can be seen in different forms, for instance, time delay, refraction and reflection; see, e.g., Davies (1990) for a detailed description of these phenomena. Hence, ionisation affects radio communications, navigational positioning, High frequency (HF) radar etc, in either an advantageous or disadvantageous way. For instance, HF communication is only possible because the associated signals are refracted and returned to the Earth. On the other hand, satellite and receiver navigational signals such as the L-band signals based on their time of propagation suffer a delay, causing errors (Mitchell and Spencer, 2003).

2.2 Data collection

One of the most important physical parameters in the ionosphere is the electron density, and accurate knowledge of its spatial distribution is essential for several purposes, such as prediction of the space weather effects on telecommunications, determination of the state of the ionospheric activities, computation and correction of propagation time delays in navigational systems.

In order to investigate the ionospheric effects on radio-signals, several instruments have been constructed and used to probe the ionisation structure, for example, the ionosonde which is a radar designed to determine the height of the ionospheric reflecting layers. Most of the traditional devices are restricted to either the bottom or the top side of the ionosphere, leaving more details about the entire F region (140km and above) required. This motivated the development of the most recent systems, namely, the Global Positioning System (GPS) and the Navy Navigation Satellite System (NNSS).

GPS and NNSS are navigation and positioning satellite-based systems that provide the user with velocity, time and location information in the following three dimensions: latitude, longitude and altitude. In both systems, satellites transmit dual-frequency radio signals through the ionosphere and these signals are then received by a chain of ground-based receivers. The systems are subject to several sources of error. These include ionospheric delays, satellite and receiver clock errors and receiver noise.

The GPS satellites orbit the Earth at high altitudes of approximately 20,200 km. These satellites transmit dual-frequency signals of 1572.42 MHz and 1227.60 MHz. The NNSS satellites are in near-circular polar orbits at altitudes of around 1100 km, transmitting phase coherent signals at 150 MHz and 400 MHz. The orbiting period of a GPS satellite is around twelve hours. On the other hand, a typical NNSS satellite time of one orbit is approximately twenty minutes, and hence the ionosphere can be assumed to be stationary during the radio wave transmission period.

The navigational systems' receivers record the phase and time delay of each propagated radio signal. These records give valuable information about the ionosphere in the form of ray-path integrations of electron-density measurements. For the dual-frequency based systems of navigation, the so-called differential Doppler technique is used to obtain these measurements. The method is mainly based on the differential phase of the dual radio signals. For a full detail on this method, the reader is referred to Davies (1990). Ionospheric observations obtained by such techniques are called *Total Electron Content* (TEC).

2.3 The Total Electron Content (TEC)

In satellite-to-receiver navigation systems, the dispersive effect of the ionosphere on the dual frequencies results in propagation time delays and a phase difference between the two signals. These effects are related to the electron density present along the propagation ray path, which is a fundamental ionospheric quantity known as the Total Electron Content. TEC is defined as the line integral of the electron density in a column of unit cross-sectional area along a radio-wave path. It can be expressed as

$$I = \int_r^t f(s)ds \tag{2.1}$$

where $f(s)$ is the free electron density that correspond to the vertical electron-density profile at a specific latitude θ and longitude ϕ, and s is the distance along the straight ray path between the receiver, r, and the satellite transmitter, t. The TEC is expressed in terms of TEC Units (TECU). One TECU is 10^{16} el/m^2, where el is an abbreviation of the word electron. Due to the dispersive nature of the ionosphere, the radio signals suffer from phase delays causing errors in the TEC measurements, so it is helpful to map the ionisation spatial distribution of the ionosphere over an observed geographical region of interest in order to correct these errors.

The classical approach for the ionospheric imaging problem builds a two-dimensional grid in the vertical ionospheric plane that lies above the receiver chain. This grid is subdivided into N columns and r rows. The columns are equally spaced in degrees

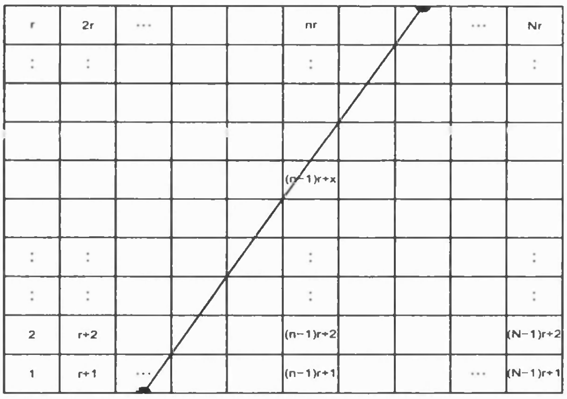


Figure 2-2: *A simple grid of pixels showing a typical satellite-to-receiver ray path.*

longitude and, due to the Earth's curvature, grow wider as the height increases. Each column contains r pixels at equally spaced altitudes. These pixels are labelled systematically by a single index $j = 1, 2, \ldots, Nr$, as shown in Figure 2-2: the pixel in row x of column n has index $j = (n-1)r + x$, for $x = 1, \ldots, r$ and $n = 1, \ldots, N$. The amount of ionisation within this pixel is assumed to be constant but is considered as an unknown quantity. Here it is denoted by d_j. We will follow this notation in our approach.

As Figure 2-2 demonstrates, for each TEC record, the transmitted signal ray traverses a finite number of pixels in the grid. Therefore, the TEC integral in Eq (2.1) can be represented by a finite sum of shorter integrals along segments of the propagation path length. With the grid of pixels covering the ionospheric plane of interest, this can be approximated by a weighted summation leading to the formation of a system of linear equations in the unknown densities of the form

$$Y_i = \sum_{j=1}^{Nr} w_{ij} d_j, \tag{2.2}$$

for $i = 1, \ldots, I$, or generally in matrix notation,

$$Y = Wd, \tag{2.3}$$

where Y is the vector of the I measurements which most engineers believe that they are always recorded without any noise, W is the known $I \times Nr$ matrix with w_{ij} being the segment length of the ith ray that lies in pixel j, and d is the vector of unknown local electron densities d_j.

It is obvious that any individual TEC measurement contains no information about the spatial variation of the amount of ionisation or electron density along the propagation path. Thus, to find the spatial distribution of the electron density a set of many TEC measurements should be recorded. This information must then be "inverted" to recover the values of the unknown number in each pixel in the grid, i.e. $\{d_j\}$. In almost all ionospheric studies, the problem is known as *Ionospheric tomography.*

Tomography is a well known technique for determining the internal structure of an object, or more precisely for determining the distribution of a physical quantity of interest. It is mainly based on using a set of measurements recorded along many intersecting paths that traverse the object in different directions. Each path is a line integral of the space-dependent quantity of interest and is known as a projection. Reconstruction based algorithms combine these projections, and the inferred estimates are used to obtain a tomographic image of the object parameter of interest.

Tomography is commonly used for medical imaging. For example, in Computer Aided Tomography, two-dimensional cross sectional images are derived from measurements of attenuation of X-rays passing through the body from several angles. Other common medical imaging techniques are Positron Emission Tomography and Single Photon Emission Computed Tomography. Some other areas to which tomography has been applied are geophysics (Gustavsson *et al.* 1986; Takauchi and Evans, 1995), oceanography (Munk and Wunch, 1979) and relatively recently to the ionosphere, as in Austen *et al.* (1988), and Pryse and Kersley (1992).

In ionospheric tomography, the total electron content (TEC) measurements are the required projections of the process. Figure 2-3 shows a ray path geometry that may be used for tomographic TEC observations. Here, the receivers are ground based and a satellite passes over the chain of receivers.

Unlike in medical tomography, for instance, in ionospheric tomography the geometry of the ionosphere and the distribution of receivers limits the angular range of the propagation rays. For example, there is a lack of paths that can run horizontally through the ionosphere with this geometry because of the curvature of the Earth, as

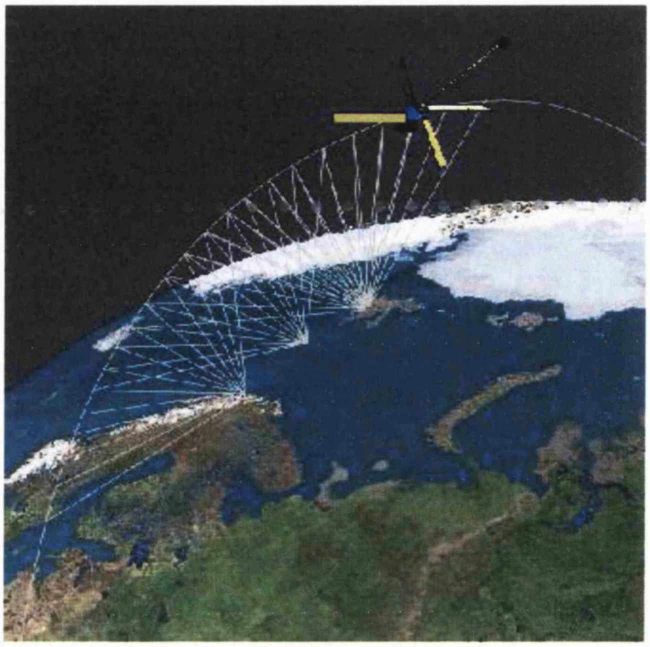


Figure 2-3: *A demonstration of the radio-signal path geometry that can be used for ionospheric tomography using the associated TEC measurements. The semi-circle line represents the trajectory of the orbiting satellite. All three receivers are ground based and the lines from these receivers to the positions of the satellite represent signal paths. This picture has been downloaded from the following website: www.aber.ac.uk/propag/images /experimental/geom_small.gif.*

can be seen in Figure 2-3. This leads to an incomplete set of tomographic TEC data in many ionospheric applications. The limited geometrical coverage condition requires the incorporation of more information to overcome the problem of the missing data in many reconstruction ionospheric studies. Accordingly, the retrieval of the ionospheric image cannot be applied using direct inversion algorithms. There is no unique approach that has been shown to handle all ionospheric problems. However several reconstruction algorithms have been constructed and applied and these have been shown to be partially successful. Some of these approaches are outlined and discussed in Chapter 3.

Chapter 3

Review of ionospheric tomography

3.1 Regularisation based tomographic reconstruction techniques

3.1.1 Regularisation

Recall that in ionospheric tomography studies, the aim is to obtain values of the unknown electron densities. An ionospheric tomography application can be viewed as an inverse problem for which Eq (2.3), for instance, could be solved for d. In practice, W is a known highly singular matrix, and the number of unknown parameters is much bigger than the number of available TEC measurements, i.e. the data set is incomplete. A problem is said to be well-posed when it has a unique solution and this solution depends continuously on the initial condition. If one or more of these conditions fail to hold, then the problem is said to be ill-posed; see Marroquin *et al.* (1987). With an incomplete set of TECs, the ionospheric inversion problem is ill-posed because it has infinitely many solutions. As a result, and also due to the high singularity of W, the standard matrix inversion approach of the form $d = W^{-1}Y$ is not possible. To overcome this difficulty a regularisation technique is required. Solutions to the general inversion problem may also be required to satisfy some known physical features of the underlying scene. For example, the electron density values should be positive, and the reconstructed image should reveal a high degree of smoothness. Again, this may be accomplished by regularisation.

A set of values which solves an equation of interest with the same properties as Eq (2.3), and at the same time captures the expected features of the problem will be described as a "good" solution in this chapter.

In order to generalise the problem of finding a solution to any model similar to that given in Eq (2.3), assume that

$$y = X\theta + \epsilon, \tag{3.1}$$

where y is an $n \times 1$ vector of observations, X is a known $n \times p$ fixed matrix, usually representing a discretised version of a mathematical transform, θ is a $p \times 1$ vector of unknown values with $p > n$, and ϵ is the associated data noise.

To solve the above system of equations for θ, several regularisation methods are available in the literature. Some of these methods are described below.

Least-squares regularisation approaches

There are two main types of regularisation techniques that are based on minimising the residual sum of squares (RSS), $(y - X\theta)^T(y - X\theta)$. In the first type, sets of basis functions are used, whereas in the second type a penalty function is added to the RSS term. The aim in the latter approach is to find an estimate, $\hat{\theta}$, that minimises the resulting penalty-based function. The two types of method will now be discussed in more detail.

1. **Basis functions approaches**

   To implement this technique, we need to define a space Θ of possible solutions, i.e. a set of θ's, say of the form

   $$\theta = \sum_{j=1}^{q} a_j \phi^{(j)}, \tag{3.2}$$

   where $\phi^{(j)}$ denotes the jth chosen basis function, and a_j is its associated but unknown weight. A basis is a set of vectors that, in a linear combination, is able to represent each vector in an associated space of interest. Every element of the basis is known as a basis function.

   The reason behind combining each set of basis functions together via Eq (3.2) is to hopefully produce a "good" overall solution, $\theta \in \Theta$, that can capture the required features of the true scene, for instance. Then the selection of the most "good" solution, $\hat{\theta}$, is based on the fact that this solution minimises the residual sum of squares.

For the ionospheric tomography problem, for example, there are two main choices for the basis functions that can be used to represent the unknown structure of electron density. These are as follows:

A pixel indicator function: This type of basis function was used by Austen *et al.* (1988). Given a pixel's latitude u, and height above the surface of the Earth r, the above authors expressed the basis function as

$$\phi^{(j)}(u,r) = \begin{cases} 1, & \text{if } (u,r) \text{ is in the } j\text{th pixel;} \\ 0, & \text{elsewhere.} \end{cases}$$

That is, each basis function is a vector of zeros except the jth pixel entry which is one. This forms vectors of basis functions that are orthonormal. These vectors can be combined together to form the basis functions matrix, ϕ. In this case, each basis vector represents a column in this matrix and the associated weights are simply the required pixel electron densities.

Expansion in terms of observed or model ionosphere basis functions: Suppose a set, ξ, of ionospheres each having J pixels are observed for several local times, solar activities, etc. Denote each observed ionosphere in ξ by s_j, then s_j represents a column in ξ. Here the matrix ϕ of basis functions is written as an expansion in terms of ξ. To simplify the necessary computations in all ionospheric studies based on this approach ξ is constructed with orthogonal columns. In this case,

$$\phi = \xi.$$

An alternative approach for deriving a single s_j vector is to create an artificial ionosphere. This can be achieved by using an ionospheric model, such as the Chapman layer model, Davies (1990) or the IRI-90 model, Bilitza (1990). The derivation approaches of observed and model ionospheres are discussed in the next section.

For a simple demonstration on the latter basis functions, and for some more detail on the above types of basis functions used in ionospheric tomography, the reader is referred to Raymund *et al.* (1994) and the references therein.

In the above basis function approaches, these functions can be obtained from *a priori* information and the solution of the tomographic problem consists of calculating the weight a_j associated with each basis function. Hence determining the number of basis functions and their corresponding weights are crucial steps in any basis functions based method. For more detail on these approaches, see Section 3.2.

2. **Penalty function approaches**

   Penalty function based approaches are techniques that are usually used to invert a model of the form given by Eq (3.1) for the unknown values. To apply any of these approaches, a "penalty" function for θ, say, $g(\theta)$ should be defined. The goal is then to find θ which minimises

   $$(y - X\theta)^T(y - X\theta) + \lambda g(\theta), \tag{3.3}$$

   where λ is an arbitrary regularisation parameter that can be chosen to control the trade-off between the faithfulness to the data (small value of RSS) and the regularisation degree of a solution.

   One standard form that the penalty function may take is given by

   $$g(\theta) = \sum_{i=1}^{p} \theta_i^2, \tag{3.4}$$

   that is, $g(\theta)$, is an L_2-norm penalty, for more details on this approach; see, for example, Green (1998). The above penalty function is one of the standard regularisation methods, mainly due to Tikhonov (Tikhonov and Arsenin, 1977) with a stabilising functional $||\theta||^2$. Poggio *et al.* (1985) showed that this approach can be used for solving computational vision problems when the physical constraints of the problem are embodied through a vector q. In this case, the aim is to minimise the stabilising functional given by $||q\theta||^2$.

   An alternative penalty function to the quadratic one given in Eq (3.4) is that based on the log function, $g(\theta) = -\sum_{i=1}^{p} \theta_i \log \theta_i$, known as an entropy measure. In this case, the inversion method is known as the method of *maximum entropy*. The motivation for this criterion, as in many other regularisation techniques, is to convert an ill-posed problem into a well-posed one, and hence solve the problem for the unknowns, θ. This aim is to be achieved by adding *a priori* assumptions. Obviously, the technique is only applicable when the components of the unknown vector, θ, are all positive. This condition is expected to hold in many applications of inverse problems because in such problems θ is usually a vector of intensity measurements or other quantities that are strictly non-negative. For an application of this approach, see for example, Gull and Daniell (1978).

It is obvious that regularization approaches based on both the L_2 norm and the entropy penalty functions, displayed above, do not account for smoothness between neighbouring objects, e.g., neighbouring pixels in a reconstruction grid, but rather minimise the RSS based on the best overall expected value of all components of θ.

A third type of penalty functions are those known as the *roughness penalties.* Given the data set, y, of Eq (3.1), the method aims to estimate the entire curve, $G(\theta)$, where y can be re-written in terms of this function as follows:

$$y = G(\theta) + \epsilon. \tag{3.5}$$

Here, the penalty is a certain functional of the considered function, G. When the penalty function belongs to this category, the inversion process is described as a *roughness penalty approach.* To demonstrate this approach, suppose that we wish to fit a model of the form given in Eq (3.5) with the data points shown in the left panel of Figure 3-1. Applying the standard piece-wise method of least squares, without considering any constraints on the curve $G(\theta)$ that may fit the data, simply joining all data points, as in the second panel of Figure 3-1, would give a zero residual sum of squares, but does not explain the underlying phenomenon adequately. This is because it does not account for high variability between nearby points in the curve to allow for the variation in the data values. Even if a slightly more smoothed curve is selected to fit the data points, a curve that "wiggles", e.g., show some high degree of roughness may also be described as an inappropriate fit.

Given a function G with continuous second derivative, a common choice of roughness functional is the integrated squared second derivative, $\int_{-\infty}^{+\infty} G''(\theta)^2 d\theta$. This functional is used as a measure of roughness of the chosen curve. In this case, the penalty approach is based on minimising the *penalised sum of squares* given by

$$(y - G(\theta))^T(y - G(\theta)) + \lambda \int_{-\infty}^{+\infty} G''(\theta)^2 d\theta, \tag{3.6}$$

where T denotes the transpose.

As λ approaches ∞, the minimiser, $\hat{G}$, reveals no clear curvature and the integral

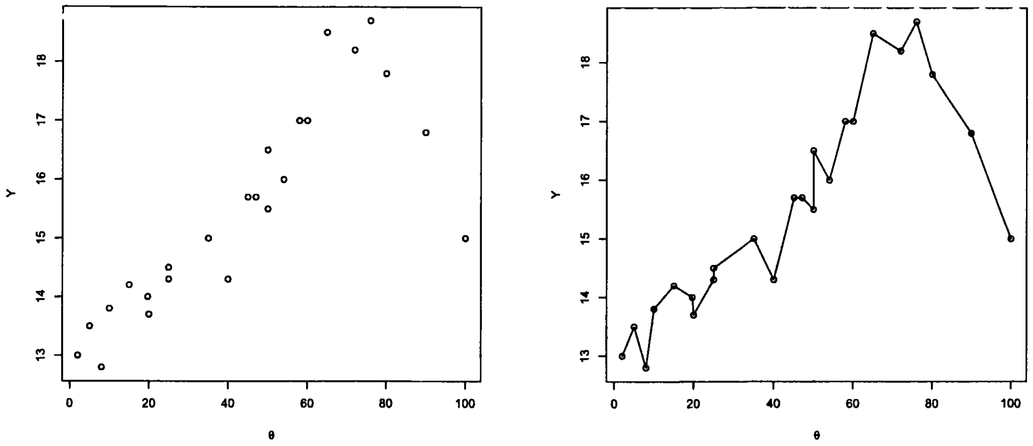


Figure 3-1: *Data values y (left). The piece-wise least squares fit to y (right).*

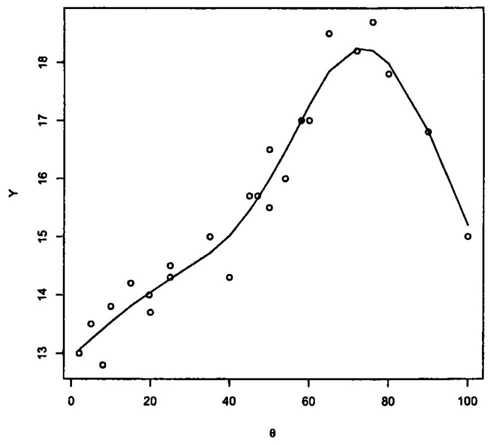


Figure 3-2: *A penalised least squares fit to y obtained when* $\lambda = 0.56$.

in Eq (3.6) tends to zero. Therefore, $\hat{G}$ tends to be approximately the linear regression fit. In contrast, when the smoothing parameter value approaches zero, $\hat{G}$ turns to be almost the piece-wise least squares fit to the problem. To overcome both problems, λ should be tuned until a reasonable fit is obtained. For example, for the data set of Figure 3-1, one plausible curve that fits the data pretty well is derived when λ is assigned a value of 0.56. This fit is shown in Figure 3-2.

Penalised likelihood regularisation approach

Another technique in which roughness penalties are used is the *penalised maximum likelihood approach.* This approach is widely used in applied statistics; see, for instance, Green and Silverman (1994), Wahba (1990), and Kristensson (1986). In this approach, the roughness penalty term is subtracted from the log-likelihood of the data. Suppose that the data y in Eq (3.5) are independently normally distributed with mean $G(\theta) = X\theta$ and common variance σ^2, then the log-likelihood function takes the form

$$l(y, \theta) = -\frac{1}{2\sigma^2}(y - G(\theta))^T(y - G(\theta)).$$

The penalised likelihood solution is that which maximises the penalised log-likelihood function, $l_p(y, \theta)$, given by

$$l_p(y, \theta) = l(y, \theta) - \lambda \int_{-\infty}^{+\infty} G''(\theta)^2 d\theta. \tag{3.7}$$

The Bayesian alternative approach

An alternative method to those frequentist approaches described above for the inverse problem of interest given in Eq (3.1) is that based on Bayes' theorem.

The Bayesian paradigm consists of four main contributions:

1. to define a prior distribution of the unknown variables, $\pi(\theta)$, in order to support the prior beliefs about them.

2. to construct the likelihood function, $L(y, \theta)$, for the given data y.

3. to form the posterior distribution, $\pi(\theta|y)$, by combining the likelihood and the prior densities using Bayes' theorem; i.e.

$$\pi(\theta|y) \propto L(y, \theta)\pi(\theta), \tag{3.8}$$

where $\pi(\theta|y)$ is usually known up to a normalising constant which is generally not available.

4. to obtain inferences about θ given y from the posterior distribution.

By taking the logs of the posterior distribution given in Eq (3.8), we obtain

$$\log \pi(\theta|y) = l(y, \theta) + \log \pi(\theta) + \text{constant}. \tag{3.9}$$

Comparing the terms of Eq (3.9) with those of the penalised likelihood function shown in Eq (3.7), one can see that the prior density term in the former equation plays the role of the penalty function in $l_p(y, \theta)$, so that the maximum penalised likelihood estimator, $\hat{\theta}$, maximising l_p, turns to be the mode of the associated posterior distribution, i.e. the *maximum a posteriori* estimator.

Unlike almost all non-Bayesian regularisation approaches where the solution space is restricted by the *a priori* knowledge, in this method, the *a priori* information is represented in terms of an appropriate probability density, Marroquin *et al.* (1987).

3.2 Review of some ionospheric tomography approaches

In this section, we will discuss several existing reconstruction algorithms that implement some of the above regularisation techniques to map the ionosphere's spatial distribution of electron concentration, in particular, the ART, MART, MIDAS and a Bayesian algorithm approach. All these algorithms require one or more initial starting images. The initial images are usually known as *background ionospheres.* These are very important, especially for the non-Bayesian approaches. This is due to the fact that because only limited projection angles are available, and only a few widely spaced monitoring receivers are used in most ionospheric experiments, so many pixels in the reconstruction grid are not traversed by any ray path. Hence path segments in such pixels are zero. The *a priori* information represented in the background ionospheres in this case plays a significant role in stabilising the solution obtained from these algorithms. In most cases, this can be achieved by assigning reasonable values of electron densities for such pixels, in addition to those crossed by one or more rays.

There are two ways of obtaining initial configurations. The first is experimental, whereas the second is artificial. To obtain an initial state by the first approach, a set of satellite-to-receiver signals are observed at a specific time and location. Each TEC value is then divided between the pixels that were intersected by the associated datum ray path but in proportional to the length of the ray segment within each pixel. In general, a pixel electron-density value, d_j, is obtained by dividing the overall TEC values assigned to this pixel by the total length of the associated ray segments. The term *model ionosphere* refers to the generation of an artificial ionosphere by using a set of simulated TEC rather than measured TEC data. In this technique, it is necessary to provide *a priori* information on the vertical profile Raymund *et al.* (1993). For this purpose, one can use the Chapman layer model, Davies (1990), or the IRI-90 model described by Bilitza, (1990). These models mainly combine the prior knowledge on a vertical profile parameters to produce an estimate of the electron density at a particular altitude, e.g., a d_j value.

3.2.1 The Algebraic Reconstruction Technique (ART)

ART is an iterative approach for solving a linear equation. It was first introduced as a reconstruction algorithm in 1970 by Gordon, Bender, and Herman. These authors show that the algorithm converges to the least-squares solution. In 1983, Censor *et al.* modified this approach by considering a sequence of relaxation parameters.

For the ionospheric tomography problem defined in Eq (2.3), a starting image of pixel electron densities, d^0, is required as an initial guess. In any iterative step, k, the current image is modified to produce a new image by considering only a single ray path, i, and changing the electron density estimate of all pixels that are intersected by this ray. The discrepancy between the TEC measurements, Y_i, and the corresponding projection estimated value, $\sum_{j=1}^{J} w_{ij} d_j^k$, obtained from the current image d^k is redistributed among all pixels intersected by the ith ray in proportion to the length of intersection within each pixel, e.g., w_{ij}. That is, pixel values along the ith projection are refined to conform with the ith TEC record without changing the remaining pixel densities. At each iteration, the algorithm cycles through all ray paths according to the following formula:

$$d_j^{k+1} = d_j^k + \lambda_k \frac{Y_i - \sum_{j=1}^{J} w_{ij} d_j^k}{\sum_{j=1}^{J} w_{ij}^2} w_{ij}, \qquad \text{for} \quad i = 1, \ldots, I, \tag{3.10}$$

where $\{\lambda_k\}$ is the relaxation parameter set which may consists of one number or a series of numbers. Different values of the relaxation parameter have different effects. Herman *et al.* (1975) studied this effect and concluded the following: suppose that H_i represents the set of all d values such that the $\sum_{j=1}^{J} w_{ij} d_j = Y_i$, known as the hyperplane of Y_i, then if

$$\lambda_k \text{ is } \begin{cases} < 0, & \text{new move is away from } H_i; \\ = 0, & \text{no new move;} \\ \in (0,1), & \text{new move is towards } H_i, \text{ but does not quite reach it;} \\ = 1, & \text{new move is to } H_i \text{ exactly;} \\ \in (1,2), & \text{new move is past } H_i, \text{ but } d^{k+1} \text{ is nearer to } H_i \text{ than } d^k \text{ was;} \\ = 2, & d^{k+1} \text{ is the mirror image of } d^k \text{ in } H_i; \\ > 2, & d^{k+1} \text{ is on the other side of } H_i, \text{ farther from } H_i \text{ than } d^k \text{ was.} \end{cases}$$

Therefore, usually a value of λ_k is chosen to be within the interval $(0, 2)$.

The ART was first introduced in Computerised Ionospheric Tomography by Austen *et al.* (1988). A drawback associated with this approach is that some unrealistic negative estimates of free electron densities may be generated. However, it has been successfully implemented for some experimental ionospheric tomography; see, for example, Andreeva *et al.* (1990).

3.2.2 The Multiplicative Algebraic Reconstruction Technique (MART)

MART is an iterative reconstruction algorithm that was proposed by Gordon *et al.* (1970) and is considered as a modified version of ART.

To implement MART for ionospheric tomography, a starting image, d^0 is required. The algorithm then iterates by changing values of the pixel electron densities based on the difference between the measured and the recovered TEC value for each path, as in ART. However, the changes in this case are multiplicative rather than additive. Here, d_j^k is updated via the following equation:

$$d_j^{k+1} = d_j^k \left(\frac{Y_i}{\sum_{j=1}^{J} w_{ij} d_j^k} \right)^{\lambda_k w_{ij}}, \quad \text{for} \quad i = 1, \ldots, I, \tag{3.11}$$

where λ_k is the relaxation parameter at the kth iteration with $0 < \lambda_k < 1$. That is, the electron densities' vector, d, is corrected through an iteration by a ratio of the measured TEC and the calculated TEC with a relaxation parameter λ_k.

Lent (1977) proved that this algorithm converges to the maximum entropy solution where the aim is to maximise the function, $f(d)$, given by

$$f(d) = -\sum_{j=1}^{J} d_j \log d_j.$$

The MART algorithm was initially applied to experimental ionospheric tomography in 1990 by Raymund *et al.* In several experimental cases, this algorithm's behaviour is not well understood, in particular, when the data set is incomplete; see, for example, Censor (1983). However, other authors such as Kersley *et al.* (1993) and Pryse *et al.* (1995) found it to perform well for their experiments.

An advantage of MART over ART in estimating the electron densities is its ability to avoid negative values of the *d*s.

3.2.3 The Multi-Instrument Data Analysis Software (MIDAS)

In 1992, a non-iterative reconstruction algorithm for two-dimensional imaging was introduced by Fremouw *et al.* Here, a set of empirical orthonormal basis functions derived from an ionospheric model with a wide range of possible vertical profiles across the grid are used to represent the vertical structure of the ionosphere, while the horizontal structure is selected from a Fourier basis. A large set of possible background ionospheres are therefore created to provide the required initial knowledge. The technique allows the *a priori* information to be fitted based on the least-squares approach to estimate the pixel electron density. The authors' approach is extended into three-dimensions by Spencer and Mitchell (2001). In addition, a time-dependence has been included into the tomographic approach to allow for changes in the ionosphere's electron densities when the observing period is long, such as that required by the GPS signals. The modified version of the algorithm is called MIDAS.

As in the other approaches to the ionospheric problem, the aim is to solve the equation $Y = Wd$, for the vector of electron densities, d, and consequently to map the associated spatial distribution. The matrix, W, is known to be highly singular and contains no prior information about the true solution. To overcome this difficulty a mapping matrix, X, is used to transform the problem from a voxel-based representation with unknown electron densities into a basis functions representation with the unknowns being the coefficients of the basis functions, represented by the matrix, M, in the following mathematical expression:

$$Y = WXM. \tag{3.12}$$

Now Eq (3.12) needs to be solved for M. Re-writing the latter equation in terms of this matrix gives

$$M = (WX)^{-1}Y.$$

Here $(WX)^{-1}$ is a generalised inverse matrix. The inversion is then solved by applying the Singular Value Decomposition to the matrix WX, Mitchell *et al.* (2003). Once M is determined, the electron densities within each voxel can be then retrieved by setting

$$d = XM.$$

Obviously, the selection of the orthonormal basis functions from a set of vertical profiles is a critical factor in this approach. Here, as in the ART algorithm, the problem of obtaining negative electron densities arises. However, this approach is computationally cheaper since it is non-iterative.

It is obvious that the above well-known described three regularisation approaches for imaging problems of the ionosphere are mainly dependent on the *a priori* information used to start the algorithms. Thus, the closer the starting condition is to the true image, the more accurate reconstruction will be. Therefore, the background ionospheres should be chosen to span the entire space of all plausible solutions to the tomographic problem. This makes the *a priori* information very critical factors in the solutions produced by these approaches.

3.2.4 A Bayesian algorithm

Markkanen *et al.* (1995) developed a new reconstruction algorithm based on a Bayesian approach to the ionosphere imaging problem. In this method, the unknown variables are effectively the pixel electron densities together with the unknown phase constant associated with each receiver. All unknown parameters are arranged in one column vector denoted by X. Then each TEC measurement is represented as a linear combination of the associated unknown electron density values and phase constants. Moreover, an error term is added on this combination. Thus the inversion problem is defined by the following equation

$$Y = AX + \epsilon, \tag{3.13}$$

where ϵ is a vector of errors associated with the TEC data. Note that A in Eq (3.13) is not exactly the matrix W in Eq (2.3) because it also contains contributions of the phase constants.

Since this approach is Bayesian, Y, X and ϵ, are all treated as random variables and a set of distributions are used to model these parameters. The measurements are assumed to be recorded with a zero mean Gaussian error, ϵ, but are not necessarily conditionally independent. That is, if Σ denotes the unknown variance matrix of ϵ, then Σ may not be a diagonal matrix. Under these assumptions, $Y \sim N(AX, \Sigma)$. The authors only considered either completely flat prior distributions on the pixel electron densities and the phase constants, or a simple Gaussian prior. The prior distribution is denoted by $D(X)$. This gives the following posterior distribution:

$$D(X|Y) \propto D(X)|\Sigma|^{\frac{-1}{2}}\exp\left(-\frac{1}{2}(AX-Y)^T\Sigma^{-1}(AX-Y)\right). \tag{3.14}$$

The inversion is stochastic and therefore the idea is to obtain the most probable values of the electron densities and phase constants. However, because of the big size of the matrices involved, a solution requires some sort of regularisation. By choosing the specified prior distributions, the authors aim to formulate maximum a posteriori estimation as a piece of matrix algebra.

The Bayesian approach was initially tested on a set of simulated data in which several Chapman profiles peak height values were used in the retrieval procedure via a prior distribution. The results indicated that this approach mainly depends on the regularisation variables, but the choice of their best values may be based on visible artifacts and the lowest negative electron densities as well as the residual sum of squares. Further, the authors implemented the approach with experimental data. Here, two tomographic images were obtained of which one revealed a reasonable agreement with a vertical profile obtained from the EISCAT radar. Nevertheless, the choice of the regularisation profile remained an outstanding problem, although this approach was found to be less sensitive than the iterative methods of ART and MART, for instance, to their initial conditions.

Chapter 4

Bayesian model formulation

4.1 Modelling the TEC data

In this work we mainly focused on the analysis of 572 TEC measurements recorded by a chain of four ground-based NNSS receivers located approximately along a line of longitude in Italy, where radio signals from an approaching satellite are detected. The measurements are carried out during a single satellite passage over the chain of receivers at discrete instants of time, and in sequential locations. Figure 4-1 illustrates the pattern of links from the four ground receivers to the NNSS satellite. We will now discuss how we modelled our data in this study.

Hajj *et al.* (1994) assumed that the variance of the TEC noise is constant for all observations; these authors considered discretisation effects as a second source of noise. Markkanen *et al.* (1995) assumed that the measurement errors are Gaussian with zero mean, and considered the case of equal error variances for all TEC measurements. Since the ionosphere affects all satellite-to-receiver radio signals in a similar manner, we simply accept the view of Hajj *et al.* (1994) that the resulting data noise variance should be constant for all TEC measurements, and follow Markkanen *et al.* (1995) in modelling each record error, e_i, by a Gaussian distribution with mean zero and variance, ϵ^2. According to this Eqs (2.2) and (2.3) can be re-written as

$$Y_i = \sum_{j=1}^{Nr} w_{ij} d_j + e_i, \tag{4.1}$$

for $i = 1, \ldots, I$, or

$$Y = Wd + e, \tag{4.2}$$

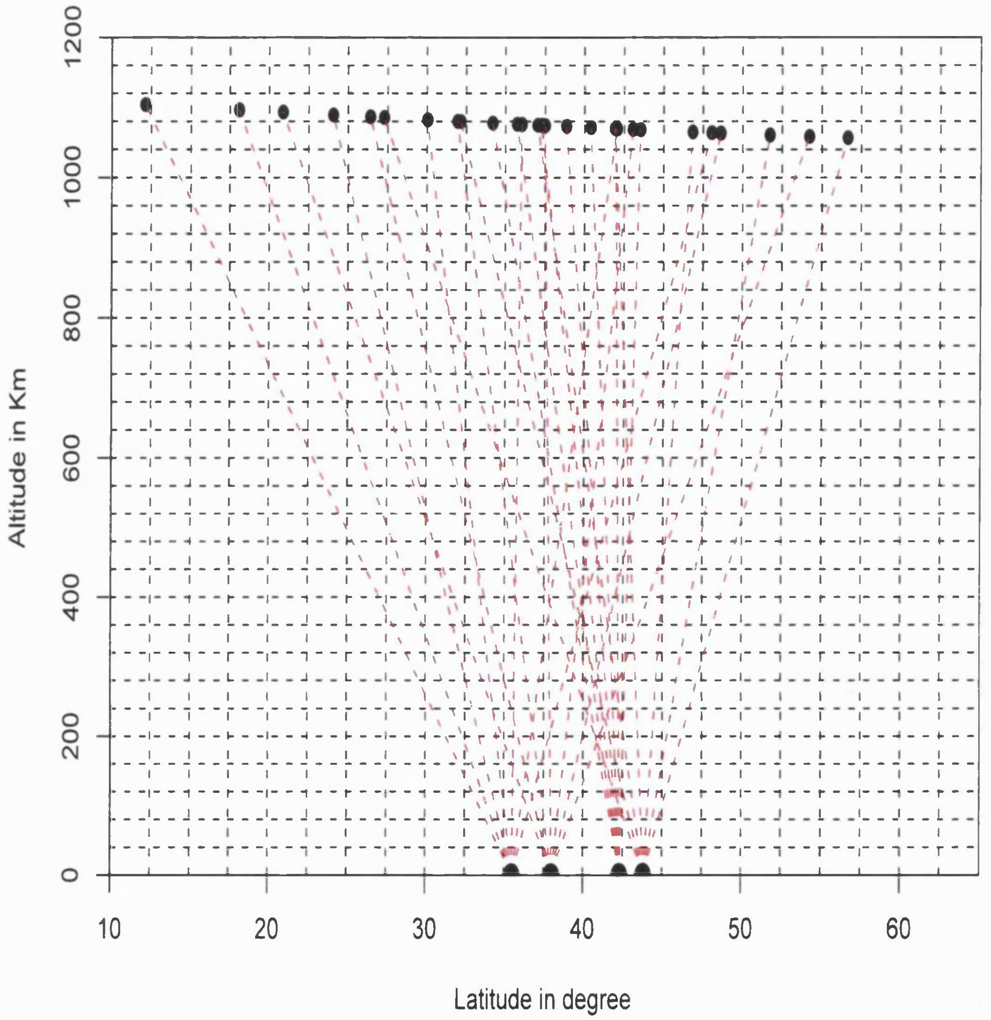


Figure 4-1: *The approximate two-dimensional geometry of the radio-wave paths as the Navy Navigation Satellite System (NNSS) satellite passes at a subset of the positions over a chain of four receivers on the ground.*

where e is the measurements noise vector whose elements, e_i, quantify the uncertainty due to ionospheric dispersive noise and other detecting process noise, and will be described as the data noise.

The errors are also assumed to be independent. Thus, by the above assumptions, the overall model for the TEC measurements $Y_1, \ldots, Y_I$ given d and ϵ^2 is

$$Y_i | d, \epsilon^2 \quad \sim \quad N(E(Y_i), \epsilon^2), \qquad i = 1, \ldots, I, \tag{4.3}$$

where

$$E(Y_i) \quad = \quad \sum_{j=1}^{Nr} w_{ij} d_j, \tag{4.4}$$

and these measurements are conditionally independent given d and ϵ^2.
The choice of modelling the TEC measurements with a common variance is worthy of comment. The Poisson distribution has been employed in several similar cases in which signals are transmitted through an object and then detected to give observations of the form of counts, such as in Single Photon Emission Computed Tomography, see e.g., Winkler (2003, Part VII) and Higdon *et al.* (2003). However, in ionospheric studies we have no evidence of dependence between the TEC expected value $E(Y_i)$ and the variance of the associated noise.

The above data model gives us a basic starting point for Bayesian inference about the spatial pattern of the ionospheric electron density. Under the assumption that the TEC records are conditionally independent, the likelihood function for these records is

$$\begin{aligned} L(Y|d, \epsilon^2) &= \prod_{i=1}^{I} \frac{1}{\sqrt{2\pi\epsilon^2}} \exp\left\{ -\frac{1}{2\epsilon^2} (Y_i - E(Y_i))^2 \right\} \\ &= \left(\frac{1}{\sqrt{2\pi\epsilon^2}} \right)^I \exp\left\{ -\frac{1}{2\epsilon^2} \left(\sum_{i=1}^{I} (Y_i - E(Y_i))^2 \right) \right\}. \end{aligned} \tag{4.5}$$

4.2 Modelling the electron density values directly

Given the linear model in Eq (4.2) and the unknown random variable vector $(d_1, \ldots, d_{Nr}, \epsilon^2)$, the classic approach to the ionospheric tomography problem is based on solving the model for this vector. Very often the number of local electron densities, Nr, is greater than the number of TEC observations, I. This makes the inversion process impracticable and therefore, methods of regularisation are needed to overcome this problem. Many such approaches have been developed, but several drawbacks have been found. In addition to those discussed earlier in Chapter 3, one main drawback is that almost all these methods do not take into consideration the uncertainty about the true value of the physical quantity of interest.

An alternative method that overcomes the uncertainty problem is the Bayesian approach introduced in this chapter. The analysis of this method requires the specification of a prior distribution for d and ϵ^2.

Our prior knowledge about the underlying true pattern of electron densities informs us that the ionosphere possesses a certain degree of smoothness. One approach to modelling the prior distribution is to work with the pixel densities directly, choosing a model which stipulates that neighbouring pixels in the grid have similar densities of free electrons. With the regular 2D grid considered in this approach, two pixels are specified as neighbours if they are directly adjacent so that interior pixels have four neighbours, edge pixels have three, and corner pixels have two. For tractability, we might assume that under this neighbourhood structure, densities of any two pixels that are not neighbours are conditionally independent given the values of other pixels. Gaussian Markov Random Fields (GMRFs), also known as conditional auto-regressions, introduced by Besag (1974, 1975) define the spatial correlations between pixels in a way that satisfies this conditional independence property. Within this class of models, smoothing parameters that control the horizontal degree of smoothness, $\beta_h \geq 0$, and the vertical degree of smoothness, $\beta_v \geq 0$, between the adjacent pixels are required. These models are briefly introduced in section 4.4.

Under the assumption that ϵ^2 is independent of the local electron densities, d, and that the smoothing parameters β_h and β_v are fixed, the joint distribution of d and ϵ^2 is

$$\pi(d, \epsilon^2 | \beta_h, \beta_v) = \pi(d | \beta_h, \beta_v) \pi(\epsilon^2), \tag{4.6}$$

where $\pi(\epsilon^2)$ represents the prior distribution of ϵ^2.

4.3 A parametric model for electron density values

In this section, we will introduce an alternative parametric model and justify this choice for modelling the ionospheric problem. The new model will be constructed within a Bayesian framework, i.e. prior distributions are assigned to all parameters and a posterior distribution is constructed allowing uncertainty about the parameters to be examined. However, the electron densities are not the unknowns for which the model will be inverted. Instead, inferences about another related set of random variables will be obtained. The definition of the new set of variables is based on the famous theorem of Chapman (1931). Under this theorem, every vertical section of the underlying ionospheric scene is represented by a Chapman profile, also known as the vertical electron-density profile, see Figure 2-1. Moreover, each vertical profile is described by three quantities, namely, the peak height which is the altitude of the peak in the vertical profile where maximum amount of ionisation is found; the spread, also known as the scale height, which measures the width of the profile; and the electron density in the area under the profile curve, i.e. the total number of free electrons under the vertical profile. We shall apply this result. Therefore, we will represent each grid column n by a vertical profile. Further, the nth vertical profile will be described by three parameters. These parameters are: the profile peak height, μ_n, the spread, σ_n^2, and the total number of electrons within the area under the profile curve, γ_n, for $n = 1, \dots, N$, where N represents the total number of columns in the grid.

Fremouw *et al.* (1994) postulate that the vertical profile has the shape of a Gaussian distribution. These authors considered the vertical profiles as background profiles and used them to obtain inferences about the local electron densities. By using the above ionospheric profile shape assumption, we approximated the shape of the nth vertical profile by a Gaussian curve centred at the corresponding peak height, μ_n, and the electron density variation along the profile, σ_n^2, is assumed to be controlled by this curve variance, for all $n = 1, \dots, N$. Since every vertical profile is assumed to be affected by the amount of electron concentration under the curve, and approximately normally shaped, therefore, the electron density at height h within the nth column is

$$\gamma_n \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left\{-\frac{1}{2\sigma_n^2}(h - \mu_n)^2\right\}. \tag{4.7}$$

The above formula represents the case where height is measured continuously. However, this form is easily modified to satisfy the discrete assumption of the 2D ionospheric grid. Suppose that we are interested in the jth pixel electron density and that this pixel lies in the nth column, then the electron density can be written as

$$l_{nj} = \frac{\gamma_n}{\sqrt{2\pi\sigma_n^2}} \exp\left\{-\frac{1}{2\sigma_n^2}(h(j) - \mu_n)^2\right\}, \tag{4.8}$$

where $h(j)$ is the corresponding pixel mid-height. Note that in this discrete approximation we assume a constant density within each pixel.

By involving the new vertical profile parameters set in the TEC data model through Eq

(4.8), Eq (4.1) is converted from a linear model in pixels electron densities into a nonlinear function of the corresponding profile parameters. This produces a new vector of random variables, $X = (\mu, \sigma^2, \gamma, \epsilon^2)^T$, where $\mu = (\mu_1, \ldots, \mu_N)^T$, $\sigma^2 = (\sigma_1^2, \ldots, \sigma_N^2)^T$ and $\gamma = (\gamma_1, \ldots, \gamma_N)^T$.

Thus, to tackle the ionospheric problem, in a Bayesian analysis we can work with either:

1. A GMRFs prior model for the local electron densities, d, or

2. A prior defined as a "higher level" parametric model for the vertical profile parameters, namely, μ, σ^2 and γ.

In the second approach, there will still be the opportunity to specify that the vertical profiles alter smoothly from column to column.

We chose to work with the second approach in reconstructing the map of electron density in the ionospheric region of interest for the following two main reasons:

- The ionospheric region of interest is huge. It extends approximately from 10° to 65° in latitude and over 1000 km in altitude. Therefore, a 2D grid of many pixels is required to "adequately" reconstruct the spatial distribution over this region. Let the grid, for instance, be of size 22×20 then by choosing the usual linear model, the inversion process would need to infer 440+1 local electron densities beside the error parameter. On the other hand, if the parametric model is used, inferences of only $(20 \times 3) + 1$, i.e. 61 parameters are needed.

- The geometry of the TEC observations in Figure 4-1 shows that many pixels are not intersected by any ray path. Thus, inferences about these pixels will be dominated by the weak prior distribution in the case of the first form of Bayesian modelling. In contrast, in the parametric modelling method, it is not necessary that each pixel should be intersected by one or more rays to have a "good" estimate of its free electron content, rather, it may be adequate to have only few ray paths passing through the corresponding column.

4.4 The parametric model prior distribution

4.4.1 Spatial prior model for the profile parameters

In the higher level parametric model, a distribution that reflects our prior belief that geographically adjacent vertical sections of the ionosphere are similar in term of the amount of ionisation present is needed. Since every vertical section in the reconstruction grid is represented by a vertical profile, a spatial dependence between neighbouring profiles is expected. In other words, we assume that there exists a local spatially structured variation in each of the three parameters, namely, the peak height, μ_n, the spread, σ_n^2, and the electron concentration under the vertical profile curve, γ_n, for $n = 1, \dots, N$.

In this case, we will assume that any two vertical profiles separated by a single profile or more within the ionospheric grid are conditionally independent of all other vertical profiles given a small set of geographical neighbouring profiles.

A suitable class of spatial models that can handle the variation structure inherent between neighbouring sites, whether these sites are peak heights, profile spreads or column electron densities are the spatially-dependent Gaussian Markov Random Fields models mentioned in section 4.2. In statistics, GMRFs are widely used, with important applications in structural time-series analysis, analysis of longitudinal and survival data, graphical models, image analysis and spatial statistics. For references and illustrations, see Rue and Held (2005, Ch. 1).

A GMRFs model has several components: a set of p sites $P = \{1, 2, \dots, p\}$; a neighbourhood system $\delta = \{\delta_n; n \in P\}$ where each member of δ is a subset of neighbouring sites of the nth site; and a random field which is a set of random variables $X = \{X_n; n \in P\}$ defined on a set Ω. A configuration of X is a Markov field on Ω if it satisfies the following two conditions:

- positivity: $P(X = x) > 0, \forall x \in \Omega$,
- Markovianity: $P(X_n = x_n | X_{-n} = x_{-n}) = P(X_n = x_n | X_m = x_m, m \in \delta_n)$, $\forall n \in P$, where X_{-n} denotes the variable consisting of all the components of X except the nth component.

The latter condition states that the value of the nth site X_n given the rest of sites depends only on the value of its neighbouring sites in the set, δ_n.

The key result concerning GMRFs is the Hammersley-Clifford theorem. This theorem states that a particular distribution $P(X = x)$ is a Markov random field if, and only if,

$$P(X = x) \propto \exp\{-\sum_{c \in C} \theta_c(x)\}, \tag{4.9}$$

where C is the set of cliques c and a "clique" is defined as either a singleton site or a set of sites all of whom are neighbours, and $\theta_c(x)$ is a function that depends only on values of x_j for $j \in c$. Parameters that control the degree of smoothness between neighbouring sites are known as smoothing variables and will be usually involved in

$\theta_c(x)$, therefore, $\sum_{c\in C}\theta_c(x)$ could be considered as a smoothness function.

We shall propose a model for the profile parameters in which GMRFs prior distributions of the form specified by Eq (4.9) are used. For this purpose we shall assume that $\beta_\mu \geq 0$, $\beta_{\sigma^2} \geq 0$ and $\beta_\gamma \geq 0$ are the necessary smoothing parameters of μ, σ^2 and γ, respectively.

Our prior beliefs, given by most electronic engineers, about the vertical profiles inform us that these profiles have peaks at altitudes lying between 250 km and 500 km above the Earth's surface. According to this information and the size of the reconstruction grid, we bounded each peak height, μ_n, between 80 km and 1200 km. Since the TEC observations give measurements of concentration of free electrons present in columns of unit cross-sectional areas and the latter are measured in meters^2, the selected boundaries are modified to produce the following peak heights prior distribution:

$$\pi(\mu|\beta_\mu) \quad \propto \quad \exp\left\{-\beta_\mu \sum_{a\sim b}(\mu_a-\mu_b)^2\right\}, \tag{4.10}$$

where $\sum_{a\sim b}$ represents the sum over all pairs of neighbours, with each pair counted once only, and $80,000 < \mu_a < 1,200,000$ given in meters, for all $a = 1,\ldots,N$.

A suitable range for the spreads is derived mainly from a set of values we obtained from a preliminary study, as explained in Chapter 6. These values indicate that the spreads are roughly between 10^{10} m^2 and 10^{11} m^2. Therefore, the upper boundary of the required interval is picked to be 10^{12} m^2. Moreover, since these parameters cannot be negative, a lower suitable boundary may be zero. Hence, the spreads prior distribution can be written as:

$$\pi(\sigma^2|\beta_{\sigma^2}) \quad \propto \quad \exp\left\{-\beta_{\sigma^2} \sum_{a\sim b}(\sigma_a^2-\sigma_b^2)^2\right\}, \quad 0 < \sigma_a^2 < 10^{12}, \tag{4.11}$$

where $a = 1,\ldots,N$.

Finally, and in a similar manner to above, the prior distribution of the electron densities under the profile curves is chosen to be:

$$\pi(\gamma|\beta_\gamma) \quad \propto \quad \exp\left\{-\beta_\gamma \sum_{a\sim b}(\gamma_a-\gamma_b)^2\right\}, \quad 0 < \gamma_a < 10^7, \tag{4.12}$$

for $a = 1, \ldots, N$, measured in units of $10^{11}\mathrm{el/m}^2$, because this unit is consistent with that of the local electron densities given by $10^{11}\mathrm{el/m}^3$.

It is worth mentioning that all the above boundaries are chosen to be big enough to easily cover the expected realistic ranges in order to avoid having inferences influenced by the boundaries given by the priors.

Note that in the models, the set of profiles extends along just one dimension, therefore a site will only have one or two neighbours, depending on whether it is an edge or an inner site.

The prior distributions given by Eqs (4.10) - (4.12) are convenient from a computation and theoretical point of view: they possess the Markovian property and are jointly Gaussian. The Markov property is important for models implemented by Markov Chain Monte Carlo methods, as it will also ensure rapid computation of the conditional densities; see Rue (2001).

In this study, we will further assume that μ, σ^2 and γ are independent and that ϵ^2 is independent of these parameters. Hence, the prior distribution $\pi(X)$, where $X = (\mu, \sigma^2, \gamma, \epsilon^2)$ is

$$\pi(\mu, \sigma^2, \gamma, \epsilon^2 | \beta_\mu, \beta_{\sigma^2}, \beta_\gamma) = \pi(\mu|\beta_\mu)\pi(\sigma^2|\beta_{\sigma^2})\pi(\gamma|\beta_\gamma)\pi(\epsilon^2). \tag{4.13}$$

4.4.2 The noise prior model

In our approach, the error parameter is an important variable because it reflects the uncertainty within the model due to several natural and technical source of noise. Since we do not know much about it, it is useful to specify a prior distribution that provides only little information. This is the criterion for a good model as it will allow information about the parameter ϵ^2 to be extracted primarily from the observed data.

Prior distributions that satisfy the above aim are known as Jefferys' priors and are usually described as flat distributions. The prior distribution we adopt to model the precision, $\tau^2 = \frac{1}{\epsilon^2}$, is the improper Gamma distribution, $\mathrm{Ga}(0, 0)$. This distribution is widely used to model unknown variances; useful applications can be found in Gilks *et al.* (1996). Although the prior has an infinite integral and is improper, when formally combined with the data likelihood it can yield an acceptable proper posterior distribution; see Gelman (2005). The Gamma prior is particularly convenient since its full conditional posterior distribution is also Gamma, and so can easily be handled in Markov Chain Monte Carlo simulations by, say, the Gibbs sampler of Geman and

Geman (1984). Let us denote this improper Gamma prior by $\pi(\tau^2)$, then

$$\pi(\tau^2) \propto \frac{1}{\tau^2}, \qquad 0 < \tau^2 < \infty. \tag{4.14}$$

Thus, by Jeffreys' invariance principle, see Gelman *et al.* (2004), ϵ^2 is modelled by the improper Inverse-Gamma prior distribution given by

$$\pi(\epsilon^2) \propto \frac{1}{\epsilon^2}, \qquad 0 < \epsilon^2 < \infty. \tag{4.15}$$

4.5 The ionospheric image posterior distribution

As mentioned in the previous section, the Bayesian approach allows us to model the spatial distribution of interest producing the following posterior distribution for X

$$\begin{aligned} \pi(X|Y, \beta_\mu, \beta_{\sigma^2}, \beta_\gamma) \;\; &\propto \;\; (\epsilon^{-2})^{\frac{I}{2}+1} \exp\left\{ -\frac{1}{2\epsilon^2} \sum_{i=1}^{I} (Y_i - E(Y_i))^2 \right\} \\ &\times \;\; \pi(\mu|\beta_\mu)\pi(\sigma^2|\beta_{\sigma^2})\pi(\gamma|\beta_\gamma). \end{aligned} \tag{4.16}$$

Our Bayesian inference about the underlying ionospheric scene given the TEC data is based on the above posterior distribution. This distribution's normalising constant, $\pi(Y)$, cannot be calculated due to the high dimensionality and complexity of the model. In this case, a random field $F = \{F_1, \ldots, F_N\}$ configuration, $f = \{f_1, \ldots, f_N\}$, of any type of the vertical profile parameters can, in principle, be obtained by exploring the posterior distribution via Markov Chain Monte Carlo (MCMC) techniques. This is because with these techniques the normalising constant is not required as it will be automatically cancelled in the acceptance ratio. Once inferences about these parameters are collected, estimates of the pixels electron densities are calculated using Eq (4.8).

A key issue in any Bayesian modelling is the nature of the resulting posterior distribution, for example, an improper posterior model is not acceptable. Such a model may emerge if unsuitable improper prior distributions are used. One foolproof way of avoiding the impropriety problem is to use proper priors (Hobert and Casella, 1996).

A GMRFs prior distribution for x where $x \in \mathbb{R}^N$, is called an intrinsic Gaussian autoregressions or intrinsic GMRFs (IGMRFs). These type of GMRFs are known to be improper as mentioned by Besag (1989) and Rue *et al.* (2005). Such a prior if used in building up a posterior model may lead to an improper posterior distribution. However, the impropriety of the model in this case could be removed by converting the improper prior distribution into a proper one. This could be perfectly achieved by restricting x_i's to any finite integral; see, for example, Besag (1989) and Besag *et al.* (1991). Obviously, implementing the latter suggestion in the above example makes the integration in Eq (4.18) finite.

Fortunately, the nature of our application and the expected realistic ranges of μ_n, σ_n^2 and γ_n given in Eqs (4.10) - (4.12), make our prior distribution proper.

Chapter 5

Implementing MCMC using simulated data

5.1 Markov Chain Monte Carlo (MCMC)

Markov chain Monte Carlo has had a profound influence on statistics, especially but not only in Bayesian inference; see, for example, Smith and Roberts (1993), Besag and Green (1993), Gilks *et al.* (1996).

MCMC is a sampling based simulation technique that can be used to explore a high-dimensional distribution, $\pi(X)$, defined on a sample space Ω. The exploration process aims to build a stationary Markov chain on the state space Ω, with an aperiodic and irreducible transition kernel, $P(X, X')$, which has $\pi(X)$ as its stationary distribution, where X' denotes a new state given the current state X. In this context, after a burn-in period specified to remove dependence of the chain on its starting state, the simulated sequence of samples converges in distribution to the desired model, $\pi(X)$. Although, the values along the Markov chain are dependent, summary features of the target distribution, such as point and interval estimates for unknown quantities of interest are then estimated using the converged part of the chain.

An essential step in any MCMC-based inference problem is the determination of a Markov chain for which $\pi(X)$ is its equilibrium distribution. A tractable way to ensure convergence to $\pi(X)$ is to construct a transition kernel $P(X, X')$ that not only satisfies the ergodicity conditions: irreducibility and aperiodicity, but also exhibits *detailed balance*, i.e.

$$P(X, X')\pi(X) \;=\; P(X', X)\pi(X') \qquad \forall X, X' \in \Omega. \tag{5.1}$$

Detailed balance is also called *time reversibility*, because it means that the Markov chain is invariant whether it was run forwards in time or backwards. If the detailed balance condition is maintained then the *global balance* $\int \pi(X)P(X, X')dX = \pi(X')$ is also satisfied.

Hastings (1970) designed the so-called Metropolis-Hastings algorithm that provides a way to accomplish detailed balance and global balance. The algorithm is iterative; given a state $X \in \Omega$ and an arbitrary proposal density $q(X, X')$, a new realisation, X', is proposed according to $q(X, .)$. The definition of the proposal distribution is essentially arbitrary, subject to the condition that the Markov chain is irreducible and aperiodic. In practice, it is convenient to select a proposal density for which sampling is considered a relatively fast and easy task, and for which calculation of the acceptance probability is not awkward. Once a new candidate X' is generated from $q(X, .)$, it is then accepted as the chain's new state with probability

$$\alpha(X, X') = \min\left\{1, \frac{\pi(X')q(X', X)}{\pi(X)q(X, X')}\right\}. \tag{5.2}$$

If X' is accepted, then the transition kernel $P(X, X')$ is given by

$$P(X, X') = q(X, X')\alpha(X, X') \qquad \text{if } X' \neq X. \tag{5.3}$$

However, if the new state was rejected, then the chain remains in the present state, X with probability

$$r(X) \;=\; 1 - \sum_{X' \neq X} q(X, X')\alpha(X, X'). \tag{5.4}$$

Obviously, $\pi(X)$ and q are related through the acceptance probability ratio, and this relationship has a strong effect on the chain convergence rate, as a "good" proposal density will lead to an efficient sampler.

A key feature of this algorithm is that determination of the model normalising constant is not required, as this will be cancelled in $\pi(X')/\pi(X)$, thus knowledge of the model only up to a constant of proportionality is adequate for implementation. For more details, see Smith and Roberts (1993). A special case of the Metropolis-Hastings sampler that is widely used in practice is the original Metropolis algorithm of Metropolis *et al.* (1953). This is based on the selection of a symmetric proposal distribution, i.e. $q(X, X') = q(X', X)$, such as a Gaussian centred on the current chain state. The acceptance probability in this case is simplified to

$$\alpha(X, X') = \min\left\{1, \frac{\pi(X')}{\pi(X)}\right\}. \tag{5.5}$$

Given a state vector $X = (X_1, \ldots, X_{i-1}, X_i, X_{i+1}, \ldots, X_k)$, a new state X' can be generated using different updating schemes. A well known scheme is single-site updating, also known as component-wise updating. Here only a single element of X, say X_i, is updated. That is, q proposes a new single value X_i' leaving the remainder of X, X_{-i}, unaltered. X_i may be updated randomly or systematically. However, only one element of X is updated at each iteration, producing

$$X' = (X_1, \ldots, X_{i-1}, X_i', X_{i+1}, \ldots, X_k). \tag{5.6}$$

The Gibbs sampler given by Geman and Geman (1984) is an example of a component-wise based MCMC algorithm. It was firstly used in statistical image processing and restoration in the field of spatial statistics. Later Gelfand and Smith (1990), amongst others, implemented this approach in other areas such as missing data and Bayesian computation problems. The Gibbs algorithm iterates by sampling from the full conditional distribution of the selected element, X_i, given the values on the complement, X_{-i}. In this case, the new candidate X' is always accepted, i.e. $\alpha(X, X') = 1$. Therefore, the Gibbs sampler can also be considered as a special case of the Metropolis-Hastings algorithm but with an acceptance probability of one.

A different type of updating strategy is the multiple-site or the block-wise scheme. Under this strategy, either all elements of X or a small subset of components of X are updated jointly in each single iteration. Again Hastings algorithm can be used with suitable proposal densities, such as the multivariate Gaussian, to generate Markov chains with limiting distribution, $\pi(X)$. Several studies based on such blocking schemes were implemented on different problems. Knorr-Held and Rue (2002) found block updates methods can sometime give misleading estimates even for long runs in a disease mapping problem unless the block is big enough. Roberts and Sahu (1997) emphasized that blocking can also make an algorithm converge more slowly and proved this by two different examples. On the contrary, other investigators, such as Liu *et al.* (1994) deduced that block sampling often improves the properties of the MCMC algorithm. Amit and Grenander (1991) found that the larger the blocks that are updated simultaneously, the faster the convergence. Indeed, the choice of the MCMC algorithm updating strategy can often dramatically affect its efficiency and therefore a sampling scheme should always be carefully chosen.

5.1.1 Convergence and accuracy

An issue of primary concern associated with the implementation of MCMC is the assessment of the convergence of the Markov chain. In practice, three pivotal questions arise. Firstly, how to decide how long should the burn-in period be in order for the chain to escape from the initial state influence? Secondly, how a chain should be monitored to decide whether it has converged to the posterior distribution? Finally, how large should a sample be to give inferences to a specified accuracy?

Theoretically, any Markov chain with a unique stationary distribution and an appropriate transition mechanism will eventually attain equilibrium. However it is usually impossible to say in advance when this equilibrium will be reached. Additionally as the simulated values come from a Markov chain, typically they will be serially correlated; Kass *et al.* (1998). Therefore, it is necessary to run the Markov chain for an adequate period of time before one can obtain posterior averages with sufficient accuracy. The high dependency between successive states can create a slow mixing Markov chain which causes slow convergence to the posterior model. Many convergence diagnostics have therefore been developed, some of which are discussed and used in the present study.

One easy diagnostic tool is a trace plot of the Markov chain. This is a time series plot which indicates whether the chain apparently converges as well as whether it is mixing properly. The mixing behaviour is assessed by the chain's "hairiness" about the chain's ergodic mean. Even if the chain is apparently mixing well, convergence might not be guaranteed. For example, if the posterior distribution was bimodal and the Markov chain was effectively sampling from a single mode then its time series plot will reveal a hairy trace but also a chain that is trapped within one of the two modes. Here, more iterations should be considered to allow for sampling from the second mode and hence accomplish convergence.

Another commonly used method for monitoring convergence is based on the output obtained from several independent realisations of the MCMC sampler, each of length $2n$, generated by using different random seeds and/or widely dispersed initial values. If the inferences drawn from a large number of chains, after discarding the first n iterations, are consistent and virtually identical with one another, then one may assume that convergence has been achieved. Nevertheless this is still not guaranteed.

By proposing several parallel chains, it is also possible to diagnose Markov chain convergence via the corresponding time-series plots, e.g., if any two chain traces do not meet within a bounded region of the sampling space beyond the burn-in discarded period then lack of convergence is suggested. Thus examining successive batches within a single chain, or between several parallel chains, can provide us with "negative" evidence that the process is not sufficiently long. However, there is no "positive" evidence that equilibrium has been reached. Gelman and Rubin's (1992) multiple chains convergence diagnostic compares the between and within chain variances, and the ratio of these two values is used to quantify convergence.

Many authors have discussed different convergence diagnostic tools, including Brooks and Roberts (1998), Brooks and Gelman (1998) and Cowles and Carlin (1996). In any particular MCMC application, a general acknowledgement is that no one convergence diagnostic tool is "best" and furthermore, different approaches might give inconsistent decisions.

Once convergence is reached, the next task is to decide when it is safe to stop the sampler. An important issue at this stage is the Markov chain autocorrelation. The chain's autocorrelation can be graphically represented by a time series plot known as the autocorrelation function (ACF); Chatfield (2003). This plot exhibits correlations between successive states of the chain at various lags. If the amount of dependence between the chain samples decreases rapidly, the ACF correlation diminishes fast. Accordingly, in this context, we would say that the chain is mixing well.

Although the ACF plot acts as an indicator of good mixing, it does not measure the speed of this mixing. To quantify this, the *integrated auto-correlation time* is used. In any MCMC approach, the aim is usually to estimate the the expectation $E_\pi(g)$ of a function of interest $g(X)$ under the target distribution π. However, due to the complexity of π the corresponding expectation integral given by $\int_{X\in\Omega} g(X)\pi(X)dX$, is analytically intractable, therefore it may be estimated by the empirical average of M values of g, e.g., $\{g(X^{(m)})\}$, for $m = 1,\ldots,M$, where $\{X^{(m)}\}$ are realisations of X sampled from the chain; $\overline{g}_M = \frac{1}{M}\sum_{m=1}^{M} g(X^{(m)})$, as discussed by Ripley (1987). A key issue is that MCMC samples are dependent. To counteract this problem, a larger sample is required for a respectable estimate of $E_\pi(g)$. The mean square error is commonly used to assess the quality of this estimator, it is given by $\text{MSE}(\overline{g}_M) = \text{var}(\overline{g}_M) + \text{b}^2(\overline{g}_M)$ where the first term denotes the variance of $\overline{g}_M$ and b represents the associated bias. These terms can be formulated as follows

$$\text{b}^2(\overline{g}_M) \;=\; [E(\overline{g}_M) - E(g)]^2 \approx \frac{1}{M^2}\left(\sum_{m=1}^{M}\left[E\left(g(X^{(m)})\right) - E\left(g\right)\right]\right)^2 \qquad (5.7)$$

$$\text{var}(\overline{g}_M) \approx \frac{\text{var}_\pi(g)}{M}\tau(g), \qquad (5.8)$$

where $\text{var}_\pi(g)$ represents the equilibrium variance of $g(X)$, and $\tau(g)$ is the required integrated autocorrelation time and is defined as

$$\tau(g) \quad = \quad 1 + 2\sum_{m=1}^{\infty} \rho_m(g), \tag{5.9}$$

where $\rho_m(g)$ denotes the autocorrelation of $g(X^{(m)})$ at lag m, calculated at equilibrium; see for example, Green and Han (1992).

It is clear that the MSE$(\overline{g}_M)$ is dominated by the asymptotic variance of the sample mean, var$(\overline{g}_M)$ because it is of order M^{-1} while that of the squared bias is M^{-2}. Thus, a key issue in any MCMC approach is the assessment of the sampling variance, $\text{var}_\pi(g)$. The integrated autocorrelation time value when independent realisations from π are generated is one. Thus if $\tau(g)$ is small, then we can conclude that the Markov chain is rapidly mixing. In contrast, a large value of $\tau(g)$ indicates that the chain is still slowly mixing and hence, we may be forced to use larger number of iterations or modify the sampling process in order to improve the accuracy of the resulting ergodic averages.

Several approaches have been introduced in literature to reduce the sampling variance, for example, methods of antithetic variables (Green and Han, 1992), importance sampling, model reparameterisation (Gelfand *et al.*, 1995) and altering jumping rules for Metropolis-Hastings algorithms (Gelman *et al.*, 1996).

5.2 A standard MCMC sampler

Our methodology is based on the reconstruction of an ionospheric region from a collection of TEC data by simulating the posterior distribution, $\pi(X|Y)$, defined in Eq (4.16). Initially, standard MCMC algorithms that may be capable of generating realisations of $X = (\mu, \sigma^2, \gamma, \epsilon^2)$ are used. To update the Markov chain, we select the basic single-site updating approach, so we may need to consider each parameter's full conditional distribution.

Since we have modelled the precision, τ^2, with an improper Gamma prior density, the full conditional posterior of τ^2 is

$$\pi(\tau^2|\mu, \sigma^2, \gamma, \beta, Y) \propto L(Y|\mu, \sigma^2, \gamma)\pi(\tau^2), \tag{5.10}$$

where $\beta = (\beta_\mu, \beta_{\sigma^2}, \beta_\gamma)$, thus after some simplifications we obtain

$$\pi(\tau^2|\mu, \sigma^2, \gamma, \beta, Y) \propto (\tau^2)^{\frac{I}{2}-1} \exp\left(-\frac{\tau^2}{2}\sum_{i=1}^{I}(Y_i - E(Y_i))^2\right). \tag{5.11}$$

It follows that τ^2 is distributed as a Gamma random variable, i.e.

$$\left(\tau^2|\mu, \sigma^2, \gamma, \beta, Y\right) \sim Ga\left(\frac{I}{2}, \frac{\sum_{i=1}^{I}(Y_i - E(Y_i))^2}{2}\right), \tag{5.12}$$

which is a well known distribution to sample. Thus, new values of ϵ^2 can be proposed by taking the reciprocal of the precision samples drawn by Gibbs sampling steps of Geman and Geman (1984).

In contrast, the complicated form of the likelihood function under the non-linear data model when combined with any vertical profile parameter GMRFs prior distribution, forms an awkward conditional distribution. This precludes any further Gibbs sampling. Instead, we turn to Metropolis sampling, Metropolis *et al.* (1953). The Metropolis proposal distributions are chosen to be univariate Gaussian with suitably scaled variances.

The algorithm described above is called the standard MCMC algorithm in this study. It begins by initialising the unknown variable values and fixing the smoothing parameters $\beta_\mu, \beta_{\sigma^2}, \beta_\gamma$. Then the algorithm iterates through the following steps:

1. For $n = 1, \ldots, N$, at each time t, update the vertical electron density profile parameters, systematically, by Metropolis sampling, e.g., if $Z_n \in \{\mu_n, \sigma_n^2, \gamma_n\}$ then

   - draw Z_n' from

   $$Z_n' \sim N(Z_n^{(t)}, \zeta_Z^2), \tag{5.13}$$

   where $Z_n^{(t)}$ denotes the current parameter value and ζ_Z^2 is the proposal density scale.

   - Accept Z_n' with probability $\alpha(Z_n^{(t)}, Z_n') =$

   $$\min\left\{1, \frac{L\left(Y|Z_n', X_{-Z_n}\right)\pi\left(Z_n'|Z_{\delta_n}\right)}{L\left(Y|Z_n^{(t)}, X_{-Z_n}\right)\pi\left(Z_n^{(t)}|Z_{\delta_n}\right)}\right\}, \tag{5.14}$$

   where Z_{δ_n} denotes the set of all orthogonal neighbours of Z_n.

   - If Z_n' is accepted then set $Z_n^{(t+1)} = Z_n'$, otherwise, $Z_n^{(t+1)} = Z_n^{(t)}$.
   - Sample τ^2 from its conditional distribution using the Gibbs sampler, and then set $\epsilon^2 = \frac{1}{\tau^2}$.

Table 5.1: *The synthetic plane vertical profiles parameter values.*

Profile	Peak height μ	Spread σ^2	Electron density under the curve γ
1	1.00	0.50	35.0
2	2.00	0.25	30.0
3	1.25	1.00	25.0
4	1.50	0.30	20.0
5	2.50	0.10	29.0

It is worth mentioning that in computing the contribution to the posterior distribution from the likelihood in the second step, only the nth column pixel electron densities need to be used to calculate $\{E'(Y_i)\}$, however, $E'(Y_i)$ only needs to be calculated if the ith ray passes through this column. This could be efficiently done by adding the new corresponding column weight terms and subtracting the current terms of Eq (4.4). After completion of each cycle, values of the updated parameters are stored.

5.3 Example

In order to test the standard MCMC algorithm defined in the last section and illustrate our methodology, we considered sampling the posterior model given in Eq (4.16) using a set of simulated TEC data. We adopted a 3×5 grid of pixels to represent a synthetic ionospheric plane with five vertical electron-density profiles. The corresponding profiles parameter values are given in Table 5.1. Thirty six propagation ray paths from a satellite to a chain of three ground-based receivers are constructed to intersect the synthetic plane as depicted in Figure 5-1. As mentioned earlier, each pixel electron density is determined at the mid-height of the corresponding pixel level.

The electronic engineers believe that the error associated with the observations recording process is negligible therefore the Gaussian noise associated with our procedure is chosen to be small with zero mean and variance $\epsilon^2 = 2000^{-1}$. In order to simulate the artificial TEC records that correspond to the 36 propagation paths, the error of each measurement, e_i, is sampled from this Gaussian model and then added to the record calculated expected value, $E(Y_i)$, using Eq (4.4), for $i = 1, \ldots, 36$. Preliminary tuning runs of length 10,000 iterations indicated that a "good" degree of spatial smoothness may be accomplished by assigning the interaction parameters $\beta_\mu, \beta_{\sigma^2}$ and β_γ to 0.003, 0.004 and 0.008, respectively, and the rates of accepting new values of the parameters of interest that lie between 20% - 40% may be obtained by using normal proposal kernels with the following scales: $\delta_\mu^2 = 0.05^2, \delta_{\sigma^2}^2 = 0.05^2$ and $\delta_\gamma^2 = 0.5^2$. The initial peak heights, spreads, electron densities under the profile curves and precision values are, respectively, 1.75, 0.1, 31.26 (records mean) and 1500. Figure 5-2 demonstrates some resulting time-series plots when 150,000 iterations of the algorithm are generated with random number seed 1.0. For comparison reasons, this seed will be used for all further related simulations, unless stated. The following constraints have been used to define the prior distribution:

$$0 < \mu_n \leq 3, \tag{5.15}$$

$$0 < \sigma_n^2 \leq 10, \tag{5.16}$$

$$0 < \gamma_n \leq 100, \tag{5.17}$$

for $n = 1, \ldots, 5$. The upper limit of the peak height is chosen to be 3 because this is consistent with the reconstruction grid configuration, whereas the other two upper limits are chosen to be big enough to allow the data to speak about themselves.

In part (a) of Figure 5-2, due to the effect of the "bad" starting values of the vertical profile parameters, the residual squared sums; $\sum_{i=1}^{36} (Y_i - E(Y_i))^2$ calculated in the first iteration after updating these parameters has substantially increased, therefore τ^2 has approximately dropped to zero. This reflects that the error associated with the first replication is very big. Further, all plots in this figure reveal that the Markov chain starts to converge after approximately 20,000 iterations, where most of the plots began to stabilise at their corresponding parameters true values. This is an expected result because the added simulated data white noise is very small. However, a few chains such as those for, μ_5 and γ_5, get trapped away from their true values (the matching colour solid lines). This may be due to the fact that these chains are not mixing well. As a simple diagnostic for convergence, or rather lack of it, we generated two other independent replications of the Markov chain, one by running the sampler using another seed and the other by initialising the algorithm with a different state, each of length 150,000. Figures 5-3 and 5-4 display the new time series plots.

Comparing the parallel sequences of Figures 5-2 - 5-4, a lack of convergence is evident as parameter chains such as those of σ_3^2, γ_2 and γ_5 are moving very slowly, or are stuck in separate regions in the posterior distribution. The "stickiness" of the chains can also be seen in the high correlations present either between the chain successive states, see for example Figure 5-5, or between the model parameters of interest as Figure 5-6 reveals. Thus, the problem on which we need to focus is how to reduce the existing high correlations in order to produce rapid mixing chains and hence improve the efficiency

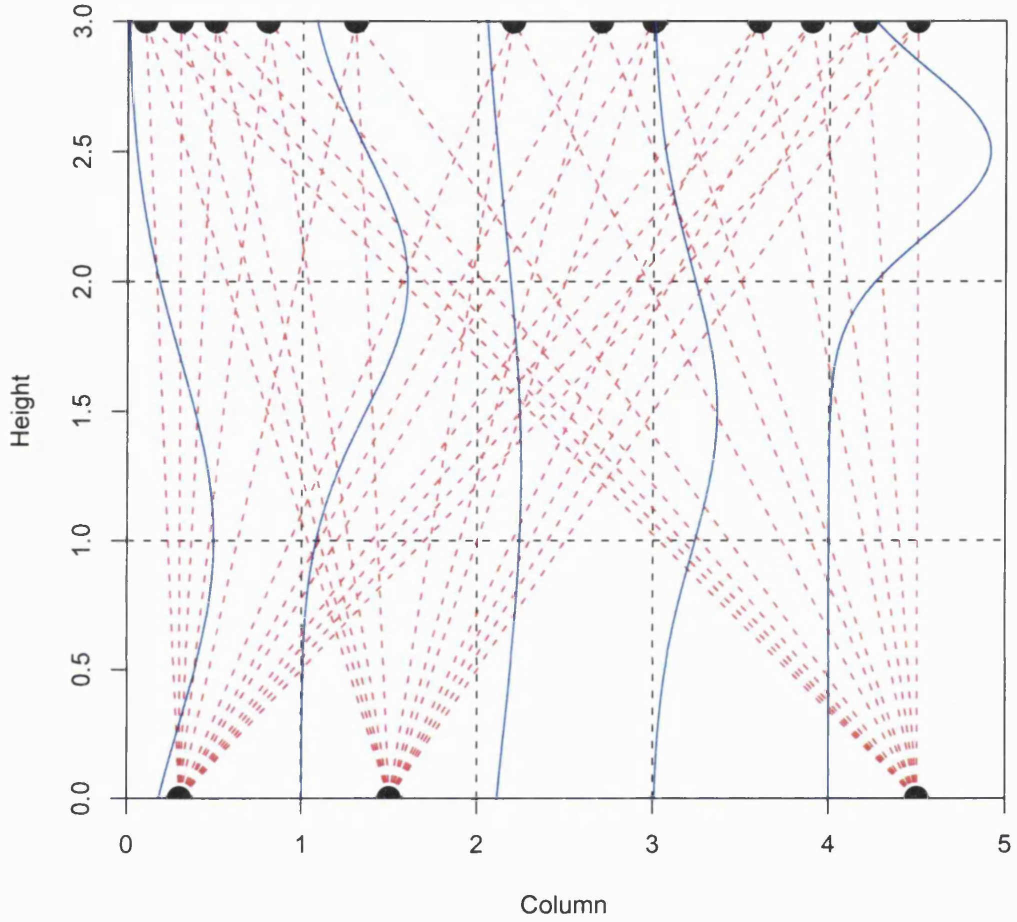


Figure 5-1: *The 2D grid of pixels showing the satellite-to-receiver propagation ray paths (red dotted lines) for an ionospheric tomography simulated data experiment. The blue curves represent the vertical electron-density profiles in each column.*

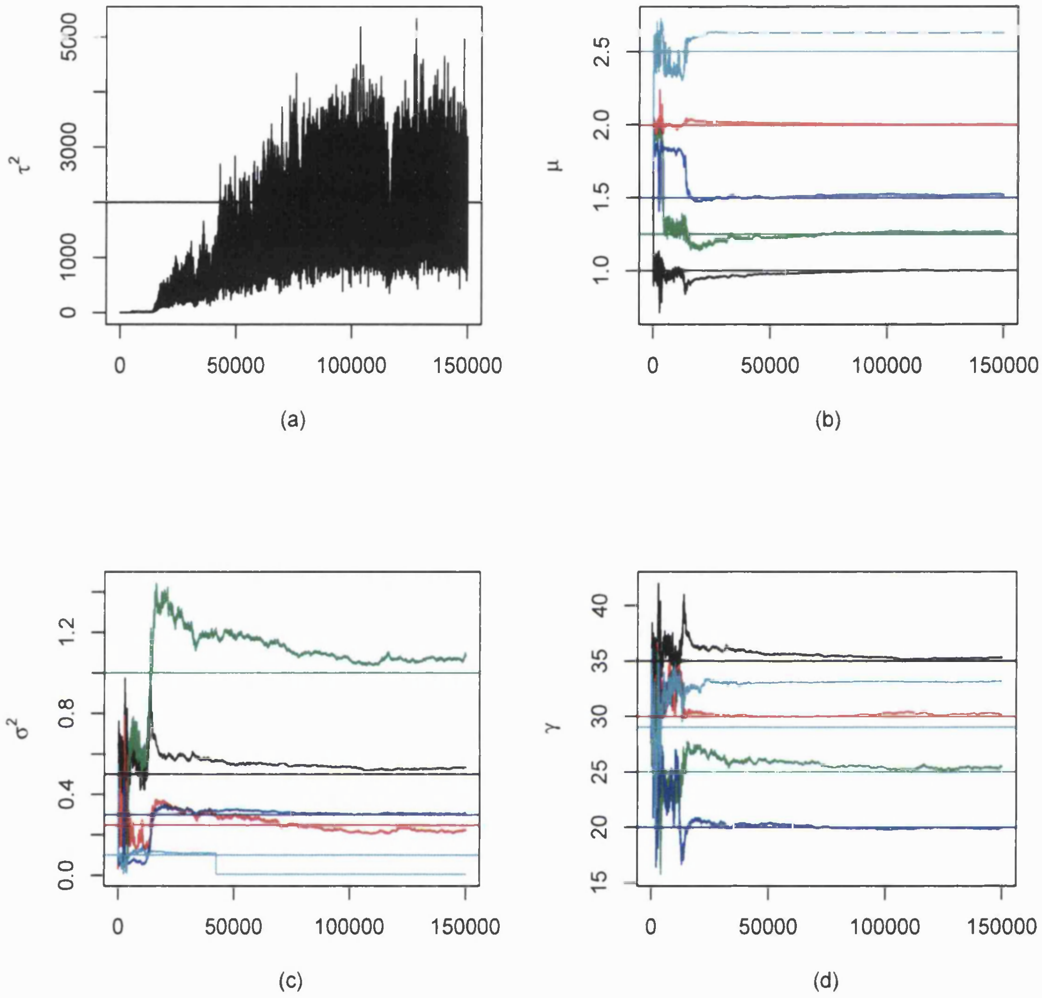


Figure 5-2: *Trace plots of the precision (a), peak heights (b), spreads (c) and the electron densities under the profile curves (d) recorded when the standard component-wise sampler is used. The profile parameters proposal distributions standard deviations are respectively, 0.05, 0.05 and 0.5, for all $i = 1, \ldots, 5$. The 1st, 2nd, 3rd, 4th and the 5th vertical profile chains are in black, red, green, blue and light blue, respectively. Solid lines represent the synthetic parameter values.*

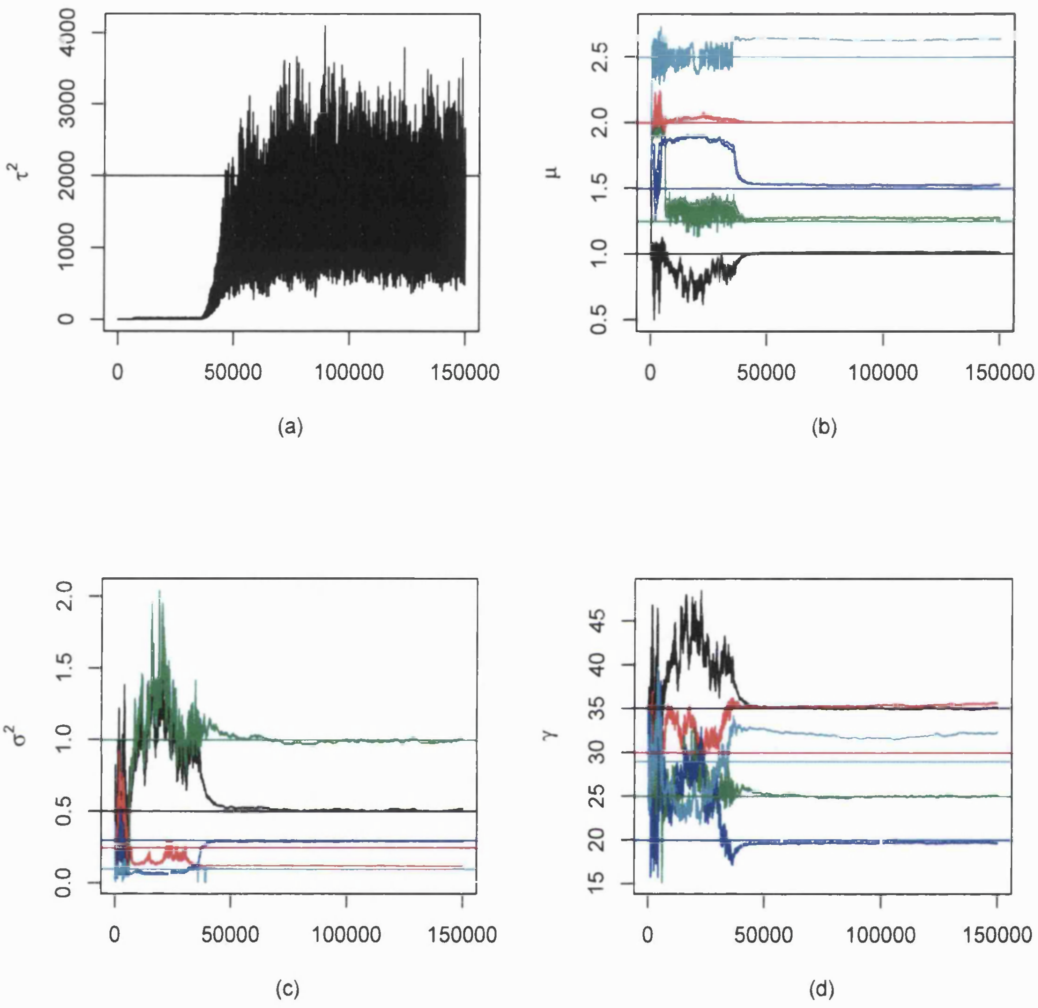


Figure 5-3: *Trace plots of the precision (a), peak heights (b), spreads (c) and the electron densities under the profile curves (d) recorded when the standard component-wise sampler is used. The profile parameters proposal distributions standard deviations are respectively, 0.05, 0.05 and 0.5, for all* $i = 1, \ldots, 5$. *The 1st, 2nd, 3rd, 4th and the 5th vertical profile chains are in black, red, green, blue and light blue, respectively. Solid lines represent the synthetic parameter values.*

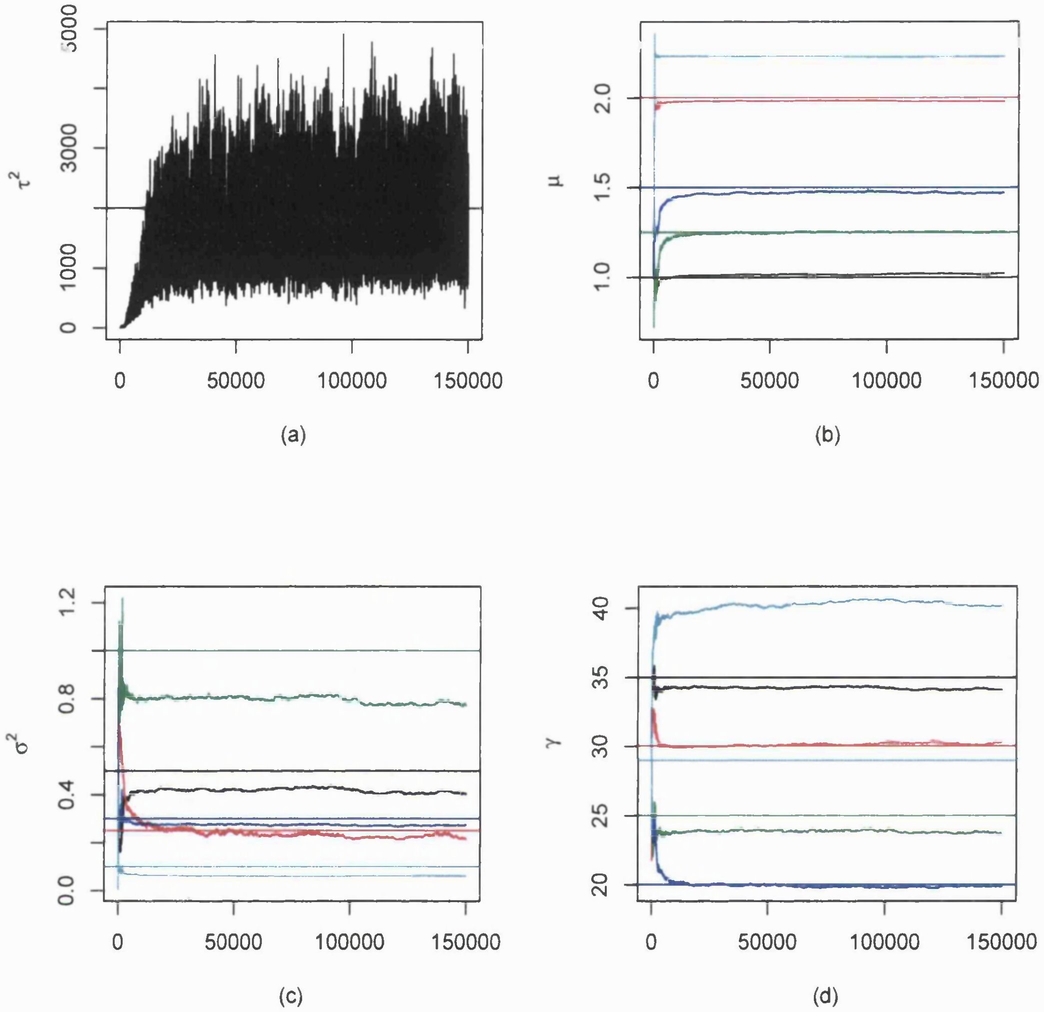


Figure 5-4: *Trace plots of the precision (a), peak heights (b), spreads (c) and the electron densities under the profile curves (d) recorded when the standard component-wise sampler is used. The profile parameters proposal distributions standard deviations are respectively, 0.05, 0.05 and 0.5, for all* $i = 1, \ldots, 5$. *The 1st, 2nd, 3rd, 4th and the 5th vertical profile chains are in black, red, green, blue and light blue, respectively. Solid lines represent the synthetic parameter values.*

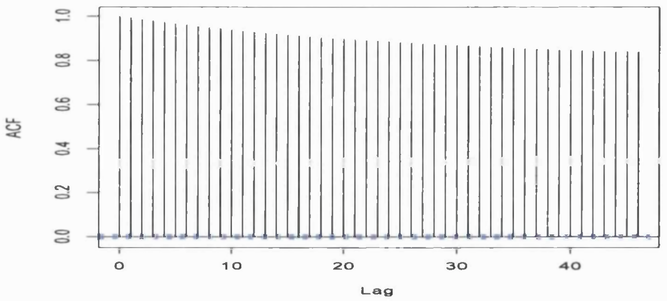


Figure 5-5: *Auto correlation plot for the last 50,000 accepted values of* γ_5 *using the standard component-wise sampler with the same input of Figure 5-2.*

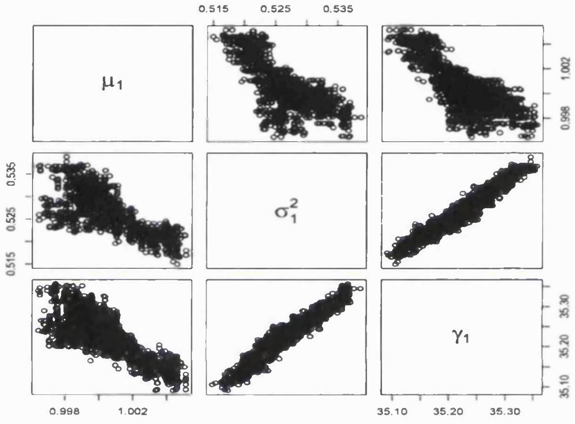


Figure 5-6: *Scatter plot of the first column profile parameters samples obtained from the last 50,000 iterations of Figure 5-2.*

of the MCMC sampler?

Recall that within this algorithm, the variables are systematically updated using a single-site scheme. This allows for the high dependencies between the parameter in turn and the other variables to influence the acceptance of the new move. Thus an alternative approach might be a block-wise scheme, where highly correlated parameters can be "blocked" and jointly updated. For this purpose we decided to run two different versions of block-wise based sampling MCMC algorithms in parallel. Within the first version, we updated parameters of the same type together, for example, update $\mu = \{\mu_1, \ldots, \mu_5\}$, then $\sigma^2 = \{\sigma_1^2, \ldots, \sigma_5^2\}$ and finally $\gamma = \{\gamma_1, \ldots, \gamma_5\}$, systematically, before updating τ^2. In this case, a new realisation of any single type, say μ' for instance, is simulated from a $N(\mu, \mathrm{I}_5\delta_\mu^2)$, where I_5 represents the identity 5×5 matrix, and then accepted with a Metropolis probability given by

$$\min\left\{1, \frac{\pi\left(\mu'|\ldots\right)}{\pi\left(\mu|\ldots\right)}\right\}. \tag{5.18}$$

Figure 5-7 shows the resulting time-series traces.

In contrast, the second block-wise sampler version updates set of parameters representing a single profile together, i.e. a proposal of the form $(\mu_n', \sigma_n^{2'}, \gamma_n')$ is generated using a Metropolis update and therefore is accepted with probability,

$$\min\left\{1, \frac{\pi\left(\mu_n', \sigma_n^{2'}, \gamma_n'|\ldots\right)}{\pi\left(\mu_n, \sigma_n^2, \gamma_n|\ldots\right)}\right\}, \tag{5.19}$$

where for example,

$$\begin{aligned} &\pi(\mu_n', \sigma_n^{2'}, \gamma_n'|\ldots) = \\ &\qquad L(Y|\mu_n', \sigma_n^{2'}, \gamma_n', \ldots)\pi(\mu_n'|\mu_{\delta_n})\pi(\sigma_n^{2'}|\sigma_{\delta_n}^2)\pi(\gamma_n'|\gamma_{\delta_n}). \end{aligned} \tag{5.20}$$

We will call this version the location-wise algorithm. As in the single-site and the first block-wise sampler, the last step in each cycle of the location-wise algorithm is a single Gibbs update of the precision parameter, τ^2. The output of this approach is displayed in Figure 5-8.

Figures 5-7 and 5-8 reveal that the suggested two block-wise updating schemes have not helped. This is because many of the resulting chains are still not mixing well since some chains got stuck in few states as can be seen in parts (b) and (c) of Figure 5-7, whereas several others are trapped in some small regions of the sample space far from the modal area as e.g., μ_4 and μ_5 chains of Figure 5-8 show.

The fact that the performances of the block-wise samplers are poor was not so surprising because generating a set of parameters candidates that all lie in a high probability region may be rare especially when the posterior distribution is narrow and many parameters are blocked and updated together as in the first block-wise sampler version.

Further, the resulting parallel nuisance variable, τ^2, realisations of Figures 5-2, 5-7 and 5-8 indicate that the component-wise standard MCMC sampler is the best updating approach to use because unlike the block-wise scheme results where 150,000 sweeps are

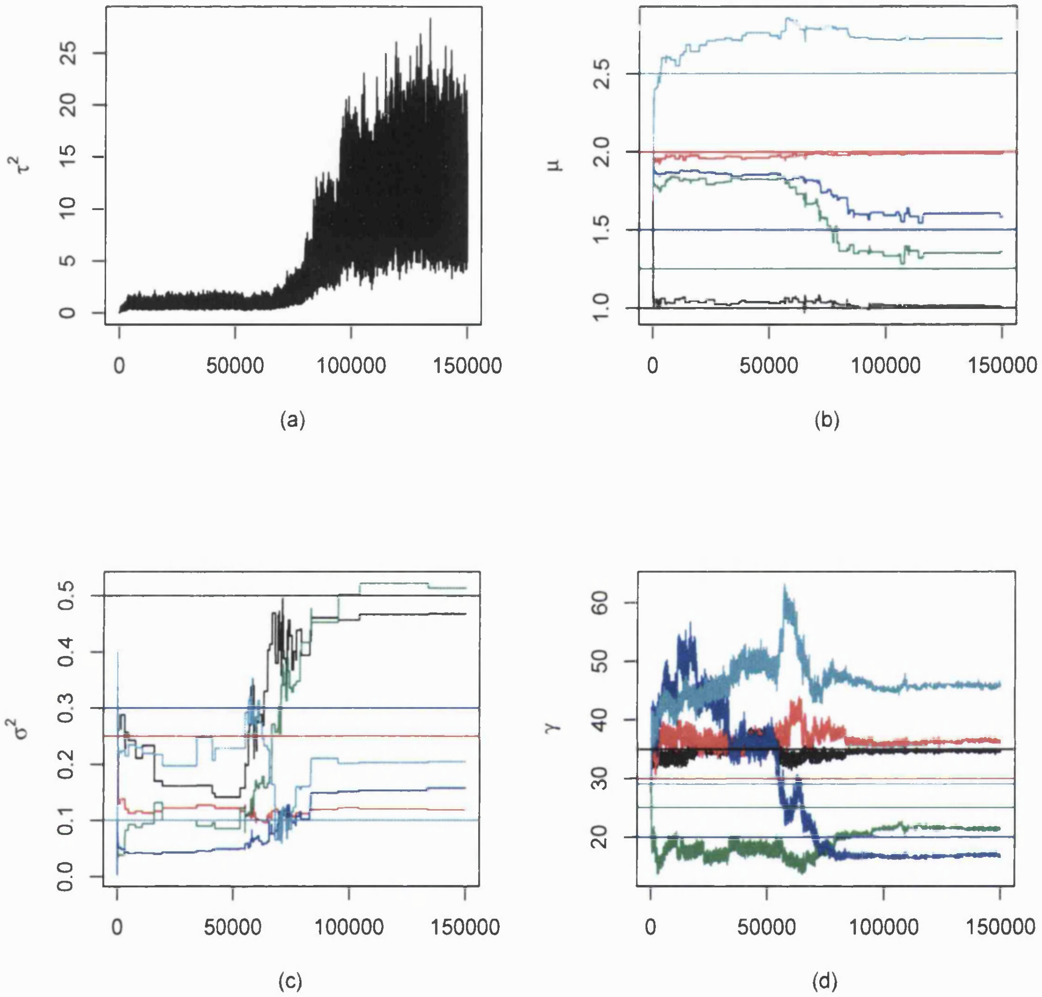


Figure 5-7: *Trace plots of the precision (a), peak heights (b), spreads (c) and the electron densities under the profile curves (d) recorded when the first version of the block-wise sampler is implemented. The profile parameters proposal distributions standard deviations are respectively, 0.05, 0.05 and 0.5, for all $i = 1, \ldots, 5$. The 1st, 2nd, 3rd, 4th and the 5th vertical profile chains are in black, red, green, blue and light blue, respectively. Solid lines represent the synthetic parameter values.*

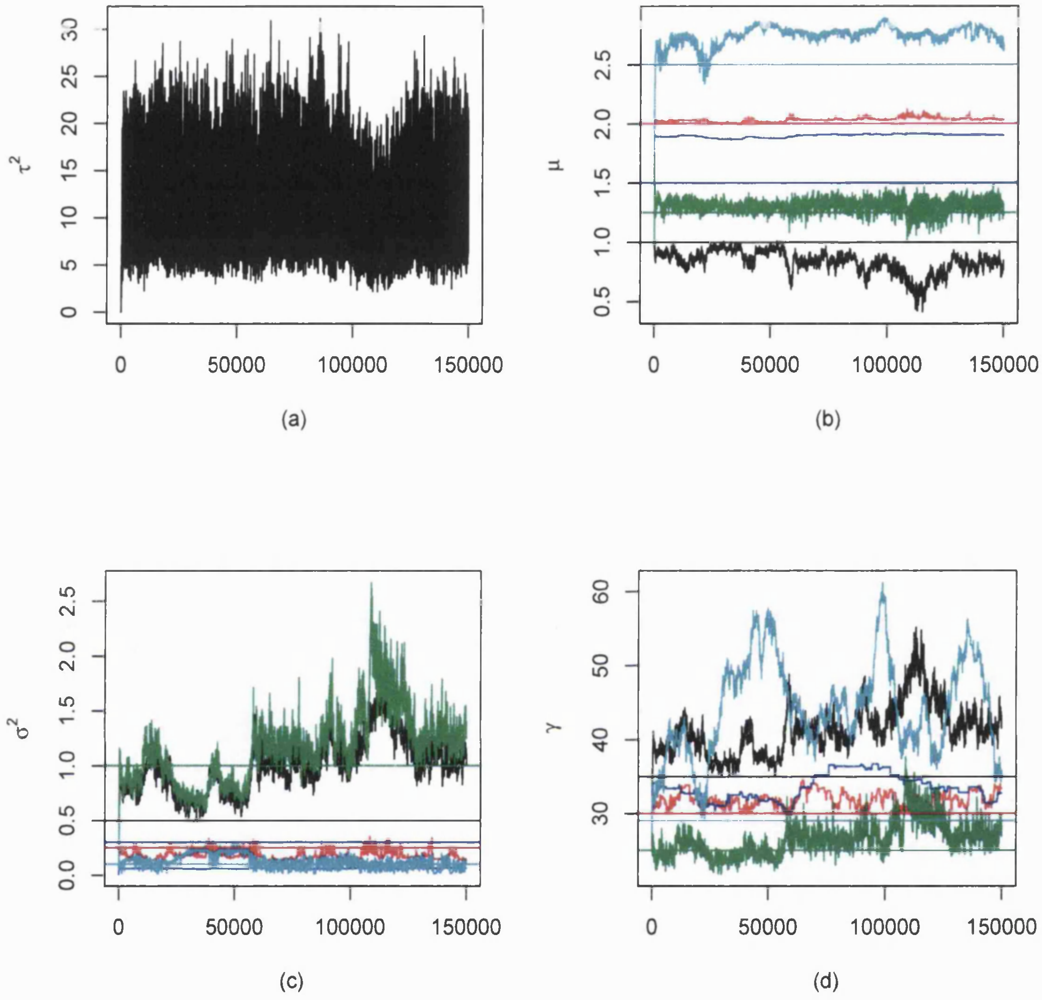


Figure 5-8: *Trace plots of the precision (a), peak heights (b), spreads (c) and the electron densities under the profile curves (d) recorded when the location-wise sampler is used. The profile parameters proposal distributions standard deviations are respectively, 0.05, 0.05 and 0.5, for all $i = 1, \ldots, 5$. The 1st, 2nd, 3rd, 4th and the 5th vertical profile chains are in black, red, green, blue and light blue, respectively. Solid lines represent the synthetic parameter values.*

insufficient as a burn-in period, the single-site approach requires approximately 20,000 iterations only to generate large and accurate values of τ^2. This reveals that in this case small residual squared sums; $\sum_{i=1}^{36}(Y_i - E(Y_i))^2$ are produced, where $E(Y_i)$ is calculated after updating each parameter. Thus, this scheme substantially increases the MCMC sampler efficiency if compared with the other two block-wise versions schemes. However, the mixing problem associated with this method reveals a more complicated structure in the posterior distribution. This issue will be followed up in the next section.

5.4 Development of a mixed MCMC algorithm

5.4.1 How to improve the sampling?

Following Bayes' theorem the target distribution is constructed by combining the likelihood distribution with the prior distribution. However, because our prior knowledge about the ionosphere internal structure is very limited the posterior distribution is strongly dominated by the likelihood. Thus to improve the MCMC sampling it is important to study what in the likelihood distribution causes difficulty in moving. This requires examining the way in which $\{\mu, \sigma^2, \gamma\}$ affect the likelihood. Since it is intuitively obvious that this effect originates from the local electron densities in the expected values terms, we decided to monitor these densities at a small number of places. For this purpose we will use the geometrical plots of the reconstructed vertical electron density profiles sketched at different stages of the simulation process versus the local electron density values, l_{n1}, l_{n2}, l_{n3}, where l_{ni} denotes the nth profile local electron density at the ith level, as demonstrated in Figure 5-9.

Let us for example consider the first and the last parts of the latter figure. Part (a) shows that the local electron densities at the three levels are well estimated and therefore the corresponding profile is almost accurately recovered whereas part (e) reveals that the third level local electron density in particular is highly overestimated and the generated profiles are approximately coincident creating a curve that apparently does not match the desired vertical profile. This indicates a poor mixing behaviour caused by the high correlations between the main parameters. These high interactions influence the whole column local electron densities to behave in a similar manner as can be seen in the scatter plot of Figure 5-10.

Moveover, although we may be able to match the local electron densities at the specified levels by values of $\{\mu_n, \sigma_n^2, \gamma_n\}$ for any column n which are far from the synthetic values, it is then hard to move away from these values because changing for example, the peak height μ_n, affects the whole column of the local electron densities and hence such a single variable change may be rarely accepted in this case.

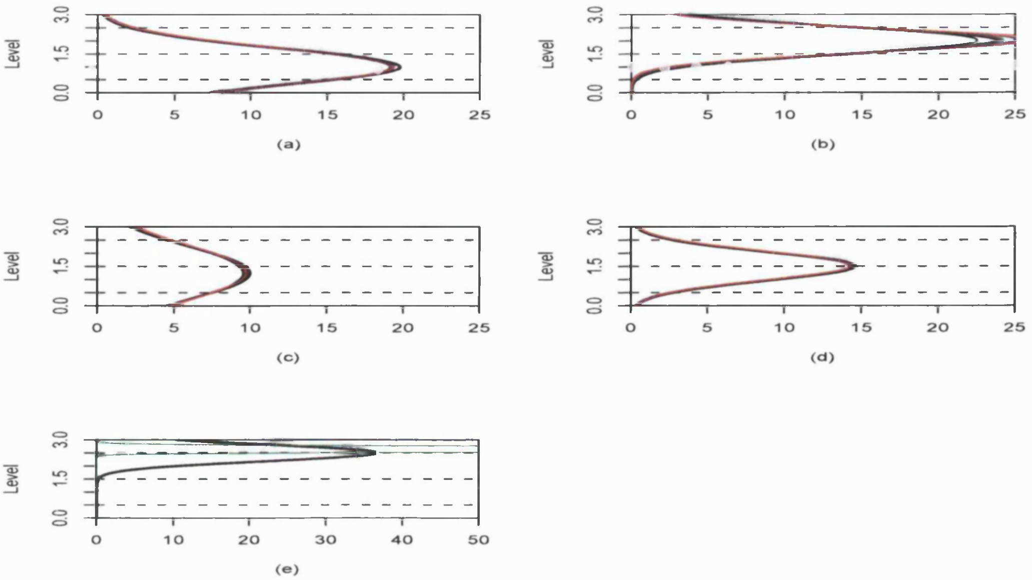


Figure 5-9: *The reconstructed vertical-electron density profiles generated at iteration 60,000, 70,000, . . . , 150,000 of a single run of the standard MCMC sampler using 3 levels at the heights specified by the dashed lines. (a),. . . , (e) represent the 1st,. . . , 5th column profiles, respectively. The thick dark black curves reveal the corresponding synthetic profiles.*

It is known that any straight line can be specified by two points but if instead a normally shaped curve was of interest, as in this study, then at least three adequately spaced points should almost be correctly fitted. Furthermore, for a reasonable fit, two of these points should roughly bound the curve whereas, the third point should approximately lie at the curve peak height. Therefore to best fit the vertical profiles it is necessary to initially fit the local electron densities that jointly create these profiles. This may be achieved by regularly generating direct moves in the horizontal direction within the vicinity regions of the local electron densities shown as the red bounded areas in Figure 5-11 for instance, in addition to those moves originally proposed in the major parameters directions.

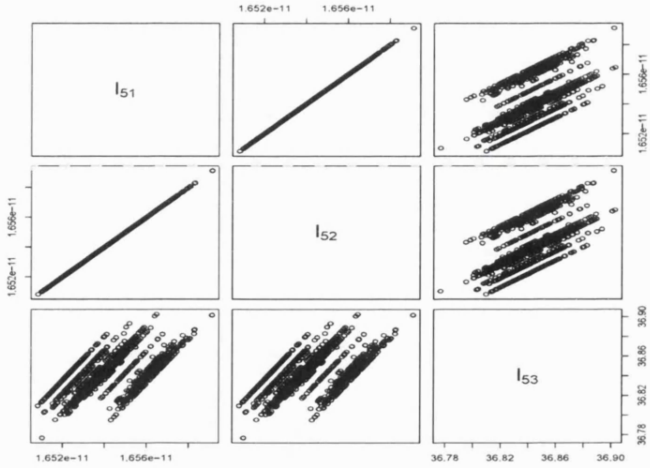


Figure 5-10: *Scatter plot of the 5th profile local electron densities obtained at the 3 distinct levels. Only the last 50,000 values of each relative chain are used. These results are associated with those chains of Figure 5-2.*

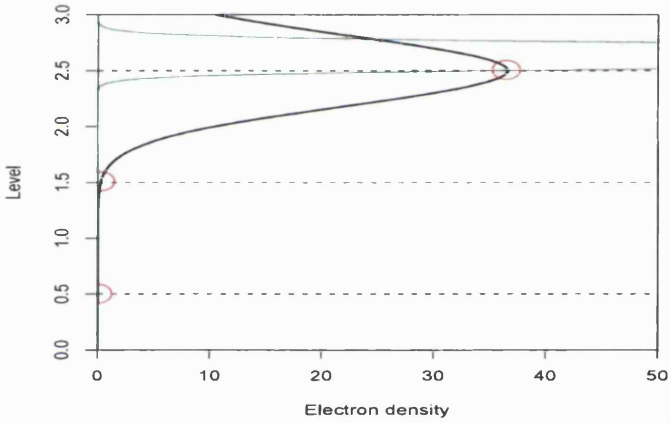


Figure 5-11: *The fifth column retrieved vertical-electron density profiles generated at iterations 60,000, 70,0000,..., 150,0000 of the single run results of the standard sampler used to plot Figure 5-2. The black curve is the synthetic profile. The bounded areas by the red shapes represent the neighbourhoods of the local electron densities at the three distinct grid levels mid-heights.*

To propose the new moves reparameterising ideas could be implemented via the following system of equations:

$$l_{n1} = \frac{\gamma_n}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2\sigma_n^2}(h(1)-\mu_n)^2}, \tag{5.21}$$

$$l_{n2} = \frac{\gamma_n}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2{\sigma^2}_n}(h(2)-\mu_n)^2}, \tag{5.22}$$

$$l_{n3} = \frac{\gamma_n}{\sqrt{2\pi}\sigma_n} e^{-\frac{1}{2{\sigma^2}_n}(h(3)-\mu_n)^2}, \tag{5.23}$$

where $h(j)$ denotes the mid-height of the jth layer, for $j = 1, 2$ and 3.

In general, reparameterisation methods are widely used in many MCMC application problems to accelerate convergence via improving the mixing of the simulated Markov chain. For such applications, the reader is referred to Gelfand *et al.* (1995).

5.4.2 The mixed algorithm

The sampler is mainly based on two simulation procedures. The first procedure generates proposals of (μ, σ^2, γ) using the component-wise standard MCMC sampler whereas the second, generates realisations of the vertical profiles local electron densities at the mid-height of the three levels and then transforms these values back to the original parameters set producing states of the form $(\mu', \sigma^{2'}, \gamma')$. Proposals of τ^2 are obtained by a Gibbs step after completion of any type of the above simulation processes. A full cycle of the new algorithm consists of only one of the two mentioned procedures. The algorithm is designed to iterate between these two procedures in turn. Therefore we called this sampler the *mixed algorithm*. The reparameterisation step will now be discussed in more detail.

Recall that when applying the standard MCMC sampler that after accepting a new move the corresponding profile local electron densities are calculated using the electron concentration function given in Eq (4.8). In contrast, a reparameterisation step involves the generation of $l'_{n1}, l'_{n2}, l'_{n3}$, for each profile. These values, if they satisfy the positivity condition, are then used to determine new estimates of the block $(\mu_n, \sigma_n^2, \gamma_n)$ by solving the associated equations similar to those given by Eqs (5.21) to (5.23) for $\mu'_n, \sigma_n^{2'}$ and γ'_n. Here every local electron density, l_{nj}, is sampled from a Gaussian distribution centred at the current density value with a fixed variance δ_n^2. Solving the system of equations with $h(j) = j/2$, see Appendix A, produces the following expressions :

$$\mu'_n = 1 + \sigma_n^{2'}\ln(l'_{n2}) - \sigma_n^{2'}\ln(l'_{n1}), \tag{5.24}$$

$$\sigma_n^{2'} = \frac{1}{2\ln(l'_{n2}) - \ln(l'_{n1}) - \ln(l'_{n3})}, \tag{5.25}$$

$$\gamma'_n = \sqrt{2\pi}\sigma'_n l'_{n1} \exp\left\{\frac{1}{2\sigma_n^{2'}}(0.5 - \mu'_n)^2\right\}. \tag{5.26}$$

If the calculated state $(\mu'_n, \sigma_n^{2'}, \gamma'_n)$ satisfies the necessary boundary conditions of Eqs (5.15) - (5.17) then it is accepted with a Metropolis-Hastings decision, i.e.

$$\alpha = \min\left\{1, \frac{\pi(\mu'_n, {\sigma^2}'_n, \gamma'_n|\cdots)q((\mu'_n, {\sigma^2}'_n, \gamma'_n) \to (\mu_n, \sigma_n^2, \gamma_n))}{\pi(\mu_n, \sigma_n^2, \gamma_n|\cdots)q((\mu_n, \sigma_n^2, \gamma_n) \to (\mu'_n, {\sigma^2}'_n, \gamma'_n))}\right\}, \tag{5.27}$$

where for example, $q((\mu_n, \sigma_n^2, \gamma_n) \to (\mu'_n, {\sigma^2}'_n, \gamma'_n))$ represents the joint proposal density of $\mu'_n, {\sigma^2}'_n$ and γ'_n given the current state $(\mu_n, \sigma_n^2, \gamma_n)$ and $\pi(\mu'_n, {\sigma^2}'_n, \gamma'_n|\cdots)$ is the conditional posterior distribution. To simplify our notations we shall use $q(\mu_n, \sigma_n^2, \gamma_n)$ and $q(\mu'_n, {\sigma^2}'_n, \gamma'_n)$ to represent $q((\mu'_n, {\sigma^2}'_n, \gamma'_n) \to (\mu_n, \sigma_n^2, \gamma_n))$ and $q((\mu_n, \sigma_n^2, \gamma_n) \to (\mu'_n, {\sigma^2}'_n, \gamma'_n))$, respectively.

We are interested in evaluating $q(\mu'_n, {\sigma^2}'_n, \gamma'_n)$. Fortunately, the unknown joint proposal density can be determined from the known joint proposal distribution by implementing the several variables transformation method and is therefore

$$q(\mu'_n, {\sigma^2}'_n, \gamma'_n) = q_{(l'_{n1}, l'_{n2}, l'_{n3})}(l'_{n1}, l'_{n2}, l'_{n3})\left|J'\right|, \tag{5.28}$$

where $|J'|$ is the transformation Jacobian given by

$$\left|J'\right| = \left|\frac{\partial(l'_{n1}, l'_{n2}, l'_{n3})}{\partial(\mu'_n, {\sigma^2}'_n, \gamma'_n)}\right|. \tag{5.29}$$

For our example, we have

$$\left|J'\right| = \frac{{\gamma'}_n^2}{(2\pi)^{\frac{3}{2}}{\sigma'}_n^9}\exp\left(-\frac{1}{2{\sigma^2}'_n}\left(3{\mu'_n}^2 - 9\mu'_n + \frac{35}{4}\right)\right).$$

The derivation of this term follows the same procedure placed in Appendix B. Thus, the new proposal state $(\mu'_n, {\sigma^2}'_n, \gamma'_n)$ is accepted with probability

$$\alpha = \min\left\{1, \frac{\pi(\mu'_n, {\sigma^2}'_n, \gamma'_n|\cdots)|J|}{\pi(\mu_n, \sigma_n^2, \gamma_n|\cdots)|J'|}\right\}, \tag{5.30}$$

since the known proposal distributions are by definition symmetric.

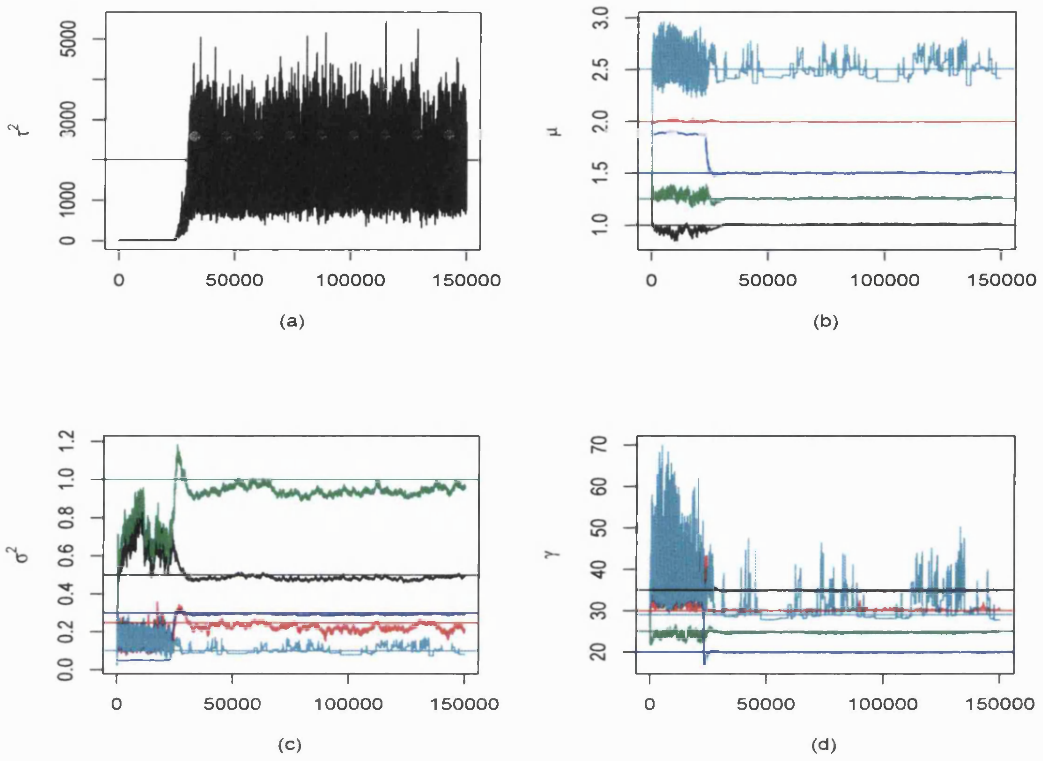


Figure 5-12: *Trace plots of the precision (a), peak heights (b), spreads (c) and the electron densities under the profile curves (d) recorded when the mixed sampler is implemented using a* 3×5 *grid. The 1st, 2nd, 3rd, 4th and the 5th vertical profile chains are in black, red, green, blue and light blue, respectively. Solid lines represent the synthetic parameter values.*

5.5 Applying the mixed algorithm

Having constructed the mixed algorithm, we now examine its performance on the simulated TEC data example of section 5.3 and discuss the associated implementation issues and results.

We began our investigation by making several preliminary runs in which the local electron density moves scaling parameters, δ_n^2, are tuned to produce reasonable acceptance rates. Figure 5-12 shows the resulting time-series plots with a proposal step size of 0.011^2 for the first four vertical profiles proposal distributions and 0.00007^2 for the last profile corresponding proposal density. The fifth profile scaling constant is

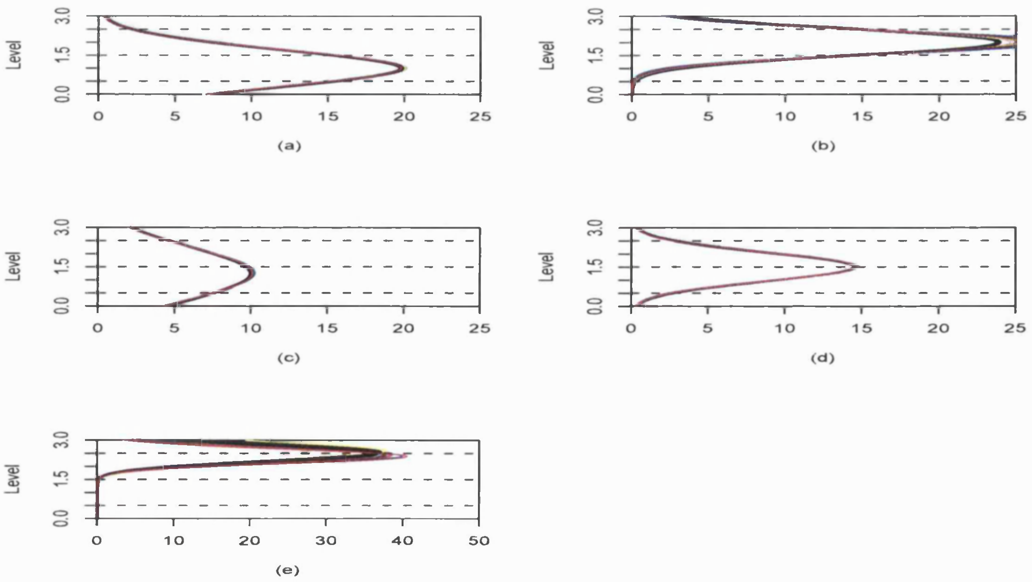


Figure 5-13: *The reconstructed vertical-electron density profiles generated at iterations 60,000, 70,000, ..., 150,000 of a single run of the mixed sampler using 3 levels. (a),..., (e) represent the 1st,..., 5th column profiles, respectively. The thick dark black curves reveal the corresponding synthetic profiles.*

chosen to be smaller than the other vertical profiles scaling values because unlike these profiles' first two levels local electron densities this profile's lower layers have negligible electron concentrations as Figures 5-9 and 5-13 reveal, therefore it is necessary to create small enough moves in order for these to be accepted. This configuration gives the following matching acceptance percentages: 33%, 32%, 36%, 37% and 10%, calculated after discarding the first 50,000 sweeps.

Comparing the new trace plots with those obtained from the matching experiment results of Figure 5-2 indicates an apparent improvement in the mixing behaviour of the produced Markov chains, especially those of the fifth vertical profile, at least from a visual perspective, because each chain is now moving more freely between many trapping states. The effect of this improvement is reflected in the nuisance parameter chain, e.g., part (a) of Figure 5-12, in the form of a rapid generation of large values of τ^2. It is also obvious from the plots of the sampled vertical profiles displayed in Figure 5-13, that in the most awkward profile of Figure 5-9, namely the fifth profile, reconstructed curves are now moving more freely within the close neighbourhood of the corresponding synthetic profile. This is due to the fact that horizontal moves in the

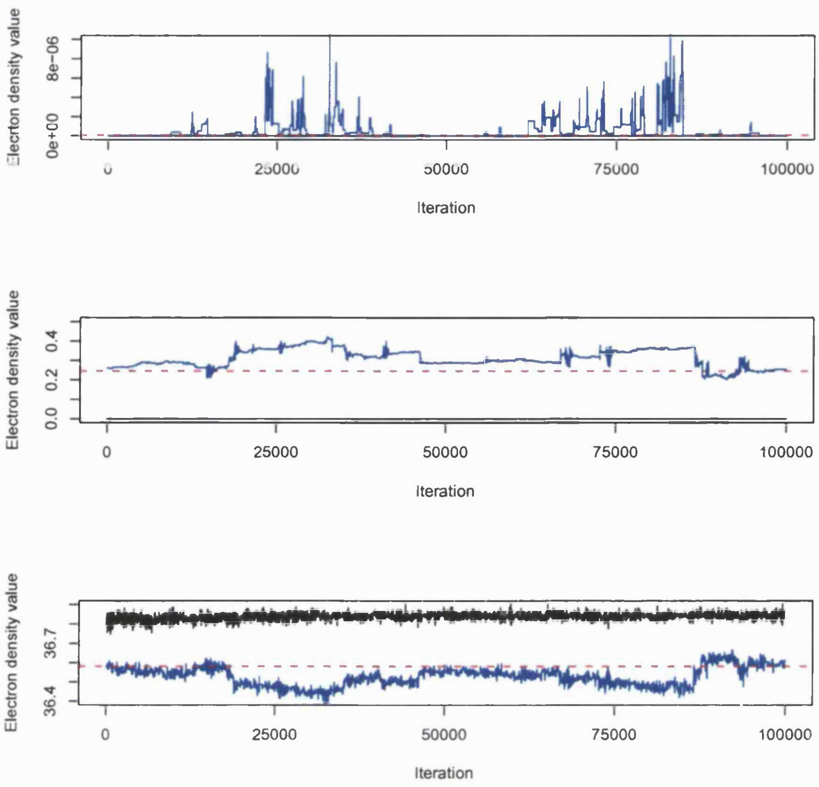


Figure 5-14: *The fifth vertical profile local electron density l_{5j} chains obtained after a burn-in phase of 50,000 from a single run of the standard sampler (black) and the mixed algorithm (blue) versus the synthetic local electron density (dashed red line) at the first (top), second (middle) and the third (bottom) grid levels.*

local electron density directions are created and accepted regularly, especially at the very low density heights. With the standard sampling such low concentrated altitude moves are rarely produced, see Figure 5-14.

This perhaps explains the "stickiness" of the last profile's restored curves, for example. Furthermore, and in contrast to the parallel scatter plot of Figure 5-10, Figure 5-15 indicates smaller correlations between the three local electron densities.

The effect of combining the reparameterisation process with the standard MCMC sampler within the new algorithm is not only noticeable in the last profile output but is also revealed in the other simulated chains results. For an example, see Figure 5-16. The three dimensional scatter plot obtained by implementing the mixed sampler

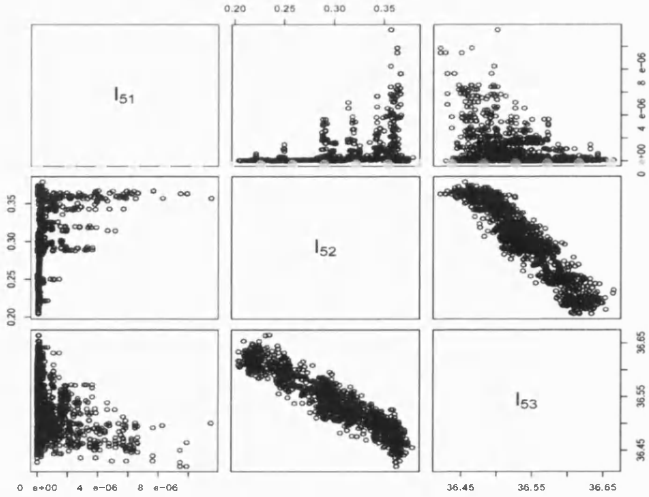


Figure 5-15: *Scatter plot of the fifth column vertical profile parameters l_{5j} samples obtained from the last 50,000 iterations using the mixed sampler with three levels based grid.*

displayed in the second panel of the latter figure is much more open than that derived by employing the standard MCMC algorithm alone (the first panel of the same figure). This indicates reduction of correlation between the components of the first vertical profile.

In general, the improvement in the results is due to the fact that by switching from the standard MCMC sampling to the reparameterising based sampling, the updating process is converted from a component-wise scheme into a block-wise scheme in which parameters of each individual profile are updated together but in a more sophisticated way than that used in the previous case of the location-wise updating approach. This effectively reduces the existing high correlations and hence accelerates convergence by improving the mixing of each chain. However, some big excursions and sticky regions that separate good mixing areas within a single chain are observed in a few chains, in particular those of the last profile, as Figure 5-12 reveals. This suggest that some chains are still not mixing adequately although other chains do. Raftery and Lewis (1992) highlighted the fact that the number of iterations required by the different chains to produce accurate ergodic averages of the quantities of interest can be dramatically different within the same problem. We are interested to know what is special about

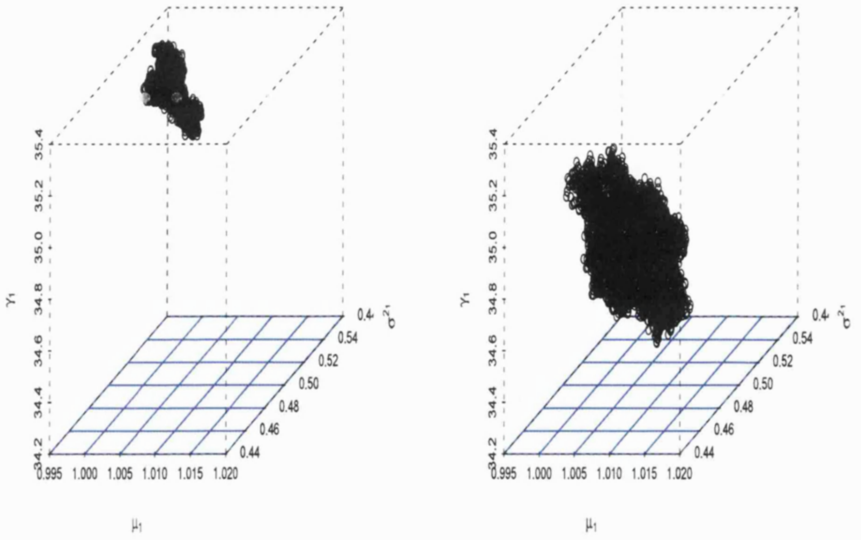


Figure 5-16: *The 3D scatter plots of the 1st profile parameters produced when the standard sampler (left) and the mixed algorithm (right) are employed with 3 levels. Only the last 50,000 values from the corresponding chains are used.*

this problem, and more specifically about the fifth profile, that make good mixing a difficult goal to achieve because it seems from the results obtained so far that the high posterior correlations are not the only cause of slow mixing. For this purpose we decided to take another look at the geometrical plot of the synthetic vertical profiles displayed in Figure 5-13, and in particular the last vertical profile curve. Although the latter profile is fitted at three different altitudes or points, Figure 5-13 reveals that two of these points lie at the curve's lower boundary, whereas the last point is located at the peak area leaving the upper boundary free. This inflates the degree of freedom present at this altitude as can be seen in the corresponding plot of Figure 5-13, and hence only a rough estimate of the corresponding local density is produced. This in general affects the fitting quality.

To overcome this problem, the idea of increasing the necessary number of restoration levels is considered and the mixed algorithm is modified and implemented accordingly, as discussed in the next section.

5.6 The mixed algorithm with more levels

In reality we do not know how many grid levels are necessary to produce good inferences for the ionospheric model parameters of interest. Almost certainly three layers are insufficient and therefore more levels may be considered. Hence, the present version of the mixed algorithm cannot be implemented. To modify it we propose to use only three suitably chosen layers d values for every vertical profile because fitting each requires the estimation of three parameters, namely, μ_n, σ_n^2 and γ_n which in this case necessitates solving a system of three equations. Moreover, the same layers of each profile should be used whether moving forward or backward within the simulation process. The latter constraint, if fulfilled, theoretically ensures convergence to the desired posterior distribution because the corresponding Markov chain detailed balance condition is satisfied. That is, if n_a, n_b and n_c are the nth profile chosen heights with current local densities l_{n_a}, l_{n_b} and l_{n_c} and current state X, then a new accepted state $X' \neq X$ with accepted proposals l'_{n_a}, l'_{n_b} and l'_{n_c} given the present state X and transformation kernel $P(X, X')$ satisfies the time reversibility condition given by Eq (5.1). The choice of the three layers needed in the reparameterisation procedure is entirely deterministic and can differ from one data set to another.

To hopefully improve the mixing of the simulated data example chains discussed in the last section, we re-discretised the two dimensional synthetic plane grid by adding an extra level, that is the reconstruction grid is converted from a 3 × 5 into a 4 × 5 grid of pixels. We then implemented the new version of the mixed sampler by assigning 0.011^2 for the first four vertical profiles proposal scalings and 0.002^2 for δ_5^2. Only the upper three levels local electron densities are used for reparameterising the model. The choice of these levels is based on the fact that within each column at least two of these layers lie in non-negligible electron concentration regions. For example, see Figure 5-17. The resulting acceptance rates are respectively, 0.3, 0.37, 0.29, 0.36 and 0.33. Figures 5-18 and 5-19 show the new inversion output.

By comparing the matching generated paths of Figures 5-12 and 5-18, we deduced that the inclusion of an extra level has obviously improved the mixing of the generated chains in general. As a result of this improvement a good fit of the vertical profiles local electron densities has been achieved as Figure 5-19 traces reveal.

We are also interested in testing the performance of the standard MCMC algorithm in this case. Therefore, we decided to run the component-wise based sampler using the same number of layers of the above experiment. The results of this run are displayed in Figure 5-20. Unfortunately, adding an extra level in this case did not help the sampler

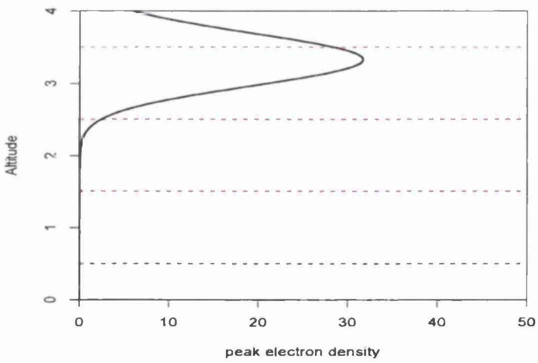


Figure 5-17: *The 5th vertical electron-density profile re-scaled into the* 4 × 5 *reconstruction grid. The red dashed lines represent the altitude of each local electron density involved in the reparameterisation process.*

to perform better and, unexpectedly, increased the burn-in period to slightly above 100,000. Further, the fifth vertical profile peak height and spread chains got stuck again in one state or more as the corresponding chains traces reveal while γ_5 converges very slowly. This is due to the following: despite the fact that the vertical profile is fitted using an extra local electron density point, this point lies in almost a zero concentration region as displayed in Figure 5-17, giving very little extra information about the density at higher altitudes. This necessitates fitting more layers.

In contrast to the equivalent mixed sampler experiment results of Figure 5-18, we concluded that the mixed algorithm performs much better than the standard algorithm in this case because all chains are mixing much faster.

Having obtained some well mixing chains using the mixed sampler we assessed the accuracy of the resulting expectations by calculating the corresponding parameters mean square errors. These errors are displayed in the first column of Table 5.2. From these values we deduced that all variables of interest are almost well estimated except γ_5 which relatively has a high MSE (defined in Section 5.1.1) of approximately 18. In general, since these normally curved profiles are fitted at only four points a better fit may be achieved by increasing the number of layers to be fitted. Therefore, we modified the grid to include 6, 12 and 24 layers instead to find out how many levels are adequate to produce estimates of a reasonable accuracy. The required reparameterisation scaling constants are fixed at 0.011^2 for the first four vertical profiles and 0.00007^2 for the last one. All acceptance percentages lie between 20% and 40%. Some of the new results

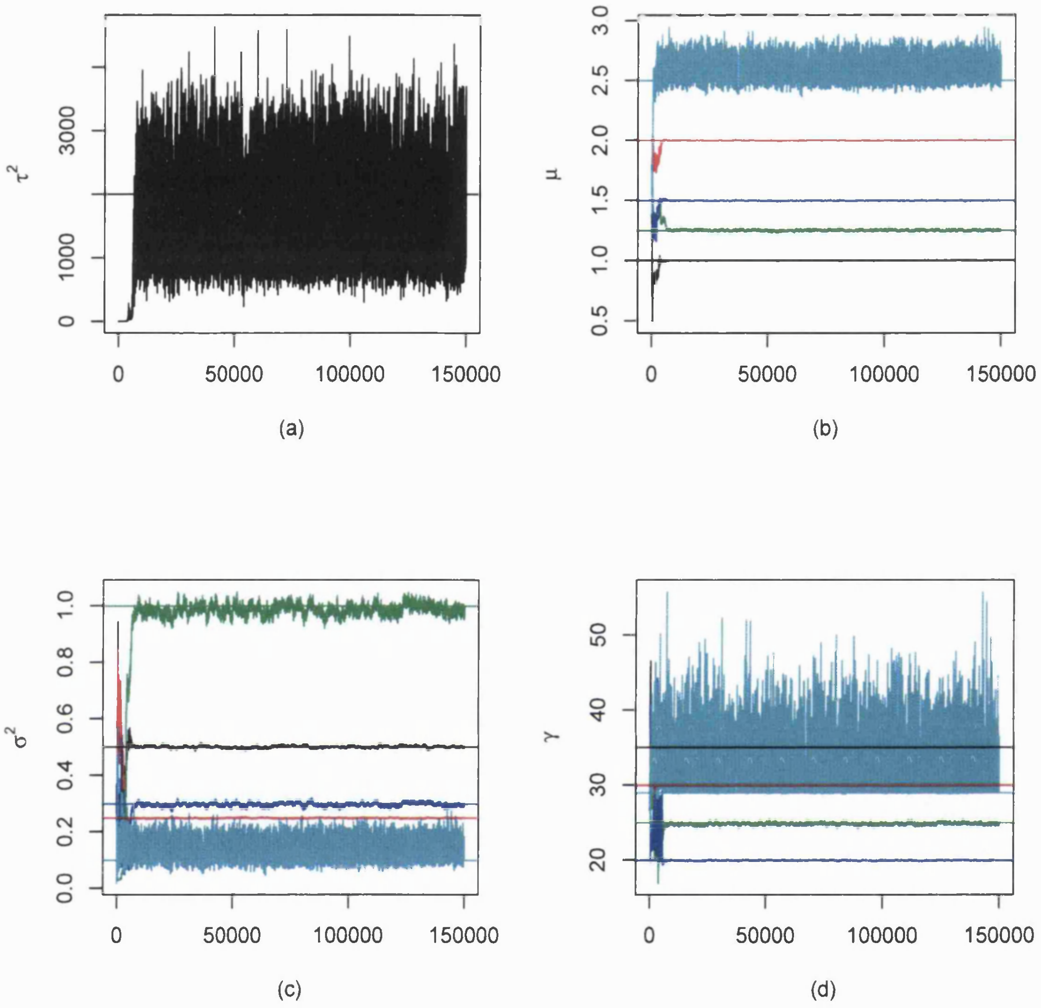


Figure 5-18: *Trace plots of the precision (a), peak heights (b), spreads (c) and the electron densities under the profile curves (d) recorded when the mixed sampler is implemented using a* 4×5 *grid of pixels. The 1st, 2nd, 3rd, 4th and the 5th vertical profile chains are in black, red, green, blue and light blue, respectively. Solid lines represent the synthetic parameter values.*

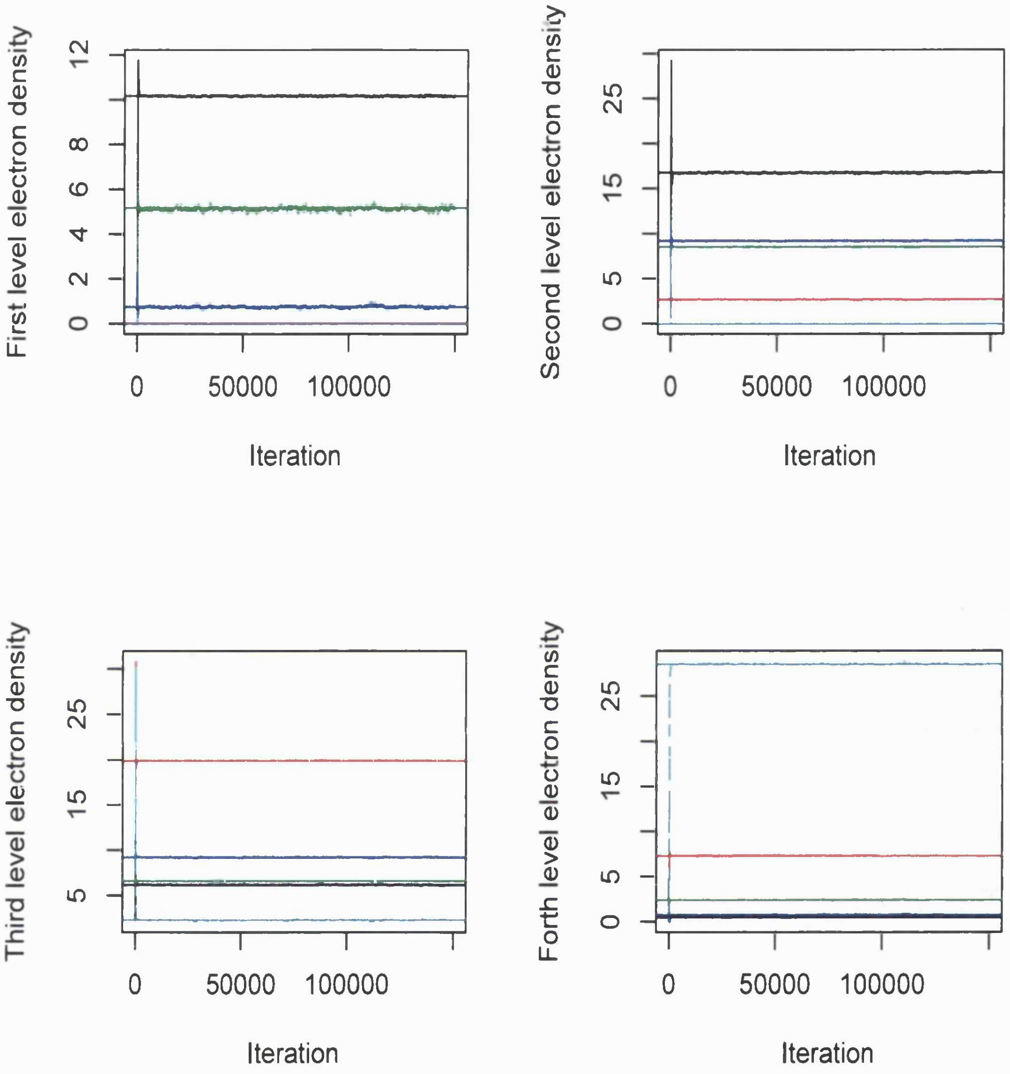


Figure 5-19: *Trace plots of the first (a), second (b), third (c) and the forth (d) levels local electron densities recorded throughout a run time of length 150,000 obtained by employing the mixed algorithm with 4 levels. The solid lines represent the synthetic electron density values at these heights.*

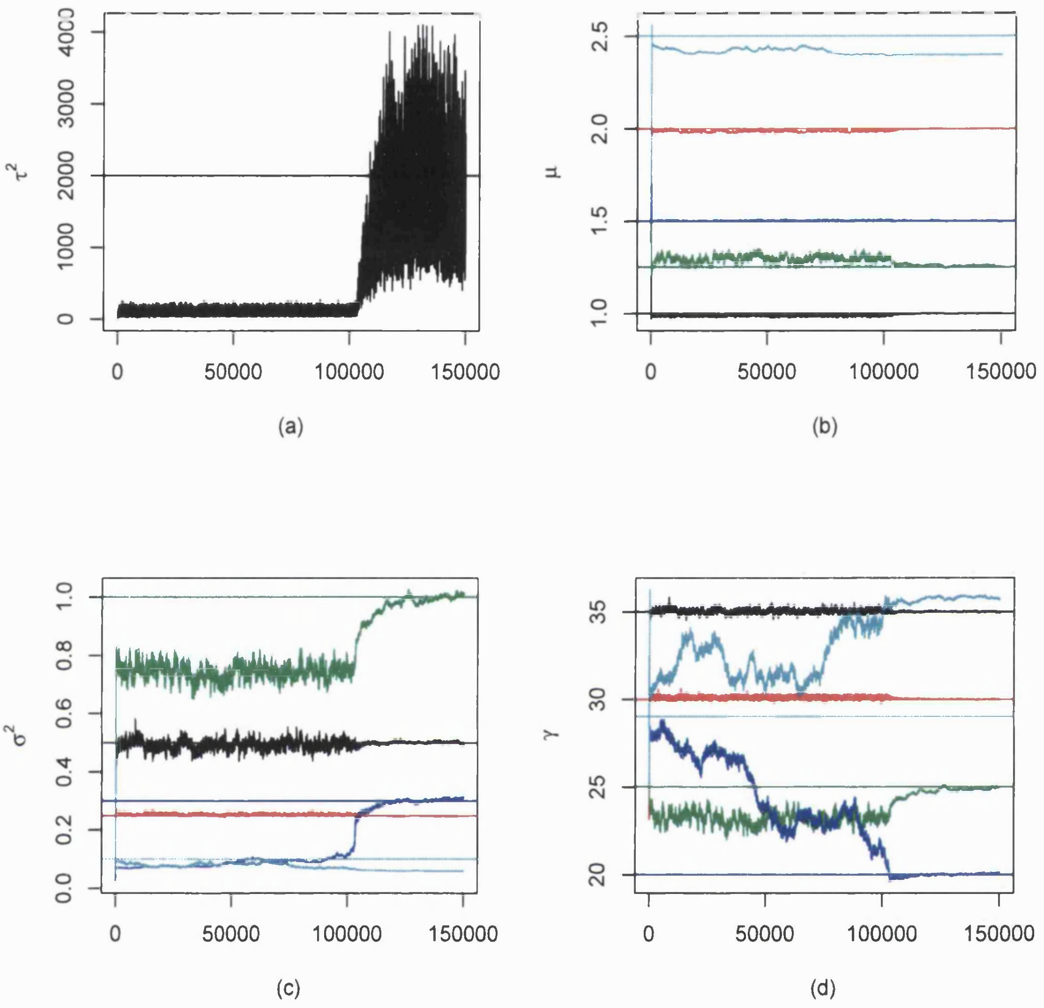


Figure 5-20: *Trace plots of the precision (a), peak heights (b), spreads (c) and the electron densities under the profile curves (d) recorded when the component-wise standard sampler is implemented using a* 4×5 *grid. The 1st, 2nd, 3rd, 4th and the 5th vertical profile chains are in black, red, green, blue and light blue, respectively. Solid lines represent the synthetic parameter values.*

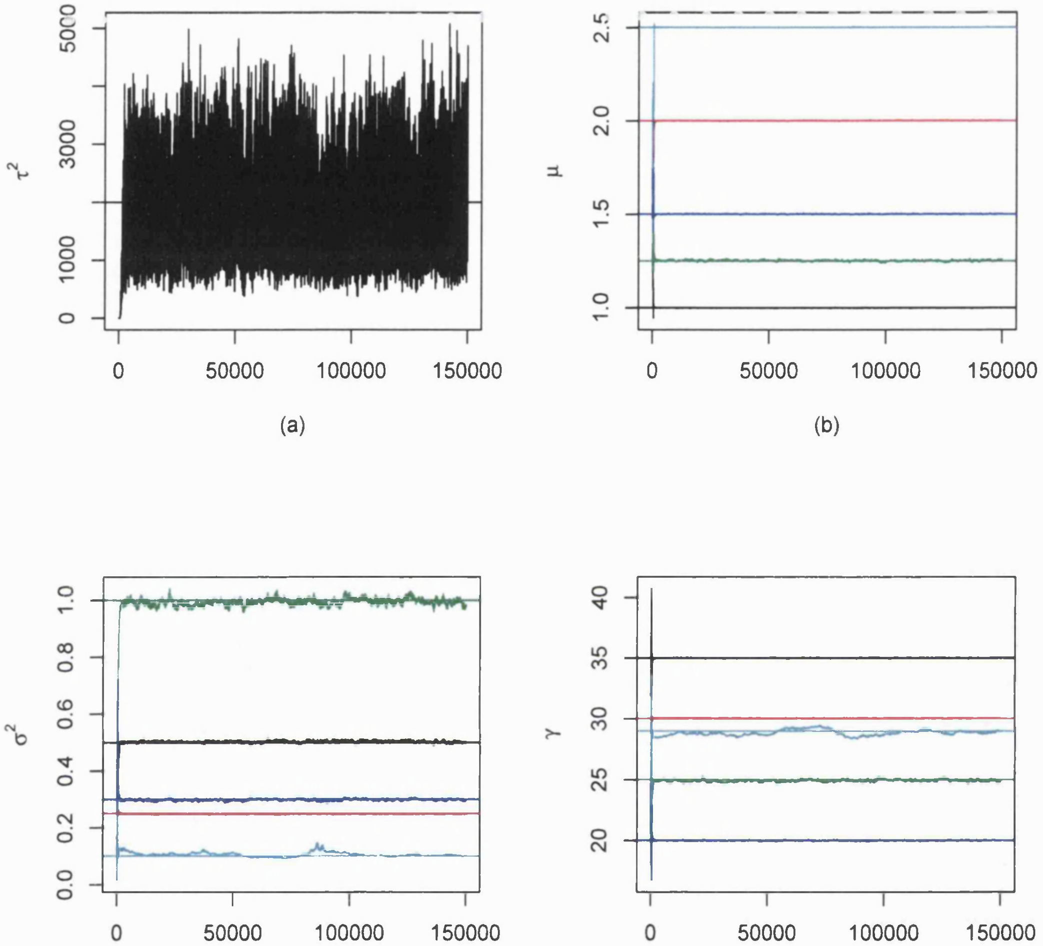
Table 5.2: *The mean square error values obtained when the mixed algorithm is implemented with the specified number of layers, n_j.*

Var	MSE ($n_j = 4$)	MSE ($n_j = 6$)	MSE ($n_j = 12$)	MSE ($n_j = 24$)
μ_1	0.0000011	0.00000039	0.00000022	0.000000088
μ_2	0.0000027	0.00000059	0.00000018	0.000000086
μ_3	0.000064	0.000015	0.0000062	0.0000026
μ_4	0.0000037	0.000002	0.0000012	0.00000063
μ_5	0.014	0.00000033	0.000000045	0.000000038
σ_1^2	0.000021	0.000023	0.0000028	0.0000042
σ_2^2	0.0000016	0.0000044	0.0000011	0.0000023
σ_3^2	0.0005	0.00016	0.00024	0.00018
σ_4^2	0.000052	0.000013	0.000017	0.000011
σ_5^2	0.0023	0.000081	0.0000014	0.0000012
γ_1	0.001	0.00035	0.00012	0.00013
γ_2	0.00059	0.0005	0.00076	0.00057
γ_3	0.044	0.013	0.0125	0.0056
γ_4	0.0024	0.00088	0.00028	0.00027
γ_5	17.6	0.0637	0.00023	0.00025

time series plots are demonstrated in Figures 5-21 and 5-22.

The last two figures show that by fitting more layers during the reconstruction process the sampler significantly improves its performance which is seen in the form of speedy mixing chains and extremely short burn-in phases since the model mode is approximately found after only a few iterations. This reflects the fact that a better fit of the simulated data has now been produced because more accurate estimates of the desired distribution parameters are obtained as the corresponding mean square error values recorded in Table 5.2 indicate. All values in this table are calculated using the last 100,000 accepted proposals from each chain.

Moreover, the latter table compares the MSE when the number of layers considered are 4, 6, 12 and 24. From this table and the plots in Figures 5-18, 5-20 and 5-21 it can be concluded that fitting six layers is adequate to obtain accurate inferences of all vertical profile parameters especially those of the 5th profile since the MSE of γ_5 for example has dropped from almost 18 to slightly less than 0.1. Further, there is no big difference between the MSE values obtained with $n_j = 6$ or 24. Thus fitting the



(a)

(b)

(c)

(d)

Figure 5-21: *Trace plots of the precision (a), peak heights (b), spreads (c) and the electron densities under the profile curves (d) recorded when the mixed sampler is implemented using a* 6 × 5 *grid. The 1st, 2nd, 3rd, 4th and the 5th vertical profile chains are in black, red, green, blue and light blue, respectively. Solid lines represent the synthetic parameter values.*

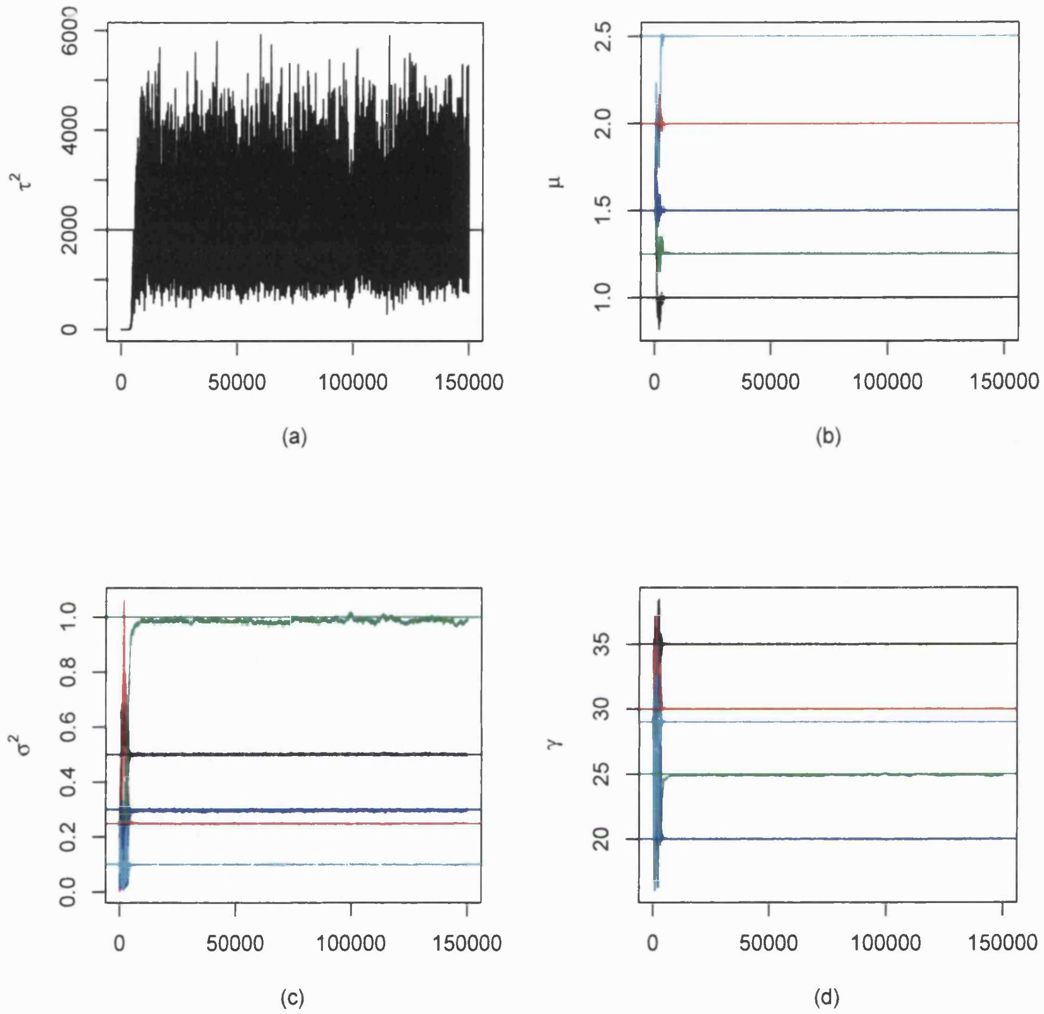


Figure 5-22: *Trace plots of the precision (a), peak heights (b), spreads (c) and the electron densities under the profile curves (d) recorded when the mixed sampler is implemented using a* 12×5 *grid. The 1st, 2nd, 3rd, 4th and the 5th vertical profile chains are in black, red, green, blue and light blue, respectively. Solid lines represent the synthetic parameter values.*

artificial data using six layers is sufficient to produce good inferences for all parameters of interest. This will also reduce the computing time required to effectively sample the posterior distribution of interest.

5.7 Summary

In this chapter, we have considered a simulated TEC data set example for which we implemented MCMC based algorithms to sample the underlying posterior distribution. The standard Metropolis algorithm results suggest that this sampler can be very slow to converge especially if the reconstruction grid is discretised with a small number of levels, e.g., 3 or 4. The new approach developed, the mixed algorithm, in which ideas of reparameterisation are used, gave promising results even when the grid has only a few rows. Since this algorithm can only be implemented using three levels local electron densities, the choice of which three layers to use is important. A general rule of thumb is to select three levels within each column such that two of them bound the vertical profile at non-negligible electron density altitudes and the third lies within the peak height region. However, a suitable set of layers is not always known in advance. One way that may help in choosing these levels is to make use of the available *a priori* information. An alternative approach which we recommend is to generate a preliminary run of the standard MCMC sampler using a "good" initial state. Based on the resulting profiles' geometrical plots the required layers could then be chosen.

Although, the mixed sampler performs generally well with small number of layers, the empirical estimates obtained from this approach are more accurate when sufficient number of layers are considered.

For the real TEC data problem, the mixed algorithm will be used in the next chapter. Since the ionospheric region of interest extends vertically from 80 km to 1200 km, it is important to choose a sensible number of rows, n_j, to discretise the 2-dimensional grid. This number should not be very small nor very large. For very small n_j, the inferences is expected to be not so accurate. On the other hand, if n_j is set to be very large the algorithm sampling CPU time will increase. This will badly affect the efficiency of the algorithm.

Chapter 6

TEC data for tomographic reconstruction via MCMC

6.1 Introduction

The simulated TEC data example studied in the previous chapter gave promising indications that the new MCMC sampler, namely the mixed algorithm, can produce accurate restorations of high-level images that deals with more complex features of the scene. In this study, the high level image model specifies the image as a collection of vertical profiles, each covers a whole column set of pixels in the grid. Capturing the attributes of these profiles based on spatial patterns of electron densities, rather than the low level image task of estimating each pixel's content of free electrons, is considered a high level image task. The next step is to implement the mixed sampler with real TEC measurements. The NNSS TEC data introduced in Section 4.1 will be used in this work. Figure 6-1 displays these measurements. In this figure, each set of points clustered to approximately form a curve represents a set of TEC observations that is collected by only one of the four receivers.

Table 6.1 gives the receivers' approximate locations in degrees of latitude, the number of records collected by every individual receiver, and the approximate detecting range of each. Since the radio signals were received between 12°N and 61°N in latitude, and the satellite orbits at an altitude of about 1100 km, the two dimensional grid is designed to cover all signals ray paths and therefore it extends horizontally from 10°N to 65°N, and vertically from the surface of the Earth to 1200 km above it.

This grid has a resolution of 2.5° latitude by 40 km altitude. The configuration discretises the ionospheric plane of interest into a 30 × 22 grid of pixels as shown in Figure 4-1.

Table 6.1: *Real TEC data geometry.*

Receiver Number	Location	Number of records	Detecting range
1	38°N	176	21°N - 60°N
2	42°N	50	34°N - 45°N
3	44°N	157	27°N - 61°N
4	36°N	189	12°N - 54°N

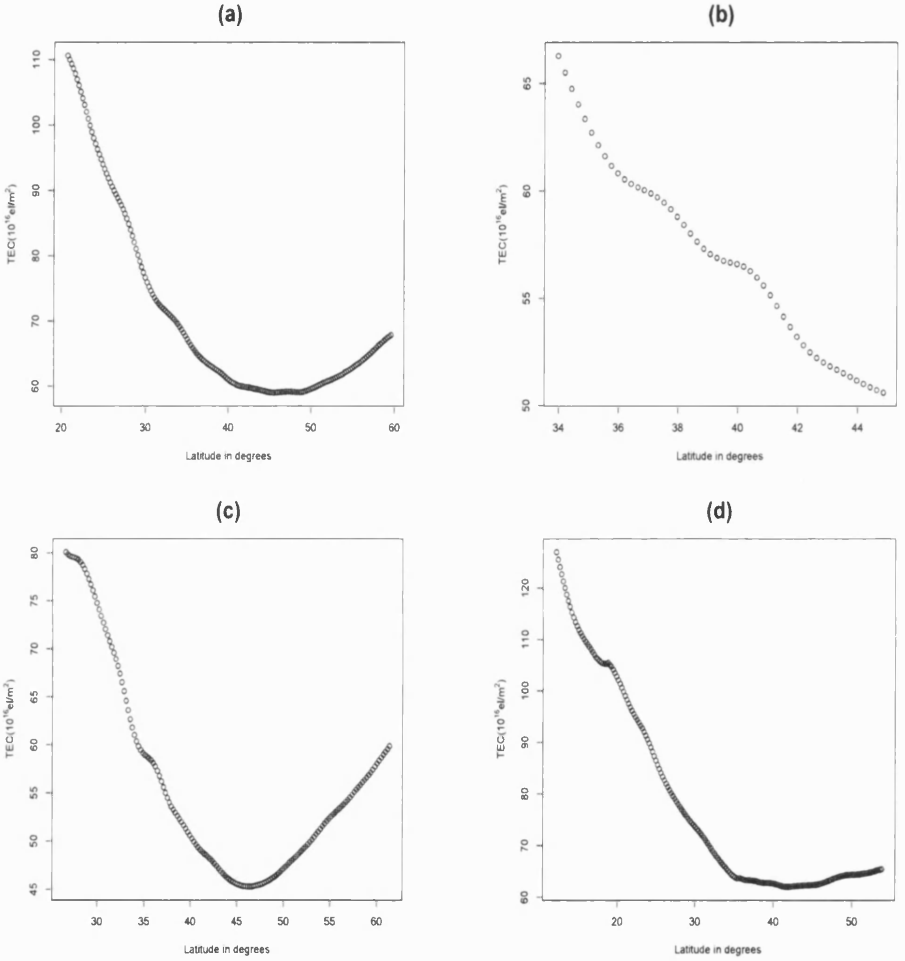


Figure 6-1: *The 572 total electron content observations collected by receiver number one (a), two (b), three (c) and four (d) recorded during a single satellite passage over the receiver chain.*

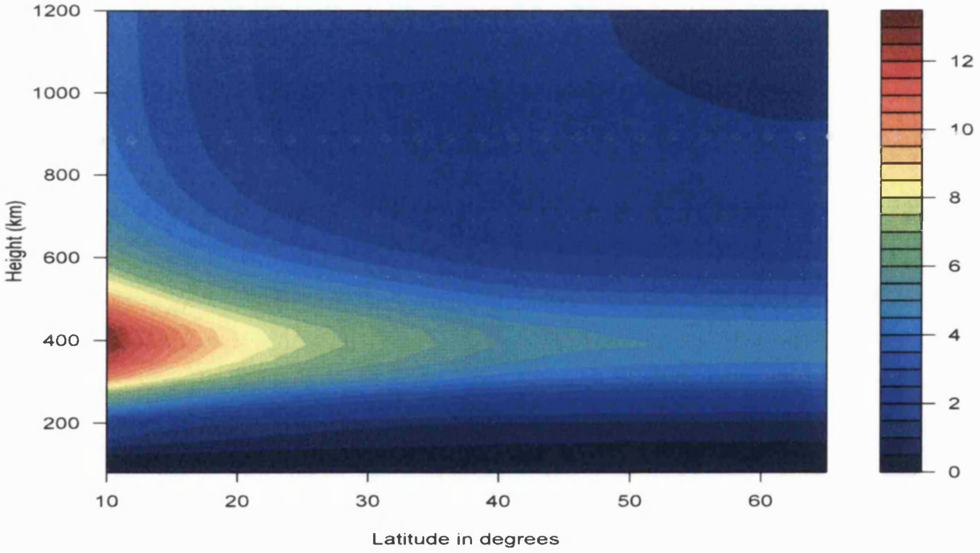


Figure 6-2: *The retrieved image of free electron densities measured in units of* $10^{11}\mathrm{el/m}^2$*obtained from MIDAS.*

Having constructed a "suitable" tomographic grid, the mixed algorithm could be implemented starting with any initial state. However, we seek a good starting point; a point that lies near the mode of the posterior distribution. Although, theoretically, the specification of such a starting point is unimportant as given any starting state the ergodic Markov chain will eventually converge, in practice, such a good initial state is crucial because it may shorten the convergence time, especially if the chain is mixing slowly.

6.2 Obtaining a good initial state

Generally, in this study any state X^0 is described by a set of N vertical electron-density profiles with an associated data noise variance. That is, $X^0 = (\{\mu_n^0, \sigma^{2^0}_n, \gamma_n^0\}, \epsilon^{2^0})$. Since our information about the ionosphere is very limited and mainly available in the form of TEC measurements, an initial guess of a good starting state requires either the contribution of the knowledge of an expert or the use of the data. For this purpose, we implemented the algorithm, MIDAS described in Chapter 3, using the same geometric grid and TEC data values, giving the tomographic image displayed in Figure 6-2. This algorithm is comparatively cheap computationally.

A difficulty that arises with estimating the required initial values directly from MIDAS results is that it operates at the pixel level producing a low level free-electron density map, whereas we seek a high-level image. To overcome this problem, we develop a method that allows us to extract a high level initial state from estimates of electron densities gathered at the low level. The derivation proceeds in four steps described in the next sub-sections.

6.2.1 Obtaining the peak height μ_n initial value

The peak height of a Chapman profile is defined as the altitude at which the maximum amount of ionisation is present within a vertical section of the ionosphere, therefore a good choice of this parameter starting value say, μ_n^0, may be the mid-height of the pixel that has the largest electron concentration within the MIDAS profile grid column. For example, assume that $l_n = \{l_{n1}, \dots, l_{n30}\}$, is the set of the nth column local electron densities, where l_{n1} for instance denotes the electron density of the pixel in the 1st row. Furthermore, suppose that

$$\arg\max(l_n) = k, \tag{6.1}$$

for some $k \in \{1, \dots, 30\}$, then we choose to set

$$\mu_n^0 = h(k), \tag{6.2}$$

where $h(k)$ represents the mid-height of the pixel at the kth row. By applying this approach using MIDAS output an initial peak height of 340,000 m above the Earth's surface is obtained for all vertical profiles for this data set.

6.2.2 Obtaining the spread σ_n^2 initial value

The spread, σ_n^2, represents the width of a normally shaped curve. Thus a good approximation of it may require the knowledge of three location values on the profile curve. These locations should be adequately dispersed but also should not lie at very low, nor very high altitudes because at these heights the local electron densities are almost zero. A sensible set of the required three points may then be that of the peak height layer, l_{nk}, i.e. the value of the electron density d_j of the kth pixel in the nth column, and any two neighboring layers such that one lies above the peak layer and the other below it, for example, $l_{n(k-1)}$ and $l_{n_(k+1)}$. Eq (4.8) allows us to express these densities in terms of the required profile spread as follows:

$$l_{nk} \;=\; \frac{\gamma_n}{\sqrt{2\pi\sigma_n^2}}\exp\left\{-\frac{1}{2\sigma_n^2}\left(h\left(k\right)-\mu_n\right)^2\right\}, \tag{6.3}$$

given $\mu_n = h(k)$ and a height increment of 40000 m, $h(k-1) = \mu_n - 40000$ and $h(k+1) = \mu_n + 40000$, hence

$$l_{nk} = \frac{\gamma_n}{\sqrt{2\pi\sigma_n^2}}, \tag{6.4}$$

and

$$l_{n(k-1)} = l_{n(k+1)} = \frac{\gamma_n}{\sqrt{2\pi\sigma_n^2}}\exp\left\{-\frac{1}{2\sigma_n^2}(40000)^2\right\}. \tag{6.5}$$

In order to obtain an initial estimate of σ_n^2, Eqs (6.4) and (6.5) need to be combined into a single equation that can be easily solved for this variable. Therefore, we will first consider the average of the neighboring layers electron densities and denote this average by, $\bar{l}_{n((k-1),(k+1))}$, so

$$\bar{l}_{n((k-1),(k+1))} = \frac{\gamma_n}{\sqrt{2\pi\sigma_n^2}}\exp\left\{-\frac{1}{2\sigma_n^2}(40000)^2\right\}. \tag{6.6}$$

Then a suitable expression of σ_n^2 in terms of all three layers could be derived from the ratio $Ra_n = \frac{l_{nk}}{\bar{l}_{n((k-1),(k+1))}}$, that is

$$Ra_n \;=\; \frac{\frac{\gamma_n}{\sqrt{2\pi\sigma_n^2}}}{\frac{\gamma_n}{\sqrt{2\pi\sigma_n^2}}\exp\left\{-\frac{1}{2\sigma_n^2}(40000)^2\right\}}. \tag{6.7}$$

Now solving the above expression for σ_n^2 gives

$$\widehat{\sigma_n}^2 = \frac{40000^2}{2\ln Ra_n}. \tag{6.8}$$

Given the values of the local electron densities obtained from MIDAS, Ra_n is found to be around 1.05, giving us an initial value, $\sigma_n^{2^0}$, of approximately $(128 \times 10^3)^2$ m^2, for $n = 1, \ldots, 22$.

6.2.3 Obtaining the γ_n initial value

The total number of free electrons within the nth grid column, Te_n, could be estimated by summing up the local electron density values of all rectangular pixels forming this column, that is

$$\hat{Te}_n = \sum_{j=1}^{30} l_{nj}. \tag{6.9}$$

An alternative more sophisticated approach to this estimation is based on multiplying the area under the Chapman profile curve by its electron density value, γ_n. Therefore, we can equate γ_n to the estimated vertical sum of local electron densities. However since increments of size 40000 m are used in building up the reconstruction grid, a discrete approximation to it is perhaps

$$\sum_{j=1}^{30} l_{nj} \approx \frac{\gamma_n}{40000}. \tag{6.10}$$

This gives us the following estimate

$$\hat{\gamma}_n = 40000 \sum_{j=1}^{30} l_{nj}, \qquad \text{for} \quad n = 1, \dots, 22. \tag{6.11}$$

γ initial values calculated using this estimator and MIDAS results lie between $(2.5357 \times 10^6)\ 10^{11}\mathrm{el/m}^2$ and $(6.8118 \times 10^6)\ 10^{11}\mathrm{el/m}^2$.

The resulting initial image obtained by implementing the above schemes is displayed in Figure 6-3.

6.2.4 Obtaining the noise parameter ϵ^2 initial value

Having derived starting values for the vertical profile parameters using MIDAS results, an initial value of ϵ^2 can be obtained by calculating the resulting residual squared sums and setting ϵ^2 as the average of these sums, that is

$$\epsilon^{2^0} = \frac{\sum_{i=1}^{572} (Y_i - E(Y_i))^2}{572}, \tag{6.12}$$

where $E(Y_i)$ is determined using the estimated local electron densities via Eq (4.4). Following this approach the data noise value associated with the initial state is $\epsilon^{2^0} = 6.4822 \times 10^5\ (10^{16}\mathrm{el/m}^2)^2$ or TECU^2.

6.3 Starting points for the smoothing parameters

In our approach to the ionospheric tomography problem, the statistical model is designed to capture the spatial features of the ionospheric image, basically at high-level

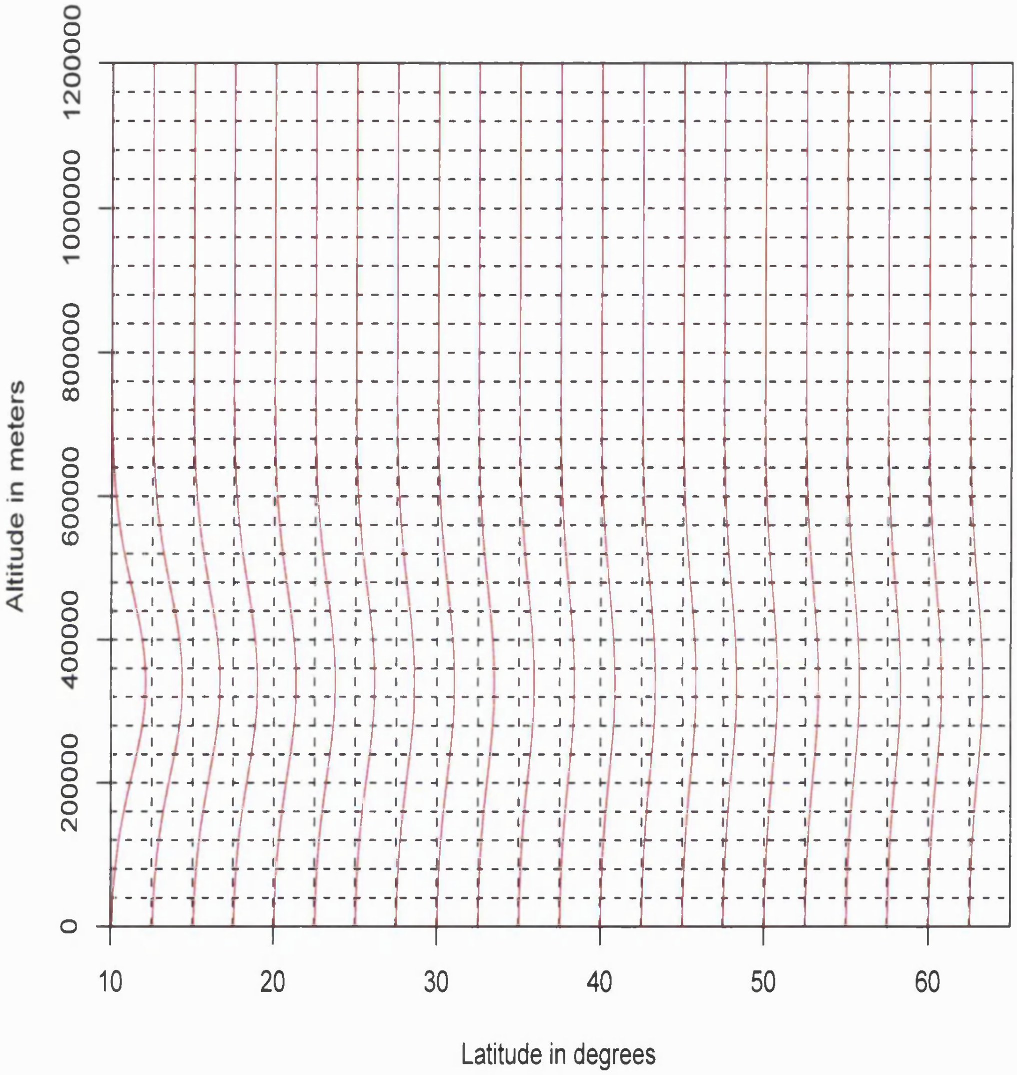


Figure 6-3: *The initial ionospheric image of vertical electron density profiles obtained from the pixel-based approach results of MIDAS.*

and consequently at a pixel-level by means of the GMRFs priors given in Eqs (4.10) - (4.12). This is because neighbouring Chapman profiles are expected to be similar with high probability. The GMRFs priors depend on the parameters $\beta_\mu, \beta_{\sigma^2}$ and β_γ because these parameters jointly control the "strength" of the interactions between the vertical profiles. Generally, this type of parameters controls the trade off between the influence of the measurements and that of the prior constraints. With the limiting case $\beta = 0$ the variables are considered to be independent, whereas with $\beta \rightarrow \infty$ they tend to cluster around a single value and therefore the resulting estimates are too faithful to the prior. Thus the choice of these parameters is critical.

It is usually impossible to estimate a smoothing parameter using a full likelihood approach. The main reason behind this is that such method requires the calculation of the normalising constant, $Z_\beta = \int_X \pi(X, \beta) dX$, of the following distribution

$$\pi(X, \beta) = \frac{1}{Z_\beta} \exp\{-\beta\phi(X)\}, \tag{6.13}$$

where X is a d-dimensional random variable and $\phi(X)$ is a function that depends on the values of X. In almost all associated problems the determination of this constant is not possible because the above integral is intractable. In an attempt to get point estimates of these types of parameters, Besag (1974, 1975) developed the *pseudo-likelihood* (PL) approximation and coding methods. For example, in the pseudo-likelihood scheme, the conditional distributions are multiplied together. Then the PL is maximised with respect to the corresponding interaction parameter. Several researchers have used these methods and found them to work reasonably well; see for example, Lee *et al.* (1995).

Following Besag's pseudo-likelihood approach, a point estimate of β_μ, for instance, is determined by maximising the corresponding pseudo-likelihood distribution written as

$$\mathrm{PL}(\mu|\beta_\mu) = \prod_{n=1}^{N} \pi(\mu_n|\mu_{-n}), \tag{6.14}$$

over β_μ.

As can be seen from Eq (6.14), the maximum pseudo-likelihood (MPL) is dependent on the conditional distribution, $\pi(\mu_n|\mu_{-n})$, which by the Markovian property of the GMRFs can be written in the form of a *local characteristic*, i.e. $\pi(\mu_n|\mu_{-n}) = \pi(\mu_n|\mu_{\delta_n})$, where μ_{δ_n} denotes the set of all neighboring peak heights of μ_n under the single dimensional neighbourhood system. Thus,

$$\mathrm{PL}(\mu|\beta_\mu) = \prod_{n=1}^{N} \pi(\mu_n|\mu_{\delta_n}), \tag{6.15}$$

where

$$\begin{aligned}
\pi(\mu_n|\mu_{\delta_n}) &\propto \pi(\mu|\beta_\mu) \\
&\propto \exp\{-\beta_\mu \sum_{m\sim n} (\mu_n - \mu_m)^2\} \\
&\propto \exp\{-\beta_\mu \sum_{m\sim n} (\mu_n^2 - 2\mu_m\mu_n)\}.
\end{aligned}$$

Let $\sharp\delta_n$ represent the number of the nth peak height neighbours and $\overline{\mu}_{\delta_n}$ denote their average, then

$$\overline{\mu}_{\delta_n} = \frac{1}{\sharp\delta_n} \sum_{m\sim n} \mu_m.$$

Hence, the local characteristic can be written in the following form

$$\pi(\mu_n|\mu_{\delta_n}) \propto \exp\{-\sharp\delta_n \beta_\mu (\mu_n - \overline{\mu}_{\delta_n})^2\},$$

which implies that

$$(\mu_n|\mu_{\delta_n}) \sim N\left(\overline{\mu}_{\delta_n}, \frac{k_n}{\beta_\mu}\right),$$

where $k_n = \frac{1}{2\sharp\delta_n}$. Thus, the required pseudo-likelihood for μ is

$$\mathrm{PL}(\mu|\beta_\mu) = \prod_{n=1}^{N} \frac{\sqrt{\beta_\mu}}{\sqrt{2\pi k_n}} \exp\left\{-\frac{\beta_\mu}{2k_n}(\mu_n - \overline{\mu}_{\delta_n})^2\right\}.$$

The MPL estimate of β_μ is obtained by maximising the *log-pseudo-likelihood* function given by

$$\begin{aligned}
\log\left(\mathrm{PL}\left(\mu|\beta_\mu\right)\right) &= \sum_{n=1}^{N} \log\left[\frac{\sqrt{\beta_\mu}}{\sqrt{2\pi k_n}} \exp\left\{-\frac{\beta_\mu}{2k_n}(\mu_n - \overline{\mu}_{\delta_n})^2\right\}\right] \\
&= \sum_{n=1}^{N} \left[\frac{1}{2}\log\beta_\mu - \log\sqrt{2\pi k_n} - \frac{\beta_\mu}{2k_n}(\mu_n - \overline{\mu}_{\delta_n})^2\right],
\end{aligned}$$

over β_μ. This produces the following estimator

$$\hat{\beta}_\mu = \frac{N}{\sum_{n=1}^{N} 2\sharp\delta_n \left(\mu_n - \overline{\mu}_{\delta_n}\right)^2}. \tag{6.16}$$

Similarly, the maximum pseudo-likelihood estimates of β_{σ^2} and β_γ are found to be

$$\hat{\beta}_{\sigma^2} = \frac{N}{\sum_{n=1}^{N} 2\sharp\delta_n \left(\sigma_n^2 - \overline{\sigma}_{\delta_n}^2\right)^2}, \tag{6.17}$$

and

$$\hat{\beta}_\gamma = \frac{N}{\sum_{n=1}^{N} 2\sharp\delta_n \left(\gamma_n - \overline{\gamma}_{\delta_n}\right)^2}, \tag{6.18}$$

respectively, where $\overline{\sigma}_{\delta_n}^2 = \frac{1}{\sharp\delta_n} \sum_{m\sim n} \sigma_m^2$ and $\overline{\gamma}_{\delta_n} = \frac{1}{\sharp\delta_n} \sum_{m\sim n} \gamma_m$.

To fix the smoothing parameters at some reasonable values using the above MPL estimators, we need to have a set of values for the associated variables; μ, σ^2, γ. To obtain this set, we have at least two options; the first is to consider the extracted estimates from the available MIDAS results obtained in the previous section, and the second is to generate new realisations approximately from $\pi(X|Y)$ via a preliminary run of the standard MCMC algorithm but under the assumption of independent profiles, i.e. we set $\beta_\mu = \beta_{\sigma^2} = \beta_\gamma = 0$. This assumption will allow the data express themselves more freely as the prior distributions will be very weak.

By considering the first option, β_γ is found to be roughly 1.64×10^{-11} in units of $(10^{11}\text{el/m}^2)^{-2}$. Unfortunately, β_μ and β_{σ^2} cannot be specified from the extracted MIDAS estimates due to the fact that under our chosen discrete configuration and method of fitting the low level image into a high level one, gave identical values of the peak heights and the spreads for this data set, which if substituted in Eqs (6.16) and (6.17) would result in undefined estimates of β_μ and β_{σ^2} e.g., $\beta_\mu = \beta_{\sigma^2} = \infty$.

Now we will apply the second approach. To shorten the burn-in period of the standard MCMC run the first option estimates of the main parameters are used as the required initial state. Here the algorithm is ran for 10,000 iterations. Some of the resulting time-series plots are displayed in Figures 6-4 and 6-5.

Inferences obtained from this run are used to reconstruct the ionospheric image depicted in Figure 6-6. This image reveals that there are different degrees of smoothness within the distinct regions of the ionospheric plane. This may be related to the amount

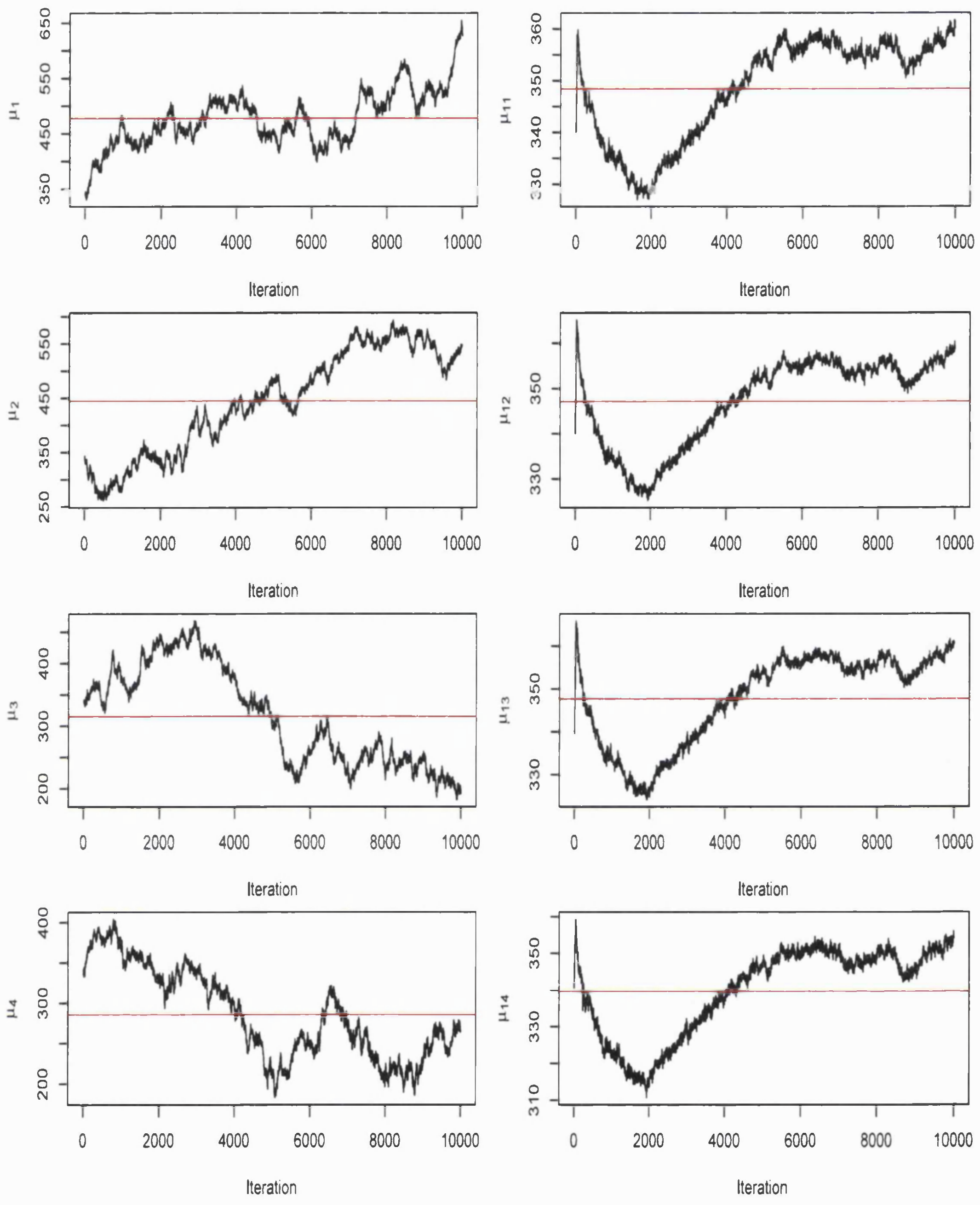


Figure 6-4: *Some vertical profiles peak height traces scaled by a factor of* 10^3 *obtained from the preliminary run of the standard MCMC algorithm with all* β*s = 0. The red lines represent the estimated averages obtained from the last 5000 sweeps.*

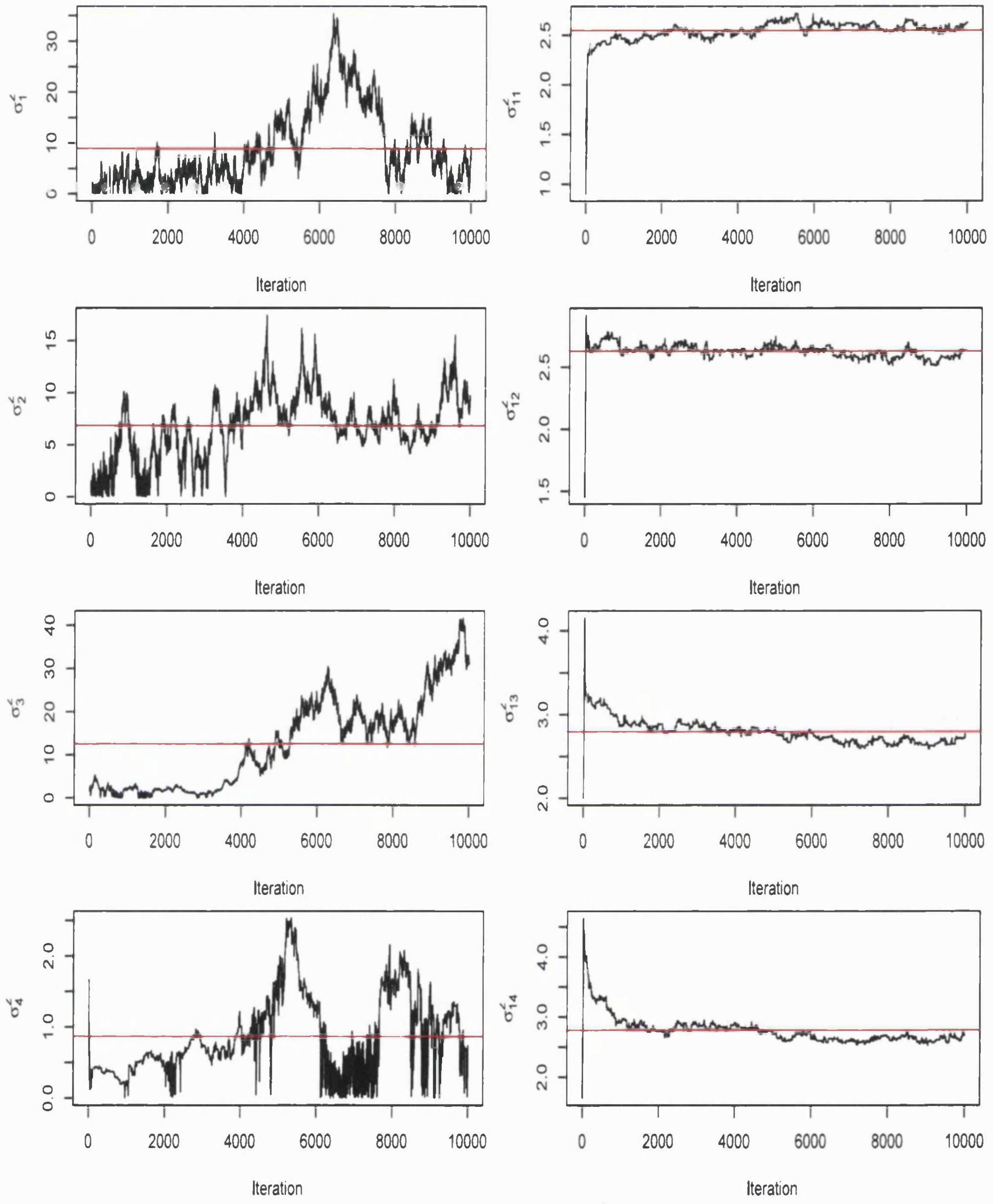


Figure 6-5: *Some vertical profiles spread traces scaled by a factor of* 10^{10} *obtained from the preliminary run of the standard MCMC algorithm with all* β*s = 0. The red lines represent the estimated averages obtained from the last 5000 sweeps.*

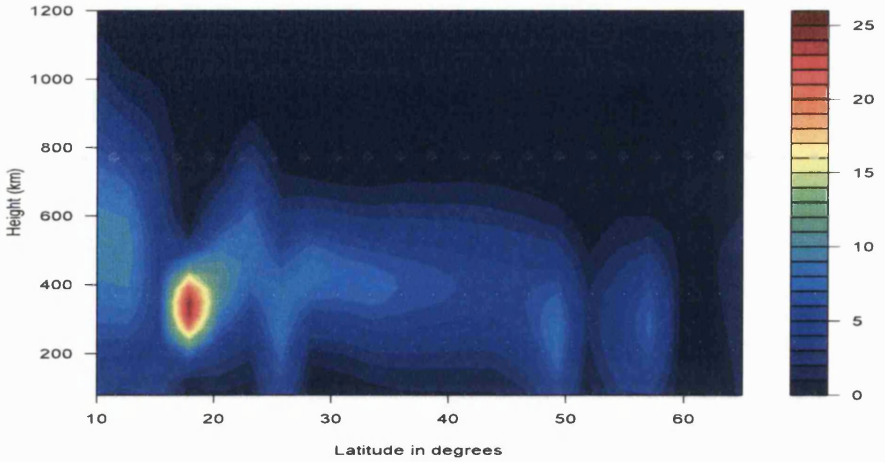


Figure 6-6: *The ionospheric retrieved image of free electron densities measured in units of* 10^{11}el/m^2*obtained from the preliminary run of the standard MCMC algorithm with all* $\beta s = 0$.

Table 6.2: *The MPL estimators obtained by using inferences for columns (from left to right)* (1 - 4), (11 - 14) and (8 - 16) *in addition to those obtained from MIDAS.*

	$10° - 20°$	$35° - 45°$	$30° - 50°$	MIDAS
Number of columns	4	4	9	
β_μ	1.38×10^{-10}	3.17×10^{-8}	2.65×10^{-9}	∞
β_{σ^2}	4.58×10^{-22}	1.79×10^{-18}	1.98×10^{-19}	∞
β_γ	2.57×10^{-14}	5.57×10^{-11}	1.67×10^{-11}	1.64×10^{-11}

of information available about each region. Therefore, we will calculate MPL estimates of $\beta_\mu, \beta_{\sigma^2}$ and β_γ for three regions within this plane and justify our choice of one of these MPL sets.

The first region extends from 10° to 20° in latitude and covers columns 1 - 4 of the grid, the second is bounded horizontally between 35° and 45° and contains columns 11 to 14, and finally the third region is wider and ranges from 30° to 50° in latitude covering all vertical sections of columns 8 to 16. Table 6.2 shows the number of columns involved in each region. Further, it compares these areas' MPL estimates obtained from the preliminary run with those derived from MIDAS.

Table 6.2 indicates that there are fairly big differences between the second and third regions' MPL estimates. However the first region MPL estimated values are much smaller than the former two regions. In terms of smoothness this means that the second region, for example is much more smooth than the first region because it has stronger spatial interactions between its vertical profiles. This can be seen in the associated chains' traces plotted in Figures 6-4 and 6-5 in the form of a parallel behaviour between the same region traces.

Since it is known that the ionosphere only allows for small variations between neighbouring regions, the roughness behaviour of the first region is odd. Further investigations of the data set led us to believe that the difference between the distinct regions of the ionospheric plane in terms of smoothness is largely due to a lack of information within some regions, especially those at the edges. For instance, the first region was intersected by only few radio signals detected by one of the four receivers as can be seen in Figures 4-1 and 6-1, whereas the second was highly covered by many signals. Therefore, the latter region's etimates of the smoothing parameters are perhaps more appropriate, especially given that the prior distributions used are defined on wide limiting ranges with no source of "artificial added" correlations because all βs are fixed at zero. This ensures that the emerged local high spatial dependency reflects a certain "real" degree of smoothness because here the results mainly depend on the likelihood of the data rather than the weak priors.

For all the above reasons, the MPL estimates of $\beta_\mu, \beta_{\sigma^2}$ and β_γ obtained from the "highly activated" region covering columns 11 to 14 are used as the required smoothing parameters' values, especially that the MIDAS MPL estimate of β_γ also supports our choice. We will refer to these values as the MPL fixed estimates henceforth.

6.4 Implementing the mixed MCMC algorithm for the real problem

Having derived a "good" initial state with reasonable interaction constants, the next stage is to apply the mixed algorithm for the TEC data set. This requires the specification of three grid layers at which reparameterisation is to be performed. To do this, inferences accumulated from a preliminary run of 100,000 iterations of the standard MCMC sampler are used. In this run, the MPL fixed estimates of $\beta_\mu, \beta_{\sigma^2}$ and β_γ are considered to identify the GMRFs priors. This produced the high-level image shown in Figure 6-7.

The last image reveals that the majority of vertical profiles are bounded at almost

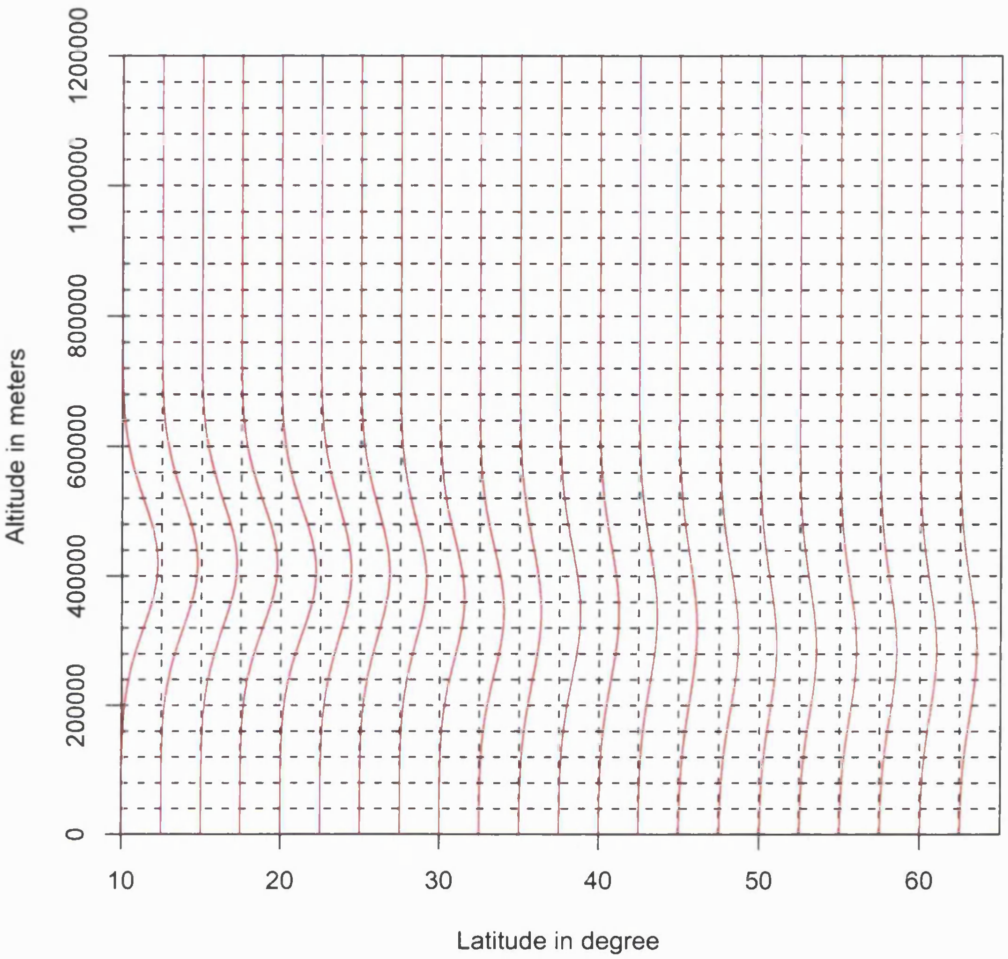


Figure 6-7: *The high-level image obtained from the standard MCMC preliminary run expectations for the real TEC measurements.*

negligible electron density values outside the 5th and the 12th rows. Thus these layers and an extra level between these two may be a good choice. Therefore we selected the 5th, 8th and 12th.

Several initial runs each of length 100,000 iterations of the mixed algorithm are carried-out by tuning the proposal distributions' standard deviations so that acceptance rates between 20% and 40% are mainly obtained for parameters within the inner region of ionospheric plane. Based on the results of these runs we assigned 2000, 5×10^9 and

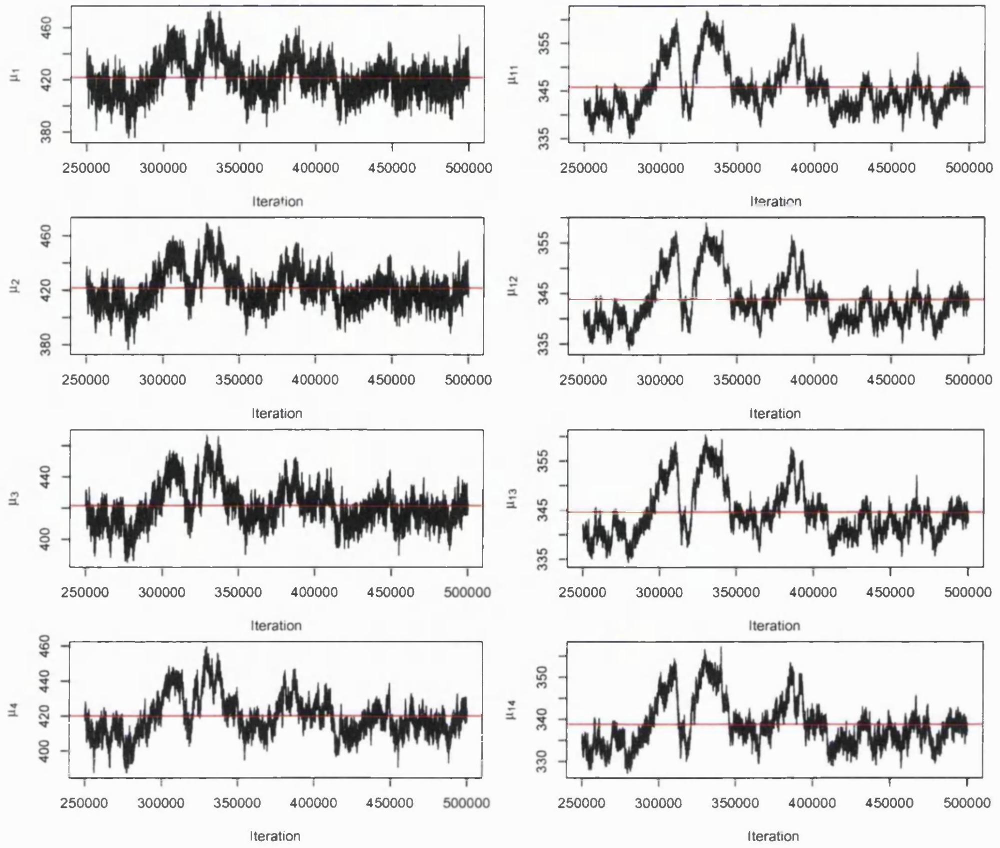


Figure 6-8: *Some selected peak heights time series plots scaled by a factor of* 10^3 *produced by implementing the mixed algorithm using the MPL fixed estimates of the smoothing parameters. The red line represents the corresponding chain empirical average.*

20000 for $\delta_\mu, \delta_{\sigma^2}$ and δ_γ respectively, for the direct updates of μ, σ^2 and γ, and 0.02 for the indirect reparameterised update step size, δ. We then ran the mixed algorithm for 500,000 sweeps.

Figures 6-8 - 6-10 reveal some parameters' time-series plots obtained from the last 50% of iterations.

For illustration purposes, four different states low-level electron density images obtained from this run are displayed in Figure 6-11. These images reveal that the overall expected spatial smoothness of the ionosphere is now captured since almost all rough areas of Figure 6-6 have disappeared. The effect of this change is also reflected in the high-level image of Figure 6-23 where smooth profiles are noticed. Moreover,

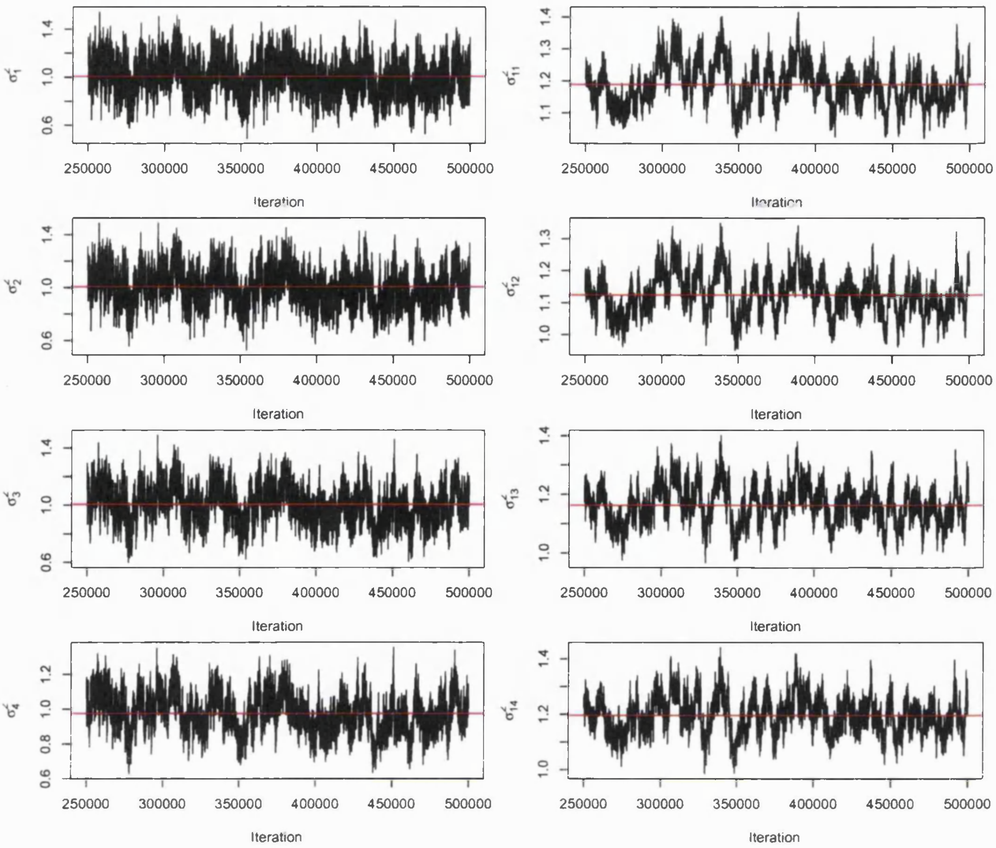


Figure 6-9: *Some selected spreads time series plots scaled by a factor of* 10^{10} *produced by implementing the mixed algorithm using the MPL fixed estimates of the smoothing parameters. The red line represents the corresponding chain empirical average.*

Figures 6-8 - 6-10 show this effect in the form of a parallel behaviour between the proposed chains. Figure 6-12 (left) displays the resulting precision trace.

The latter results show the importance of incorporating spatial priors with reasonable smoothing parameter estimates in constructing the ionospheric model. A major benefit of such priors is that vertical profiles of areas with insufficient or no data can be smoothed by "pooling strength" from neighbouring regions with similar properties.

An advantage of using fixed single point estimate of each interaction parameter is that these need to be calculated once only during the whole reconstruction process, and hence no further adaption is required. On the other hand, a drawback may be that the

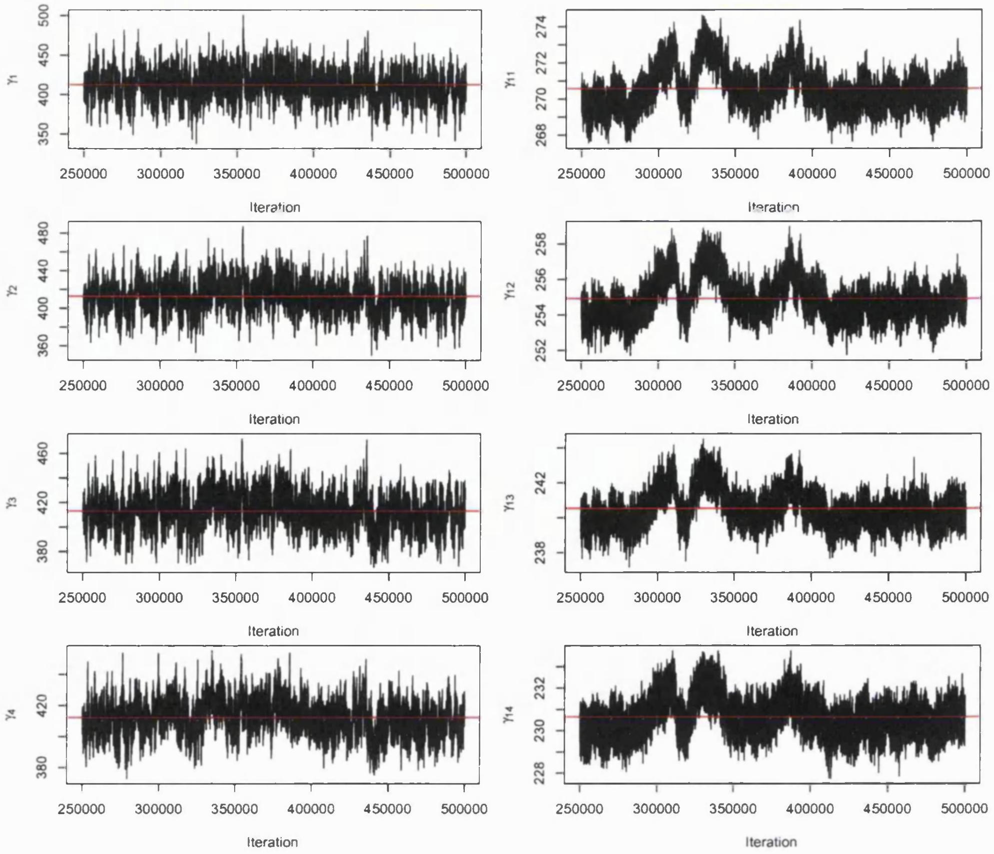


Figure 6-10: *Some selected electron densities under the profile curves time series plots scaled by a factor of* 10^4 *produced by implementing the mixed algorithm using the MPL fixed estimates of the smoothing parameters. The red line represents the corresponding chain empirical average.*

uncertainty about them is completely ignored, relying on a single initial estimate. One scheme to overcome the uncertainty problem associated with the spatial interaction parameters is to treat them as random unknown variables beside the main variables, and hence to assign a prior distribution, known as a *hyper-prior*, for each parameter. Practically, this means that the whole distribution of these parameters is to be used instead of a single set of point estimates. In this case, the spatial variables are called *hyperparameters* and the method is considered as a *fully Bayesian approach*, whereas with a unique set of point estimates the method is described as an *empirical Bayesian approach*. The latter scheme is common and was used by many researchers such as Besag *et al.* (1991) and the references therein.

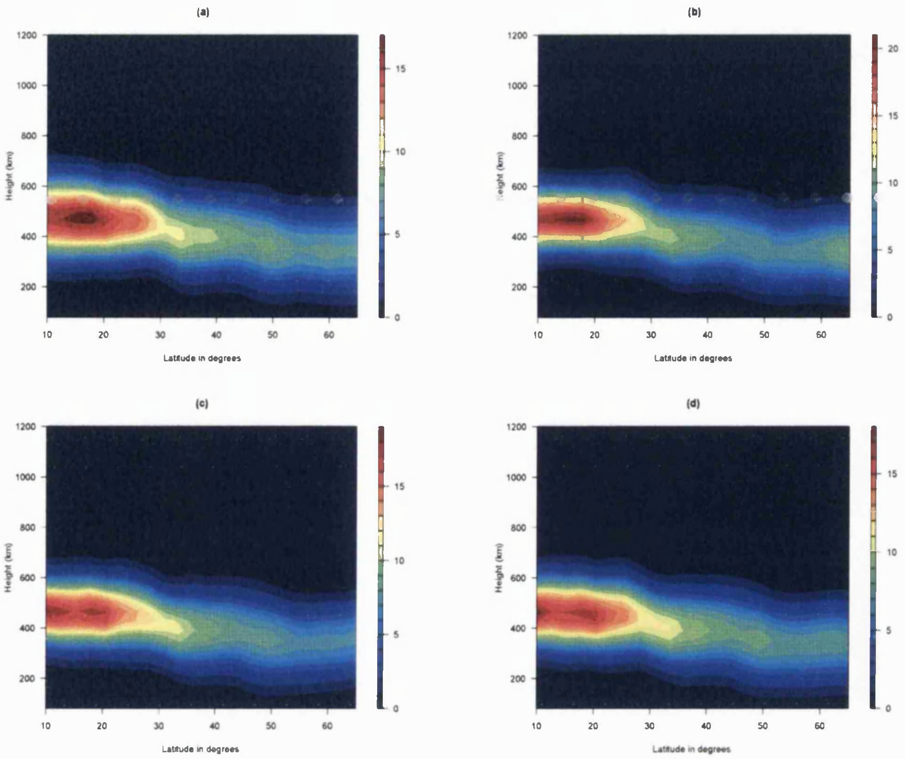


Figure 6-11: *The ionosphere maps of free electron densities in units of* 10^{11}el/m^2 *produced by the mixed algorithm at iteration (a) 300,000, (b) 350,000 , (c) 400,000, (d) 450,000 when the MPL fixed estimates of the smoothing are used.*

In this study, the fully Bayesian approach is not implemented. This is because considering $\beta_\mu, \beta_{\sigma^2}$ and β_γ as random variables requires the construction of a *hierarchical model* in which each interaction parameter is to be modelled by a hyper-prior distribution. Moreover, under the considered proper priors given in Eqs (4.10) - (4.12), the corresponding full conditional distributions of $\beta_\mu, \beta_{\sigma^2}$ and β_γ require the determination of the normalising factors, $N(\beta_\mu), N(\beta_{\sigma^2})$ and $N(\beta_\gamma)$, since the full conditional distribution of β_μ for example, is given by

$$\pi(\beta_\mu|\mu) = \frac{\pi(\mu|\beta_\mu)\pi(\beta_\mu)}{N(\beta_\mu)} \qquad \text{for} \quad \beta_\mu \geq 0, \tag{6.19}$$

where

$$N(\beta_\mu) = \int_\mu \pi(\mu|\beta_\mu)\pi(\beta_\mu)d\mu. \qquad (6.20)$$

The latter integration, if needed, requires long and complicated calculations each time β_μ changes.

Heikkinen and Högmander (1994) introduced a "combined" approach in which the restoration empirical procedure is used without ignoring the uncertainty of the attraction parameter, β. The proposal is to alternate between updating the unknown variable X using the current β values and updating the βs using the current realisation X via the MPL estimator. Thus, this scheme produces a sequence of estimates of the βs by considering them as constant parameters rather than unknown random variables. This consideration shows a special characteristic of the method. It is not crucial that each estimate in the created sequence of β is drawn from a distribution, but that they vary continuously during the restoration process; the uncertainty of β is admitted through the variation within the produced sequence.

In order to acknowledge the uncertainty of our smoothing parameters we adapted the above two authors' approach, i.e. we treat $\beta_\mu, \beta_{\sigma^2}$ and β_γ as constants to be regularly altered rather than unknown random variables. This choice is also based on the following feature of our application: the TEC data as believed by many electronic engineers are of high quality. Thus, although, estimates of the smoothing parameters cannot be derived directly from the raw data, reasonable values of them may be obtained by regularly implementing the MPL approach to the MCMC estimates, especially if these estimates correspond to the ionospheric region covered by many data signals.

We implemented two different runs of the mixed algorithm based on Heikkinen *et al.* (1994)'s approach. Each run is of 500,000 iterations and has the same configuration as the previous study. In both versions, $\beta_\mu, \beta_{\sigma^2}$ and β_γ are updated every 1000 sweeps rather than every sweep because this reduces the computing time, and gaps of this size may be reasonable with the long run considered and the high correlations noticed. Further, we chose to initially fix $\beta_\mu, \beta_{\sigma^2}$ and β_γ at zero for the same reasons indicated in the previous section.

In the first version of the mixed sampler the smoothing parameters are updated using the highly active region estimates only, whereas in the second version the whole plane results are used. The middle and the right plots of Figure 6-12 reveal the resulting

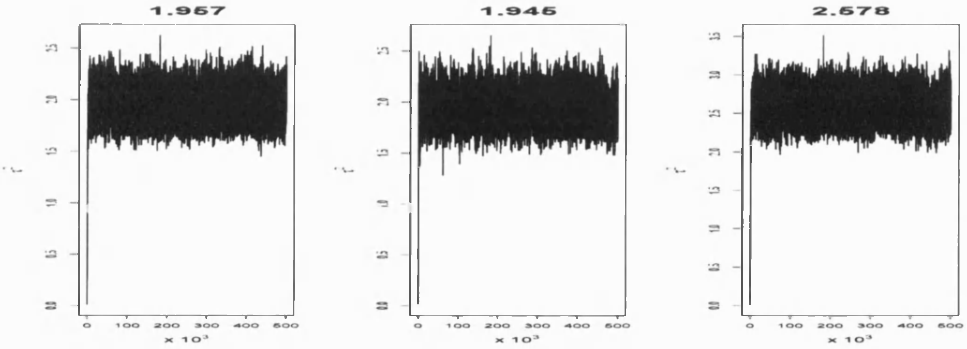


Figure 6-12: *The precision parameter in unit of $TECU^{-2}$ traces obtained from the mixed algorithm results when the smoothing parameters are: set at the MPL fixed values (left), updated using the second region results (middle), updated using the whole plane results (right). The numbers above each plot represent the relevant empirical mean.*

Table 6.3: *The maximum pseudo-likelihood estimated averages of the interaction parameters obtained from the last two versions of the mixed algorithm.*

Parameter	Fixed values	Version 1	Version 2
β_μ	3.17×10^{-8}	3.28×10^{-8}	5.04×10^{-9}
β_{σ^2}	1.79×10^{-18}	2.02×10^{-18}	2.03×10^{-19}
β_γ	5.57×10^{-11}	5.68×10^{-11}	8.87×10^{-12}

traces of the precision parameter τ^2, whereas Figures 6-13 - 6-18 depict both versions analogous traces of Figures 6-8 - 6-10. In all Figures from 6-13 to 6-18, the left panels represent the results obtained from the first version of the mixed algorithm, whereas the right panels reveal the second version's results. The MPL estimates of the attraction parameters obtained from the two runs are shown in Figure 6-19 and the corresponding uncertainty intervals are revealed in Figure 6-20. Table 6.3 compares the resulting averages of the MPL estimates with the fixed ones derived by our approach.

6.4.1 Results

The precision time-series plots of Figure 6-12 indicate that by implementing the mixed algorithm, whether the smoothing parameters are unaltered from the MPL fixed estimates or updated regularly, smaller residual square errors than those we started with are obtained rapidly since bigger values of τ^2 are generated after short burn-in periods. This has largely decreased ϵ^2 associated with the initial state by approximately 10^5 TECU^2 by all runs of the mixed algorithm.

The left panel of Figure 6-19 shows that implementing the first version of the mixed algorithm has produced MPL estimates of the interaction parameters that vary within small ranges during the whole process after the burn-in period reflecting a small amount of uncertainty, as can be also noticed in Figures 6-21 and Figure 6-24. However, the variation between the resulting MPL estimates of β_{σ^2} is slightly larger than the other two interaction parameters, as the associated histogram of Figure 6-20 reveals. Nevertheless, the resulting MPL averages of this version are, in general, close to the selected MPL fixed values as displayed in Table 6.3. Therefore this version results are fairly similar to those obtained from the fixed estimates run, with only small variations seen in the tomographic images of Figures 6-11 and 6-21, and the high-level images of Figures 6-23 and 6-24.

In contrast to the first version MPL results, the second version's histograms displayed in the right panel of Figure 6-20, and the relevant traces of Figure 6-19, show that the uncertainty in this case is generally larger because wider ranges for β_μ and β_γ are obtained and big excursions within the created chains are noticed as in β_{σ^2} trace plot. This indicates that more variation is present in this case. The uncertainty of this process has a noticeable effect on the results; at the low imaging level this effect can be seen in the form of the widely varying electron density images of Figure 6-22, for instance, whereas at the high imaging level it can be seen in the varying vertical profiles of Figure 6-25. At both levels of imaging the low and the high, the uncertainty associated with the restoration is mostly revealed within the regions that are either not intersected by any radio signal rays or only crossed by few of them. This is because these regions are now more influenced by the weak priors since smaller interaction constants are generated. Consequently, wider estimation ranges are obtained and the resulting output vary more often.

6.4.2 Conclusion

In this chapter, we have considered the real TEC data problem mainly using the mixed MCMC algorithm developed in this thesis. It have been noticed that the algorithm is very sensitive to the values of the smoothing parameters, β_μ, β_{σ^2} and β_γ. Therefore, we investigated two different approaches to deal with these parameters.

In the first approach, they are held fixed at some reasonable values. These values are obtained by implementing the MPL technique for a preliminary MCMC run using the central region of the ionospheric plane where many radio signals passed.

In the second approach, the smoothing parameters are allowed to vary during the MCMC sampling process by regularly calculating their MPL estimates. Here two versions of the mixed sampler are ran in parallel. Within the first version, all βs'

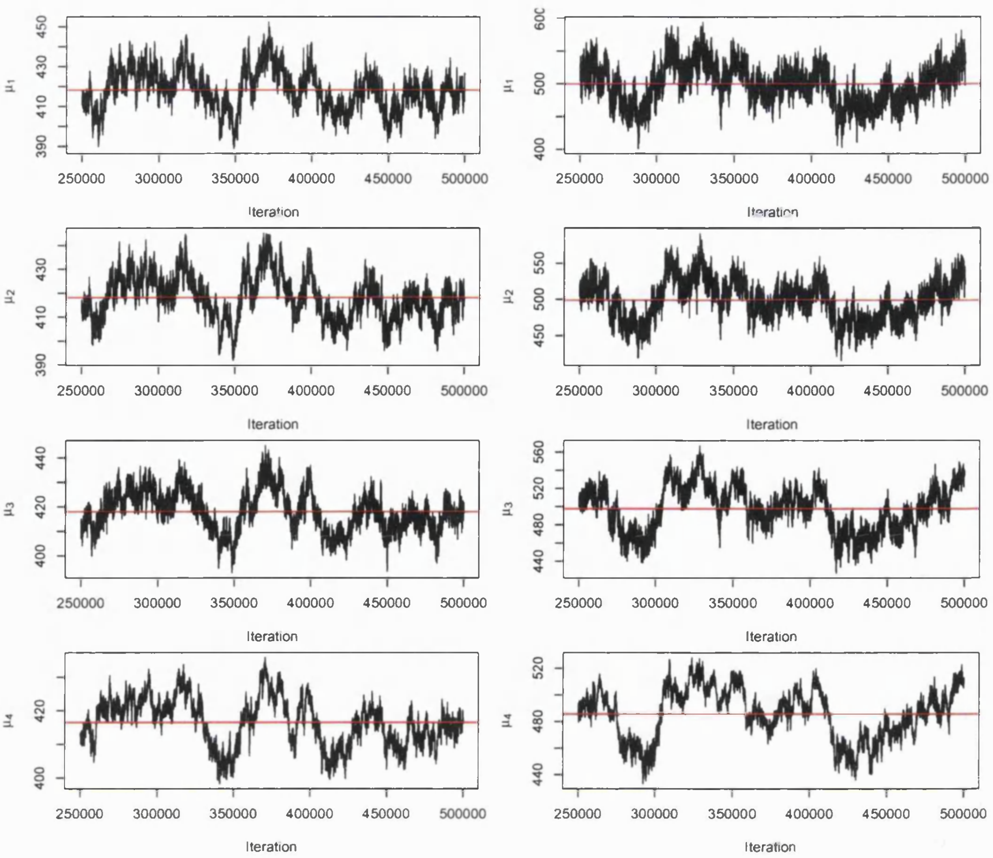


Figure 6-13: *The first region peak height traces obtained from (left) the first version and (right) the second version of the mixed algorithm when the smoothing parameters are updated regularly. The empirical average of each chain is represented by a red line. All values are scaled by a factor of* 10^3.

estimated values are extracted from the central region estimates, whereas for the second version, the whole plane is used in the MPL estimation process.

As expected, the first version results reveal more stable restorations for both the low and the high level images than the second version, as can be noticed by comparing Figures 6-21 and 6-24 with the analogous plots of Figures 6-22 and 6-25, respectively.

Although most of the images produced by the mixed sampler captured the expected degree of ionospheric smoothness, almost all time-series plots indicate slow mixing, as clearly noticed in Figures 6-8 and 6-13 for instance. To quantify this behaviour we present in Table 6.4 estimates of the integrated autocorrelation time for some chains obtained from the three distinct runs of the mixed algorithm. The aim is to obtain small

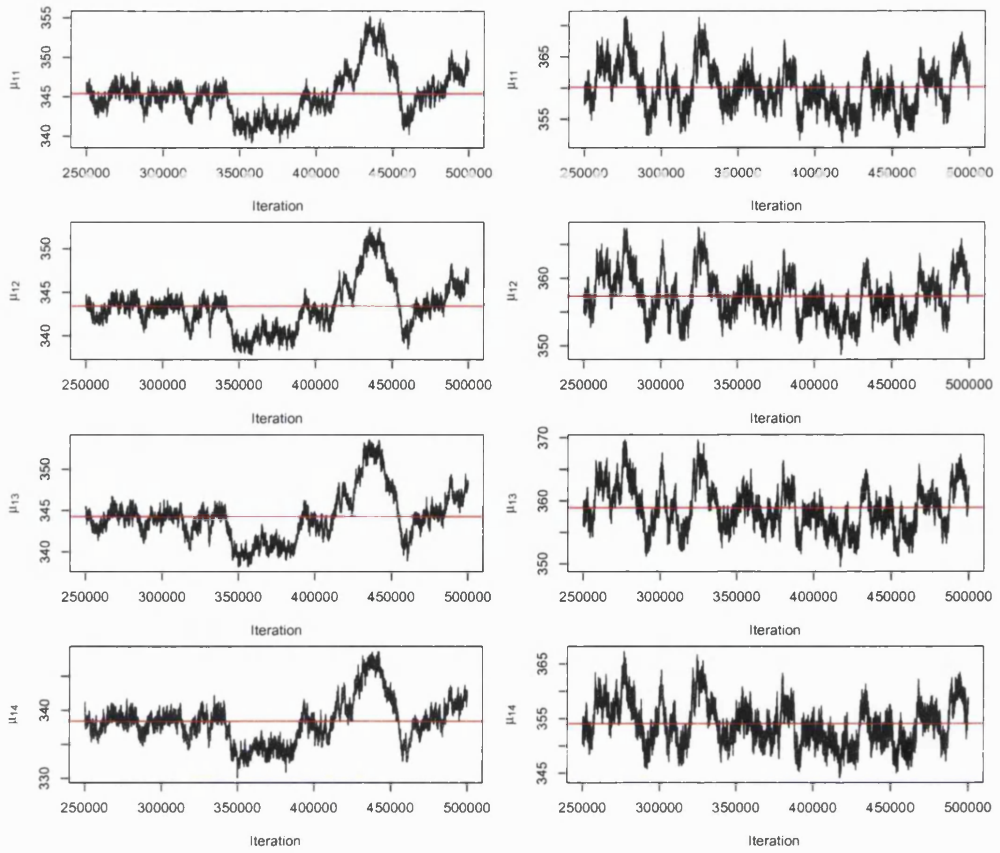


Figure 6-14: *The second region peak height traces obtained from (left) the first version and (right) the second version of the mixed algorithm when the smoothing parameters are updated regularly. The empirical average of each chain is represented by a red line. All values are scaled by a factor of* 10^3.

values of the integrated autocorrelation time, say 10 or less. The required estimates are calculated using Geyer's (1992) *initial positive sequence estimator.* This estimator will be used for all further similar calculations in this thesis.

It is clear from Table 6.4 that all three runs of the mixed algorithm perform poorly. However, as the uncertainty about the interaction parameters increases, the efficiency of the sampler decreases as measured by the integrated autocorrelation time estimates. In general, these estimates indicate that the produced chains are not mixing well.

This is probably due to two main reasons: the first is that the step sizes chosen to update the parameters, despite the fact that they produce reasonable acceptance

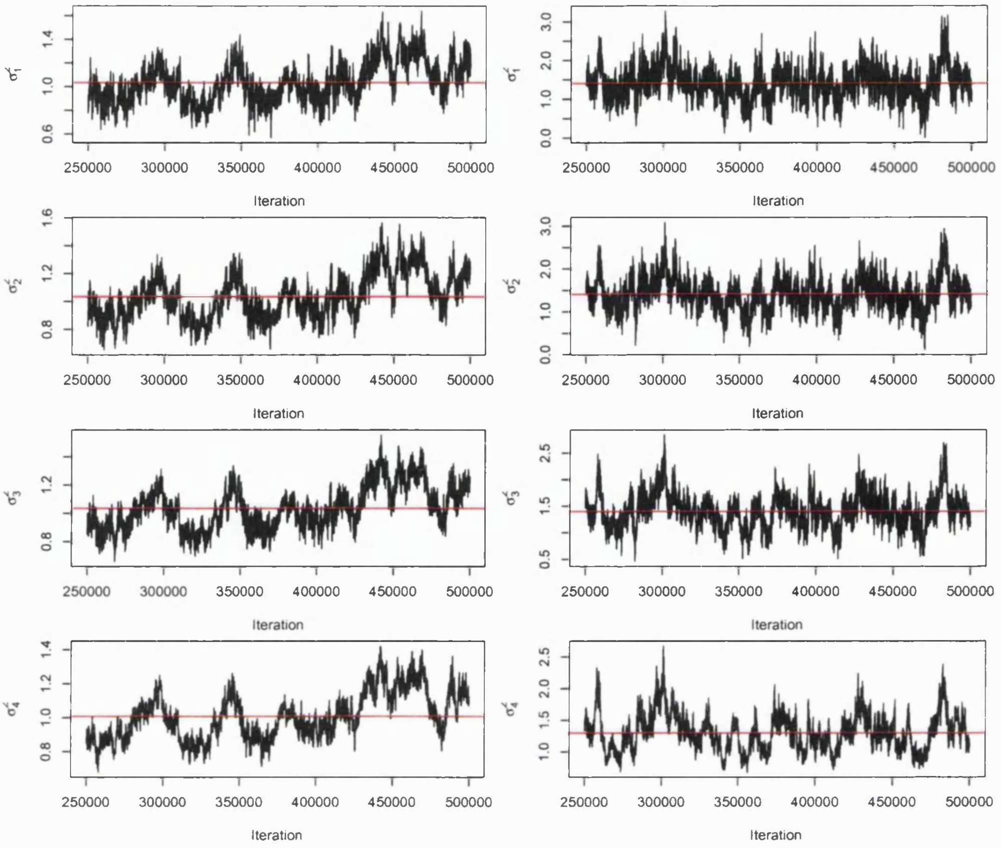


Figure 6-15: *The first region spread traces obtained from (left) the first version and (right) the second version of the mixed algorithm when the smoothing parameters are updated regularly. The empirical average of the chain is represented by a red line. All values are scaled by a factor of* 10^{10}.

rates for the central columns, produced very big acceptance rates for regions with few or no radio signals. This can be easily solved by using different step sizes for different columns. The second, and very common reason in many MCMC based problems, is the high correlations present between the parameters of interest.

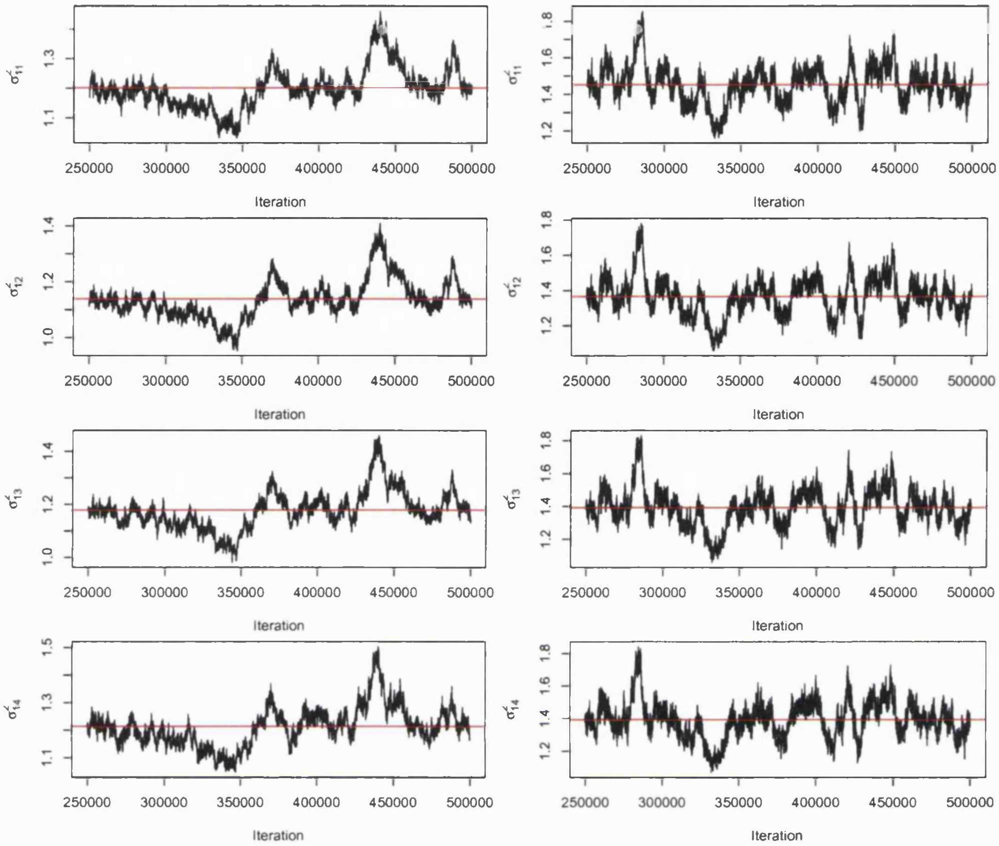


Figure 6-16: *The second region spread traces obtained from (left) the first version and (right) the second version of the mixed algorithm when the smoothing parameters are updated regularly. The empirical average of the chain is represented by a red line. All values are scaled by a factor of* 10^{10}.

A new task toward a better restoration is to speed up the mixing of all chains to be generated. In the next chapter we will discuss the construction of a new MCMC algorithm that may help us to achieve this aim.

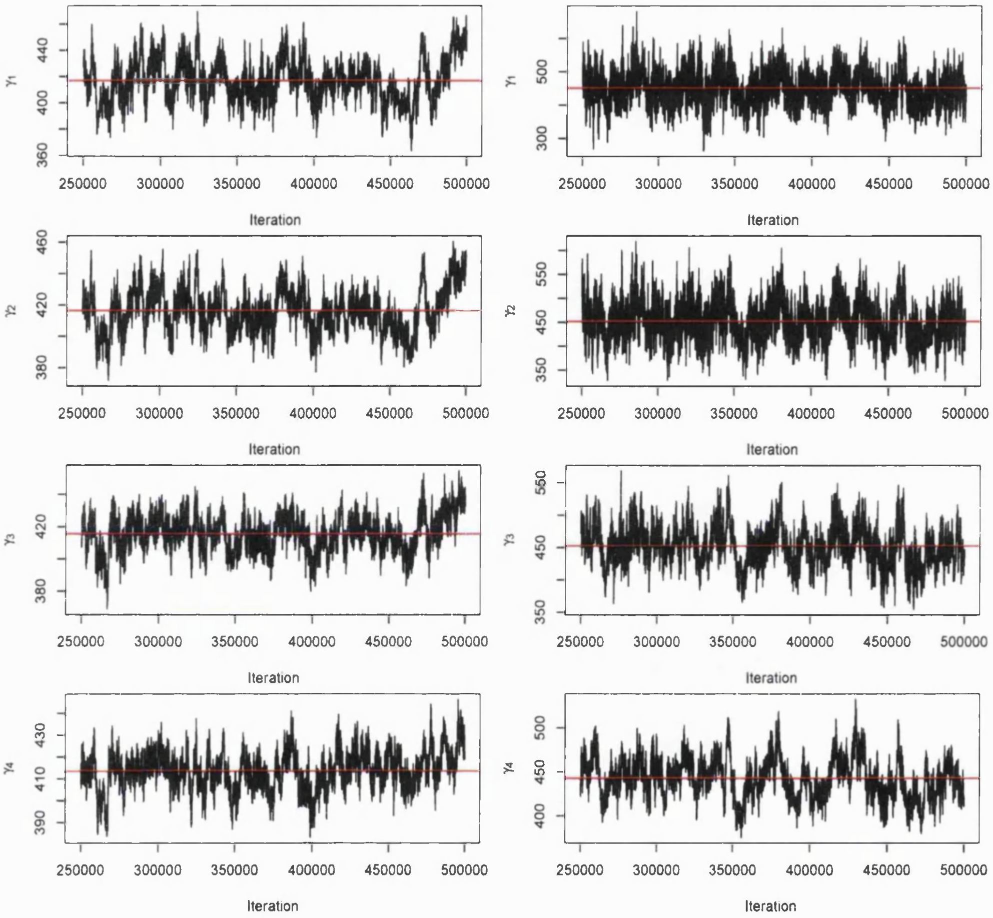


Figure 6-17: *The first region electron density under the profile curve traces obtained from (left) the first version and (right) the second version of the mixed algorithm when the smoothing parameters are updated regularly. The empirical average of the chain is represented by a red line. All values are scaled by a factor of* 10^4.

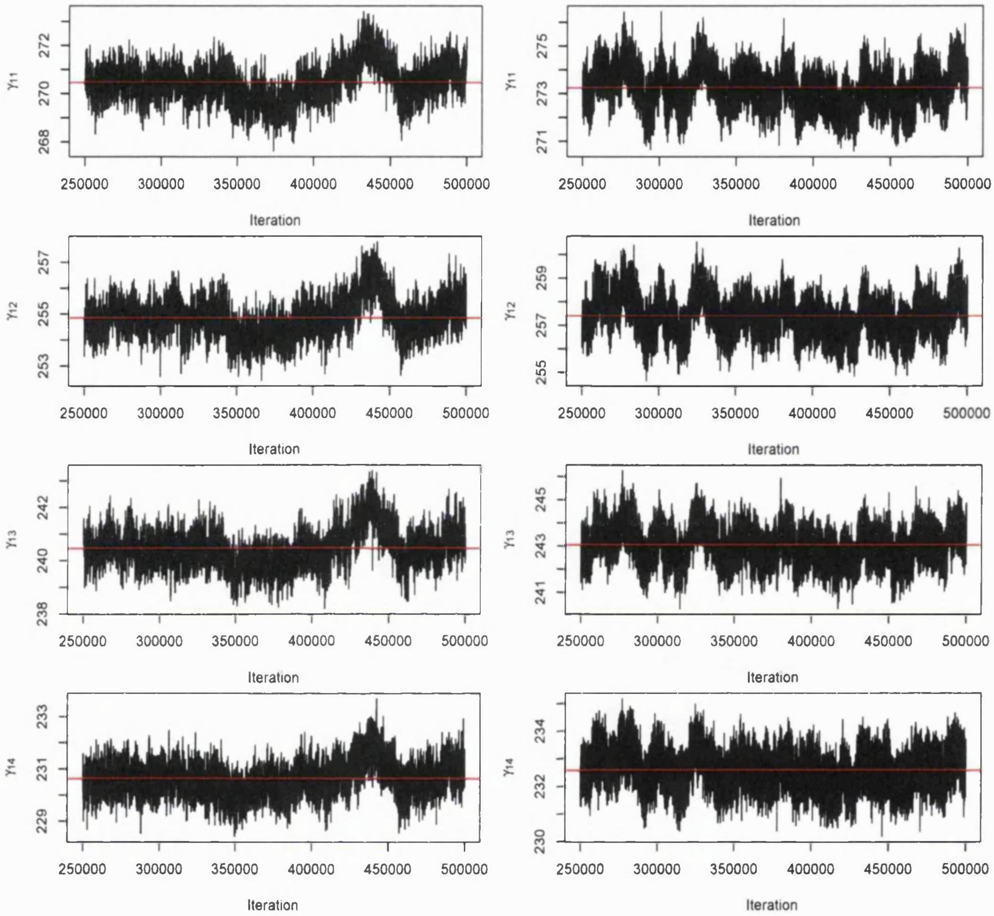


Figure 6-18: *The second region electron density under the profile curve traces obtained from (left) the first version and (right) the second version of the mixed algorithm when the smoothing parameters are updated regularly. The empirical average of the chain is represented by a red line. All values are scaled by a factor of* 10^4.

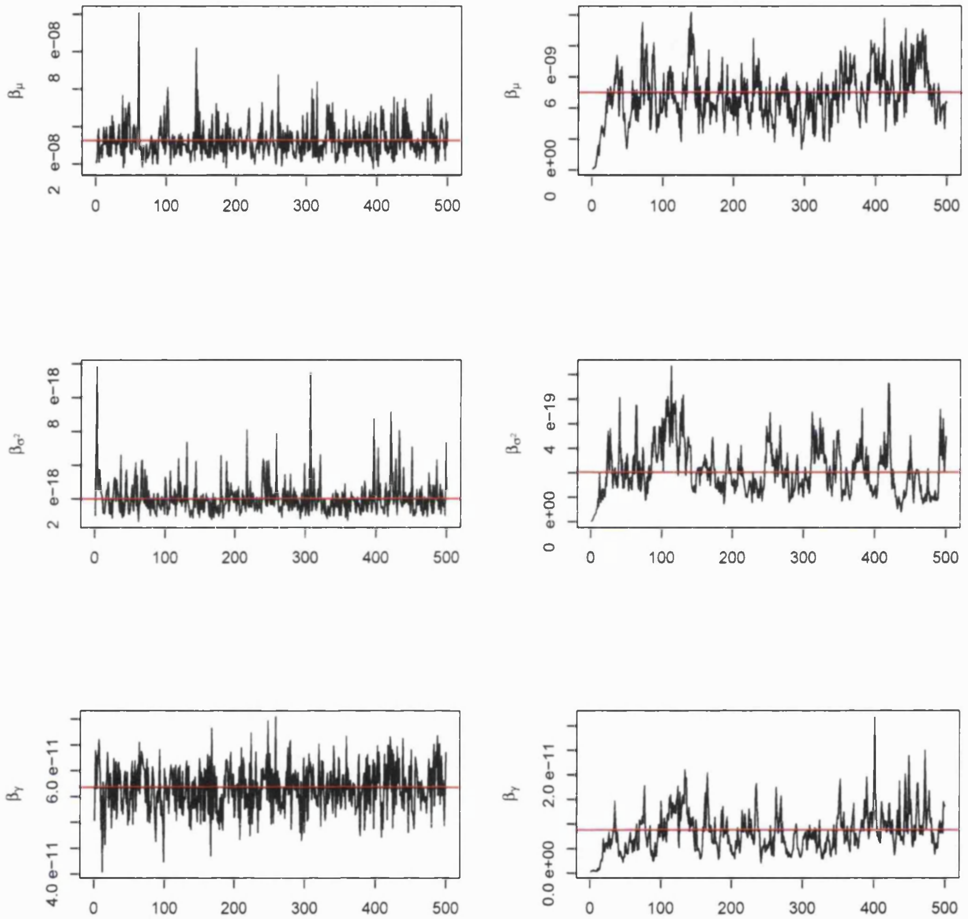


Figure 6-19: *The interaction parameter traces derived from the (left) first version and (right) second version of the mixed algorithm. Red lines represent the means of the resulting sequences.*

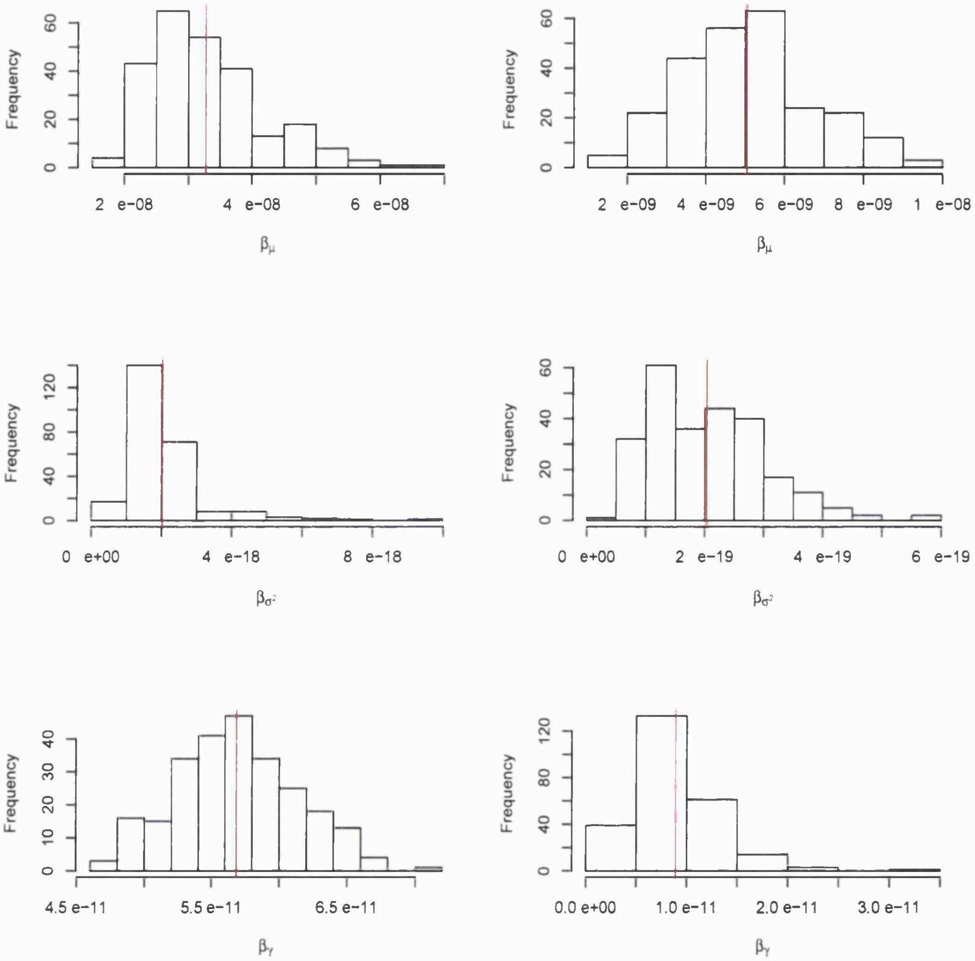


Figure 6-20: *Histograms of the smoothing parameters obtained from the last 250 MPL estimates of the corresponding chains of the above figure. The red vertical lines represent the resulting averages.*

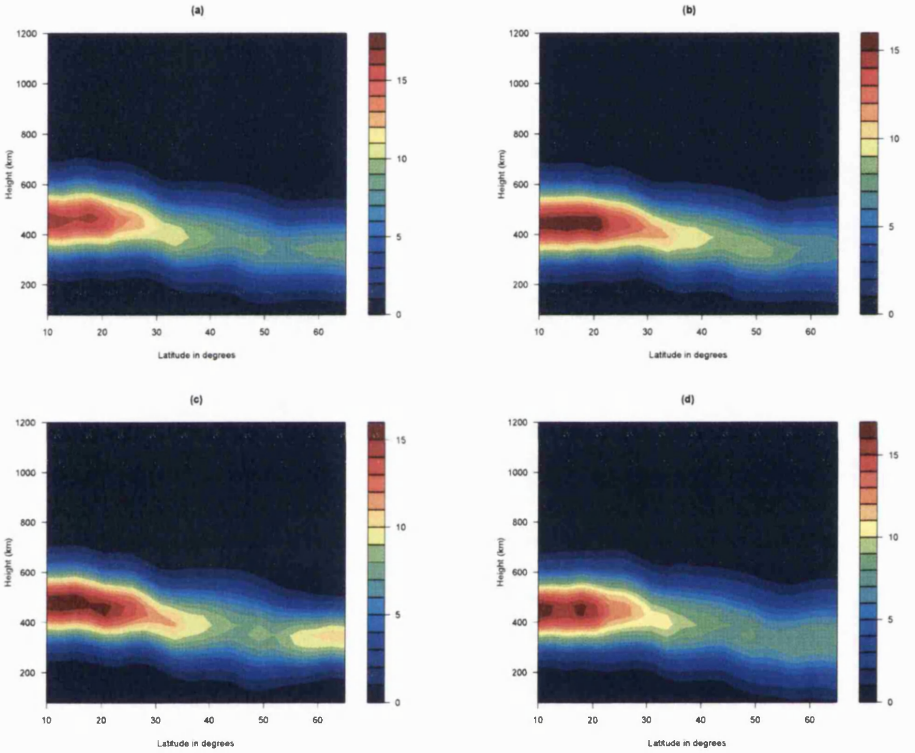


Figure 6-21: *The ionosphere maps of free electron densities in units of* $10^{11}\mathrm{el/m}^2$ *produced by the first version of the mixed algorithm at iteration (a) 300,000, (b) 350,000 , (c) 400,000, (c) 450,000.*

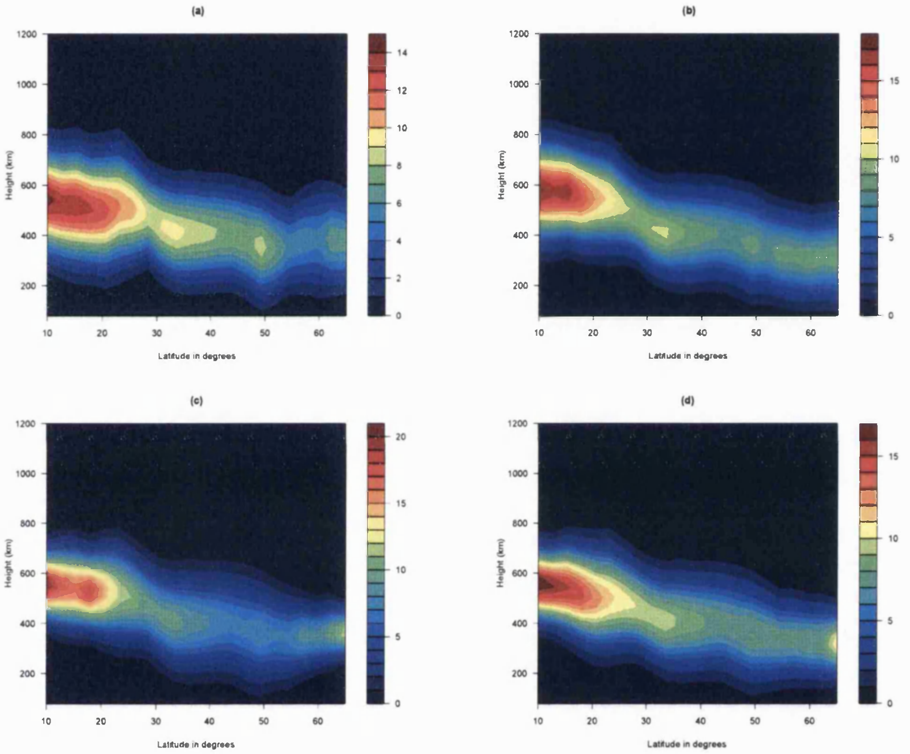


Figure 6-22: *The ionosphere maps of free electron densities in units of* $10^{11}\text{el}/\text{m}^2$ *produced by the second version of the mixed algorithm at iteration (a) 300,000, (b) 350,000 , (c) 400,000, (c) 450,000.*

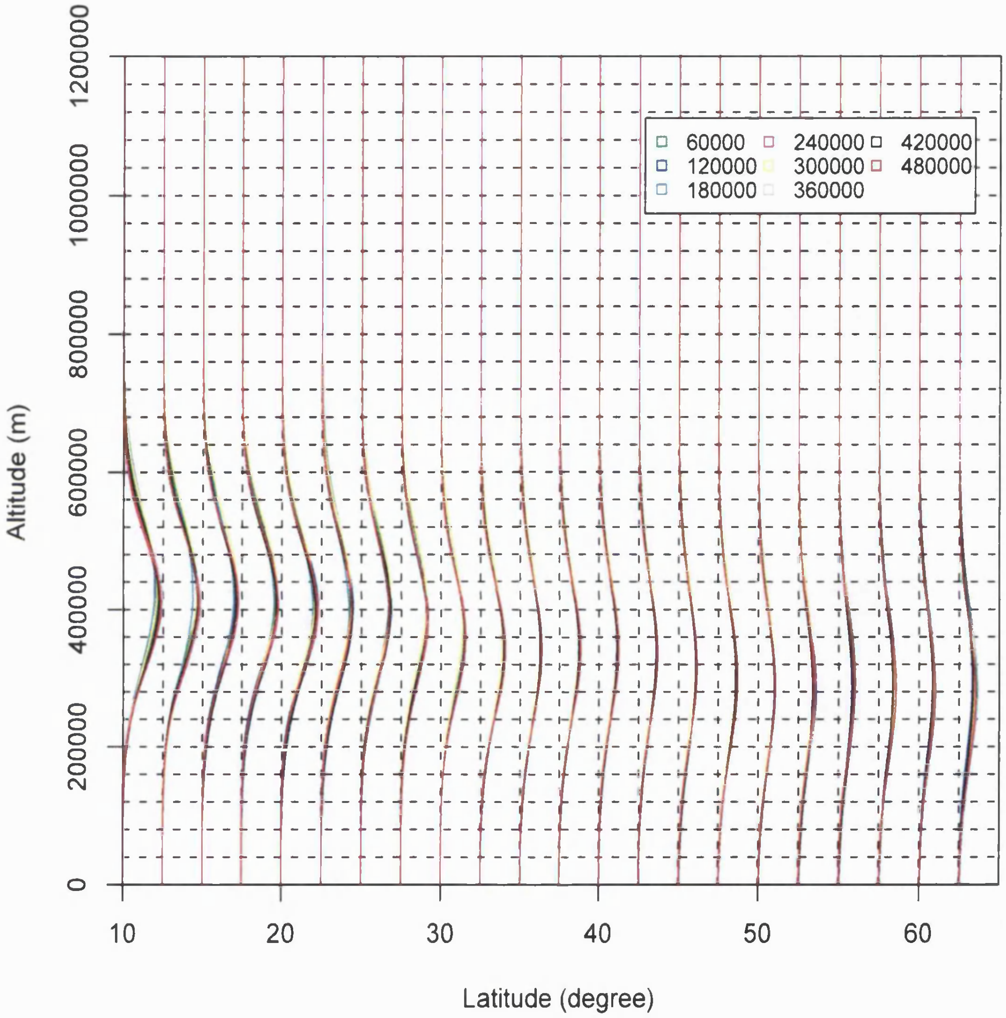


Figure 6-23: *The high-level image obtained from the mixed algorithm with the MPL fixed estimates of the smoothing parameters at the above mentioned iterations.*

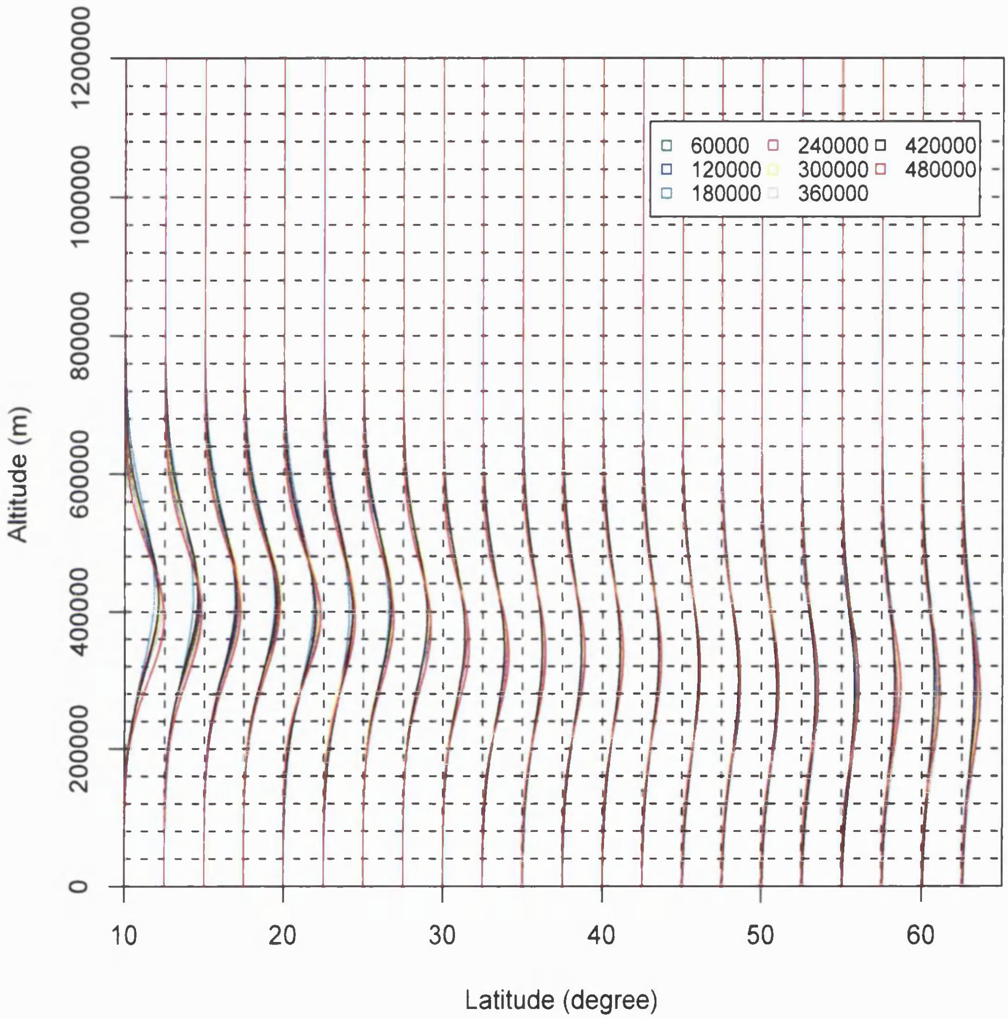


Figure 6-24: *The high-level image obtained from the first version of the mixed algorithm at the above mentioned iterations.*

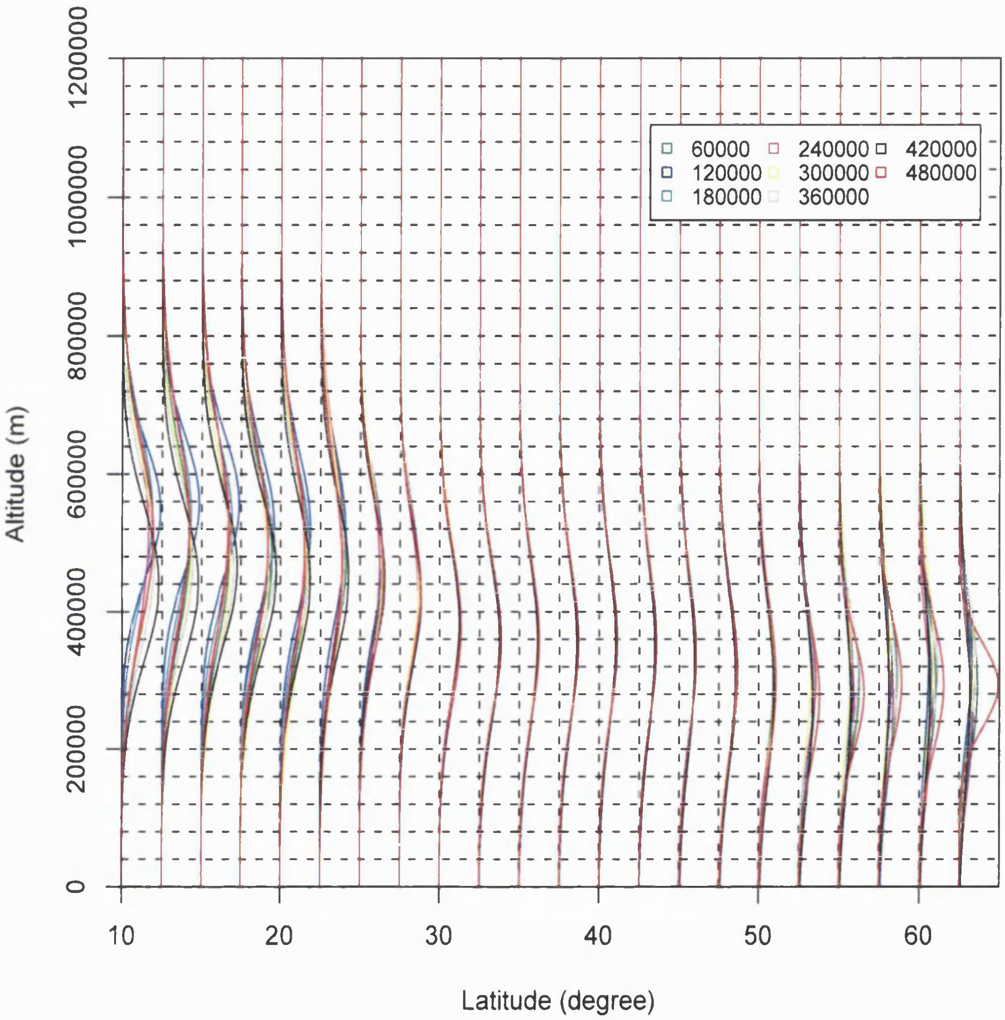


Figure 6-25: *The high-level image obtained from the second version of the mixed algorithm at the above mentioned iterations.*

Table 6.4: *Some variables' (Var) integrated autocorrelation times ($\hat{\tau}(.)$) obtained from the mixed sampler when the smoothing parameters are fixed (F) (first approach) or updated using version one (V.1) or version two (V.2) of the algorithm together with the corresponding chains empirical acceptance rates A. The integrated autocorrelation time estimates are calculated from chains' sections of length 100,000.*

Var.	$\hat{\tau}(.)$(F)	$\hat{\tau}(.)$(V.1)	$\hat{\tau}(.)$(V.2)	A % (F)	A % (V.1)	A % (V.2)
μ_1	5694	6240	6524	82	83	74
μ_2	6192	6699	6822	76	76	69
μ_3	7127	7277	7665	76	76	47
μ_4	8277	8075	10387	46	49	24
σ_1^2	1728	1793	5239	76	76	72
σ_2^2	1936	2081	6826	68	68	66
σ_3^2	2179	3369	8413	67	68	41
σ_4^2	2408	3777	11115	42	45	27
γ_1	963	950	2083	87	87	85
γ_2	1129	1138	2397	82	83	82
γ_3	1369	1441	3844	82	82	70
γ_4	1973	2566	5291	65	68	41
μ_{11}	10958	13308	15991	34	34	28
μ_{12}	10981	13419	15766	33	33	27
μ_{13}	10915	13393	15612	31	31	26
μ_{14}	10752	13106	15401	37	38	32
σ_{11}^2	4711	6914	9924	33	33	27
σ_{12}^2	3831	5783	10278	33	32	27
σ_{13}^2	2085	4608	10425	30	31	26
σ_{14}^2	1786	3953	10127	35	35	32
γ_{11}	7923	9933	11194	35	35	28
γ_{12}	7743	9709	11765	34	35	28
γ_{13}	7139	9734	9845	32	32	26
γ_{14}	5567	7922	8233	36	36	30

Chapter 7

The Principal Components MCMC algorithm: An efficient sampler

7.1 Introduction

Markov chain Monte Carlo is a method of performing implicit integration for analytically intractable distributions. A properly constructed MCMC algorithm will generate dependent samples from a posterior distribution of interest provided that it is allowed to run sufficiently long. Ergodic averages of the simulated values can be used to estimate the posterior expectations of interest.

Practical implementations of MCMC algorithms such as the Metropolis-Hasting sampler can suffer from poor mixing and consequently inefficient estimation, even when sampling from unimodal distributions.

To demonstrate the difficulty in using standard MCMC algorithms to sample from such distributions, consider a bivariate normal density centred at $\mu = (5,2)^T$ with variance matrix

$$\Sigma = \begin{pmatrix} 2.0^2 & 0.396 \\ 0.396 & 0.2^2 \end{pmatrix},$$

so that the correlation $\rho = 0.99$. This distribution can be written as

$$f(X) = \frac{1}{2\pi|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\right), \tag{7.1}$$

where $X = (X_1, X_2)^T$ and T denotes the transpose.

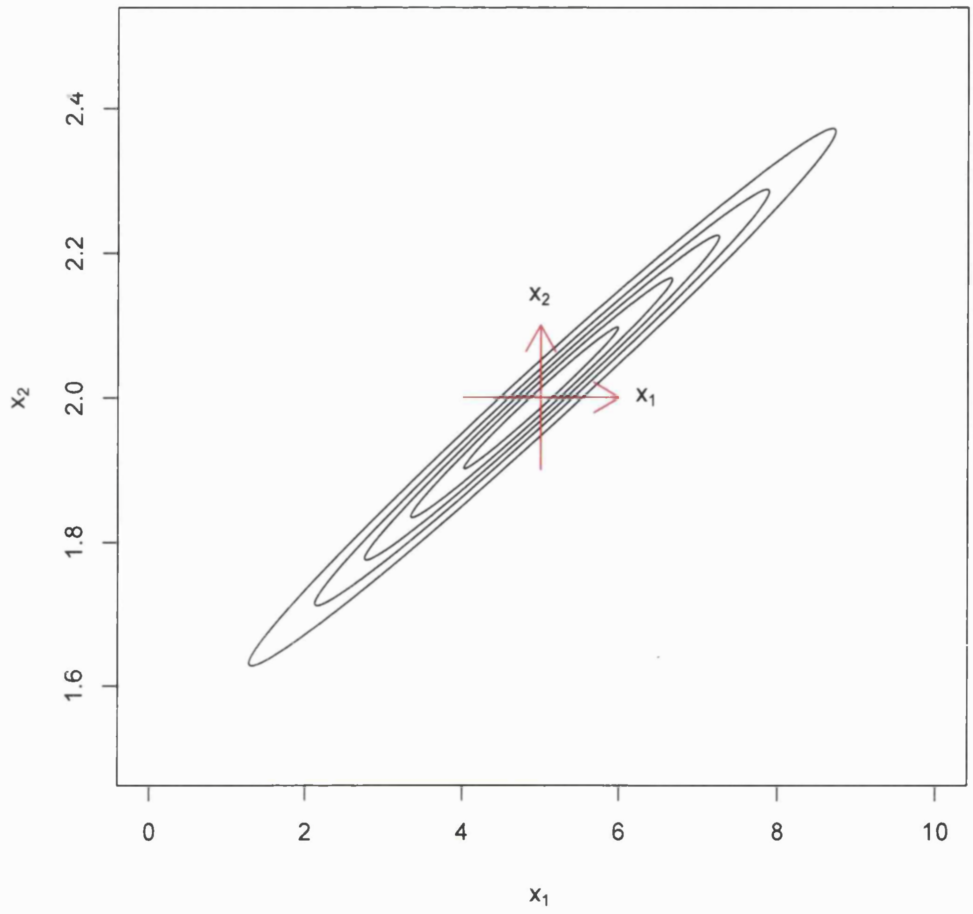


Figure 7-1: *Contours of the bivariate normal distribution $f(X)$.*

We are interested in generating draws from the above distribution. Therefore, we applied the Metropolis algorithm using a systematic single-site updating scheme so that each move is parallel to one of the coordinate axes. Figure 7-1 reveals contours of the corresponding distribution and each component sampling direction. To create these moves normal proposal densities are used. Here, we found that this sampler is most efficient with integrated autocorrelation times of about 432 when the proposal step sizes are fixed at 0.47 times the standard deviations of X_1 and X_2 under $f(X)$. However, for a good mixing chain we seek an integrated autocorrelation time estimate

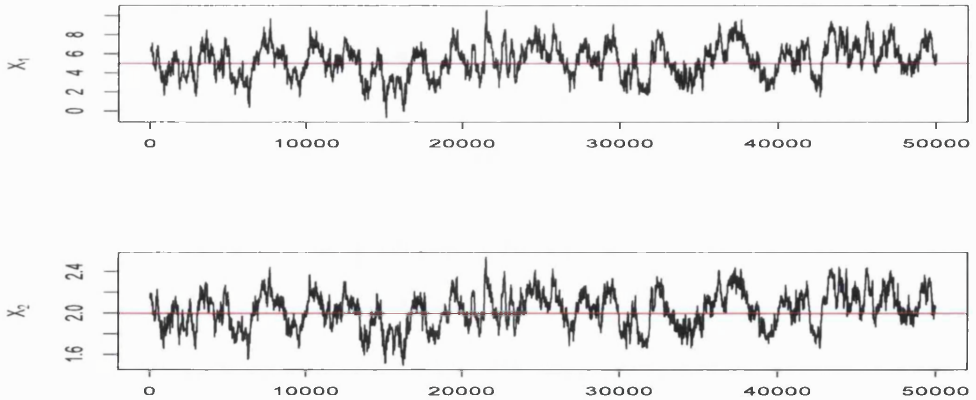


Figure 7-2: *The resulting time-series plots obtained from the last half samples produced by the Metropolis* MCMC *algorithm. The red line represents the true mean value of the corresponding variable. The sample averages of X_1 and X_2 are 4.787 and 1.978, respectively.*

that is not much greater than 1.0. The standard sampler was started from the point $(1.8, 4.5)^T$. With the selected steps, acceptance rates of about 35% are achieved and a burn-in phase of 400 sweeps is required. Figure 7-2 shows the resulting time-series plots obtained from the last 50,000 iterations of a run of 100,000 iterations. Obviously, these plots reveal a slow mixing behaviour of the resulting Markov chain. Further, the large value of the integrated autocorrelation time estimates indicate a strong dependence between successive iterations. A major reason of this behaviour is the high correlation between the two variables (i.e. $\rho = 0.99$). Therefore, the contours of this distribution as illustrated in Figure 7-1 are narrow. In this case, the probability in the sampled distribution is concentrated around the diagonal line $X_2 = 1.5 + 0.1X_1$.

A move between two states (a) and (b) that are far apart from each other needs many small steps as Figure 7-3 illustrates. The reason behind this is that any generated big move that lies away from the main diagonal area will be probably rejected because it lies in a low probability region, and therefore the chain is likely to move very slowly. A solution to the slow mixing problem is to generate moves that allow rapid movement within the region where the distribution f is concentrated. This may be achieved by proposing moves along the orthogonal directions e_1 and e_2 plotted in Figure 7-4, rather than the original coordinate axes of X_1 and X_2. Suitable directions e_1 and e_2 can be obtained by finding the eigenvectors of the variance matrix Σ of X. These eigenvectors

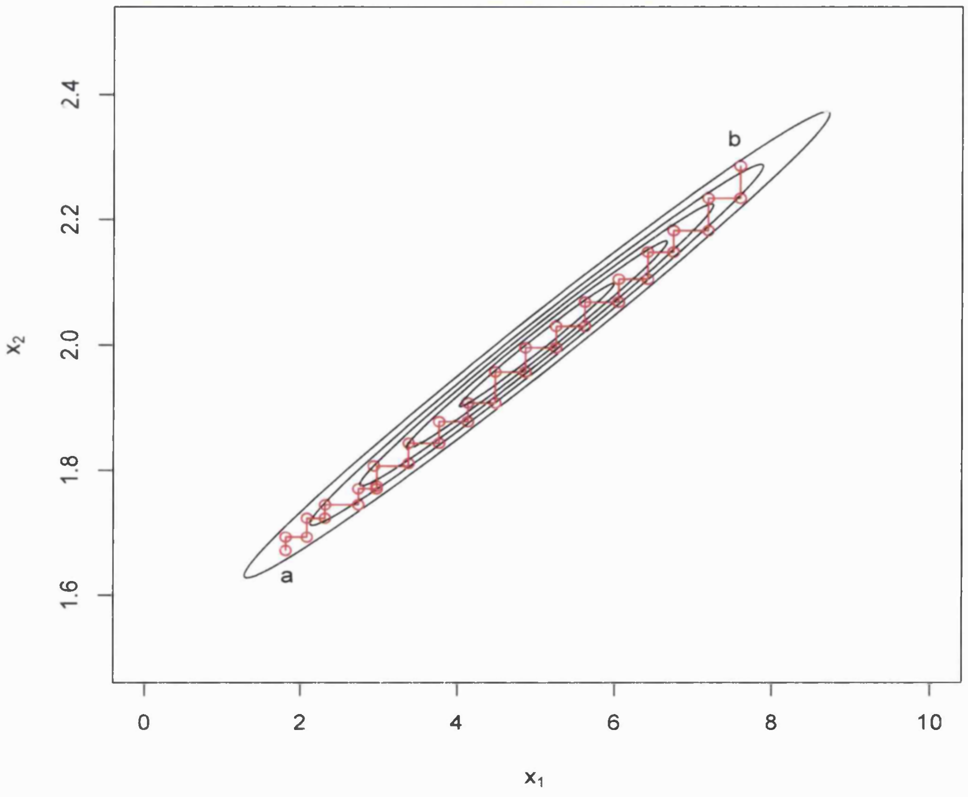


Figure 7-3: *Contours of $f(X)$ and a plausible random path between states a and b.*

are also called the *principal component loadings* or the *projection directions*. These are by construction orthogonal— although they do not appear to be in Figure 7-4 due to the different scales of the axes.

The principal components (PCs), Y_1 and Y_2, of X are linear combinations of the elements of X, that is, $Y_1 = X^T e_1$ and $Y_2 = X^T e_2$. The variances of Y_1 and Y_2 are the eigenvalues of the variance matrix of X, denoted λ_1 and λ_2, respectively. The first principal component accounts for the largest possible variance and the second component accounts for the remaining total variation. In general, the principal components are extracted in decreasing order of variance, so that the ith variate has the ith largest variance. Although the original variables are correlated, their principal components are independent.

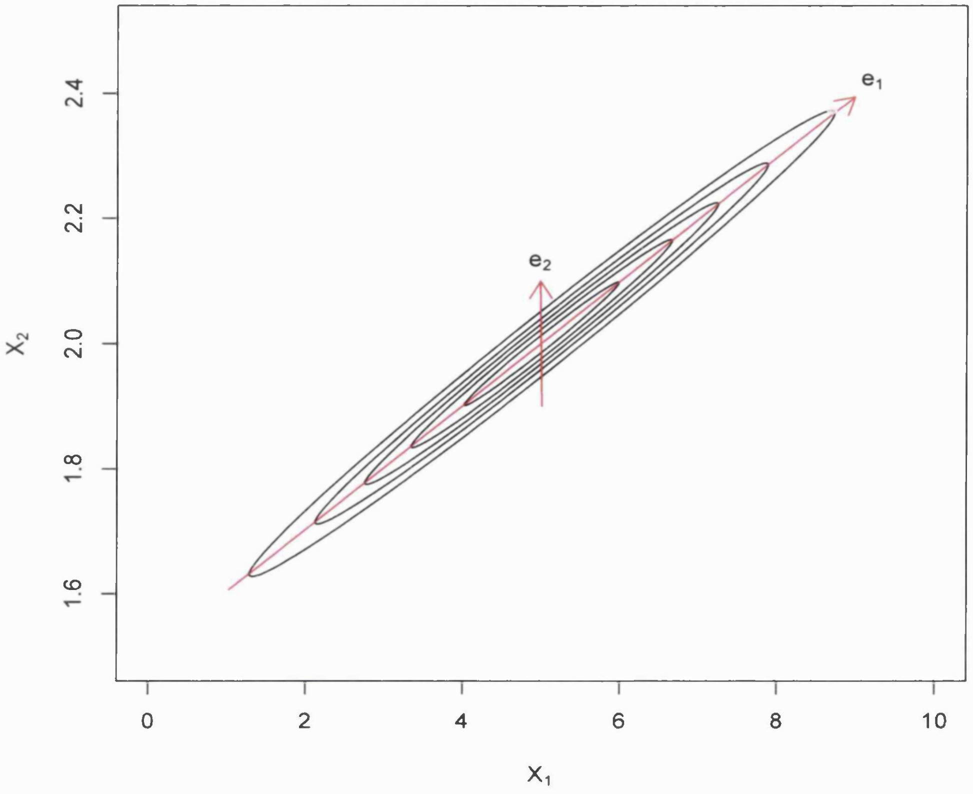


Figure 7-4: *The principal component directions of $f(X)$.*

The method for finding these components is known as the *principal components analysis* (PCA). It is a powerful technique for data analysis (Jolliffe, 1986) and has been widely used for dimensionality reduction. In this case, the general motivation for PCA is to transform the data into some reduced-dimensionality representation that capture most of the variations present in the data set in order to simplify further analysis. For more details on PCA, the reader is referred to Dillon *et al.* (1984, Ch. 2) and Chatfield *et al.* (1980, Ch. 4).

To employ the outcome of PCA in MCMC sampling, one can make Metropolis Hastings updates in the principal component directions by adding multiples of the eigenvectors of the variance of X in each updating step, rather than taking steps along the original axes. Another more appropriate approach is that based on transforming X to a scaled version, $\tilde{X}$, for which each element has variance equal to 1.0. This scaling is a common practice in multivariate analysis especially when the distinct parameters' variances differ substantially.

The MCMC algorithm developed in this thesis on the basis of the PCA approach is defined as the *Principal components* MCMC algorithm and is abbreviated to PCMCMC. The steps of this algorithm are summarised below.

7.2 The PCMCMC algorithm

Let $\pi(X)$ be a $D-$dimensional distribution with a positive-definite variance matrix, Σ. The PCMCMC sampler proceeds as follows:

1. If Σ is known then determine its eigenvectors, $(e_1, \ldots, e_D)$, and the corresponding eigenvalues, $(\lambda_1, \ldots, \lambda_D)$. Alternatively, if it is unknown then use a random sample from π to estimate Σ and derive the eigenvectors and eigenvalues of this estimated variance matrix. One way to obtain such a sample is to run a standard MCMC algorithm long enough, bearing in mind that this may be slowly mixing.

2. At each iteration t:

   - Generate a proposal, X', by creating systematic updates in each principal component direction, for example,

   $$X' = X^{(t)} + N(0, s\lambda_d)e_d, \qquad \text{for} \quad d = 1, \ldots, D, \tag{7.2}$$

   where the constant s is chosen such that a reasonable acceptance rate is attained on each updating direction.

   - Accept X' as the new realisation $X^{(t+1)}$ of the Markov chain with probability

   $$\alpha = \min\left\{1, \frac{\pi(X')}{\pi(X^{(t)})}\right\}.$$

   If X' is accepted then set $X^{(t+1)} = X'$, otherwise reject X' and set $X^{(t+1)} = X^{(t)}$.

7.3 Bivariate Gaussian examples

In this section, we demonstrate the implementation process of the PCMCMC algorithm via three illustrative examples. In the first example, we simulate the bivariate normal distribution, $f(X)$, defined in section 7.1 using the true variance matrix; in the second example, we simulate f using a rough estimate of the variance matrix; and in the last example we increase the correlation, ρ, between the two variables of f to 0.99999 and sample the modified distribution.

7.3.1 Example 1

Suppose $f(X) = N(\mu, \Sigma)$, where $\mu = (5, 2)^T$ and

$$\Sigma = \begin{pmatrix} 2.0^2 & 0.396 \\ 0.396 & 0.2^2 \end{pmatrix},$$

with time series plots generated using the standard approach shown in Figure 7-2. The latter results reveal a slow mixing problem and therefore we will try implementing the PCMCMC in an attempt to improve the results. The principal components of Σ can be obtained by determining its eigenvalues and eigenvectors. The two eigenvalues are $\lambda_1 = 4.0392$ and $\lambda_2 = 0.000788$, and the related eigenvectors are $e_1 = (0.9951, 0.0985)^T$ and $e_2 = (-0.0985, 0.9951)^T$. Here $\sum_{i=1}^{2} e_{ij}^2 = 1$, for $j = 1, 2$, where $e_1 = (e_{11}, e_{12})^T$ and $e_2 = (e_{21}, e_{22})^T$.

The PCMCMC approach with $s = 1$ in Eq (7.2) allows us to generate moves of the following type:

$$X \rightarrow X + Z_1 e_1, \qquad Z_1 \sim N(0, 4.0392), \tag{7.3}$$

and

$$X \rightarrow X + Z_2 e_2, \qquad Z_2 \sim N(0, 0.000788). \tag{7.4}$$

We implemented this approach for 100,000 iterations starting with the same initial values used for the standard approach, namely $(1.8, 4.5)^T$. The acceptance rate on each direction is found to be around 70%. The burn-in period is roughly 200 sweeps. Figure 7-5 traces the resulting last 50% of the 100,000 iterations. Comparing these traces with those obtained from the standard Metropolis algorithm run of Figure 7-2 indicates a very significant improvement in the mixing speed of each chain. Estimates

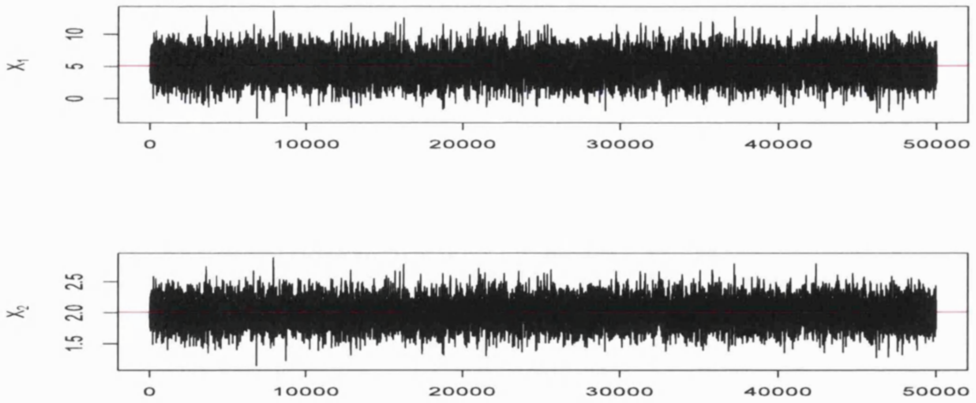


Figure 7-5: *The resulting time-series plots obtained from the* PCMCMC *algorithm based on analysing the covariance matrix, Σ. Only the last half samples of each generated chain are displayed. The red line represents the true mean of the corresponding variable. The sample averages of X_1 and X_2 are 5.018 and 2.002, respectively.*

of the integrated autocorrelation times for X_1 and X_2 are reduced from approximately 432 to 9.6.

It is worth mentioning that the first principal component, $Y_1 = 0.9951X_1 + 0.0985X_2$, is dominated by X_1 because it has the largest weight, although X_1 and X_2 are very correlated. The reason for this is that X_1 has the higher variance, i.e. $\text{Var}(X_1) = 4$ and $\text{Var}(X_2) = 0.04$. Thus a move on the first principal component direction, e_1, will be highly dominated by X_1. To make both variables equally important and directly comparable, we can scale these parameters so that each have a unit variance. This will transform X to $\tilde{X}$ with components, $\tilde{X_1} = X_1/2$ and $\tilde{X_2} = X_2/0.2$, i.e.,

$$\tilde{X} = DX,$$

where,

$$D = \begin{pmatrix} 0.5 & 0 \\ 0 & 5 \end{pmatrix}.$$

The variance matrix of $\tilde{X}$ is

$$\tilde{\Sigma} = \begin{pmatrix} 1 & 0.99 \\ 0.99 & 1 \end{pmatrix},$$

and the corresponding eigenvalues are, $\tilde{\lambda}_1 = 1.99$ and $\tilde{\lambda}_2 = 0.01$, with eigenvectors given by, $\tilde{e}_1 = (1/\sqrt{2}, 1/\sqrt{2})^T$ and $\tilde{e}_2 = (1/\sqrt{2}, -1/\sqrt{2})^T$. In this case, the principal components are

$$\tilde{Y}_1 = \frac{1}{\sqrt{2}}\tilde{X}_1 + \frac{1}{\sqrt{2}}\tilde{X}_2,$$

and

$$\tilde{Y}_2 = \frac{1}{\sqrt{2}}\tilde{X}_1 - \frac{1}{\sqrt{2}}\tilde{X}_2.$$

Hence, the Metropolis updates are now of the form

$$\tilde{X} \to \tilde{X} + Z_1\tilde{e}_1, \qquad Z_1 \sim N(0, 1.99), \tag{7.5}$$

and

$$\tilde{X} \to \tilde{X} + Z_2\tilde{e}_2, \qquad Z_2 \sim N(0, 0.01). \tag{7.6}$$

Note that updates in Eqs (7.5) and (7.6) are not the same as those given in Eqs (7.3) and (7.4) but are related via

$$X = D^{-1}\tilde{X}, \tag{7.7}$$

where,

$$D^{-1} = \begin{pmatrix} 2 & 0 \\ 0 & 0.2 \end{pmatrix}.$$

Thus, adding a move in the direction of $\tilde{e}_1$ to $\tilde{X}_1$, for example, is equivalent to adding a move in the direction of $D^{-1}\tilde{e}_1$ to X. That is,

$$X \to X + Z_1 f_1, \qquad Z_1 \sim N(0, 1.99), \tag{7.8}$$

and

$$X \to X + Z_2 f_2, \qquad Z_2 \sim N(0, 0.01), \tag{7.9}$$

where $f_1 = (2/\sqrt{2}, 0.2/\sqrt{2})^T$ and $f_2 = (2/\sqrt{2}, -0.2/\sqrt{2})^T$.

The latter vectors are not unit vectors, e.g., the norm, $||f_i|| \neq 1.0$, for $i = 1, 2$. Therefore, to represent them in the same way that e_1 and e_2 are shown in Figure 7-4, we standarised them to unit vectors, $\hat{f}_1$ and $\hat{f}_2$. These are $\hat{f}_1 = (0.9950, 0.0995)^T$ and $\hat{f}_2 = (0.9950, -0.0995)^T$. Figure 7-6 plots $\hat{f}_1$ and $\hat{f}_2$ together with e_1 and e_2. In contrast to $\hat{f}_1$ and e_1 which are approximately identical, $\hat{f}_2$ and e_2 differ greatly. Moreover, unlike the original eigenvectors, the standarised vectors are not orthogonal. Therefore, the projections of X on these directions are not independent, i.e. $\text{cov}(X^T\hat{f}_1, X^T\hat{f}_2) = \hat{f}_1^T\Sigma\hat{f}_2 \neq 0$.

We refer to the PCMCMC algorithm applied to a scaled version of X and

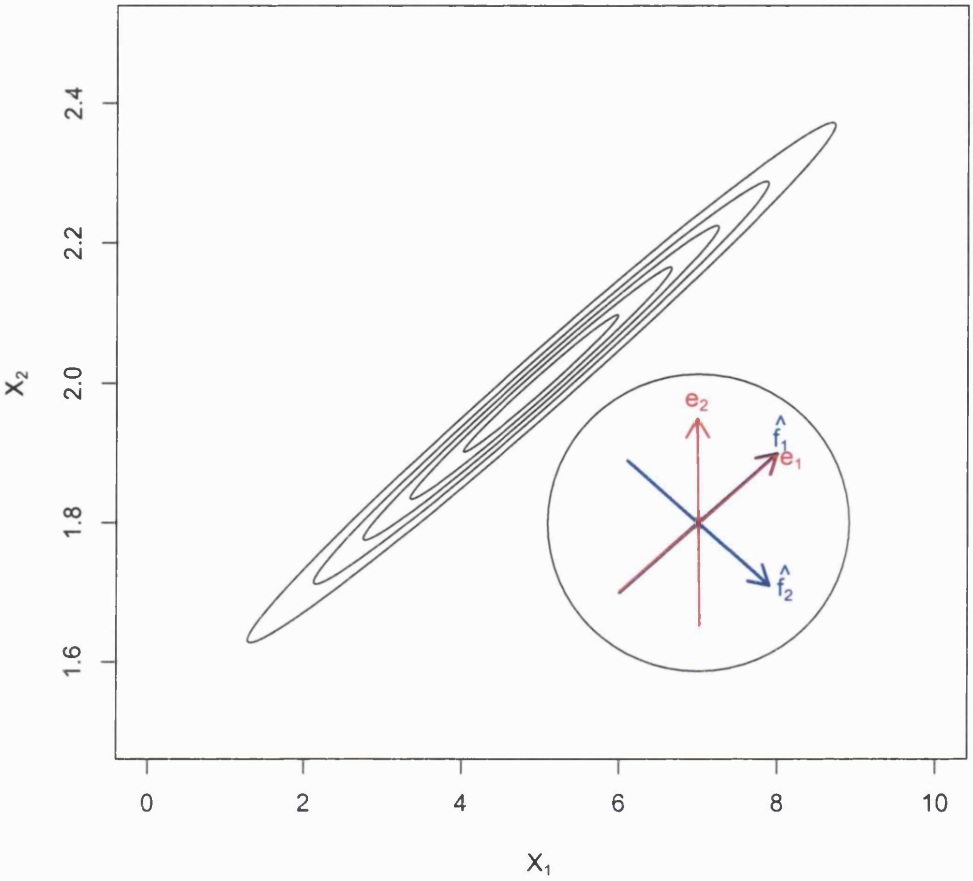


Figure 7-6: *The sampling directions of the unscaled variables obtained by analysing:* Σ *(red) and* $\tilde{\Sigma}$ *(blue) plotted with the contours for* $f(X)$.

transforming back to X as the scaled version of PCMCMC. We will now simulate the bivariate normal distribution, $f(X)$, using the scaled version of PCMCMC. Therefore, we ran the sampler for 100,000 iterations. Again around 70% of the generated proposals are accepted and the burn-in period is of length 200 iterations. Figure 7-7 shows the resulting sample traces. Here the integrated autocorrelation times are also around 9.6. The corresponding PCs time series plots are shown in Figure 7-8.

The traces of Figures 7-5 and 7-7 reveal no significant difference between the two

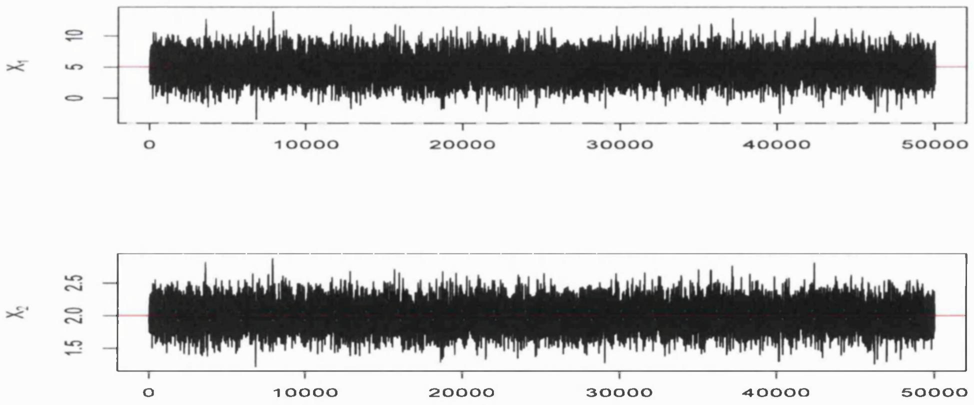


Figure 7-7: *The resulting time-series plots obtained from the* PCMCMC *run based on analysing,* $\tilde{\Sigma}$. *Only the last half samples of each generated chain are displayed. The red line represents the true mean of the corresponding variable. The sample averages of* X_1 *and* X_2 *are 5.012 and 2.001, respectively.*

PCMCMC versions for exploring $f(X)$. This is because the main principal component direction, e_1, on which most moves should be generated is captured by $\hat{f}_1$. In addition, applying PCMCMC to the scaled variable $\tilde{X}$ still produces an independence property.

The projection of $\tilde{X}$ in directions $\tilde{e}_1$ and $\tilde{e}_2$ are independent, even though this is not the case for the projections of X in directions $\hat{f}_1$ and $\hat{f}_2$. Since the sequence of $\tilde{X}$ values mixes well, so does the corresponding sequence of X values.

The results from both versions of the PCMCMC algorithm indicate that this is an efficient sampling approach because by implementing the PCA the MCMC algorithm is allowed to explore the whole sample space more freely. There are three reasons for this: first, moves that are far away from the main diagonal are now generated very less often; secondly, big jumps such as those between states (a) and (b) of Figure 7-3 can be easily performed via few moves, of which some are very large, without being rejected; and finally, the proposed estimates are based on moves in the direction of independent principal components, $Y_1 = X^t e_1$ and $Y_2 = X^t e_2$, or alternatively, $\tilde{Y}_1 = \tilde{X}^t \tilde{e}_1$ and $\tilde{Y}_2 = \tilde{X}^t \tilde{e}_2$, as can be seen in the scatter plots of Figure 7-8.

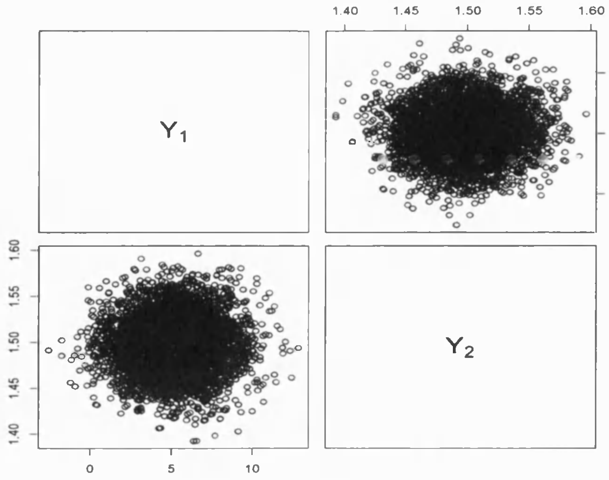


Figure 7-8: *The scatter plot of the principal components,* Y_1 *and* Y_2 of X, *obtained from samples generated by the scaled version of PCMCMC.*

7.3.2 Example 2

Now, assume that we do not know the variance matrix Σ, given in section 7.1, but we would still like to get benefits of the PCMCMC to generate realisations from $f(X)$. Further, suppose we have a set of X values obtained using the standard Metropolis algorithm to sample from $f(X)$. An approach to tackle the problem of the "missing" variance matrix is to approximate it from the given data values. Implementing this method gives us the following estimator of Var(X):

$$\widehat{\Sigma} = \begin{pmatrix} 4.02522 & 0.39884 \\ 0.39884 & 0.04031 \end{pmatrix},$$

with a correlation coefficient, $\hat{\rho} = 0.99013$. The resulting eigenvalues are $\hat{\lambda}_1 = 4.06475$ and $\hat{\lambda}_2 = 0.000783$ and the associated eigenvectors are $\hat{e}_1 = (0.9951, 0.0986)^T$ and $\hat{e}_2 = (-0.0986, 0.9951)^T$. Using these estimates the PCMCMC sampler was ran for 100,000 sweeps with $s = 1$. The resulting acceptance rates are 70% and the associated integrated autocorrelation times are approximately 9.6. Figure 7-9 displays the last half of the time-series traces.

The results obtained from this run show insubstantial difference from those of the

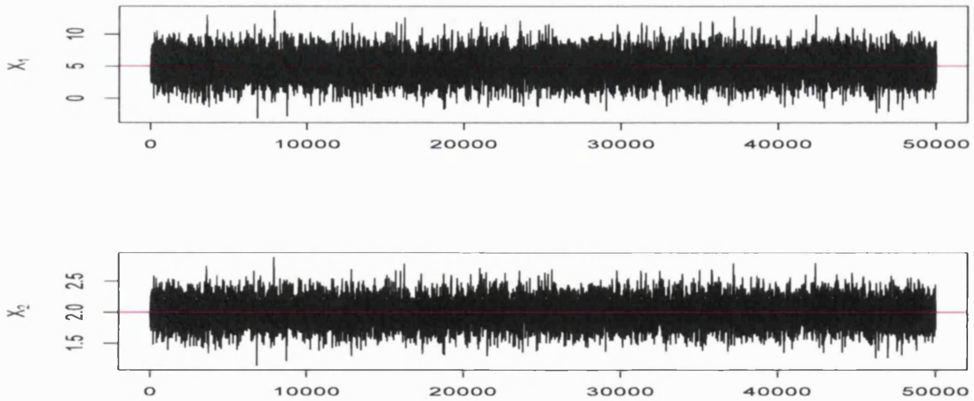


Figure 7-9: *Time-series plots of the last half of 100,000 iterations obtained from the PCMCMC algorithm when estimates of the principal components are extracted from the approximated variance matrix,* $\hat{\Sigma}$. *The red line represents the true mean of the corresponding variable. The sample averages of* X_1 *and* X_2 *are 5.019 and 2.003, respectively.*

PCMCMC run based on the true value of Σ. This is because the output derived from the standard approach gave an accurate estimate of the underlying variance matrix of X, although the chains used in the approximation step are not mixing well.

7.3.3 Example 3

Let us assume that the correlation present between the two parameters of $f(X)$ is much higher, e.g., $\rho = 0.99999$ and call the resulting distribution, $g(X)$, that is $g(X) = N(\mu, \Sigma)$, where $\mu = (5, 2)^T$ and

$$\Sigma = \begin{pmatrix} 4 & 0.399996 \\ 0.399996 & 0.04 \end{pmatrix}.$$

The eigenvalues of Σ are $\lambda_1 = 4.04$ and $\lambda_2 = 7.92 \times 10^{-7}$ and the corresponding eigenvectors are approximately, $e_1 = (0.9950, 0.0995)^T$ and $e_2 = (-0.0995, 0.9950)^T$. The tightly packed contours of this distribution are shown in Figure 7-10.

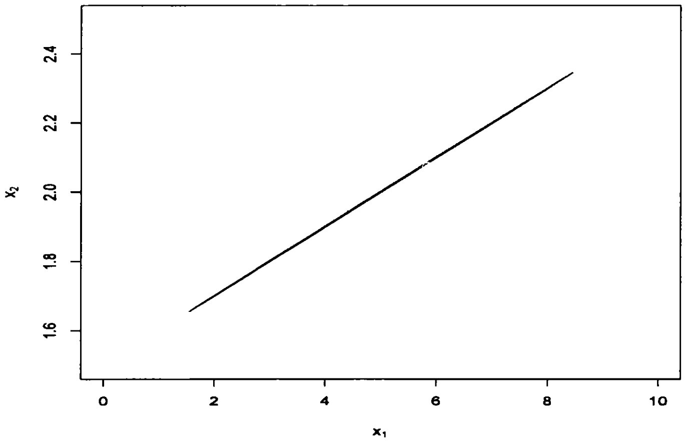


Figure 7-10: *Contours of the bivariate normal distribution $g(X)$. The contours are so close together that no white space is visible between them in the figure.*

PCMCMC for X directly

We performed a simulation study to assess the performance of the PCMCMC based on an estimate of Σ from data generated by the standard MCMC approach. We tuned the standard deviation of the proposal distribution in order to have a reasonable acceptance rate. We found that the acceptance rate on each main axis is about 35% with steps having standard deviations of 0.015 times the standard deviation of X_1 and X_2 under $g(X)$. Therefore, we used these steps.

Moreover, in an attempt to capture "good" estimates of the required variance matrix and eigenvectors, the Metropolis algorithm is allowed to run for a long time. Here we used 500,000 iterations and a starting point that is not too far away from the modal centre. This point is chosen to be $(4.8, 1.8)^T$. Figure 7-11 shows the time-series plots obtained from this run. The corresponding empirical averages for X_1 and X_2 are, respectively, 3.7053 and 1.8705 and the estimated variance of X is

$$\widehat{\Sigma} = \begin{pmatrix} 0.418 & 0.0418 \\ 0.0418 & 0.00418 \end{pmatrix}.$$

The eigenvectors of $\widehat{\Sigma}$ are, $\hat{e}_1 = (0.9950, 0.0995)^T$ and $\hat{e}_2 = (-0.0995, 0.9950)^T$, corresponding to the eigenvalues, $\hat{\lambda}_1 = 0.423$ and $\hat{\lambda}_2 = 2.445 \times 10^{-6}$.

Clearly, the estimated variance matrix is not very close to Σ. However e_1 and e_2 are accurately estimated. This is because, despite the fact that the sampler is poorly

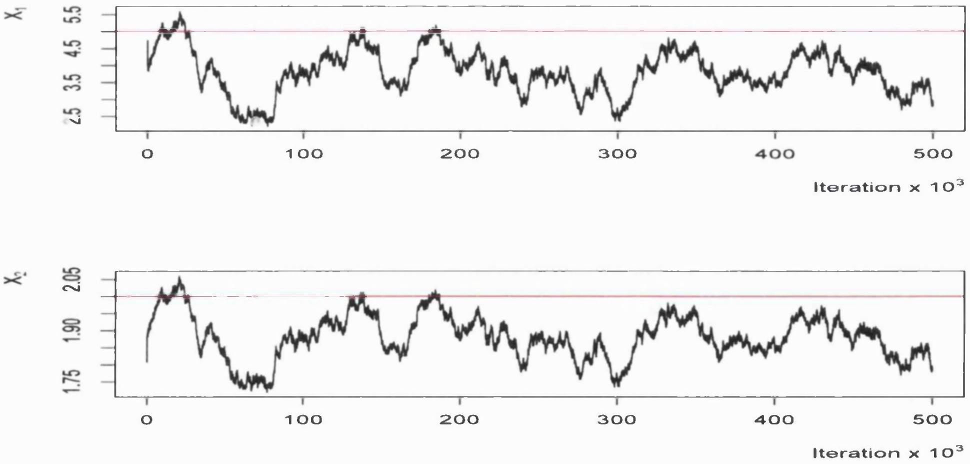


Figure 7-11: *Traces of X_1 and X_2 obtained by sampling the bivariate normal distribution $g(X)$ using 500,000 iterations of the standard Metropolis sampler. The red lines represent the true means.*

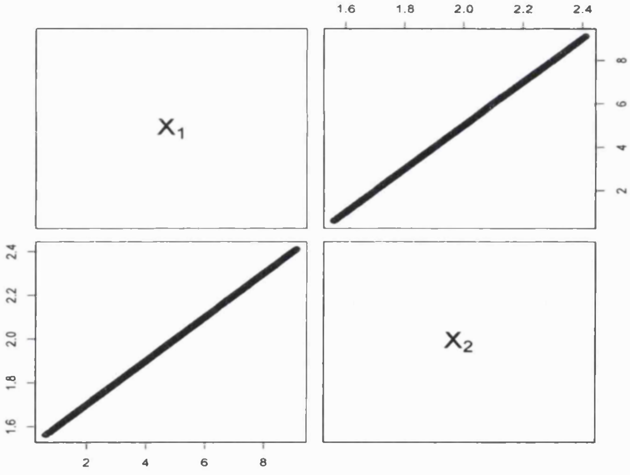


Figure 7-12: *The scatter plot of X_1 and X_2 obtained from simulating $g(X)$ using the standard MCMC algorithm.*

mixing, the accepted proposals lie around the diagonal line, $X_2 = 1.5 + 0.1X_1$, see for example, Figure 7-12, leading us to a good understanding of the direction in which the distribution $g(X)$ is concentrated.

Using the above estimates the PCMCMC algorithm was ran for 500,000 iterations with $s = 1$ and the same initial state $(4.8, 1.8)^T$. The acceptance rate on the first principal component's direction is about 90%, whereas on that of the second component it is around 54%. Figure 7-13 shows the resulting traces. The averages for X_1 and X_2 obtained from the last half iterations are, respectively, 5.007 and 2.0007, and the integrated autocorrelation time for each variable expected value is roughly 51. Moreover, the estimated variance matrix obtained from the PCMCMC run is

$$\hat{\Sigma}_{PC1} = \begin{pmatrix} 4.171 & 0.417 \\ 0.417 & 0.0417 \end{pmatrix},$$

much closer to the true value.

Scaled PCMCMC

An alternative approach is that based on $\tilde{X}$, the scaled version of X, $\tilde{X} = DX$ where,

$$D = \begin{pmatrix} \Sigma_{11}^{-\frac{1}{2}} & 0 \\ 0 & \Sigma_{22}^{-\frac{1}{2}} \end{pmatrix}.$$

Since only an estimate of Σ is available from an initial run of the standard MCMC sampler, we use this to define

$$\begin{aligned} \hat{D} &= \begin{pmatrix} 0.418^{-\frac{1}{2}} & 0 \\ 0 & 0.00418^{-\frac{1}{2}} \end{pmatrix} \\ &= \begin{pmatrix} 1.547 & 0 \\ 0 & 15.47 \end{pmatrix}, \end{aligned}$$

and transform to $\tilde{X} = \hat{D}X$. The estimate of $\tilde{\Sigma} = \text{Var}(\tilde{X})$ is then

$$\hat{D}\hat{\Sigma}\hat{D} = \begin{pmatrix} 1 & 0.99971 \\ 0.99971 & 1 \end{pmatrix}.$$

This matrix has eigenvectors, (0.70711, 0.70711) and (0.70711, −0.70711), and associated eigenvalues 1.9997 and 0.000295. The PCMCMC run in this case with $s = 1$ produced the traces displayed in Figure 7-14.

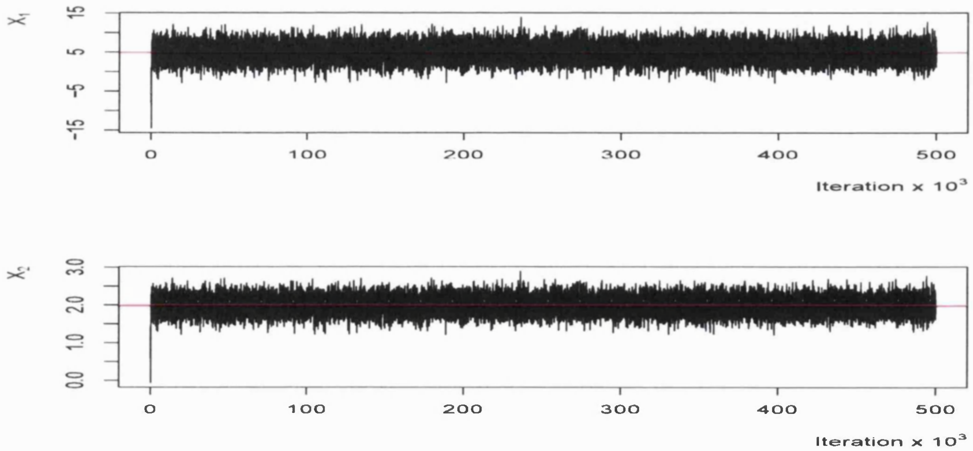


Figure 7-13: *Traces of X_1 and X_2 obtained by sampling the bivariate normal distribution $g(X)$ using the unscaled version of the PCMCMC sampler with $s = 1$. The red lines represent the true means.*

The resulting estimated variance matrix of $g(X)$ is

$$\hat{\Sigma}_{PC2} = \begin{pmatrix} 4.186 & 0.418 \\ 0.418 & 0.0418 \end{pmatrix}.$$

The corresponding averages for X_1 and X_2 are 4.9954 and 1.9995, respectively, with integrated autocorrelation times of about 51. The acceptance rate on the first and second principal component directions are 90% and 54%.

Obviously, the traces of Figure 7-11 reveal that the standard MCMC sampler is mixing very poorly since it can only move slowly within $g(X)$ due to this distribution's narrow shape. Therefore, the estimated variance matrix, $\widehat{\Sigma}$, and the corresponding eigenvalues are quite far from the truth. However, the distribution's correlation structure is reasonably well captured and the resulting eigenvectors' estimates are very accurate.

In contrast, the plots in Figures 7-13 and 7-14 reveal a significant improvement in terms of mixing. The integrated autocorrelation times estimated for X_1 and X_2 obtained from both versions of the PCMCMC results are around 51, whereas those obtained from the standard approach are about 23388. This indicates that the PCMCMC sampler is much more efficient than the standard sampler whether we scaled

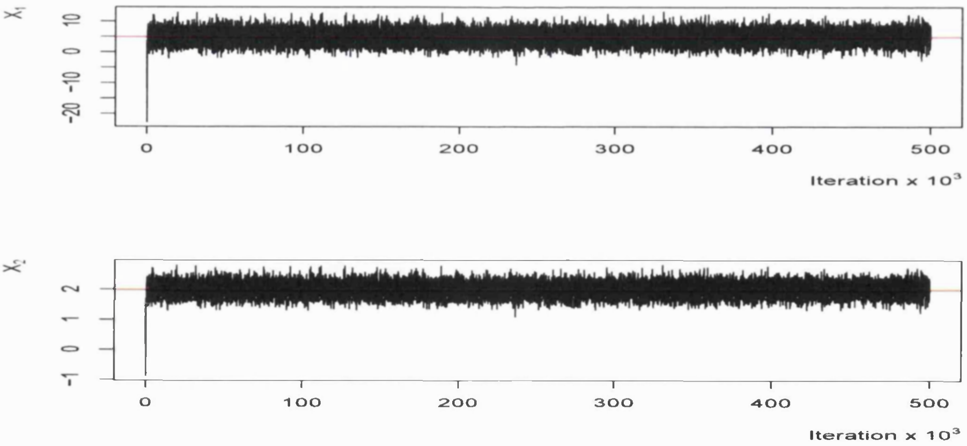


Figure 7-14: *Traces of X_1 and X_2 obtained by sampling the bivariate normal distribution $g(X)$ using the scaled version of the PCMCMC sampler with $s = 1$. The red lines represent the true means.*

the main variables or not. Therefore, accurate estimates of the variance matrix Σ and the parameters' expectations are accomplished, although $\widehat{\Sigma}$ used to derive estimates of the necessary principal components was underestimated by approximately a factor of 10.

In this example, it is clear that the high acceptance rates associated with the PCMCMC results are due to the poor choice of the eigenvalues' estimators. This is because the standard algorithm from which we obtain these estimators was mixing very slowly. As a result, the integrated autocorrelation time increased to 51 with $s = 1$. Obviously the acceptance rates can be decreased by increasing the value of s. In the next section, the effect of this constant will be studied in more detail.

The examples discussed in this section suggest that the standard Metropolis sampler performance on normal distributions is dominated by the original variance matrix eigenvectors. Even if the estimated variance matrix is incorrect, as we have seen in the last example, we are headed in the right principal component directions. However the eigenvalues may not be well estimated. Thus, the PCMCMC algorithm can be implemented without fear and is expected to be very efficient for simulating similar distributions. In all further implementations of the PCMCMC algorithm only the scaled version will be used for the purposes indicated earlier in this chapter.

7.4 Choosing standard deviations of proposals in the PCMCMC algorithm

The last section examples allowed for a good visualisation of the PCMCMC algorithm for sampling bivariate normal distributions. The estimated integrated autocorrelation times when accurate eigenvalue estimates are used with $s = 1$ are found to be 9.6. Here, the associated acceptance rates obtained on each PC direction is approximately 70%. In general, this reflects a good mixing behaviour despite the high correlations present between the random variables. However, the acceptance rate associated with each PCMCMC proposal is high. This raises the following question: what value of the proposal constant, s, of Eq (7.2) makes the PCMCMC sampler achieve optimality?

Gelman *et al.* (1996) revealed that the random walk Metropolis algorithm when implemented to sample a univariate standard normal distribution performs optimally if the overall acceptance rate is around 44% or alternatively the proposal standard deviation is set to be 2.4 times the target standard deviation. They considered the asymptotic efficiency of the Markov chain for estimating a parameter θ given by $\tau(\theta)^{-1}$, where $\tau(\theta)$ is the integrated autocorrelation time for θ, as an efficiency measuring parameter of the MCMC sampler. These authors found that for the standard normal distribution the optimal asymptotic efficiency is about 0.23, that is the optimal integrated autocorrelation time is approximately 4.4. We would then anticipate that for any D-dimensional normal distribution, the PCMCMC optimal proposal standard deviations are $\delta_i = 2.4\sqrt{\lambda_i}$, for $i = 1, \ldots, D$, because sampling is done componentwisely in the orthogonal directions of the independent principal components and each component marginal distribution is Gaussian.

For an illustrative example, consider the distribution of Section 7.1, $f(X)$, where $X = (X_1, X_2)^T$. The PCMCMC sampling algorithm can be looked at in terms of the principal components of X; $Y_1 = X^T e_1$ and $Y_2 = X^T e_2$. So the sampling problem simplifies to simulating Y_1 and Y_2 each on its own direction, where $Y_1 \sim N(E(Y_1), \lambda_1)$ and $Y_2 \sim N(E(Y_2), \lambda_2)$.

Now, to derive the optimal proposal standard deviation for the PCMCMC algorithm, we ran the sampler for a set of s values. Each run consists of 500,000 sweeps, and at each sweep Y_1 and Y_2 are updated systematically using Gaussian proposals centred at the current Y_i values with steps of size $s\sqrt{\lambda_i}$. Table 7.1 reveals the proposal steps used on the first principal component direction, the resulting acceptance rate and the integrated autocorrelation time for the estimate of $E(Y_1)$. Results of the second

Table 7.1: *The PCMCMC results obtained by tuning the Gaussian proposal scalings of the principal components of X.*

s	$\hat{\tau}(E(Y_1))$	Acceptance rate % on e_1
0.5	23.0	84
1.0	8.6	70
1.5	5.7	59
2.0	4.6	50
2.4	4.4	44
3.0	4.5	37
3.5	4.9	33
4.0	5.3	30
4.5	5.7	27
5.0	6.1	24

principal component with the same steps are almost identical to those of the first component and therefore not displayed.

The left panel of Figure 7-15 plots the resulting asymptotic efficiency measurement versus s, whereas the right panel plots the efficiency parameter against the acceptance rate associated with each principal component. Similar results (not shown) have been obtained when the scaled version of the PCMCMC is implemented instead.

The results of Table 7.1 and the plots of Figure 7-15 reveal that for sampling Gaussian distributions the PCMCMC algorithm optimal Gaussian proposal standard deviations are approximately $2.4\sqrt{\lambda_i}$ and the associated acceptance rate on each principal component direction is approximately 44%. The optimal asymptotic efficiency is around 0.23, that is the integrated autocorrelation time is optimised at a value of 4.4 approximately. These results are very consistent with those obtained by Gelman *et al.* (1996) for sampling the Gaussian distribution with $D = 1$, as expected.

As an implication on the above conclusion, we suggest using Gaussian proposals with variances $2.4^2\lambda_i$ whenever sampling is to be carried out via the PCMCMC approach. This ought to give an acceptance rate of about 44%. If this rate is not achieved, then perhaps $\hat{\lambda}_i$ is not a good estimate of λ_i. Here, one can either perform a longer run of the standard MCMC to obtain a better estimate of the underlying variation structure, or alternatively tune $\hat{\lambda}_i$ until the optimal acceptance rate is roughly obtained on each PC direction.

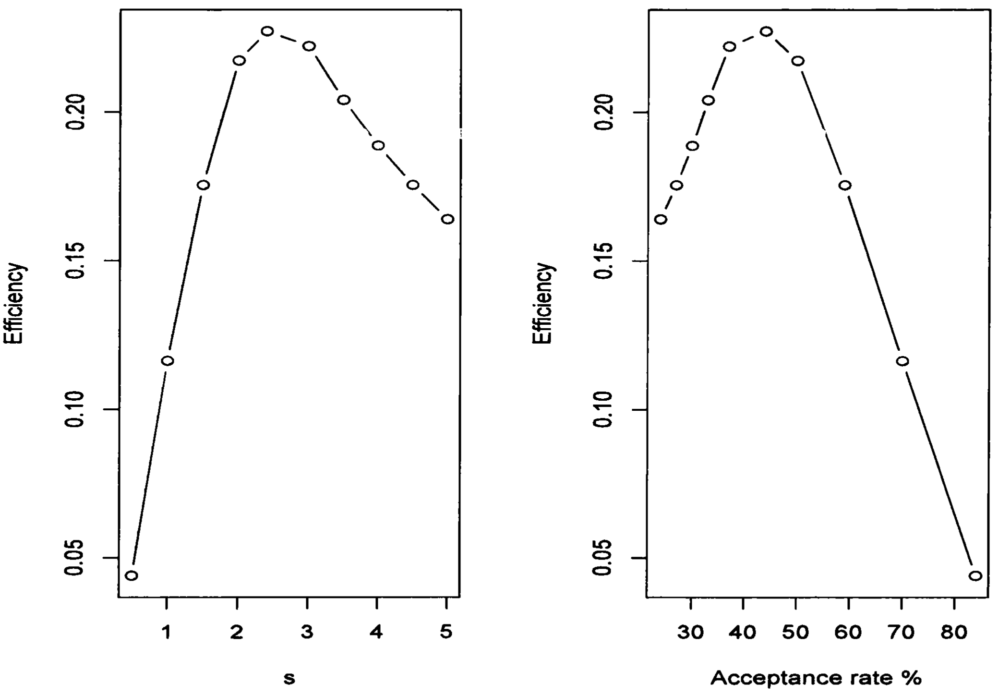


Figure 7-15: *Efficiency $\tau(\theta)^{-1}$ of the PCMCMC related to s (left) and the acceptance rate on each principal component sampling direction (right).*

7.5 Higher-dimension and non-Gaussian examples

Here we will test the performance of the PCMCMC sampler further. In the first experiment, the aim is to simulate a high-dimensional Gaussian distribution. In the second demonstration, we will apply the PCMCMC method to four problems sampling four two-dimensional but non-Gaussian distributions. In all cases, the efficiency of the PCMCMC sampler will be compared with that of the standard Metropolis algorithm.

7.5.1 Example 1

In this example, we will consider the more general case in which the target model is a multi-dimensional normal density with highly dependent components. Our example is a 10-dimensional distribution, $h(X)$. The mean of h is, $\mu_h = (2, 5, 10, 1, 3, 4, 7, 6, 12, 15)^T$, with variance matrix diagonal elements

$$9, 1, 2.25, 6.25, 0.01, 0.04, 0.25, 0.0625, 4, 25,$$

and correlation matrix given by

$$L = \begin{pmatrix} 1 & 0.95 & 0.95^2 & \dots & 0.95^9 \\ 0.95 & 1 & 0.95 & \dots & 0.95^8 \\ 0.95^2 & 0.95 & 1 & \dots & 0.95^7 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.95^9 & 0.95^8 & 0.95^7 & \dots & 1 \end{pmatrix}.$$

We performed a simulation study to compare the efficiency of the PCMCMC sampler to that of the standard Metropolis algorithm when sampling $h(X)$. In each case, we generated 100,000 iterations. For the ordinary component-wise run we tuned the proposals' standard deviations in order to make the algorithm more efficient. We found that acceptance rates of about 27% suitable for this goal. The first 10,000 sweeps are discarded as a burn-in phase. Figure 7-16 reveals the time-series plots of the last 10,000 samples gained from this approach.

The estimates of $\text{Var}(X_1), \dots, \text{Var}(X_{10})$ from simulation output of the standard MCMC run are close to the true values as displayed below

$$8.69, 0.96, 2.17, 6.04, 0.0098, 0.04, 0.25, 0.063, 4.05, 25.3. \tag{7.10}$$

Moreover, the set of sampled values gave us an accurate estimate of the target correlation structure as the 2D scatter plots illustrated in Figure 7-17 reflect.

We will now test the performance of the new approach, namely PCMCMC, for sampling the multivariate normal density $h(X)$. In an attempt to make the problem of applying the PCMCMC more realistic, we supposed we do not know the true variances nor the correlation matrix of h. Therefore, we will use the estimate of the variances from the standard MCMC to run the PCMCMC sampler. For this purpose, we define $\tilde{X} = DX$ and estimate the diagonal matrix D elements from the variance matrix estimates given in (7.10) above, as explained in section 7.3.3.

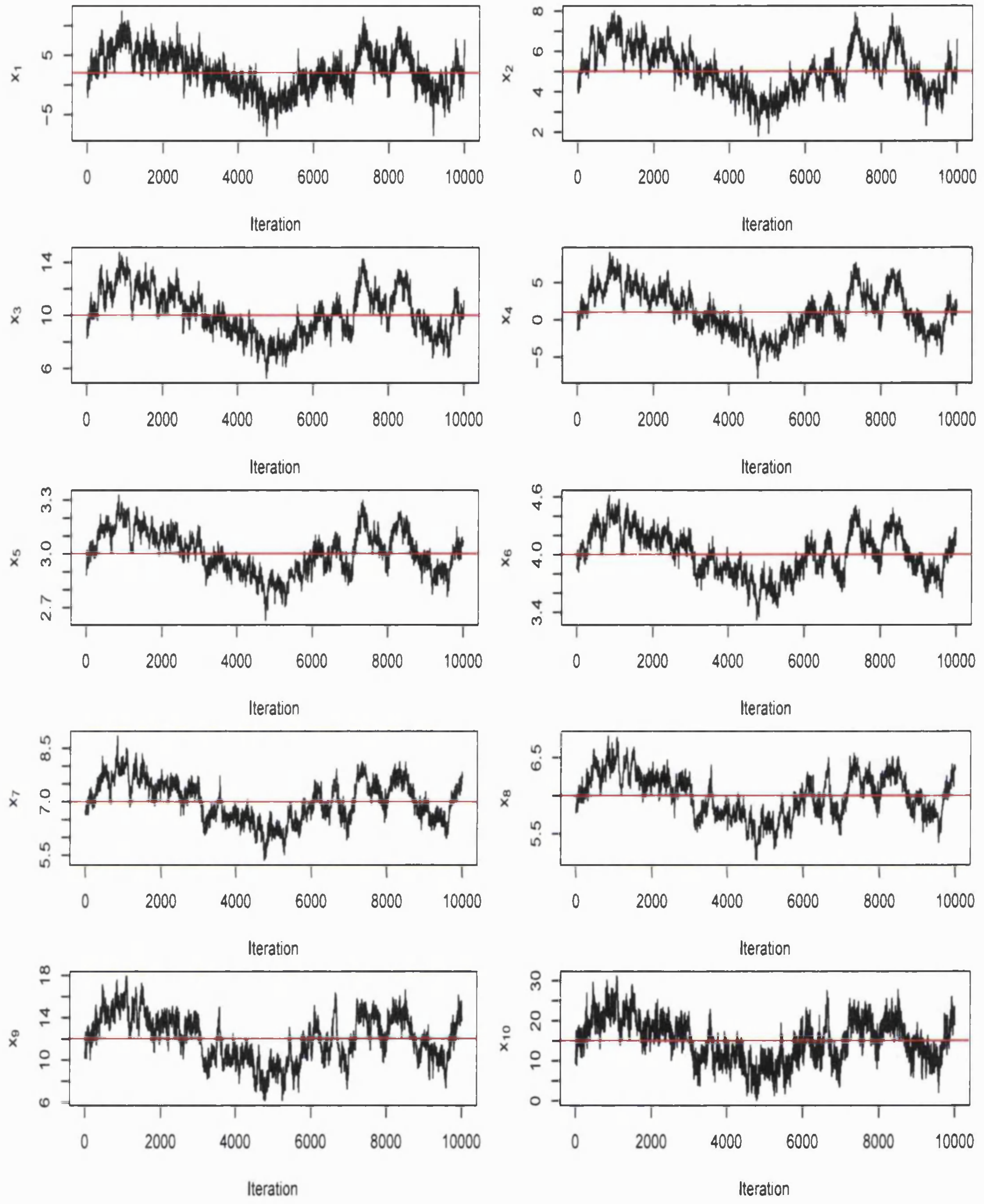


Figure 7-16: *Time-series plots of the 10-dimensional problem obtained by implementing the standard Metropolis sampler. Red lines represent the corresponding parameters' true mean values.*

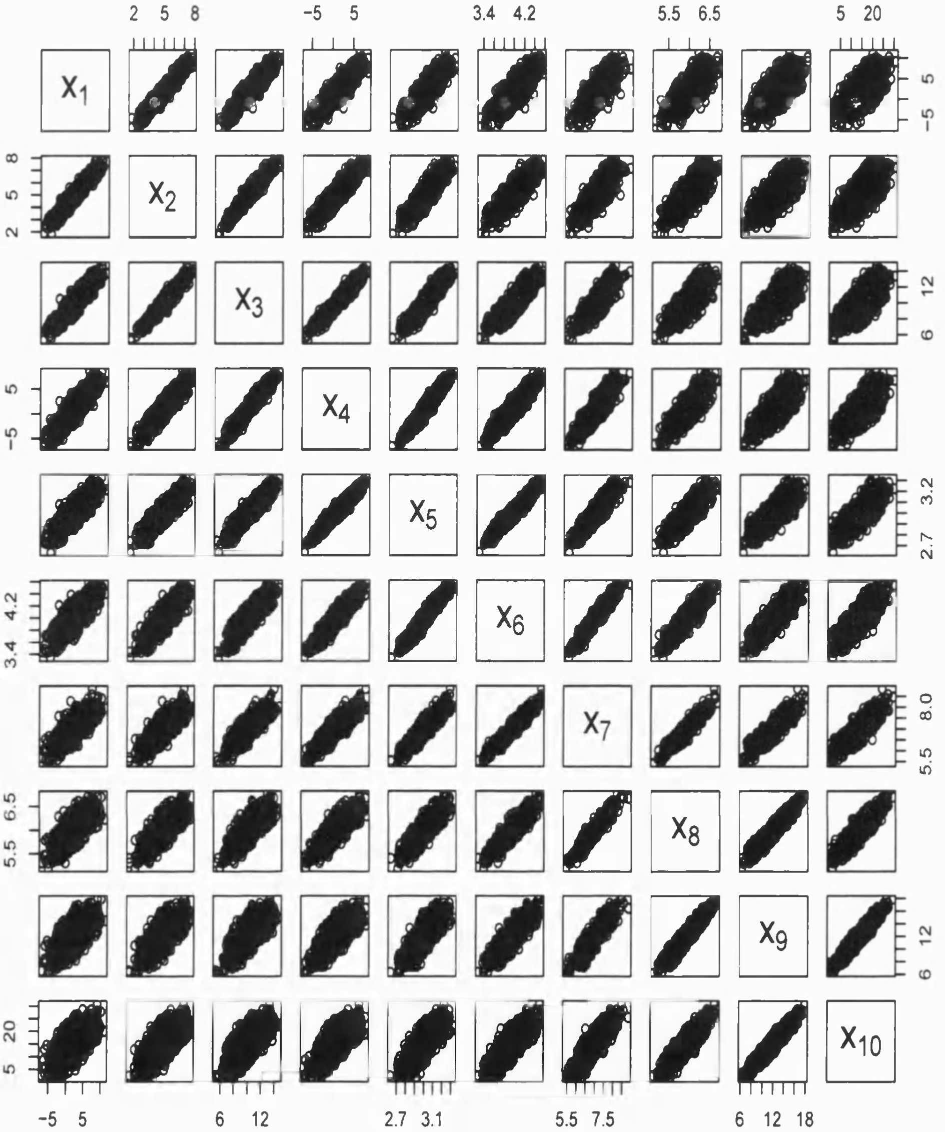


Figure 7-17: *Two-dimensional scatter plot of the highly correlated components of $h(X)$ obtained from a small set of estimated values by the standard MCMC sampler.*

Table 7.2: *The output obtained from approximately sampling $h(X)$ by the standard Metropolis approach MCMC and the PCMCMC approach. Each $\hat{\tau}_i$ is the estimated integrated autocorrelation for X_i.*

Var	TRUE	Estimate of $E(X_i)$ from MCMC	Estimate of $E(X_i)$ from PCMCMC	$\hat{\tau}_i$ for MCMC	$\hat{\tau}_i$ for PCMCMC
X_1	2	2.12	1.99	673	4.33
X_2	5	5.04	5.00	733	4.37
X_3	10	10.1	10.00	778	4.36
X_4	1	1.2	1.00	811	4.38
X_5	3	3.01	2.99	828	4.46
X_6	4	4.02	3.99	836	4.41
X_7	7	7.04	6.99	817	4.44
X_8	6	6.02	5.99	790	4.47
X_9	12	12.17	11.98	745	4.49
X_{10}	15	15.4	14.99	684	4.43

The standard deviations of the proposal distributions are set to $2.4\sqrt{\hat{\lambda}_i}$, i.e. we use the optimal value of s, obtained in the previous section, to generate scaled updates of the form given by

$$\tilde{X} \rightarrow \tilde{X} + Z_i\hat{e}_i, \qquad Z_i \sim N(0, 2.4^2\hat{\lambda}_i),$$

where $\hat{e}_i$ is the estimated eigenvector of the eigenvalue $\hat{\lambda}_i$, for $i = 1, \ldots, 10$. The acceptance rates associated with this run on all PC directions are between 44% and 45%. Figure 7-18 traces the resulting samples analogous to those of Figure 7-16. In contrast to the standard approach results, we again notice a very significant improvement in the mixing speed of the chains generated by the PCMCMC algorithm. Estimates of the integrated autocorrelation times obtained from both algorithms are reported in Table 7.2. The integrated autocorrelation time estimates of $E(X_i)$ in the latter table, reveal that the PCMCMC is mixing extremely well with integrated autocorrelation times of about 4.4. This shows that even for multivariate normal distributions with highly correlated variables, optimality is almost accomplished with the same proposal step sizes and acceptance rates obtained for the 2-dimensional problem. The small differences are due to the size of the sample set used. Accordingly, very accurate expectations are produced as displayed in Table 7.2.

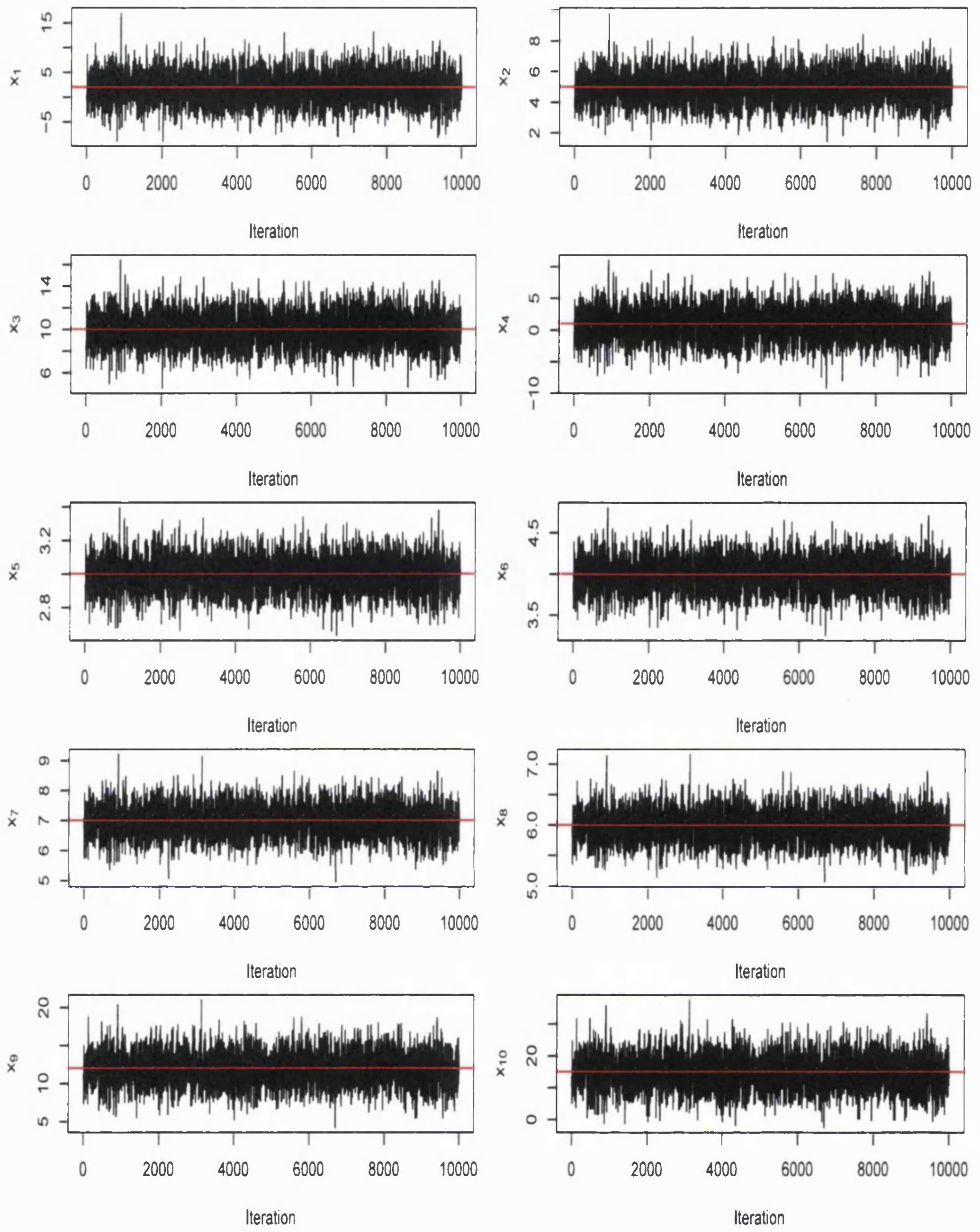


Figure 7-18: *Time-series plots of the 10-dimensional problem obtained by implementing the PCMCMC sampler. Red lines represent the corresponding parameters' true mean values.*

7.5.2 Example 2

As a final illustration in this chapter, we will implement the PCMCMC algorithm for sampling unimodal two-dimensional distributions that are not exactly Gaussian and compare the results obtained from this method with those of the standard Metropolis approach.

To construct such distributions, consider the bivariate normal density of section 7.1, namely $f(X)$ with $\mu = (5, 2)^T$ and variance matrix

$$\Sigma = \begin{pmatrix} 2.0^2 & 0.396 \\ 0.396 & 0.2^2 \end{pmatrix}.$$

Further, let $Z = (Z_1, Z_2)^T$ such that

$$Z_1 = X_1 + a(X_2 - \mu_2)^2, \tag{7.11}$$

and

$$Z_2 = X_2. \tag{7.12}$$

The Jacobian factor of this transition for any value of a is exactly one, and therefore

$$f_Z(z) = f_X(x), \tag{7.13}$$

when z and x are related by Eqs (7.11) and (7.12). That is, $f(z_1, z_2)$ is given by

$$\frac{1}{2\pi|\Sigma|^{\frac{1}{2}}} \times \\ \mathrm{e}^{-\frac{1}{2}\left(\left(z_1-a(z_2-\mu_2)^2-\mu_1\right)^2\Sigma_{11}^{-1}+2\left(z_1-a(z_2-\mu_2)^2-\mu_1\right)(z_2-\mu_2)\Sigma_{12}^{-1}+(z_2-\mu_2)^2\Sigma_{22}^{-1}\right)} \tag{7.14}$$

If $a = 0$, then $Z \sim N(\mu, \Sigma)$ otherwise the normality assumption is disrupted. Obviously, extent of departure from normality is proportional to the absolute value of a as the contour plots of Figure 7-19 reveal. We are interested in simulating all four distributions plotted in this figure using both approaches, the standard Metropolis and the PCMCMC.

The initial state chosen to run the standard MCMC algorithm is $(6, 1)^T$. For this sampler, we used Gaussian proposals with standard deviations 1.0 for Z_1 and 0.1 for Z_2. With these steps, almost 33% of the proposed values of Z_1 are accepted for all

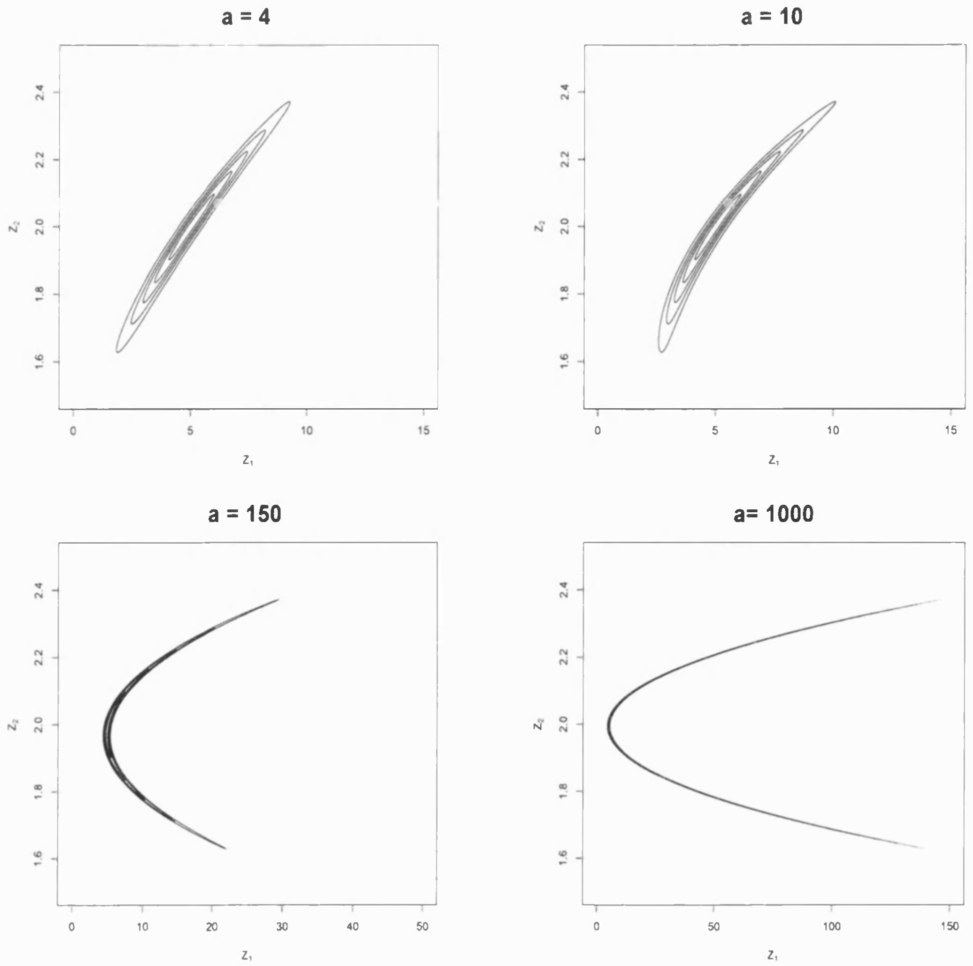


Figure 7-19: *Contours of $f_Z(z)$ with the indicated a values.*

values of a, whereas for Z_2 the acceptance rates are 34%, 36%, 11% and 4% when a is 4, 10, 150 and 1000, respectively. Each run consists of 200,000 sweeps.

Figure 7-20 reveals chains generated by this sampler for the four values of a. Scatter plots of Z_1 and Z_2 obtained from a sample of size 10000 derived from this run by thinning the chains beyond iteration 50,000 are displayed in Figure 7-21. The full run scatter plots are largely the same. The associated acceptance rates on both directions of Z_1 and Z_2, their empirical averages, the corresponding parameters' standard deviations and integrated autocorrelation times are reported in Table 7.3.

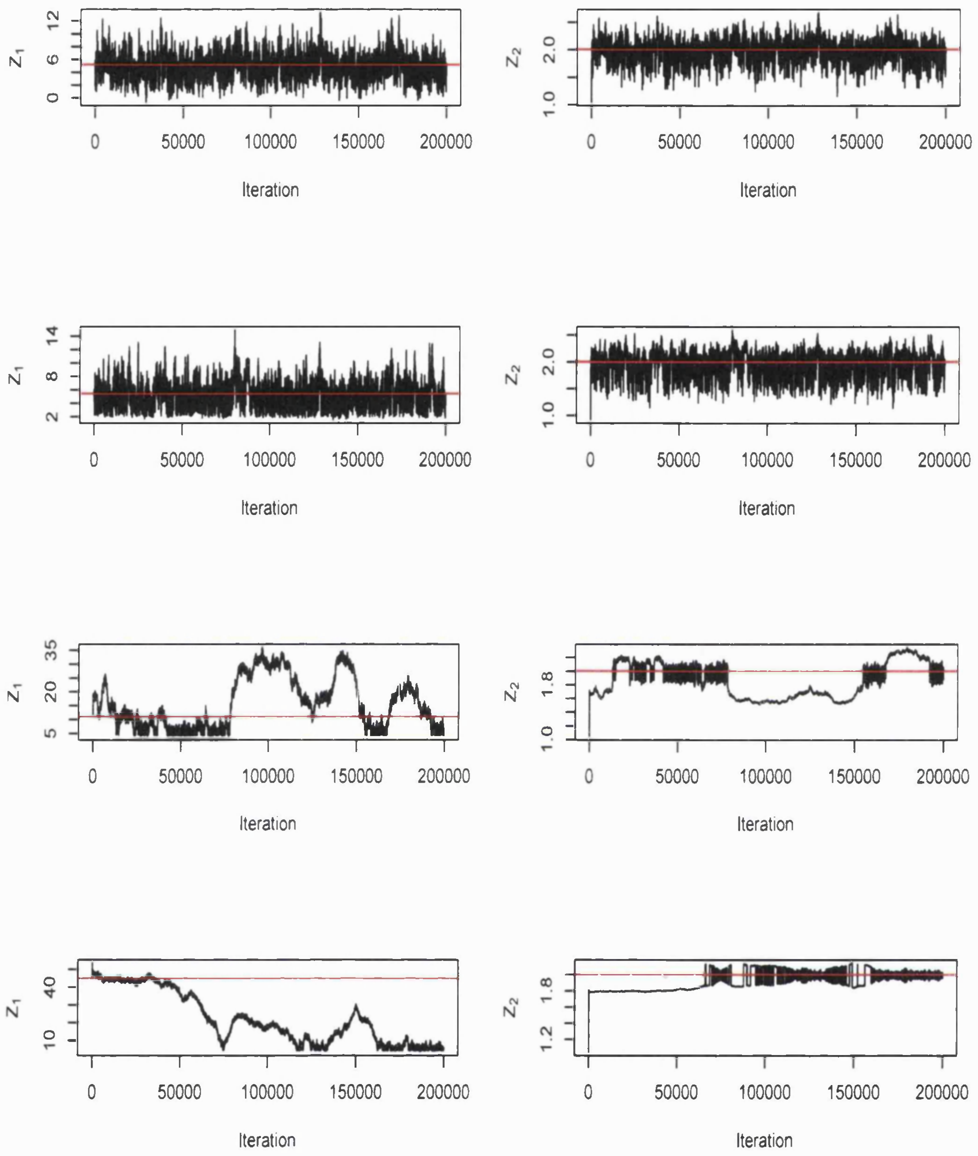


Figure 7-20: *Time-series plots obtained from the standard Metropolis run with* $a = 4$ *(first row),* $a = 10$ *(second row),* $a = 150$ *(third row) and* $a = 1000$ *(last row). The red lines represent the true mean values which are respectively: (5.16,2), (5.4,2), (11,2) and (45,2).*

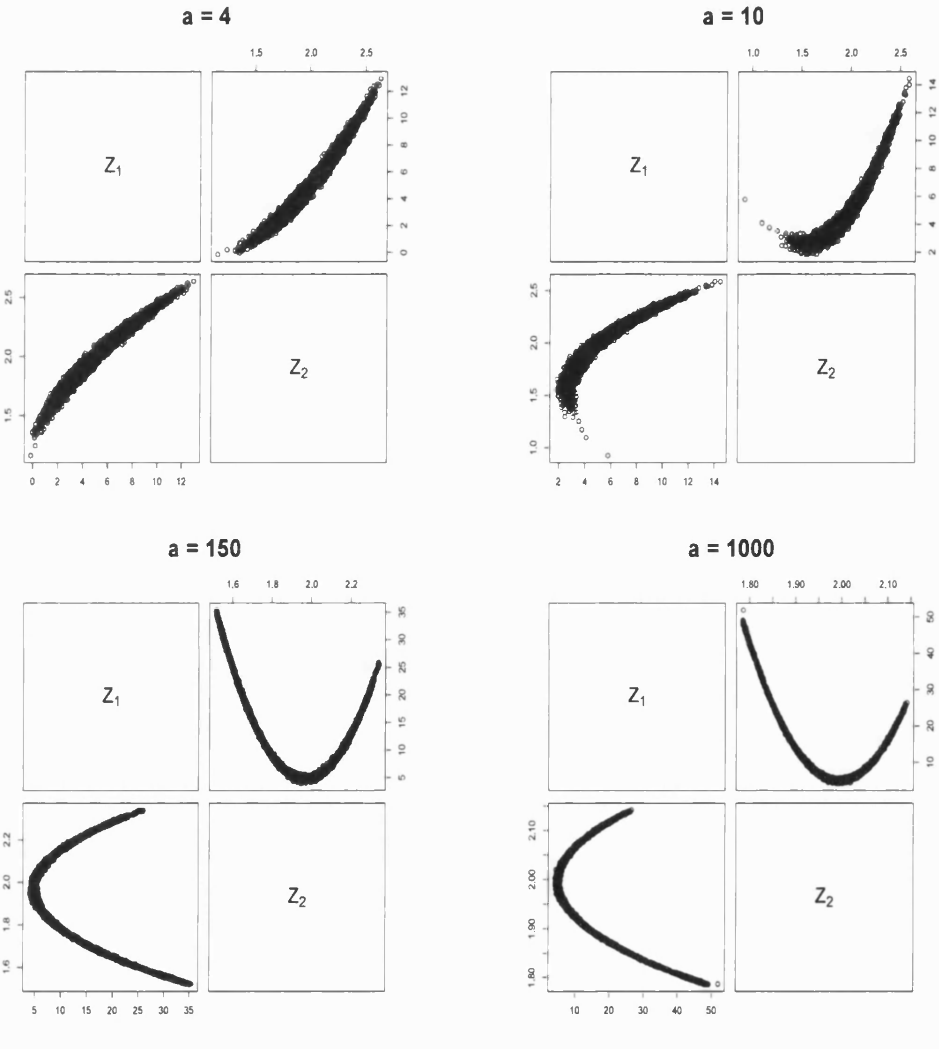


Figure 7-21: *The scatter plots obtained from a sample generated by the standard Metropolis algorithm for the specified values of a.*

Table 7.3: *Results obtained for simulating the four non-Gaussian distributions using the standard Metropolis* MCMC *and the principal components* PCMCMC *algorithms.* Y_1 *and* Y_2 *denote the associated PCs.*

$a = 4$	MCMC	PCMCMC
Acceptance rate	33%, 34%	Y_1 : 41%, Y_2 : 38%
Sample average of X_1, X_2	(5.043, 1.989)	(5.206, 1.989)
Sample standard deviation of X_1, X_2	(1.929, 0.193)	(2.007, 0.199)
Integrated autocorrelation time	(1222, 1158)	(7.9, 7.6)

$a = 10$	MCMC	PCMCMC
Acceptance rate	33%, 36%	Y_1 : 30%, Y_2 : 25%
Sample average of X_1, X_2	(5.300, 1.993)	(5.689, 1.994)
Sample standard deviation of X_1, X_2	(1.919, 0.193)	(2.09, 0.195)
Integrated autocorrelation time	(724, 563)	(52, 34)

$a = 150$	MCMC	PCMCMC
Acceptance rate	33%, 11%	Y_1 : 3%, Y_2 : 7%
Sample average of X_1, X_2	(17.465, 1.831)	(22.582, 1.814)
Sample standard deviation of X_1, X_2	(9.455, 0.256)	(6.774, 0.187)
Integrated autocorrelation time	(16136, 29295)	(574, 284)

$a = 1000$	MCMC	PCMCMC
Acceptance rate	33%, 4%	Y_1 : 2%, Y_2 : 4%
Sample average of X_1, X_2	(15.0256, 1.967)	(36.412,1.9421)
Sample standard deviation of X_1, X_2	(8.628, 0.0963)	(20.782, 0.145)
Integrated autocorrelation time	(16585, 3998)	(5105, 1667)

The variance matrices corresponding to the four distinct distributions are all unknown, therefore these matrices are estimated from the matching standard run output.

In all four cases, the standarised version of the PCMCMC sampler is implemented with starting points obtained from the standard MCMC results. For the new runs, the estimated eigenvectors and eigenvalues displayed in Table 7.4 are considered. In each case, we used variance $2.4^2\hat{\lambda}_i$ in the proposal distribution. The acceptance rates on each principal component direction and all other analogous results to those of the standard approach are summarised in Table 7.3, and the parallel time-series and scatter plots are shown in Figures 7-22 and 7-23, respectively.

The standard Metropolis first three scatter plots displayed in Figure 7-21 reveal that this algorithm has captured the general shape of these densities. However, we can see from some associated time-series plots of Figure 7-20 and the integrated autocorrelation

Table 7.4: *The approximate eigenvectors and eigenvalues estimated (Est.) from the standard MCMC algorithm for the standarised variables of the four non-gaussian distributions.*

Est.	$a = 4$	$a = 10$	$a = 150$	$a = 1000$
$\hat{e}_1$	(0.707,0.707)	(0.707,0.707)	(0.707,-0.707)	(0.707,0.707)
$\hat{e}_2$	(-0.707,0.707)	(-0.707,0.707)	(0.707,0.707)	(-0.707,0.707)
$\hat{\lambda}_1$	1.983	1.952	1.610	1.453
$\hat{\lambda}_2$	0.0170	0.0476	0.390	0.547

time values in Table 7.3 that the generated chains are not mixing rapidly. When $a = 150$, the problem of mixing is more obvious because a move between the two ends of the target density is more difficult since it requires many small steps within approximately the "U" shaped distribution. Therefore, the integrated autocorrelation times had significantly increased.

The last case in which a is increased to 1000, the corresponding time-series and scatter plots reveal that the Markov chain has got trapped in the high concentrated region around $(5, 2)$. Therefore, it did not visit the tails of this distribution shown in the corresponding plot of Figure 7-19. This is because the chain can only move slowly within the highly curved shape of this density. As a result, the acceptance rate on Z_2 axis, in particular, is very small.

In contrast, the PCMCMC results indicate that this approach is performing much better than the Metropolis standard algorithm for exploring the non-Gaussian distributions of the form given by Eq (7.14). However, this algorithm is most efficient for relatively small values of a, as the cases with $a \leq 10$.

Despite that the optimal proposal standard deviations obtained for the exact normal distributions are used for the approximately Gaussian density with $a = 4$, the associated acceptance rates and the efficiency estimates are not the same as the optimal ones derived for the perfect Gaussian distributions.

For higher amounts of perturbation, e.g., $a = 150$, benefits obtained from the PCA approach are limited because the algorithm is less efficient as the high values of the integrated autocorrelation times reflect.

The inaccurate empirical averages and the low overall acceptance rates are other indications. This is an expected result because such distributions are bent at the edges, and as a increases the normality shape vanishes, especially when high correlations are present, making moves in the estimated directions insufficient, and more precisely, inappropriate because these are not the directions on which updates should be made.

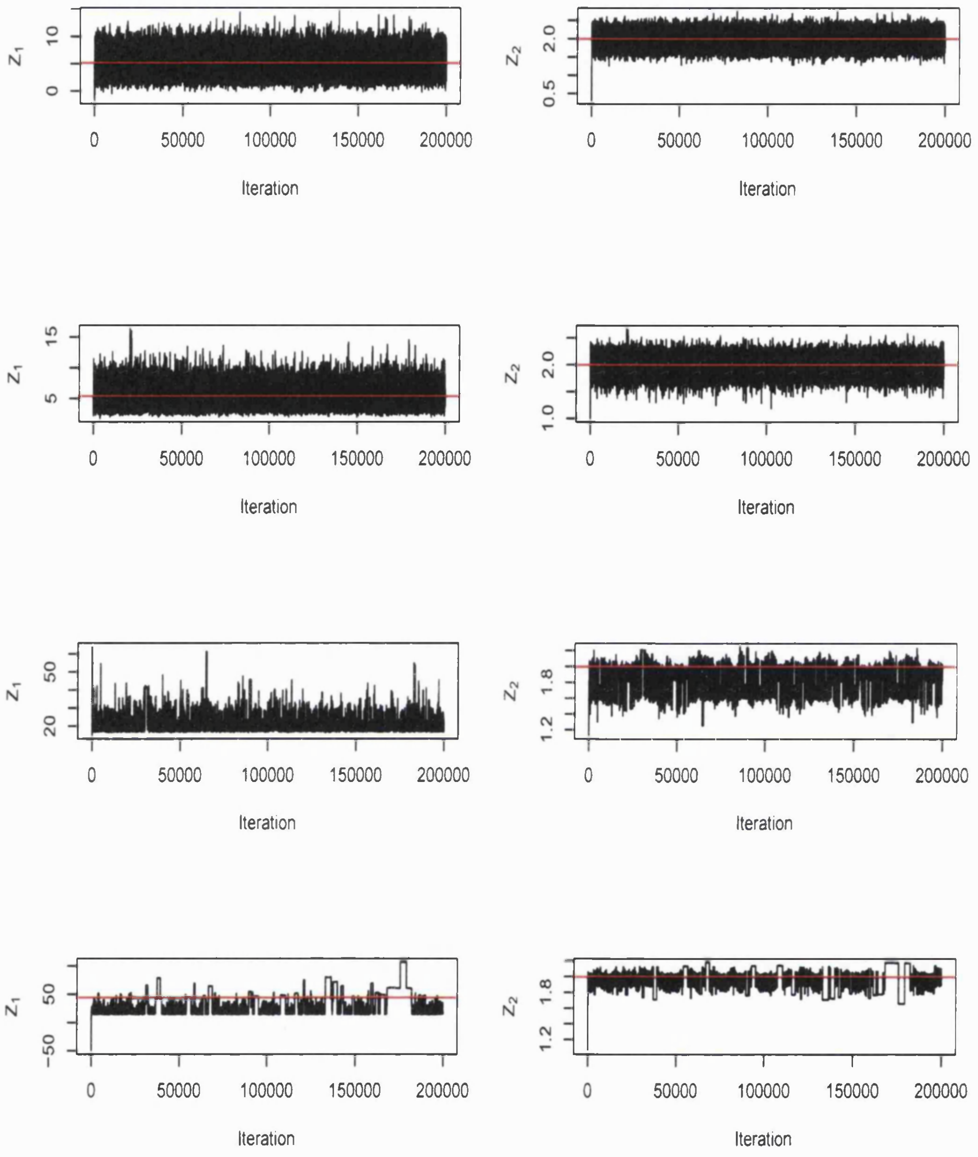


Figure 7-22: *Time-series plots obtained from the PCMCMC run with $a = 4$ (first row), $a = 10$ (second row), $a = 150$ (third row) and $a = 1000$ (last row). The red lines represent the true mean values which are respectively: (5.16,2), (5.4,2), (11,2) and (45,2).*

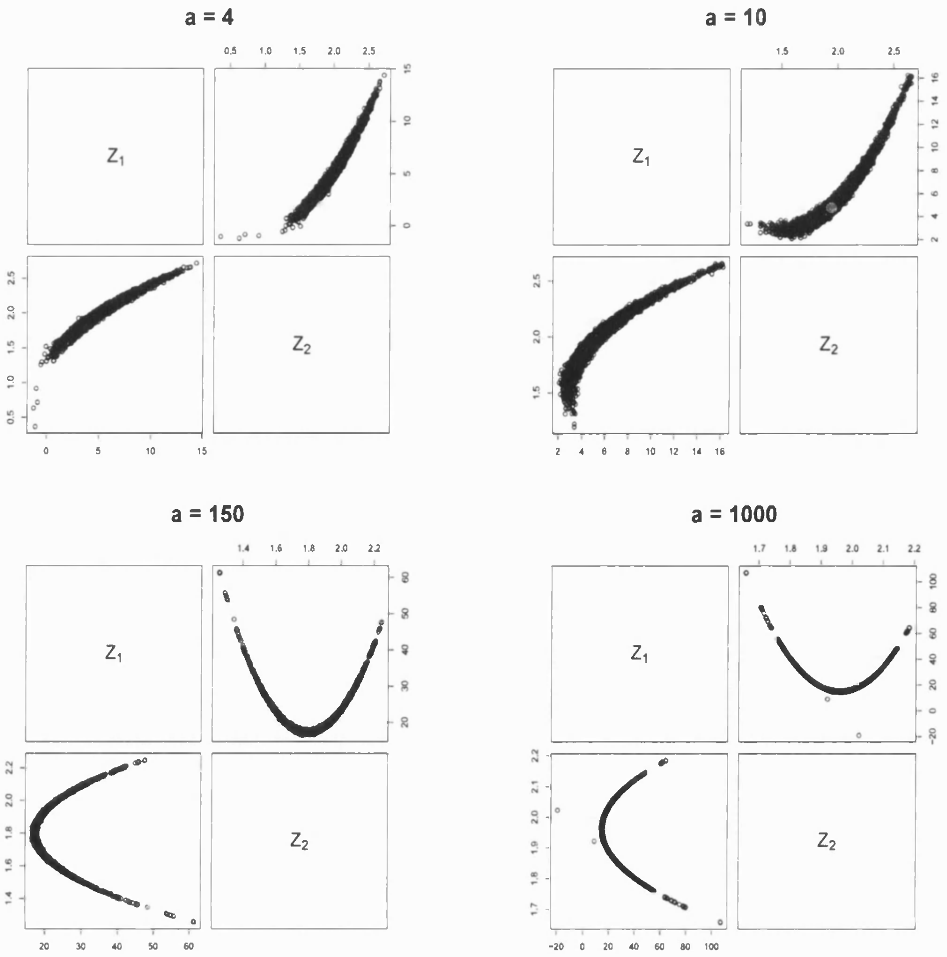


Figure 7-23: *The scatter plots obtained from a sample generated by the PCMCMC algorithm for the specified values of a.*

7.6 Conclusion

In this chapter, we have introduced the principal components MCMC algorithm PCMCMC; an efficient method that can be used to sample unimodal Gaussian or approximately Gaussian distributions with high correlation structures, by generating appropriate sized moves in the approximate directions of the target density principal components. To implement this method, it is necessary to have either the true covariance matrix or alternatively an estimate of it. If this matrix is unknown then a pilot sample roughly from the distribution of interest can be generated by the standard

Metropolis sampler and then used to estimate the required matrix. Moreover, the resulting empirical expectations can be used to initialise the PCMCMC run in order to shorten the burn-in period.

In reality, it is quite common that parameters of the posterior distribution have very different variances. In this case, the principal components with large eigenvalues will be dominated by variables with the highest standard deviations even if all variables of interest are highly correlated. To make all components equally important and comparable it is suggested to scale all original variables so that each has a variance of one.

It has been shown through the examples considered for normal target densities that this algorithm is most efficient if the Gaussian proposal distributions have variances of size 2.4^2 times the principal components' variances. The optimal acceptance rate on each principal direction is around 44% and the integrated autocorrelation time is approximately 4.4. Nevertheless, when the target density is not perfectly Gaussian the same variances give slightly lower acceptance rates which are around 40% and higher integrated autocorrelation times approximately within the range 7 - 8. This suggests that even when the distribution of interest departs quite a lot from Gaussian then this sampler may still do well. For example, in 1D, we know that the MCMC can perform well for lots of distributions. Hence, the PCMCMC would be good too, as long as directions found from the variance matrices' eigenvectors make an appropriate independent decomposition. In this case, a major advantage of this sampler is its high efficiency. This makes it a promising tool for simulating many higher-dimensional models in which high correlation structures are also expected.

Chapter 8

Implementing the PCMCMC for ionospheric tomography

8.1 Introduction

In the previous chapter, we have described the PCMCMC algorithm developed in this thesis and tested its performance for sampling Gaussian and non-Gaussian unimodal distributions with highly correlated variables. When these distributions are Gaussians or approximately Gaussians the algorithm can operate very efficiently and produce accurate results. Even when the simulated density is not normally shaped this algorithm may be abe to explore wider regions of the sampling space with a greater speed of mixing than the standard Metropolis sampler.

In this chapter, we shall return to the problem posed in Chapter 4 and implement the PCMCMC for sampling the electron density posterior distribution, $\pi(X|Y)$, displayed in Eq (4.16). As a preliminary testing of the method for the tomographic problem a set of simulated data is initially used as presented below.

8.2 PCMCMC application to simulated data

Here we apply the PCMCMC and standard MCMC algorithms to sample from the posterior distribution $\pi(X|Y)$. In this example, the synthetic ionospheric plane is represented with a 10×7 grid of pixels. Each column in this grid is assumed to contain one vertical profile. As mentioned earlier in Chapter 4, the nth vertical profile's parameters, namely the peak height, spread and electron density under the profile curve are denoted by μ_n, σ_n^2 and γ_n, respectively. The seven synthetic vertical profiles' parameter values are given in Table 8.3.

The grid is intersected by sixty five rays for which the artificial data are simulated. The derivation of these data proceeds by adding a Gaussian noise, e_i, with zero mean and variance, $\epsilon^2 = 1/2000$, to the expected value, $E(Y_i)$, calculated for each observation. The whole synthetic configuration is displayed in Figure 8-1.

The smoothing parameters involved in the prior model, $\beta_\mu, \beta_{\sigma^2}$ and β_γ, are assigned the values 0.003, 0.004 and 0.008, respectively. The constraints associated with this model are:

$$0 < \mu_n \leq 10, \tag{8.1}$$

$$0 < \sigma_n^2 \leq 10, \tag{8.2}$$

$$0 < \gamma_n \leq 100, \tag{8.3}$$

for $n = 1, \ldots, 7$. An upper limit of 10 has been placed for the peak height because μ_n cannot be any larger than the altitude of the grid. The remaining two upper limits are selected to be big enough so that all restorations are allowed to be dominated by the data rather than the weak prior.

The standard MCMC is run for 100,000 iterations starting from the true parameter values. The proposal distributions' standard deviations are selected such that acceptance rates of about 25% are obtained. The resulting traces are displayed in Figures 8-2 - 8-4. Here no attempt is made to exclude a burn-in period since the algorithm is started at the true values. Table 8.1 shows the integrated autocorrelation time estimates for the original parameters. Most of these estimates are very high indicating slow mixing, as the associated traces also reveal.

The strength of dependency between certain pairs of these variables can be visualised through the scatter plots in Figures 8-5 - 8-7. Obviously, some of these plots reveal high correlations between parameters of the same type, especially those of the peak heights and the spreads where most scatterplots are confined to a narrow region. However, the $\{\gamma_n\}$ plots are much wider indicating that these parameters are less correlated. The integrated autocorrelation times in Table 8.1 indicate that the γ_n sequences mix better although some of these values are still quite high. One possible reason behind this is that a greater degree of smoothing is applied to the γ variables by the assigned value of β_γ (relative to the scale of the γ values).

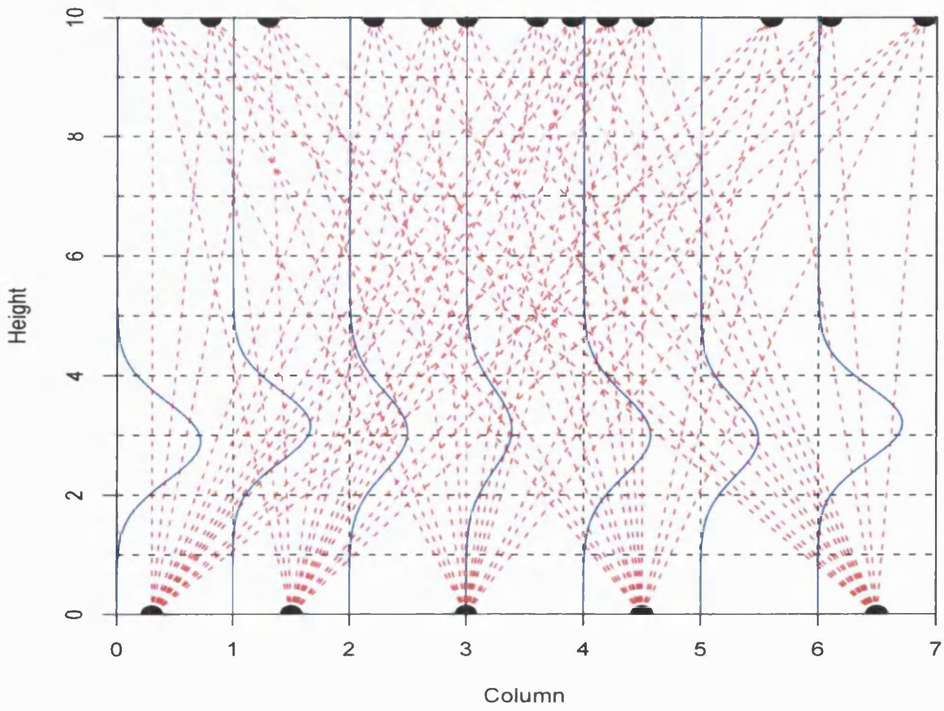


Figure 8-1: *The 2D grid showing the satellite-to-receiver rays (red dotted lines) for the simulated data inversion problem. The blue curves represent the true vertical electron-density profiles.*

Now, we will apply the PCMCMC approach. For this purpose, the variance matrix of the posterior distribution π is estimated from the standard MCMC output. All parameters are then scaled and the corresponding variance structure is consequently estimated to run the PCMCMC. The resulting eigenvalues are between 7.88 and 0.0013.

In an attempt to get the benefit from the PCA approach, we used $s = 2.4$ in Equation (7.2), as recommended in section 7.4, to generate 100,000 cycles of this algorithm. The same initial state used for the standard approach was used to run the PCMCMC sampler. The acceptance rates on the first two principal components' directions are approximately 50% but on the remaining directions they are around 44%. Figures 8-8 to 8-10 show the resulting traces of the μ_n, σ_n^2 and γ_n, respectively. Note that these figures are converted back to the original set of the unknown variables. The integrated autocorrelation times produced by this run are displayed in Table 8.2.

Table 8.1: *Simulated data example. Integrated autocorrelation time estimates obtained from the standard MCMC algorithm.*

Vertical profile number n	$\hat{\tau}(\mu_n)$	$\hat{\tau}(\sigma_n^2)$	$\hat{\tau}(\gamma_n)$
1	9104	2170	16
2	9103	8002	929
3	9280	4491	112
4	8539	6732	23
5	8729	3574	57
6	8793	1655	202
7	8361	8418	10

Table 8.2: *Simulated data example. Integrated autocorrelation time estimates obtained from the PCMCMC algorithm.*

Vertical profile number n	$\hat{\tau}(\mu_n)$	$\hat{\tau}(\sigma_n^2)$	$\hat{\tau}(\gamma_n)$
1	8.7	7.5	5.2
2	8.8	5.0	4.9
3	8.7	5.6	5.5
4	9.5	5.0	5.2
5	9.5	5.2	4.8
6	9.5	5.7	5.1
7	9.5	6.1	5.1

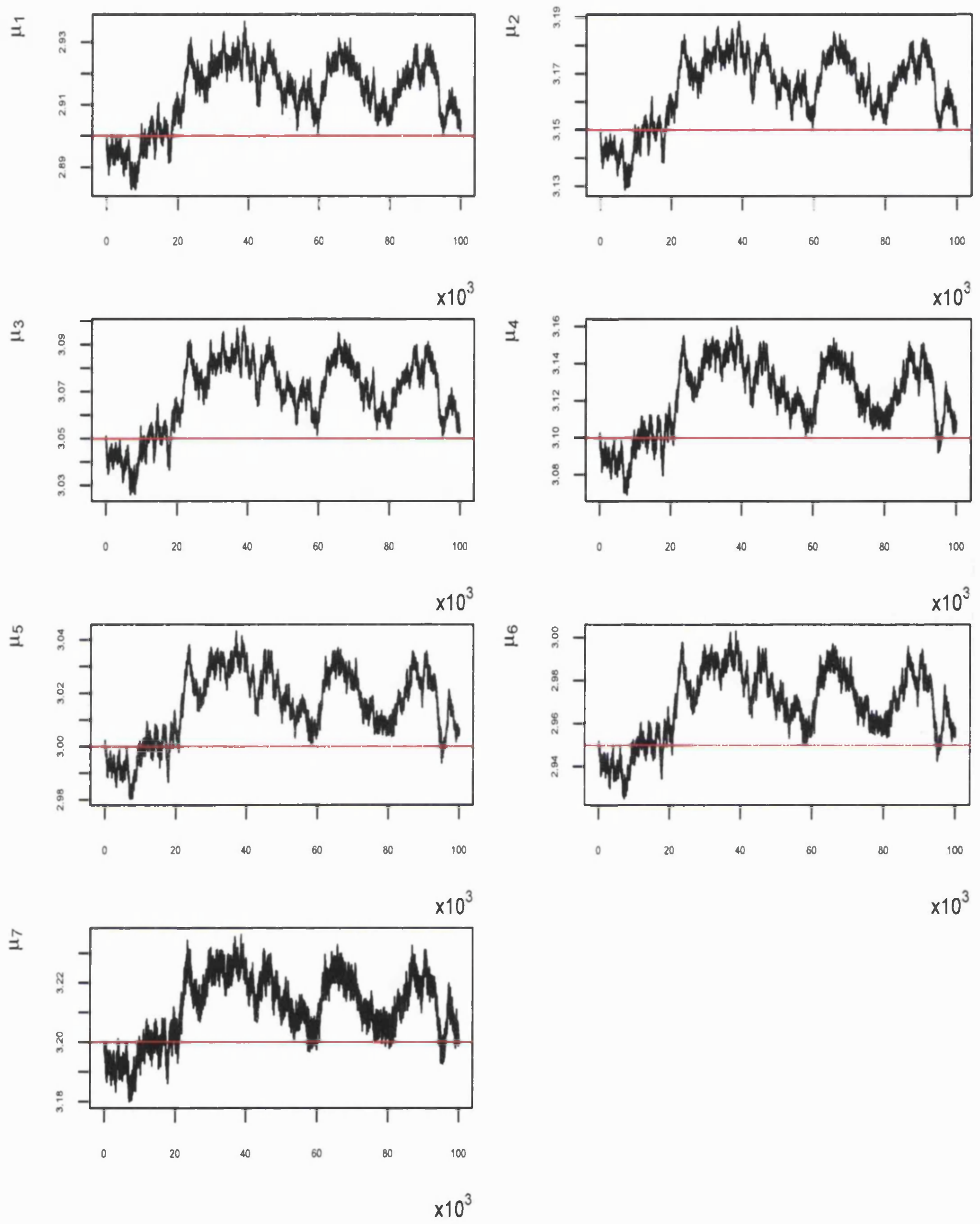


Figure 8-2: *Simulated data example. Trace plots of* $\mu_1, \ldots, \mu_7$ *produced by implementing the standard MCMC algorithm, starting from the true values of these variables (represented by the red lines).*

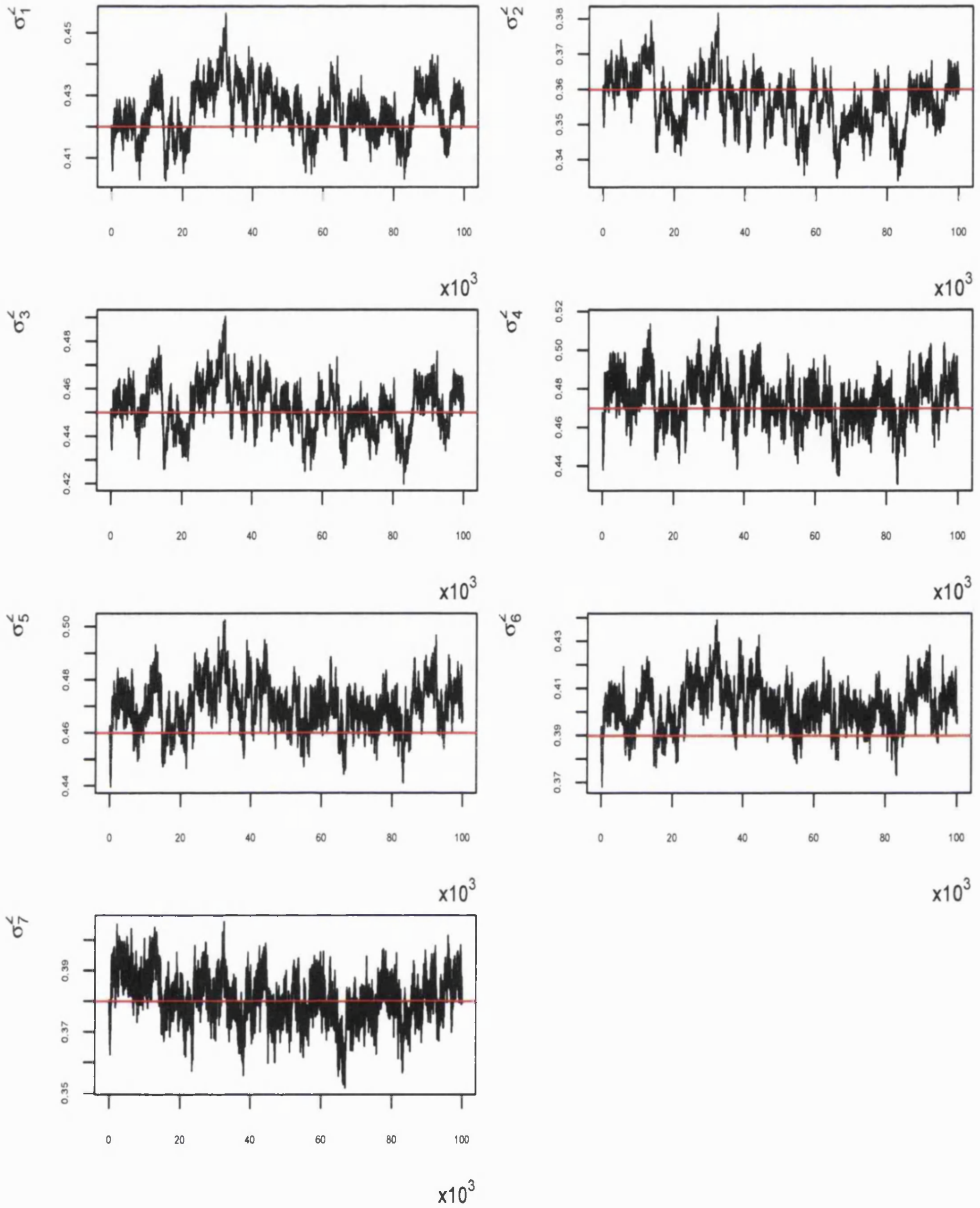


Figure 8-3: *Simulated data example. Trace plots of* $\sigma_1^2, \ldots, \sigma_7^2$ *produced by implementing the standard MCMC algorithm starting from the true values of these variables (represented by the red lines.)*

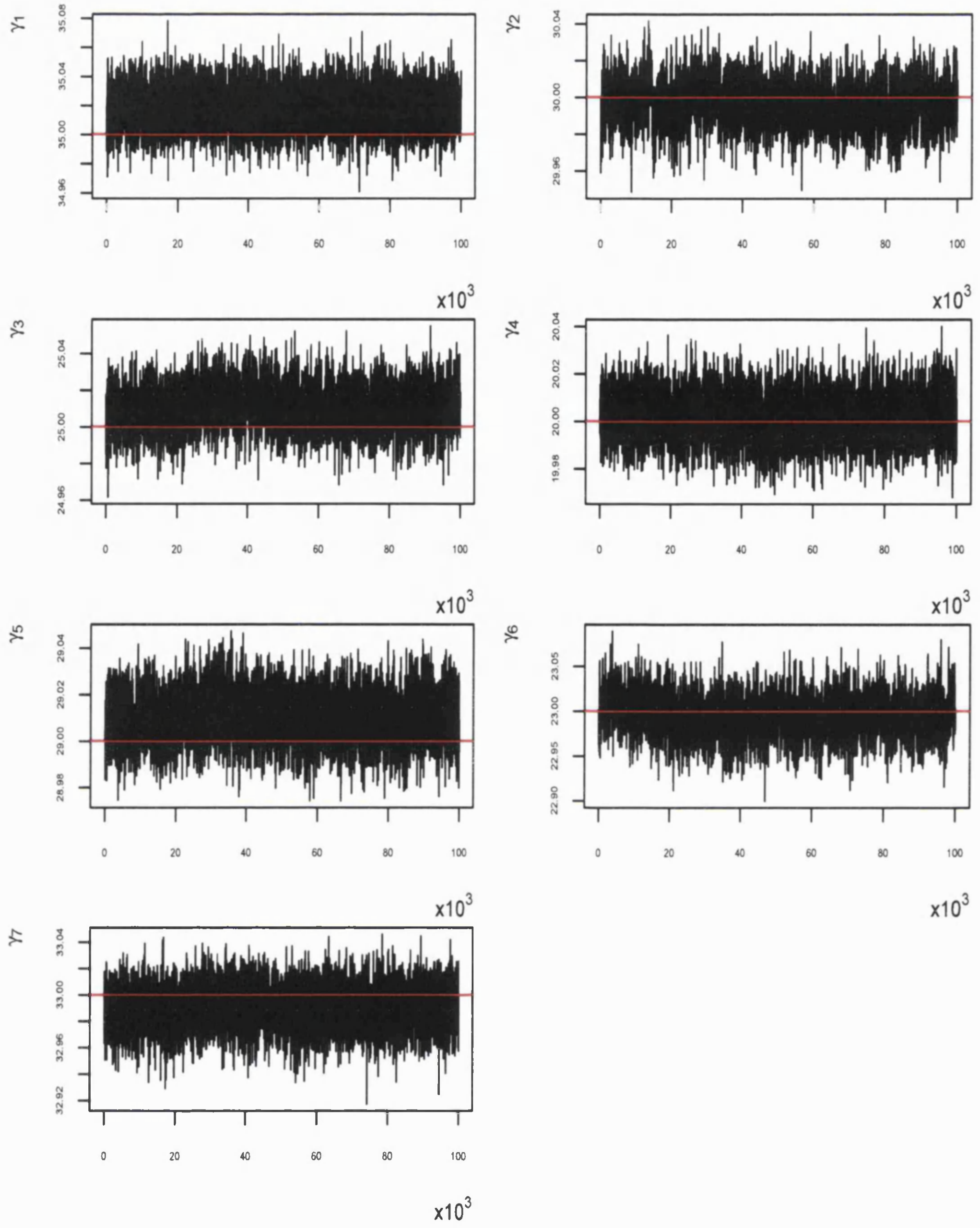


Figure 8-4: *Simulated data example. Trace plots of* $\gamma_1, \ldots, \gamma_7$ *produced by implementing the standard MCMC algorithm starting from the true values of these variables (represented by the red lines.)*

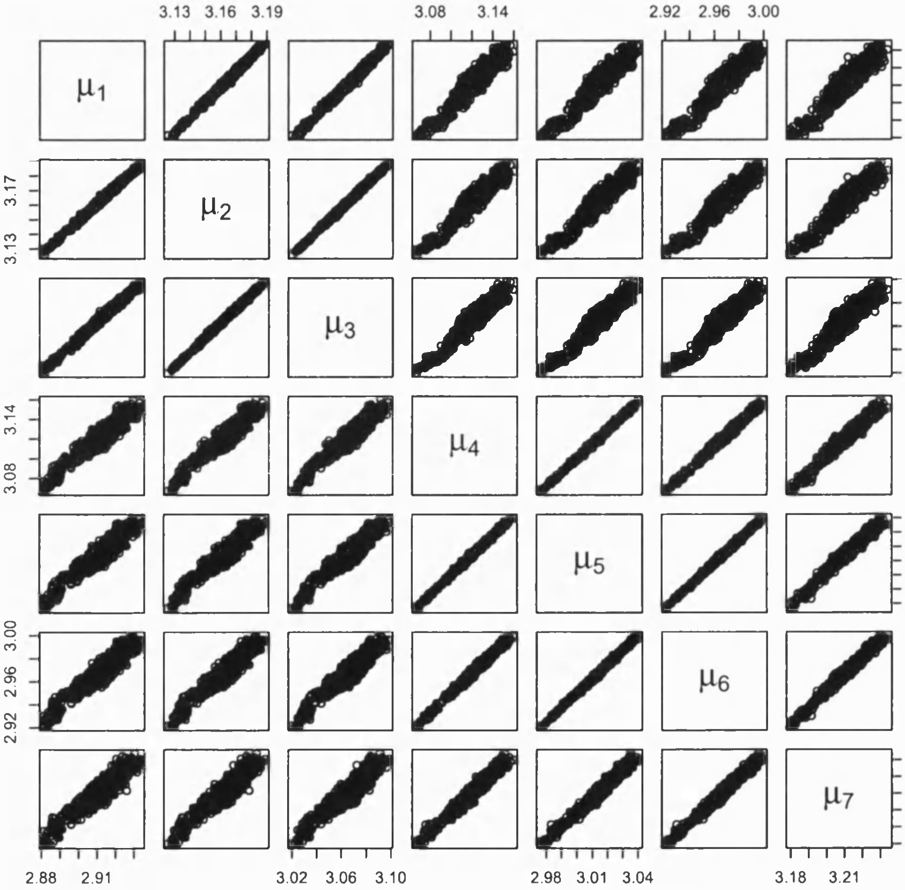


Figure 8-5: *Simulated data example. 2D scatter plots for $\mu_1, \ldots, \mu_7$ values obtained from the standard MCMC algorithm.*

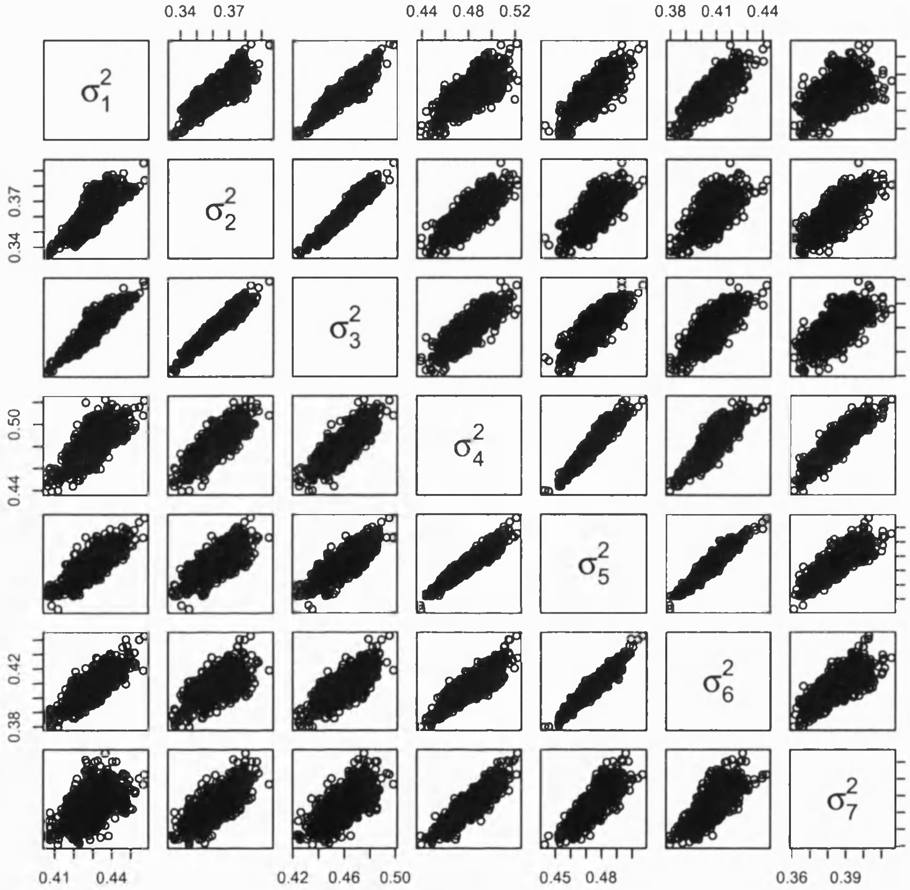


Figure 8-6: *Simulated data example. 2D scatter plots of $\sigma_1^2, \ldots, \sigma_7^2$ values obtained from the standard MCMC algorithm.*

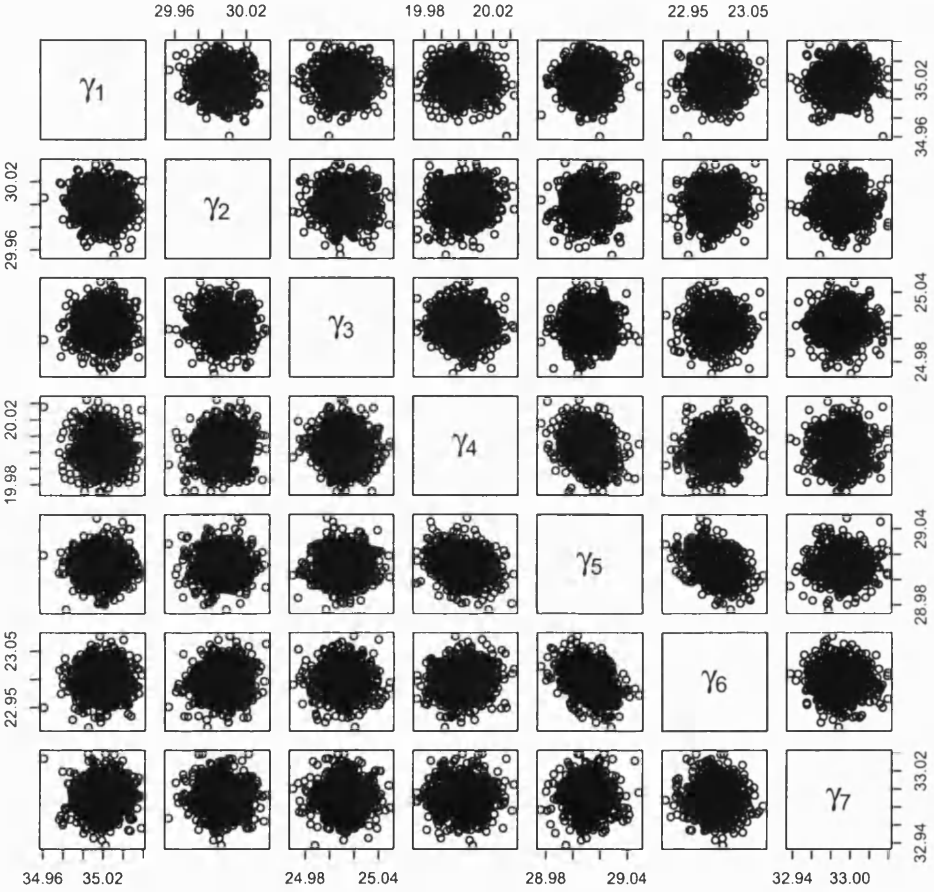


Figure 8-7: *Simulated data example. 2D scatter plots for* $\gamma_1, \ldots, \gamma_7$ *values obtained from the standard MCMC algorithm.*

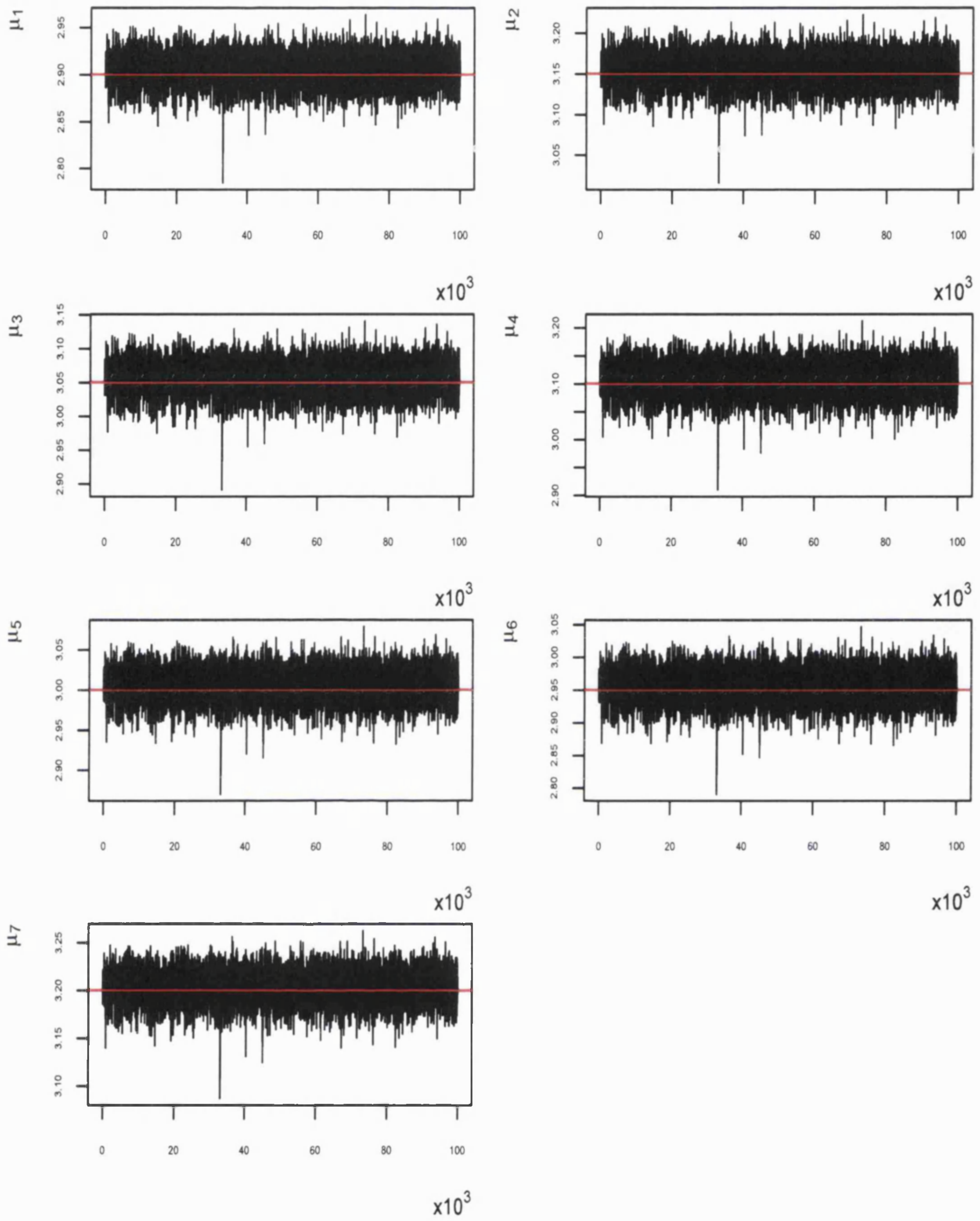


Figure 8-8: *Simulated data example. Trace plots of* $\mu_1, \ldots, \mu_7$ *produced by implementing the PCMCMC algorithm.*

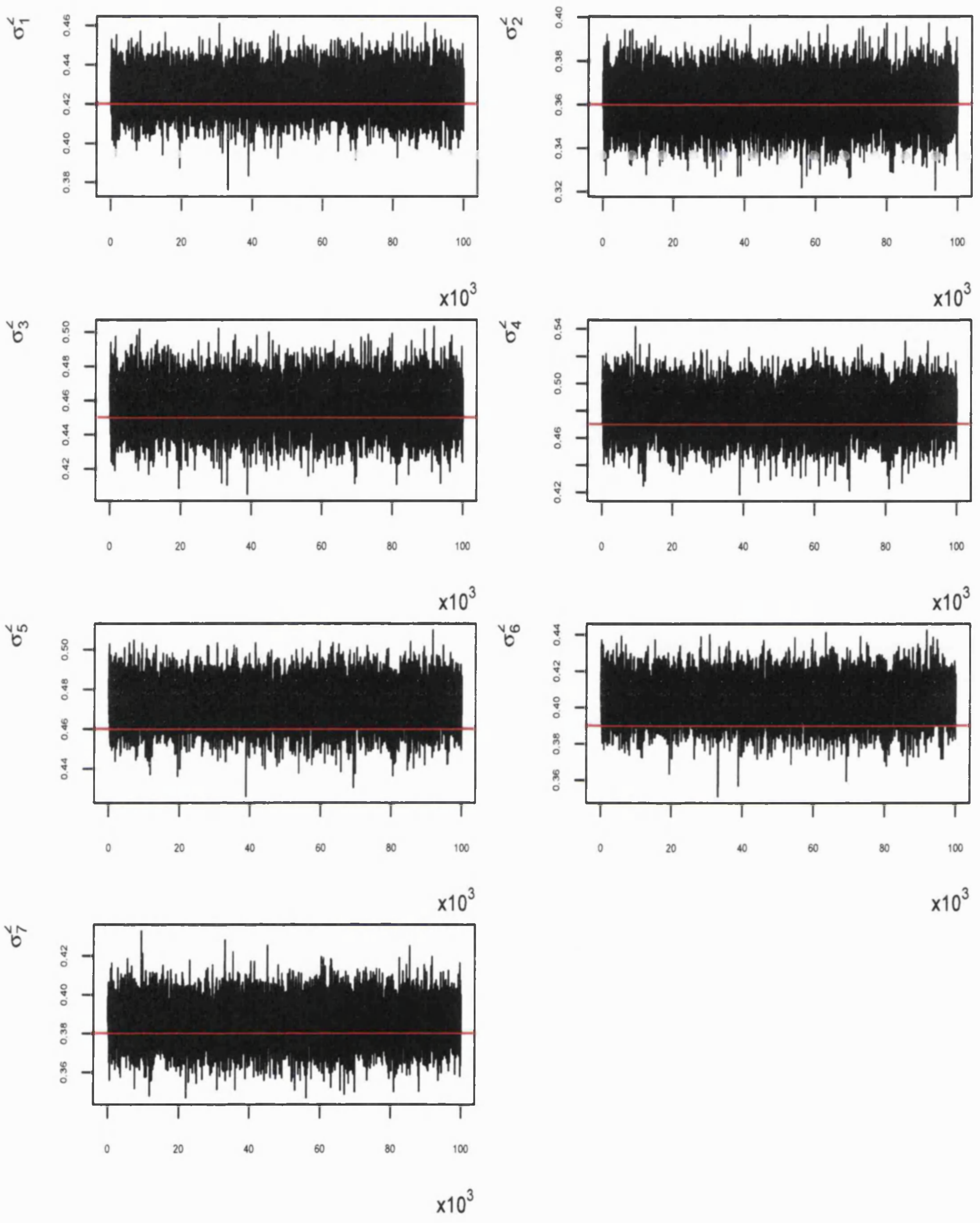


Figure 8-9: *Simulated data example. Trace plots of* $\sigma_1^2, \ldots, \sigma_7^2$ *produced by implementing the PCMCMC algorithm.*

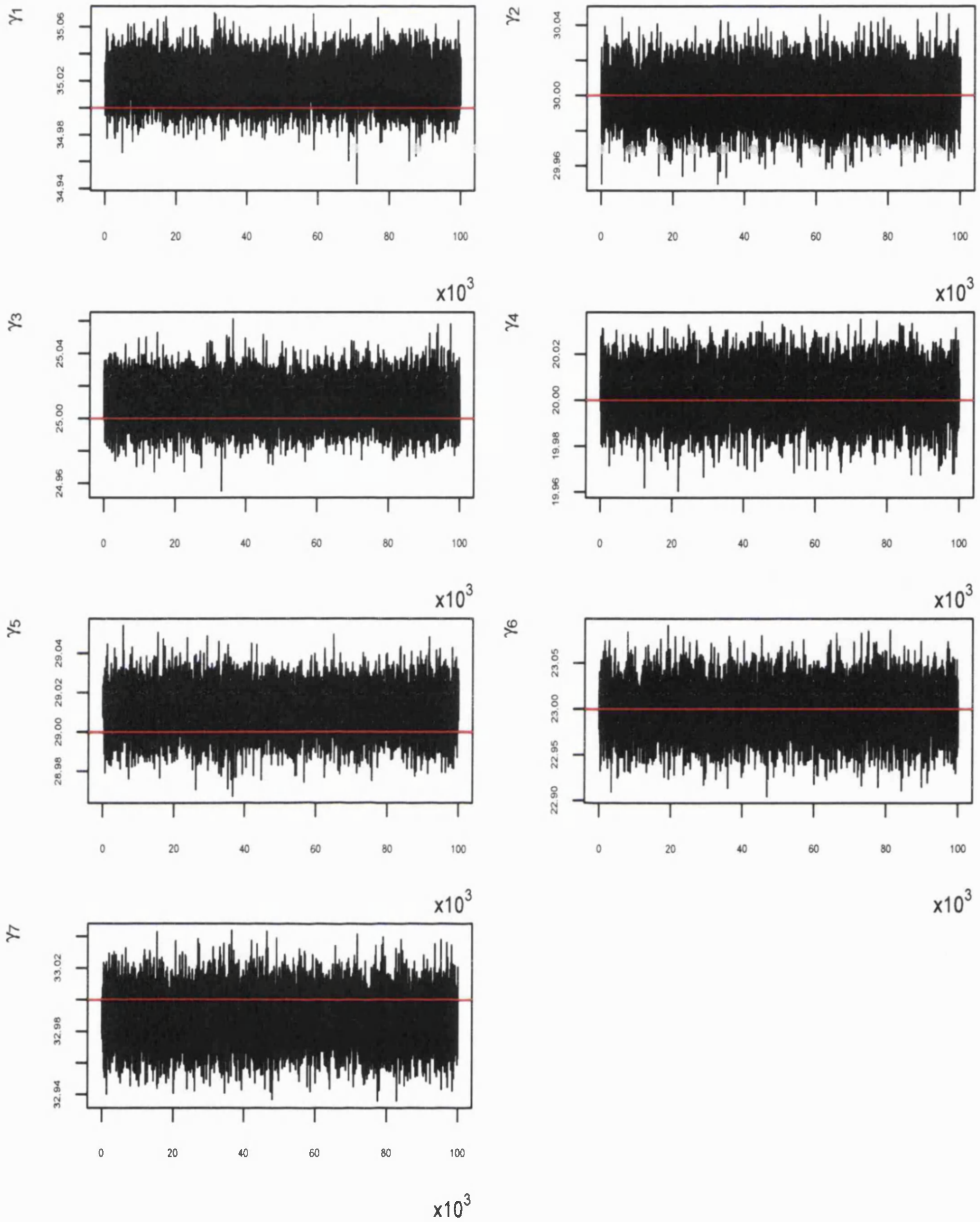


Figure 8-10: *Simulated data example. Trace plots of $\gamma_1, \ldots, \gamma_7$ produced by implementing the PCMCMC algorithm.*

Table 8.3: *The synthetic parameter values, the posterior expectations, medians and 99% credible intervals derived from the PCMCMC run.*

Variable	True value	mean	median	99% Cr.I.
μ_1	2.90	2.907	2.907	(2.887,2.926)
μ_2	3.15	3.157	3.158	(3.134,3.179)
μ_3	3.05	3.059	3.061	(3.031,3.087)
μ_4	3.10	3.113	3.114	(3.077,3.146)
μ_5	3.00	3.009	3.009	(2.984,3.032)
μ_6	2.95	2.961	2.962	(2.923,2.989)
μ_7	3.20	3.206	3.206	(3.185,3.225)
σ_1^2	0.42	0.427	0.426	(0.415,0.439)
σ_2^2	0.36	0.362	0.362	(0.349,0.375)
σ_3^2	0.45	0.458	0.457	(0.441,0.474)
σ_4^2	0.47	0.481	0.481	(0.462,0.499)
σ_5^2	0.46	0.473	0.473	(0.459,0.486)
σ_6^2	0.39	0.405	0.405	(0.389,0.418)
σ_7^2	0.38	0.386	0.386	(0.373,0.399)
γ_1	35.00	35.02	35.02	(35.00,35.03)
γ_2	30.00	30.00	30.00	(29.98,30.01)
γ_3	25.00	25.01	25.01	(24.99,25.03)
γ_4	20.00	20.00	20.00	(19.99,20.01)
γ_5	29.00	29.01	29.01	(28.99,29.02)
γ_6	23.00	22.99	22.99	(22.97,23.03)
γ_7	33.00	32.99	32.99	(32.97,33.01)

Comparing the plots of Figures 8-8 to 8-10 with the parallel plots of Figures 8-2 to 8-4 reveals a very big improvement in the speed of mixing. This improvement has been quantified by estimating the associated integrated autocorrelation times for each chain as reported in Table 8.2. All entries of this table if compared with the analogous values of Table 8.1 show that the PCMCMC dramatically outperforms the standard MCMC sampler. The new results on their own indicate that the principal components algorithm is very efficient, with all integrated autocorrelation time estimates less than ten. The values of $\hat{\tau}(\mu_n)$, $\hat{\tau}(\sigma_n^2)$ and $\hat{\tau}(\gamma_n)$ are not much higher than the $\hat{\tau}$s seen in the examples of chapter 7, suggesting the posterior distribution $\pi(X|Y)$ might be quite similar to a multivariate Gaussian distribution in 22 dimensions, with some high correlations. This assumption could be checked by plotting several sets of 3D scatter plots, for instance.

Table 8.3 shows the posterior means, medians and the 99% credible intervals obtained from the the last 50000 iterations of the PCMCMC run. The outstanding results derived for this high-dimensional simulated data problem, as can be seen from Tables 8.2 and 8.3 reveal that the PCMCMC is a promising tool for sampling from the electron density distribution with real TEC data.

8.3 PCMCMC application to TEC data

The results obtained from implementing the mixed algorithm to the actual TEC data discussed in chapter 6 reveal that this sampler suffers from very slow mixing. This suggests that there exist some high correlations between variables in the posterior distribution, $\pi(X|Y)$, displayed in Eq (4.16). One sensible set of values of the interaction parameters involved in our model are found to be those estimated from the central region of the ionospheric plane where many radio signals passed, namely the MPL fixed estimates. Therefore, to obtain a rough estimate of the unknown correlation structure of all 67 variables of interest, the results obtained from the MPL fixed estimates run given in section 6.4 are used. Some entries of the estimated correlation matrix are shown in Table 8.4.

The estimated correlation matrix obtained from the mixed algorithm run confirms that many components of the electron density distribution, $\pi(X|Y)$, are highly correlated. The high correlations are not only seen between parameters of the same type but also between different parameters, for example, μ_{11} to μ_{14} have high correlations with σ_{11}^2 to σ_{14}^2. Pairwise relations between parameters belonging to the same category are viewed via constructing 2D scatter plots of the generated samples. Some of these plots are illustrated in Figures 8-11 - 8-13. The high correlations present suggest that we may benefit from the principal components MCMC approach in reducing the effect of these correlations and consequently sample the desired distribution more efficiently.

The mixed algorithm results gave some very large estimates such as those of the spreads, σ_n^2, and at the same time produced other values that are fairly small if compared with the former ones, e.g., the peak height estimates. Typical values of μ_n, σ_n^2 and γ_n are, respectively, 30,000, 10^{10} and 200,000. Such differences can cause numerical errors if used in estimating the eigenvalues and eigenvectors required to run the PCMCMC. For example, the eigenvalues of the estimated variance matrix, $\hat{\Sigma}$, may come out negative, even though $\hat{\Sigma}$ is positive definite by construction. A change of units can resolve this problem. Therefore, we converted the pixel electron density, d_j, unit from 10^{11}el/m^3 to 10^{16}el/km^3. This unit will be used with appropriate adjustment of the

Table 8.4: *Part of the correlation matrix estimated from the real data mixed algorithm results.*

Var	μ_{11}	μ_{12}	μ_{13}	μ_{14}	σ^2_{11}	σ^2_{12}	σ^2_{13}	σ^2_{14}	γ_{11}	γ_{12}	γ_{13}	γ_{14}
μ_{11}	1	0.99	0.99	0.98	0.57	0.48	0.37	0.3	0.89	0.87	0.83	0.76
μ_{12}	0.99	1	0.99	0.99	0.57	0.5	0.39	0.33	0.88	0.88	0.85	0.77
μ_{13}	0.99	0.99	1	0.99	0.57	0.49	0.4	0.33	0.87	0.87	0.86	0.77
μ_{14}	0.98	0.99	0.99	1	0.54	0.46	0.38	0.32	0.87	0.86	0.86	0.86
σ^2_{11}	0.57	0.57	0.57	0.54	1	0.96	0.89	0.82	0.49	0.47	0.46	0.47
σ^2_{12}	0.48	0.5	0.49	0.46	0.96	1	0.96	0.9	0.38	0.46	0.41	0.43
σ^2_{13}	0.37	0.39	0.4	0.38	0.89	0.96	1	0.96	0.28	0.32	0.39	0.37
σ^2_{14}	0.3	0.33	0.33	0.32	0.82	0.9	0.96	1	0.22	0.26	0.29	0.37
γ_{11}	0.89	0.88	0.87	0.87	0.49	0.38	0.28	0.22	1	0.72	0.74	0.66
γ_{12}	0.87	0.88	0.87	0.86	0.47	0.46	0.32	0.26	0.72	1	0.66	0.67
γ_{13}	0.83	0.85	0.86	0.86	0.46	0.41	0.39	0.29	0.74	0.66	1	0.61
γ_{14}	0.76	0.77	0.77	0.86	0.47	0.43	0.37	0.37	0.66	0.67	0.61	1

other parameters to bring all numbers onto somewhat closer scales in all forthcoming applications. However, to sample $\pi(X|Y)$, we still took the further step of scaling the whole parameter vector to implement the scaled version of our PCMCMC algorithm, as in Chapter 7.

Two sampling methods from π are then considered, as we did in Chapter 6. In the first method, the smoothing parameters are held fixed throughout the whole run at the derived MPL fixed estimates, whereas in the second method these parameters are allowed to vary regularly. These simulations are discussed next.

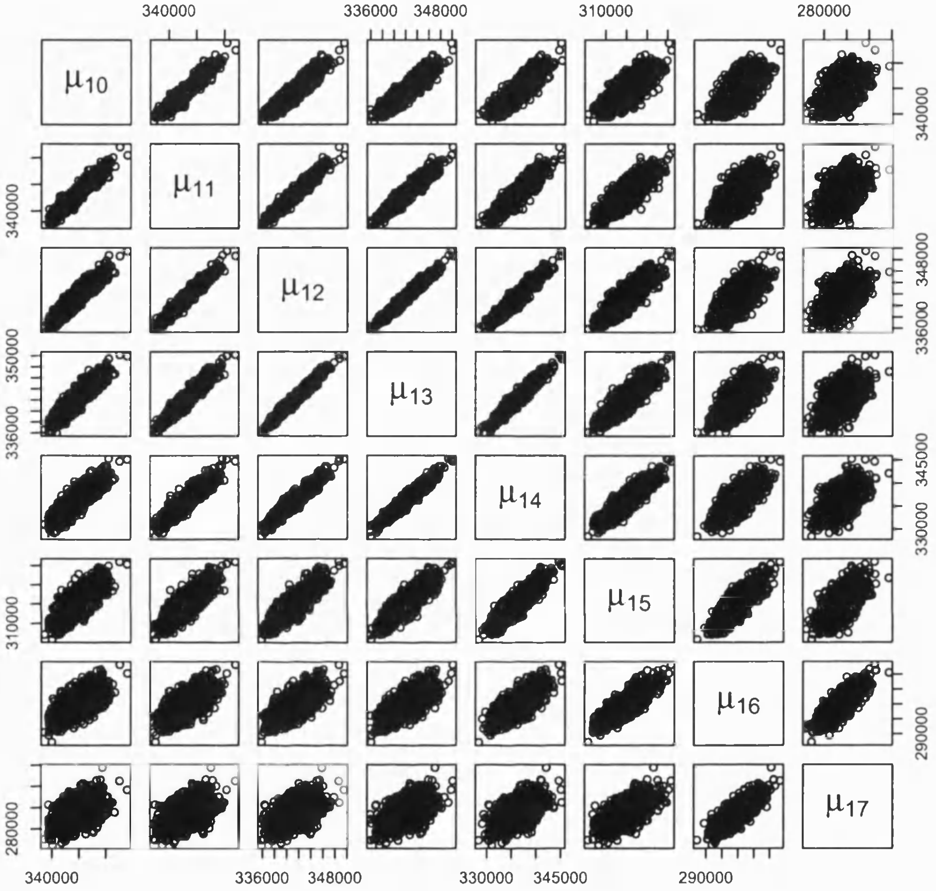


Figure 8-11: *2D scatter plots for pairs of μ_n values obtained from a sample generated by the mixed algorithm.*

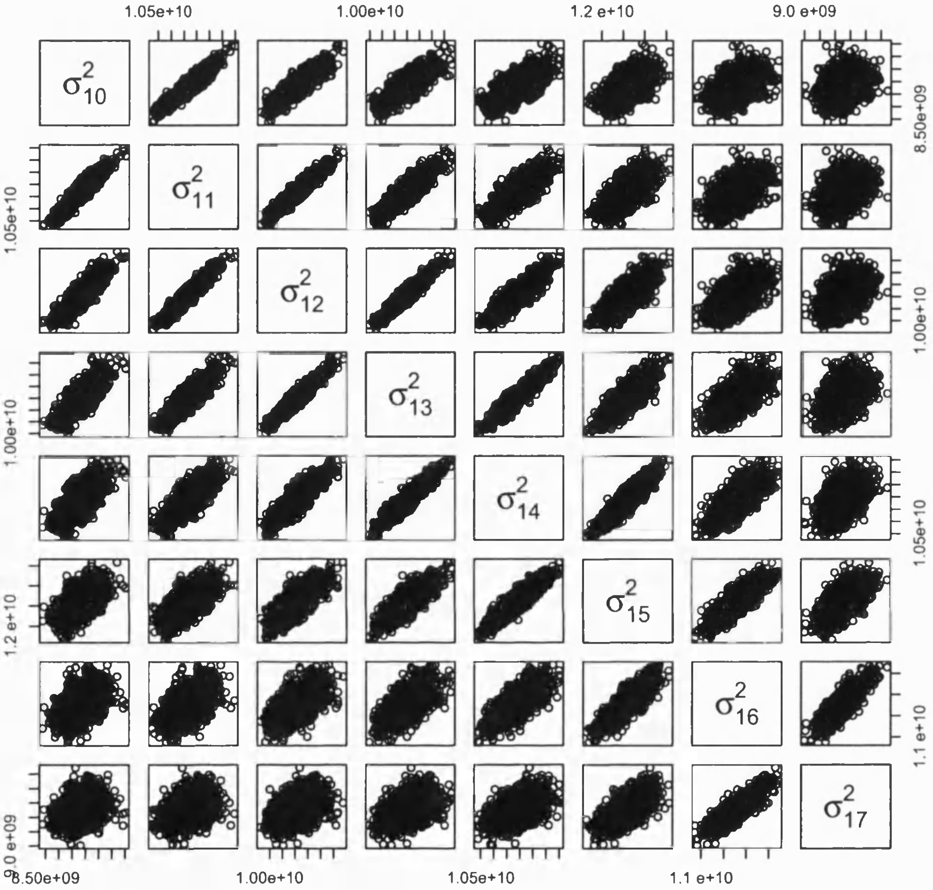


Figure 8-12: *2D scatter plots for pairs of σ_n^2 values obtained from a sample generated by the mixed algorithm.*

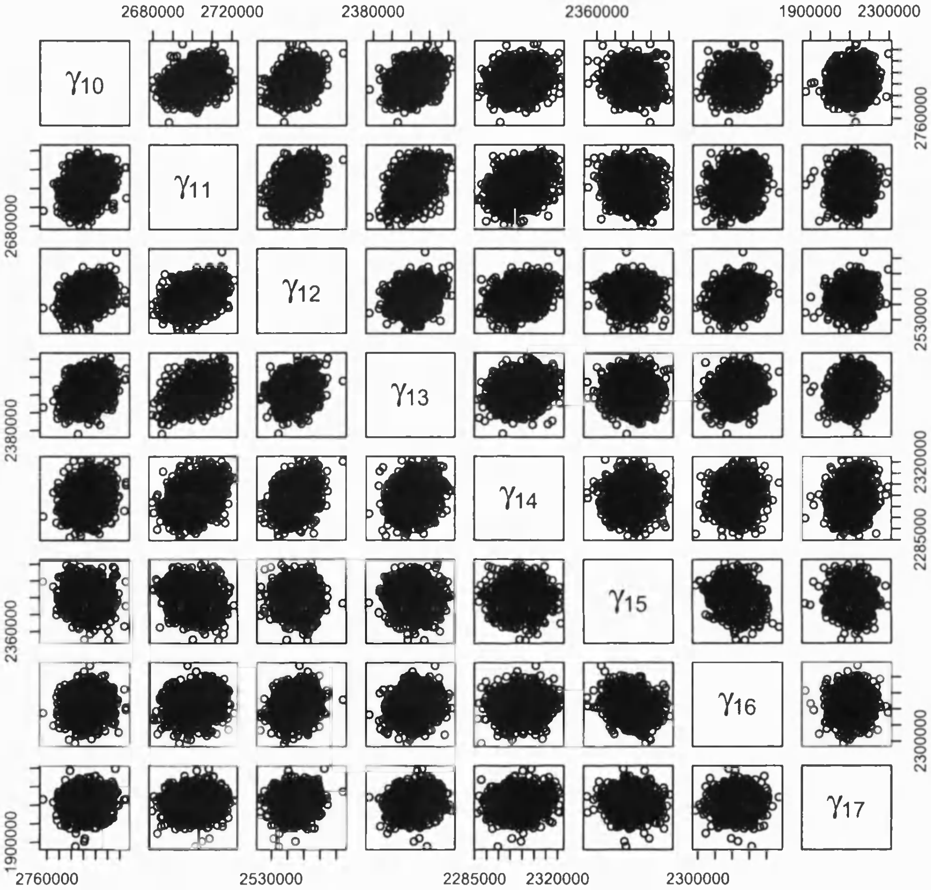


Figure 8-13: *2D scatter plots for pairs of γ_n values obtained from a sample generated by the mixed algorithm.*

8.3.1 PCMCMC with fixed smoothing parameters

The aim in this problem is to sample our posterior distribution, $\pi(X|Y)$, with fixed smoothing parameter values given by $\beta_\mu = 3.17 \times 10^{-8}, \beta_{\sigma^2} = 1.79 \times 10^{-18}$ and $\beta_\gamma = 5.57 \times 10^{-11}$, i.e. the MPL fixed estimates, using the scaled version of the PCMCMC algorithm. Therefore, the necessary principal component variances and directions are extracted from the scaled variables' estimated variance matrix derived from the mixed algorithm output using these smoothing parameter values, obtained in section 6.4. The associated 67 eigenvalues, $\{\hat{\lambda}_i\}$, are displayed in Table 8.5. The large number of very small eigen-values shows multiple, high correlations among the variables.

An alternative approach that can be implemented to obtain good or even better MPL estimates of our smoothing parameters is to apply the same technique used to derive the MPL fixed estimates described in Chapter 6 but now by using the PCMCMC rather than the mixed algorithm. However, this approach is not considered for further comparison reasons.

To start the PCMCMC run for this problem, we set the initial state to the scaled averages obtained from the long run of the mixed sampler converted to the modified units. We then ran the sampler for 100,000 iterations with proposal standard deviations $2.4\sqrt{\hat{\lambda}_i}$, for $i = 1, \ldots, 67$.

Simulation Results

Figures 8-14 to 8-16 reveal some time-series plots produced by the fixed smoothing parameters, principal components approach. Moves in the first, second and third principal component directions gave acceptance rates of approximately 39%, 34% and 35%, respectively. On the remaining 64 PCs' directions, the acceptance rates lie within the range 42% - 45%.

The resulting traces indicate that the proposed chains are mixing well. To provide a quantitative comparison between the mixed algorithm results derived in chapter 6 and the PCMCMC output found in this case, estimates of the integrated autocorrelation times are calculated for each parameter. Here segments of chains of length 50,000 obtained from the final stages of each sampler are used. Table 8.6 displays these estimates. Obviously, the results in this table reveal that the PCMCMC algorithm is performing very much better than the mixed sampler.

Table 8.5: *The estimated scaled variables' variance matrix eigenvalues derived from the mixed algorithm run with fixed interaction parameters.*

Number i	Eigenvalue λ_i	Number i	Eigenvalue λ_i
1	23.778	36	0.096
2	8.347	37	0.091
3	5.566	38	0.084
4	4.198	39	0.077
5	3.504	40	0.072
6	2.731	41	0.066
7	2.523	42	0.057
8	1.901	43	0.054
9	1.437	44	0.048
10	1.191	45	0.045
11	1.036	46	0.043
12	0.919	47	0.041
13	0.889	48	0.038
14	0.778	49	0.037
15	0.687	50	0.028
16	0.640	51	0.026
17	0.611	52	0.023
18	0.542	53	0.018
19	0.527	54	0.015
20	0.469	55	0.011
21	0.425	56	0.008
22	0.405	57	0.0077
23	0.349	58	0.0065
24	0.339	59	0.0046
25	0.322	60	0.0043
26	0.288	61	0.0041
27	0.252	62	0.0039
28	0.240	63	0.0032
29	0.204	64	0.0027
30	0.177	65	0.0023
31	0.162	66	0.0016
32	0.159	67	0.0014
33	0.155		
34	0.134		
35	0.099		

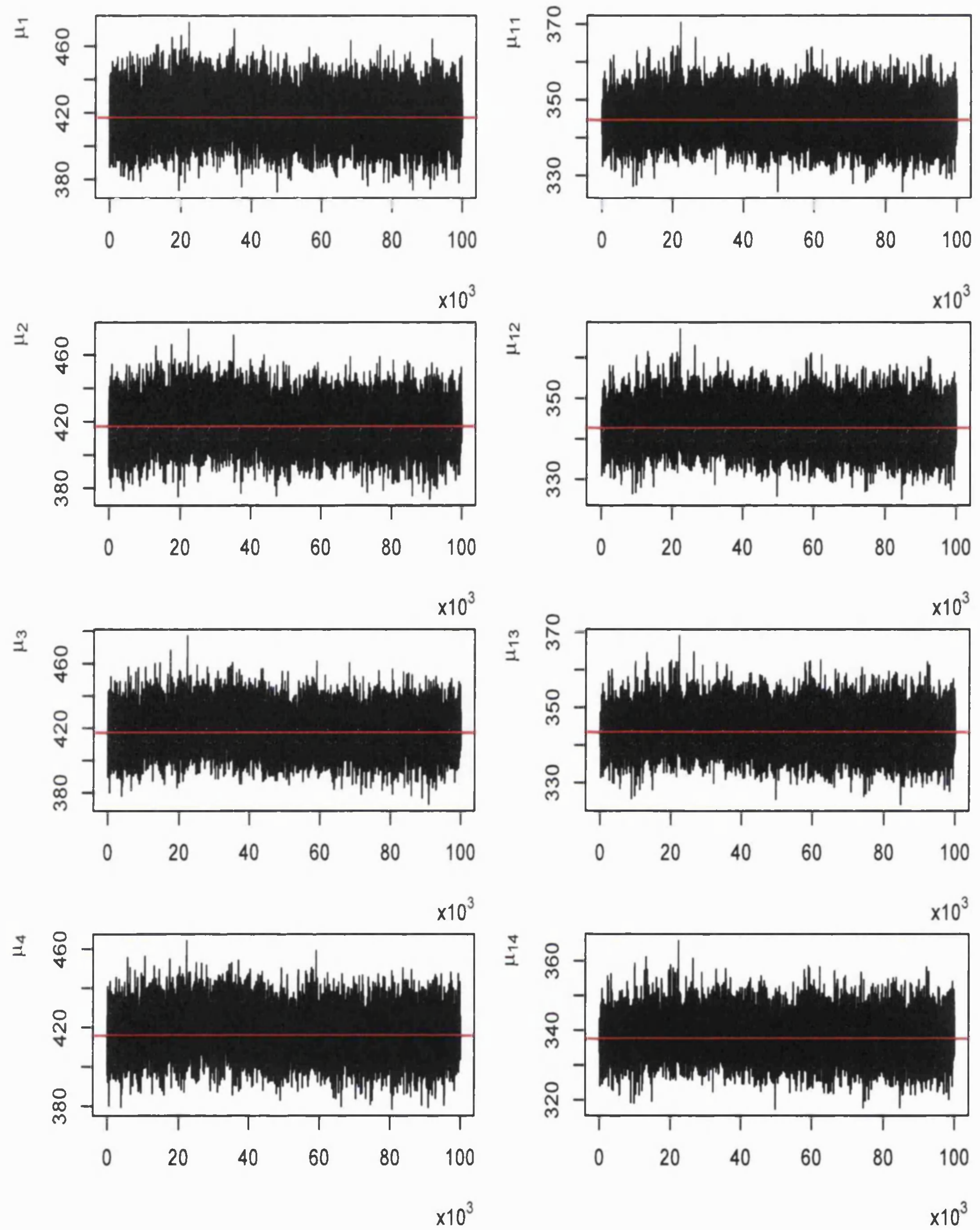


Figure 8-14: *Some selected μ_n time-series plots produced by implementing the PCMCMC algorithm using the fixed MPL estimates of the smoothing parameters. The red line represents the mean of the second half of the sampled values.*

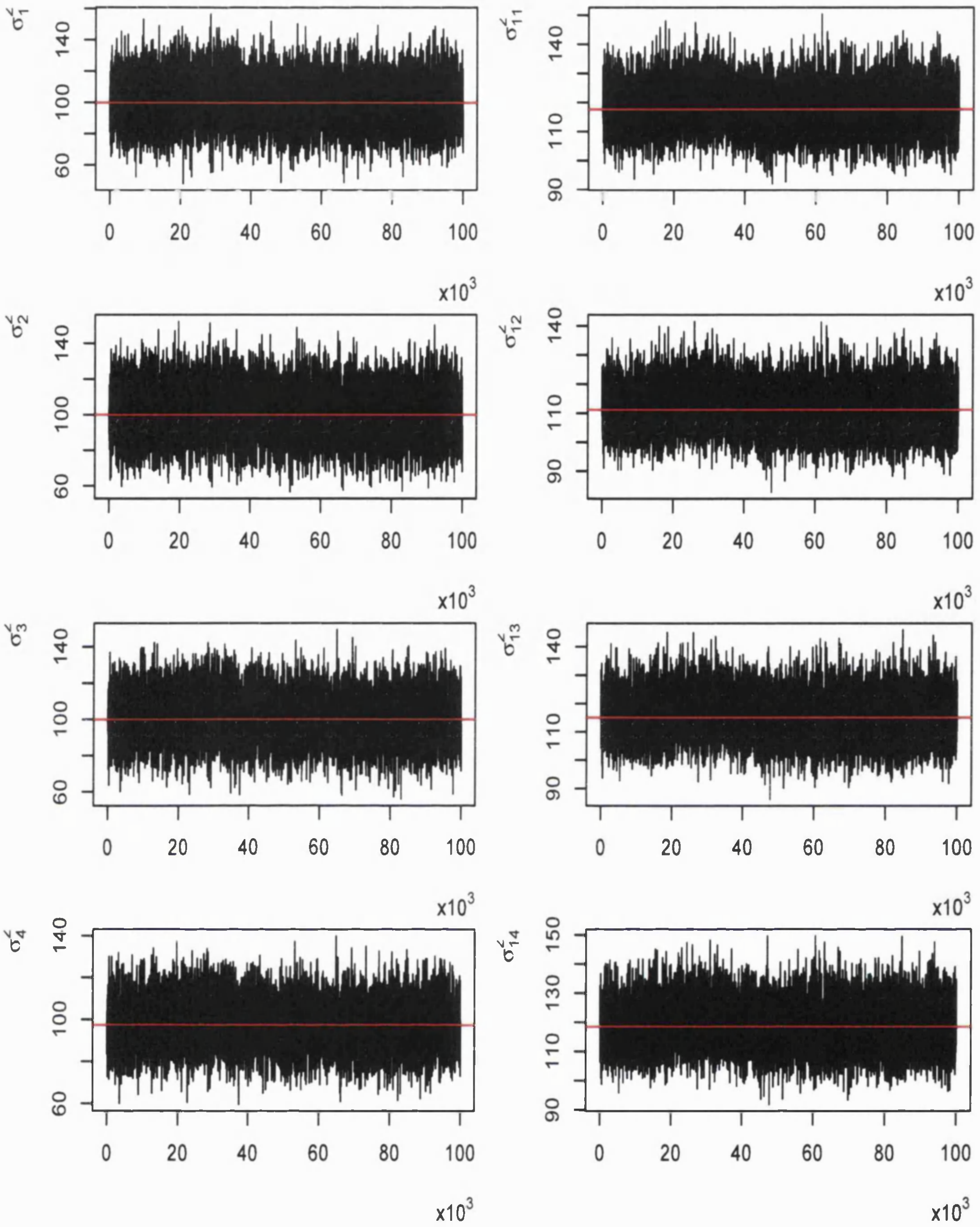


Figure 8-15: *Some selected σ_n^2 time-series plots scaled by a factor of 10^2 produced by implementing the PCMCMC algorithm using the fixed MPL estimates of the smoothing parameters. The red line represents the mean of the second half of the sampled values.*

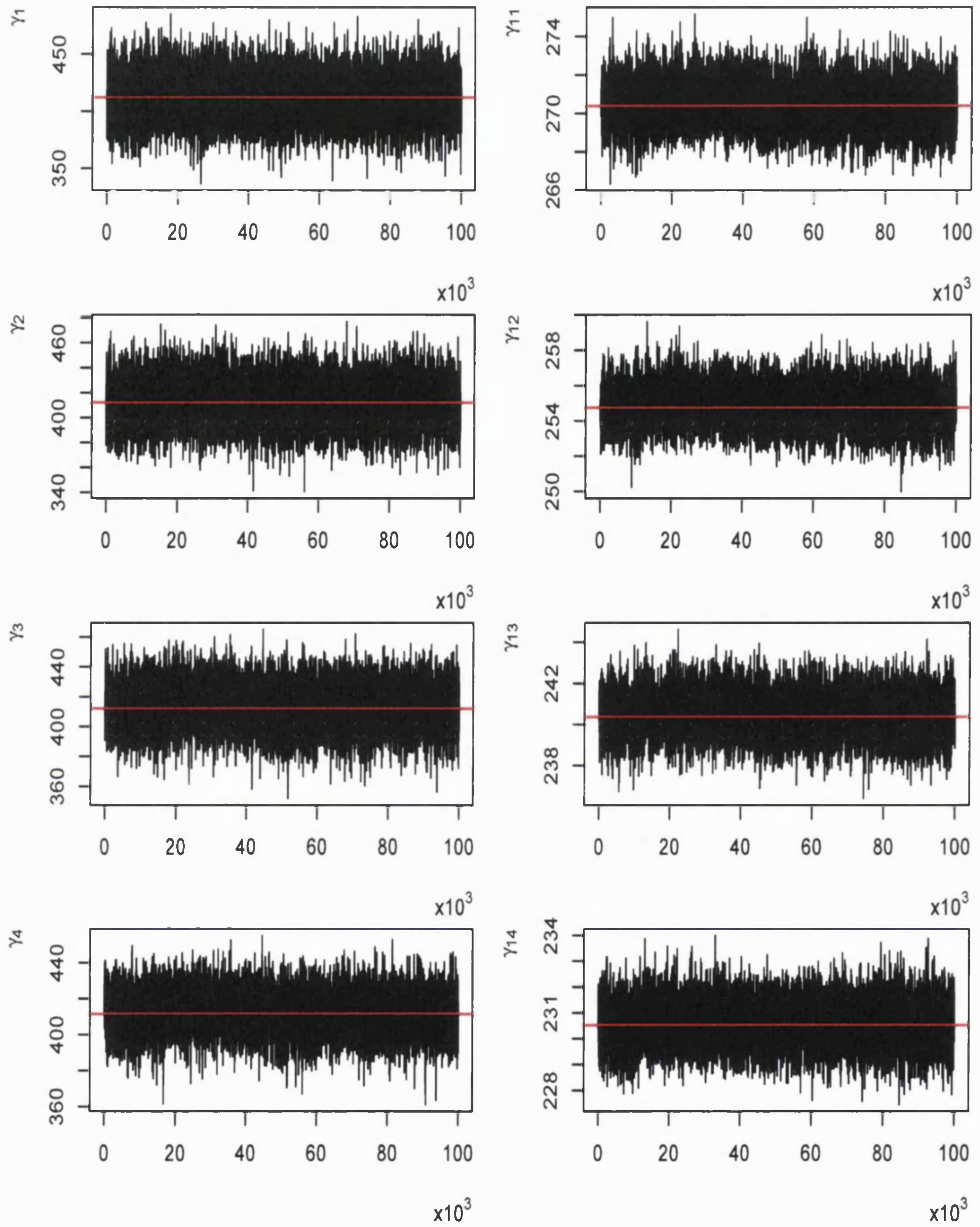


Figure 8-16: *Some selected γ_n time-series plots scaled by a factor of* 10^5 *produced by implementing the PCMCMC algorithm using the fixed MPL estimates of the smoothing parameters. The red line represents the mean of the second half of the sampled values.*

Table 8.6: *The integrated autocorrelation time estimates obtained from the last 50,000 iterations of the mixed* MC *algorithm and the* PCMCMC *algorithm with fixed smoothing parameters set at the MPL fixed estimates derived in chapter 6.*

	$\hat{\tau}(.)$MC	$\hat{\tau}(.)$PCMCMC		$\hat{\tau}(.)$MC	$\hat{\tau}(.)$PCMCMC		$\hat{\tau}(.)$MC	$\hat{\tau}(.)$PCMCMC
μ_1	3564	7.7	σ_1^2	1484	6.1	γ_1	870	5.3
μ_2	3970	7.8	σ_2^2	1693	6.6	γ_2	939	5.4
μ_3	4463	8.3	σ_3^2	1935	6.9	γ_3	1090	5.6
μ_4	4864	8.8	σ_4^2	2225	7.1	γ_4	1177	6.1
μ_5	5271	8.6	σ_5^2	2679	7.4	γ_5	545	5.6
μ_6	5564	9.3	σ_6^2	2620	7.4	γ_6	568	5.3
μ_7	5676	9.1	σ_7^2	2358	7.4	γ_7	1745	5.3
μ_8	5947	8.7	σ_8^2	1844	7.9	γ_8	2423	5.9
μ_9	5907	8.4	σ_9^2	4282	8.6	γ_9	311	4.7
μ_{10}	5913	8.4	σ_{10}^2	4575	8.9	γ_{10}	1003	6.6
μ_{11}	5832	8.4	σ_{11}^2	4091	9.2	γ_{11}	2995	7.7
μ_{12}	5736	8.5	σ_{12}^2	2053	10.0	γ_{12}	2913	7.9
μ_{13}	5622	8.6	σ_{13}^2	1745	10.3	γ_{13}	2636	8.2
μ_{14}	5331	8.6	σ_{14}^2	1485	10.4	γ_{14}	1726	7.7
μ_{15}	4513	8.2	σ_{15}^2	1244	10.3	γ_{15}	300	6.2
μ_{16}	3300	7.4	σ_{16}^2	1174	9.6	γ_{16}	757	5.4
μ_{17}	1921	6.9	σ_{17}^2	895	9.1	γ_{17}	739	5.8
μ_{18}	1789	6.5	σ_{18}^2	874	8.6	γ_{18}	742	5.8
μ_{19}	1805	6.3	σ_{19}^2	823	8.4	γ_{19}	1126	6.1
μ_{20}	1784	6.2	σ_{20}^2	768	8.1	γ_{20}	1449	5.8
μ_{21}	1697	6.0	σ_{21}^2	723	7.7	γ_{21}	1729	5.6
μ_{22}	1327	5.8	σ_{22}^2	637	7.5	γ_{22}	2041	5.6

Further, with the given fixed values of the smoothing parameters, the PCMCMC is found to be very efficient because all integrated autocorrelation time estimates are between 4.6 and 10.5. These values of integrated autocorrelation times are quite similar to those obtained for the simulated data example discussed earlier and the approximately Gaussian distribution of section 7.5.2 with $a = 4$. This may suggest that the underlying posterior distribution is possibly approximately Gaussian but with a small degree of normality disruption.

The data error, ϵ, estimated from this run is found to be small with an average of about 0.73 TECU. This is consistent with the electronic engineers' beliefs about this parameter. A comparison between the data, Y, and the new tomographic approach estimate of $E(Y)$ obtained from the posterior mean is displayed in Figure 8-17. Generally, the agreement between both sets of values is very good. This indicates that the TEC data are fitted well. Indeed, this is an expected result because the

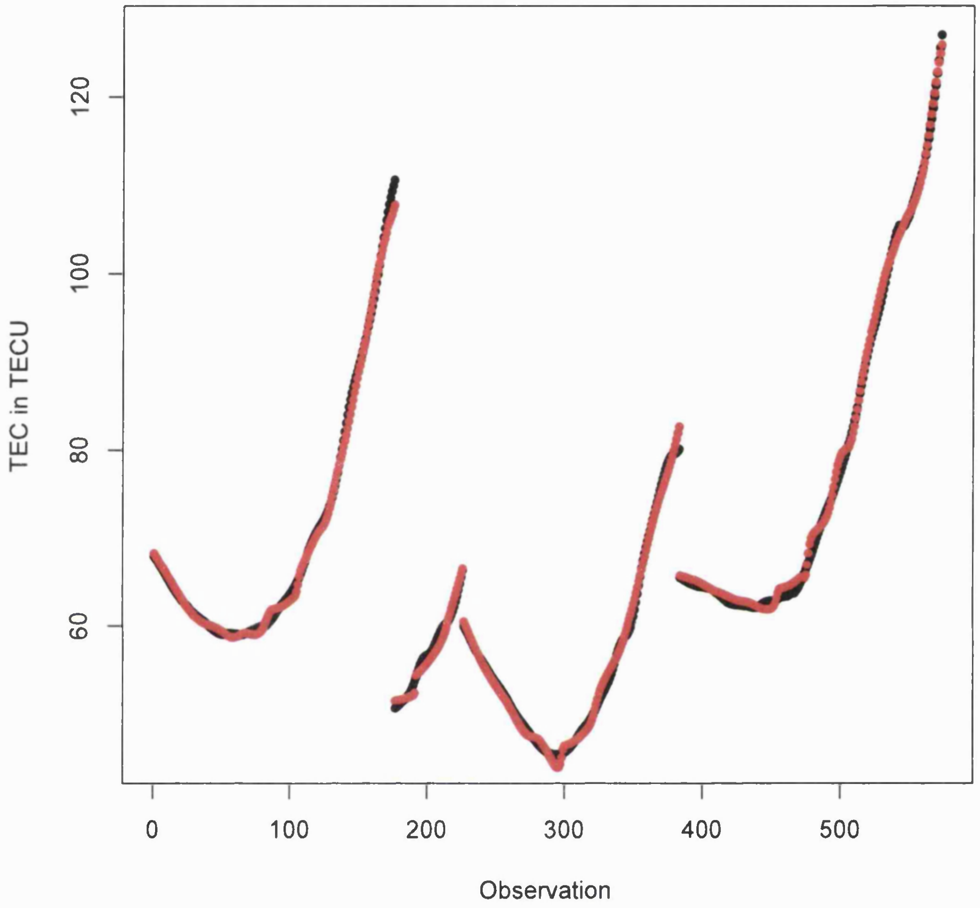


Figure 8-17: *A comparison between the 572 TEC data values Y (black) and estimates of $E(Y)$ (red) obtained from the first method PCMCMC output at the posterior mean.*

observations are recorded with a small noise, therefore the TEC data values and the values fitted by the model should be very close.

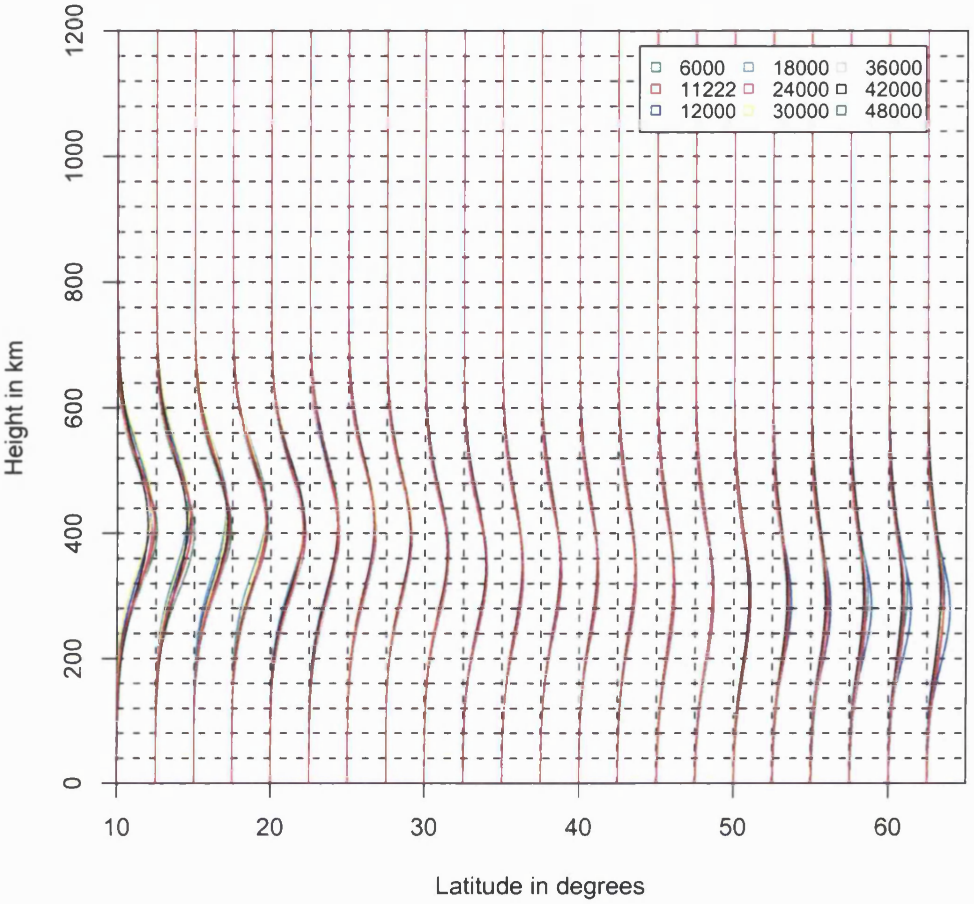


Figure 8-18: *The high-level image obtained from the first problem run of the PCMCMC algorithm at the above mentioned iterations.*

As with the mixed algorithm equivalent results, see for example Figure 6-23, the variations of the vertical profiles associated with this run are within fairly small ranges as Figure 8-18 reveals.

Both the ionospheric map of free electrons associated with the posterior mean and the length of the 95% credible interval image are shown in Figure 8-19.

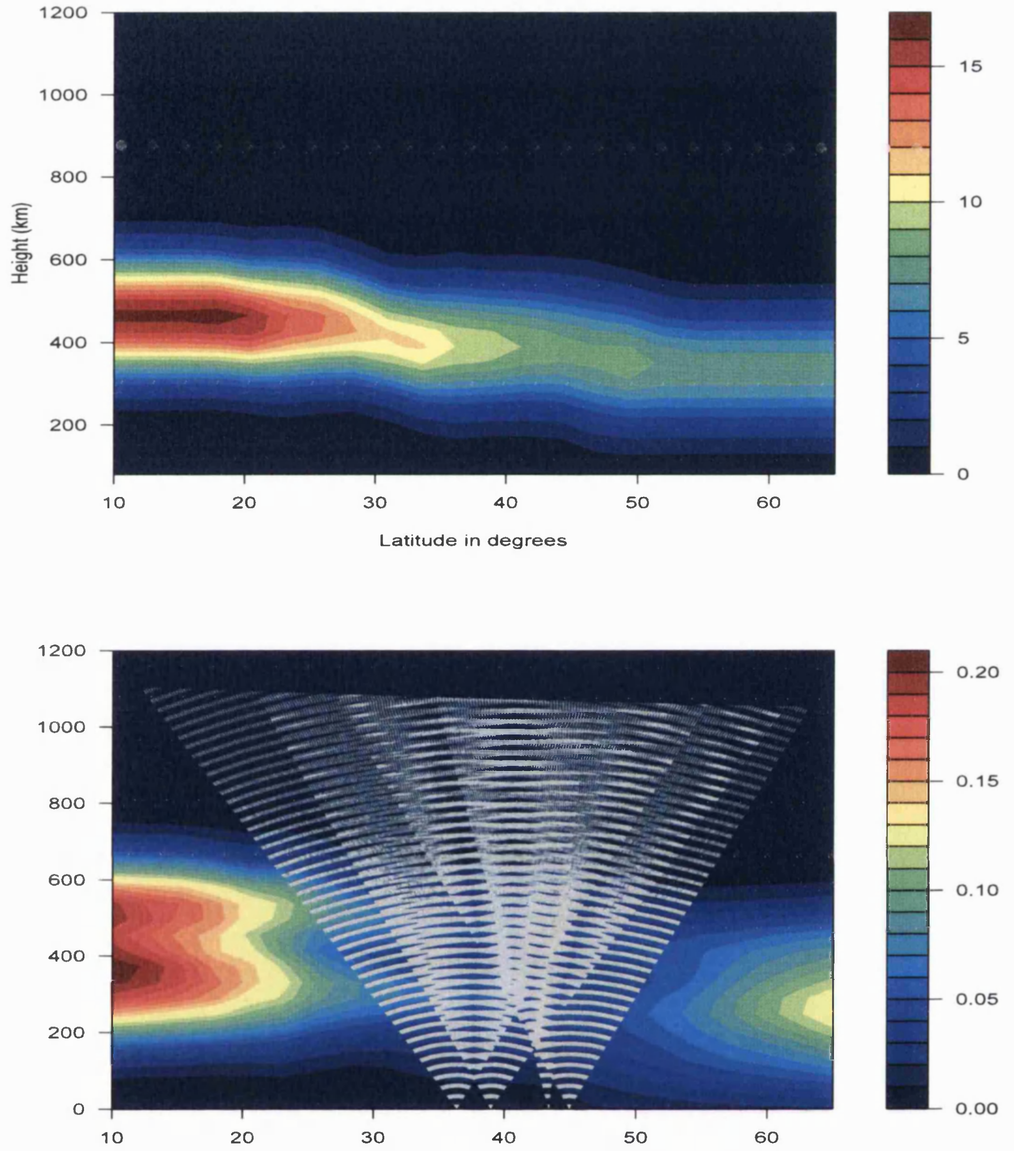


Figure 8-19: *The tomographic image of electron densities measured in units of* $10^{11}el/m^2$ *at the posterior mean (Top) and the length of the 95% credible interval overlayed by all signal paths associated with the fixed smoothing parameters PCMCMC run restorations (Bottom).*

8.3.2 PCMCMC with varying smoothing parameters

Here, the PCMCMC algorithm is applied with repeatedly updated values of the smoothing parameters. The algorithm can be summarised in the following steps:

1. Run the mixed algorithm to get a sample approximately from $\pi(X|Y)$ and then apply the MPL approach to obtain starting values for the smoothing parameters, e.g., $\beta^0 = \{\beta_\mu^0, \beta_{\sigma^2}{}^0, \beta_\gamma^0\}$.

2. With the same sample and scaled variables obtained from step 1, estimate the necessary eigenvalues, PCs, etc. This step is to be done only once.

3. Set $\beta = \beta^0$ and generate 1000 sweeps of the PCMCMC updates (as defined in step 2). Then apply the MPL approach to get a new β using this sample central ionospheric region values.

4. (Iterative step): with the present β, generate 1000 iterations of the PCMCMC algorithm to derive a new β.

For this particular case, we used the MPL fixed estimates as the required starting values. As in the first problem run, 100,000 sweeps with proposal scalings $2.4\sqrt{\hat{\lambda}_i}$, for all values of i are considered.

Simulation Results

In Figures 8-20 to 8-22, we show the equivalent traces to those displayed in Figures 8-14 to 8-16, generated by the varying smoothing parameters' PCMCMC approach.

In contrast to the first PCMCMC run acceptance rates in the first three PCs' directions, the second problem acceptance rates in these directions are slightly lower especially in the main principal component direction. These rates are, respectively, 36%, 33% and 34%. In the remaining directions, the acceptance rates are almost the same as those in the first version of this sampler.

Comparing the time-series plots displayed in Figure 8-20 with the mixed algorithm traces revealed in the right panels of Figures 6.13 and 6.14, for instance, indicates that the PCMCMC is performing much better than the mixed algorithm for this varying smoothing parameters approach. The integrated autocorrelation time estimates in Table 8.7 confirm this conclusion. However, when Figures 8-20 to 8-22 are compared with the fixed interaction PCMCMC results displayed in Figures 8-14 to 8-16, and the associated integrated autocorrelation times in Tables 8.6 and 8.7 are also compared, the varying β PCMCMC is found to be "not quite as brilliant" as for the fixed β problem.

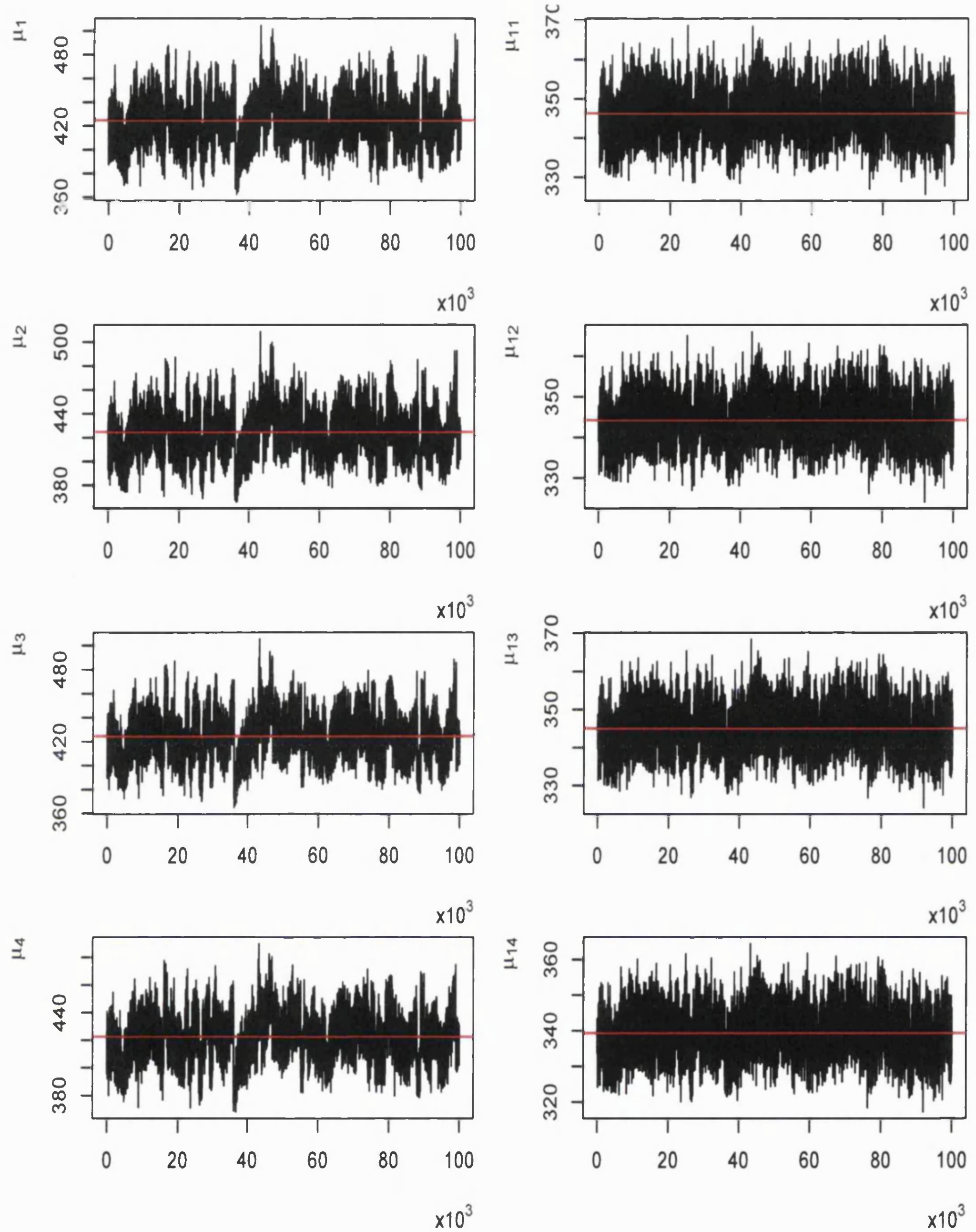


Figure 8-20: *Some selected μ_n time-series plots produced by implementing the second method PCMCMC algorithm where the smoothing parameters are updated regularly. The red line represents the average of the second half of the sampled values.*

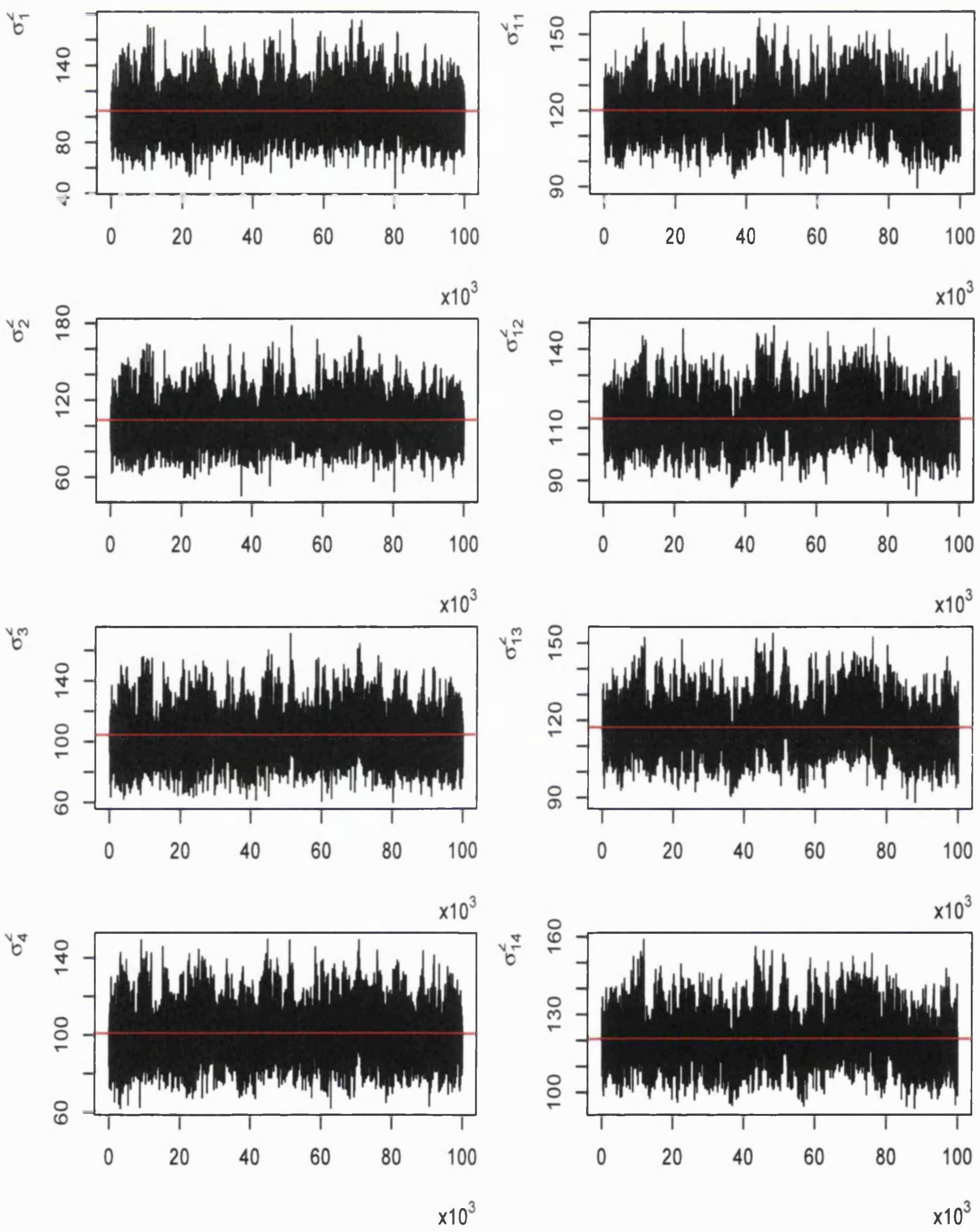


Figure 8-21: *Some selected σ_n^2 time-series plots scaled by a factor of 10^2 produced by implementing the second method PCMCMC algorithm where the smoothing parameters are updated regularly. The red line represents the average of the second half of the sampled values.*

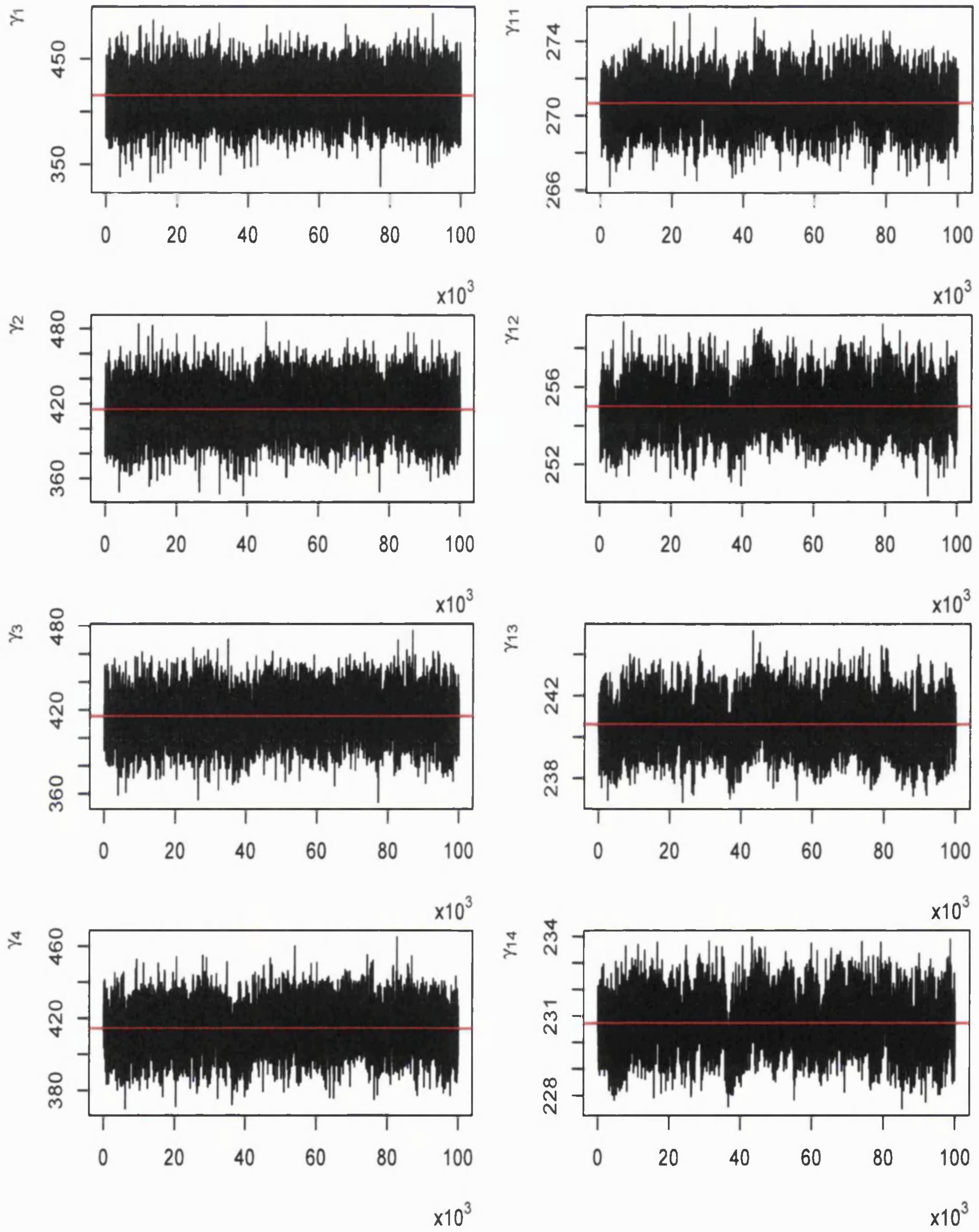


Figure 8-22: *Some selected γ_n time-series plots scaled by a factor of* 10^5 *produced by implementing the second method PCMCMC algorithm where the smoothing parameters are updated regularly. The red line represents the average of the second half of the sampled values.*

Table 8.7: *The integrated autocorrelation time estimates obtained from the last 50,000 iterations of the mixed* MC *algorithm and the second problem* PCMCMC *algorithm. Here the smoothing parameters are updated every 1000 iteration.*

μ_n	$\hat{\tau}(.)$MC	$\hat{\tau}(.)$PCMCMC	σ_n^2	$\hat{\tau}(.)$MC	$\hat{\tau}(.)$PCMCMC	γ_n	$\hat{\tau}(.)$MC	$\hat{\tau}(.)$PCMCMC
μ_1	3448	538	σ_1^2	2294	191	γ_1	1858	16
μ_2	3837	562	σ_2^2	2633	221	γ_2	2343	31
μ_3	4613	586	σ_3^2	2915	329	γ_3	3127	52
μ_4	4605	592	σ_4^2	4038	189	γ_4	3898	73
μ_5	5752	559	σ_5^2	4463	506	γ_5	3802	61
μ_6	8173	561	σ_6^2	6978	694	γ_6	2459	85
μ_7	5744	494	σ_7^2	7559	385	γ_7	5316	220
μ_8	6351	474	σ_8^2	5700	194	γ_8	6036	508
μ_9	9045	337	σ_9^2	6579	477	γ_9	2796	120
μ_{10}	9064	270	σ_{10}^2	9696	534	γ_{10}	4810	67
μ_{11}	9362	276	σ_{11}^2	9677	677	γ_{11}	7329	225
μ_{12}	9558	280	σ_{12}^2	9546	669	γ_{12}	6984	167
μ_{13}	9568	278	σ_{13}^2	9320	686	γ_{13}	6772	181
μ_{14}	9317	261	σ_{14}^2	9178	670	γ_{14}	6478	185
μ_{15}	8470	143	σ_{15}^2	8863	711	γ_{15}	5496	237
μ_{16}	7494	51	σ_{16}^2	8035	476	γ_{16}	2444	362
μ_{17}	3843	47	σ_{17}^2	6775	162	γ_{17}	2656	305
μ_{18}	2684	70	σ_{18}^2	3260	163	γ_{18}	2509	148
μ_{19}	2608	67	σ_{19}^2	2015	149	γ_{19}	5540	75
μ_{20}	2735	49	σ_{20}^2	1979	132	γ_{20}	10482	54
μ_{21}	2878	42	σ_{21}^2	1890	118	γ_{21}	11303	26
μ_{22}	2615	37	σ_{22}^2	1698	102	γ_{22}	11377	29

The empirical average of ϵ obtained from this sampling approach is about 0.7. Although the results indicate no substantial difference, most estimates of the integrated autocorrelation times differ highly. This is because the second method incorporates variation in the smoothing parameters. The effect of this can be clearly seen in the scatter plots of Figure 8-23. The irregular plots in this figure indicate that the degree of normality disruption in this case is higher than that present in Figure 8-11. This is due to the generation of some values of $\beta_\mu, \beta_{\sigma^2}$ and β_γ other than the MPL fixed estimates used to produce Figure 8-11, as Figure 6-19 reveals. This indicates that the PCMCMC sampler is very sensitive to the values of the interaction parameters used.

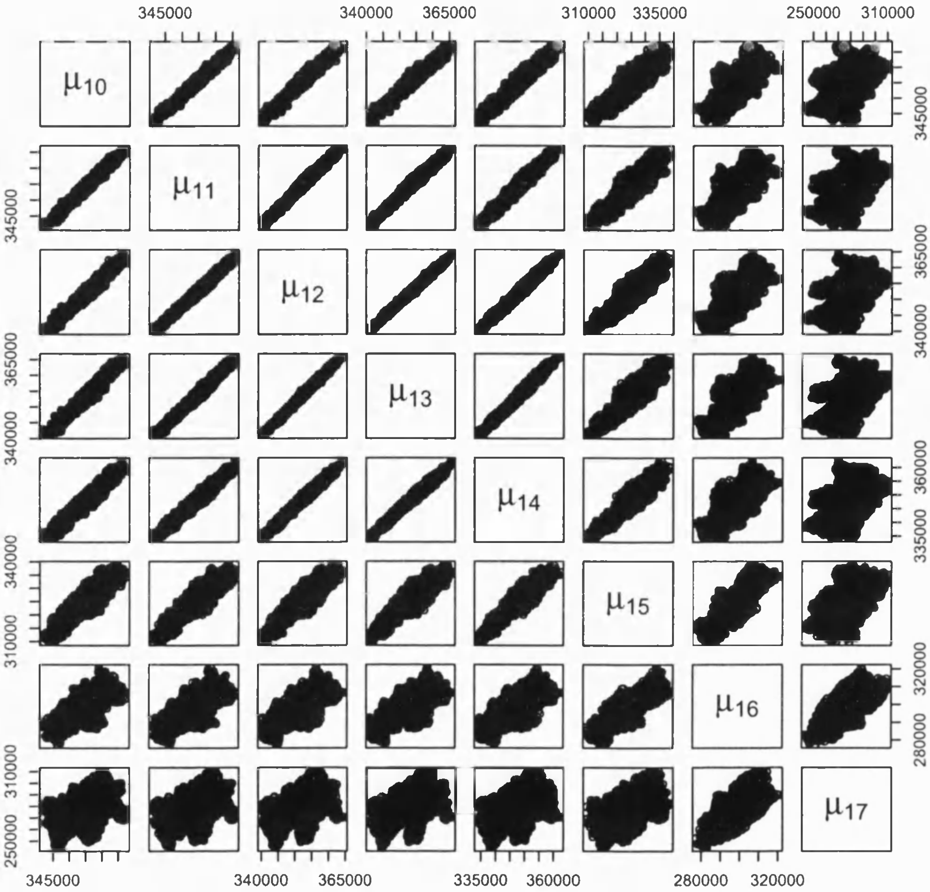


Figure 8-23: *Some profiles' peak heights scatter plots obtained from a sample generated by the mixed algorithm when the interaction parameters are updated regularly.*

The ionosphere tomographic image obtained at the posterior mean and the length of the 95% credible interval image are shown in Figure 8-24. In Figure 8.25, we compare between Y and estimates of $E(Y)$ obtained at this mean. Again a very good fit is obtained. The associated high-level image is displayed in Figure 8-26.

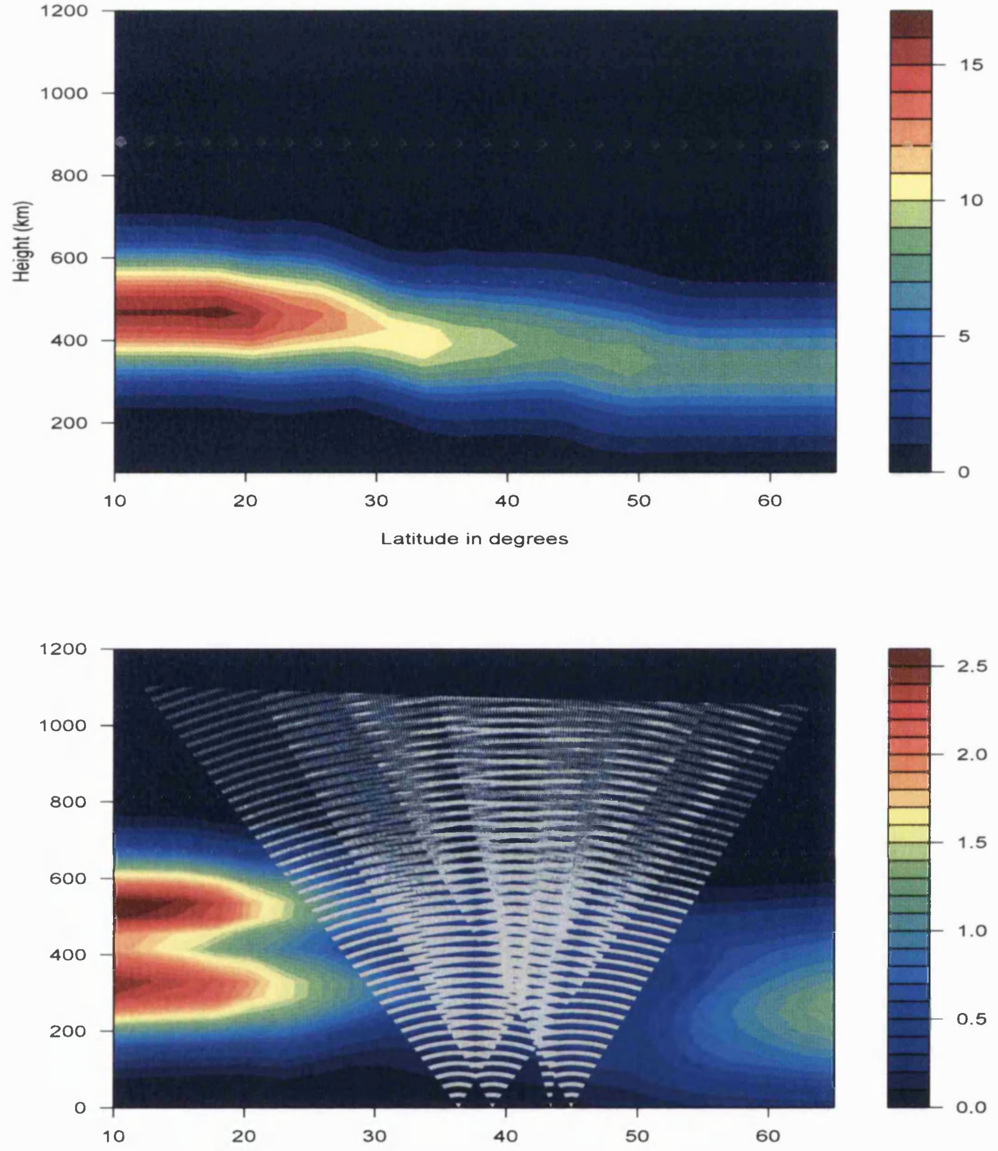


Figure 8-24: *The tomographic image of electron densities measured in units of* $10^{11} el/m^2$ *at the posterior mean (Top) and the length of the 95% credible interval overlayed by all signal paths associated with the varying smoothing parameters PCMCMC run restorations (Bottom).*

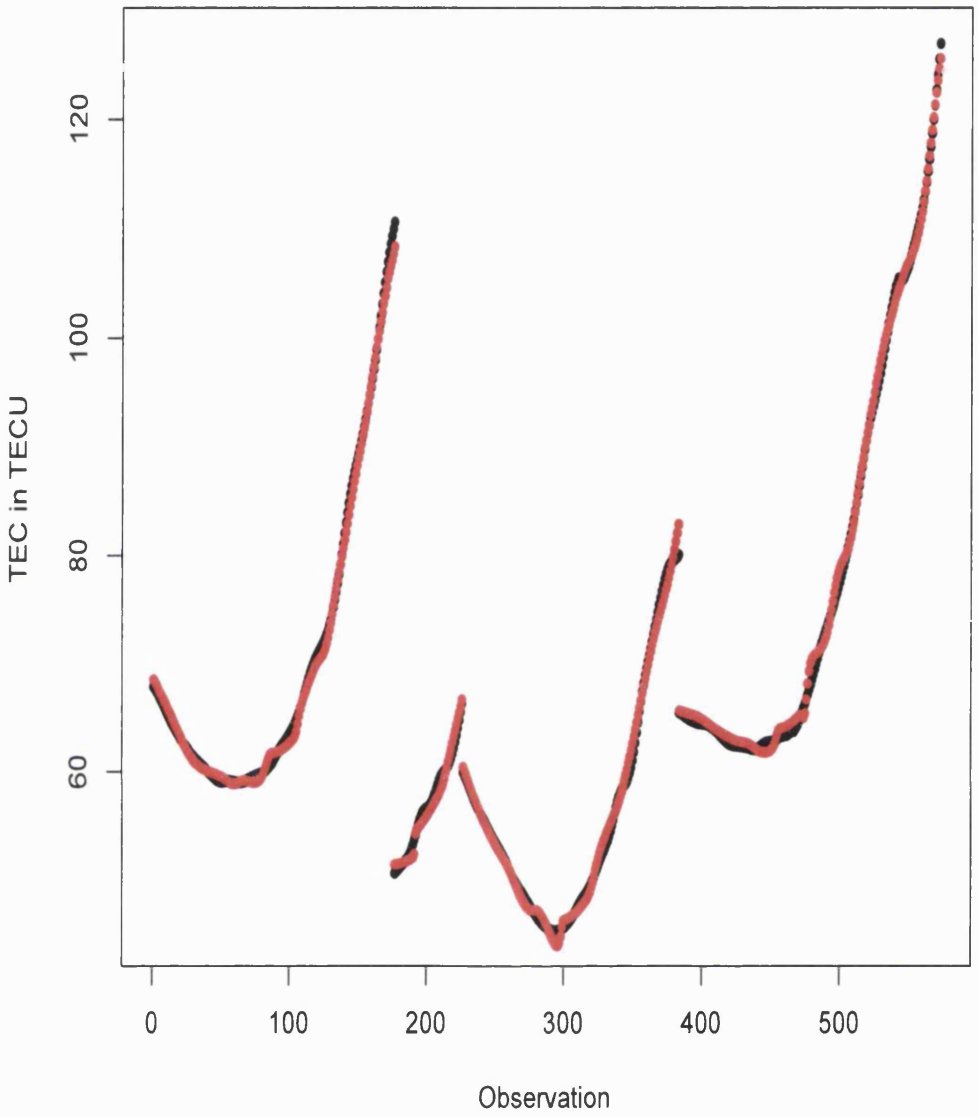


Figure 8-25: *A comparison between the 572 TEC data values Y (black) and estimates of $E(Y)$ (red) associated with the varying smoothing parameters PCMCMC at the resulting posterior mean.*

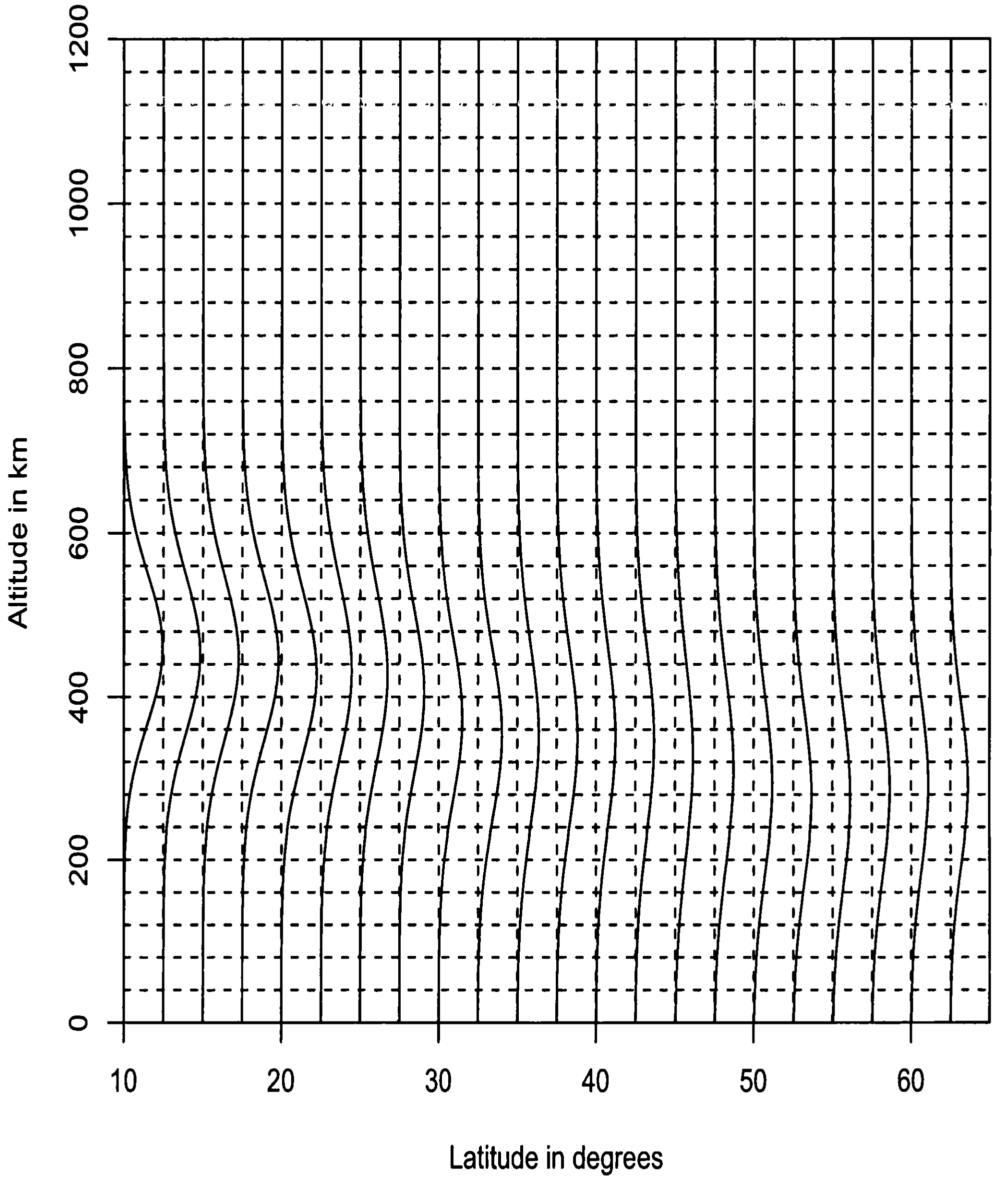


Figure 8-26: *The high-level image obtained from the second method PCMCMC run at the resulting posterior mean.*

8.4 Uncertainty and sensitivity

In the real data problem of ionospheric tomography, we are uncertain about the true values of the physical quantities of interest. Fortunately, implementing the Bayesian approach allows us to make uncertainty statements about the final outcome of the analysis. for example in the form of credible intervals as recorded in Tables 8.8 and 8.9, or images of the length of the 95% credible intervals as displayed in Figure 8-19.

In this study, we have assessed the sensitivity to the starting values and seeds using the initial values obtained from MIDAS, and other random and flat initial states. Figure 8-27 reveal the resulting images. The images in this figure do not show important differences for the posterior expectations of electron densities when different seeds and initial states were considered. The associated length of the 95% credible interval images displayed in Figure 8-28 reveal a high degree of variation within the boundary regions of the grid represented in the differences in the shape and size of these regions. This reflects the fact that there is less certainty about the free electron concentration within these areas which is consistent with the lack of data noticed in these parts of the ionosphere.

A major benefit of the Bayesian approach is the ability to incorporate prior knowledge in the form of *a prior* distribution. During the development of our algorithm we found that the results are very sensitive to the amount of smoothing selected in the prior model, especially in regions where there is a lack of data. When the smoothing parameters are allowed to vary during the simulation process, the uncertainty associated with these parameters increased the overall uncertainty for all other variables. This effect was seen in the form of some wider 95% associated credible intervals, see Tables 8.8 and 8.9.

8.5 Conclusion

In this chapter, we demonstrated the application of the new algorithm we have developed, principal components MCMC, in ionospheric tomography study to image the electron density in the ionosphere.

The PCMCMC algorithm was initially tested on a set of simulated TEC data. This algorithm revealed an outstanding performance leading to the generation of a Markov chain that mixes extremely well.

Table 8.8: *The estimated posterior expectations and credible intervals of μ in km, σ in km, and γ in $10^{16}el/km^2$ scaled by a factor of 10^5, associated with the last 50,000 iterations of the PCMCMC fix βs approach.*

	$\bar{\mu}_n$	95% Cr.I.		$\bar{\sigma}_n$	95% Cr.I.		$\bar{\gamma}_n$	95% Cr.I.
μ_1	417	(408,426)	σ_1	100	(95,105)	γ_1	412	(399,426)
μ_2	417	(408,426)	σ_2	100	(95,105)	γ_2	412	(400,425)
μ_3	417	(409,426)	σ_3	100	(95,104)	γ_3	412	(402,423)
μ_4	416	(408,424)	σ_4	99	(95,102)	γ_4	412	(404,420)
μ_5	405	(398,411)	σ_5	98	(94,102)	γ_5	397	(391,402)
μ_6	404	(397,411)	σ_6	103	(100,106)	γ_6	368	(364,373)
μ_7	404	(399,410)	σ_7	101	(98,103)	γ_7	340	(336,344)
μ_8	392	(387,397)	σ_8	96	(94,98)	γ_8	294	(290,297)
μ_9	366	(361,370)	σ_9	95	(93,97)	γ_9	269	(267,270)
μ_{10}	346	(342,350)	σ_{10}	100	(97,102)	γ_{10}	280	(279,280)
μ_{11}	345	(341,348)	σ_{11}	108	(106,111)	γ_{11}	270	(270,271)
μ_{12}	343	(339,346)	σ_{12}	105	(103,108)	γ_{12}	255	(254,256)
μ_{13}	344	(340,347)	σ_{13}	107	(105,110)	γ_{13}	240	(240,241)
μ_{14}	338	(334,342)	σ_{14}	109	(106,111)	γ_{14}	231	(230,231)
μ_{15}	317	(312,321)	σ_{15}	115	(112,117)	γ_{15}	240	(239,241)
μ_{16}	302	(297,307)	σ_{16}	112	(109,115)	γ_{16}	241	(239,244)
μ_{17}	290	(285,296)	σ_{17}	107	(103,111)	γ_{17}	211	(208,215)
μ_{18}	285	(279,291)	σ_{18}	105	(101,109)	γ_{18}	202	(196,208)
μ_{19}	285	(278,291)	σ_{19}	104	(100,109)	γ_{19}	202	(193,210)
μ_{20}	285	(278,291)	σ_{20}	104	(100,109)	γ_{20}	201	(191,212)
μ_{21}	285	(277,292)	σ_{21}	104	(99,110)	γ_{21}	201	(189,214)
μ_{22}	285	(277,293)	σ_{22}	104	(99,110)	γ_{22}	201	(187,215)

Table 8.9: *The estimated posterior expectations and credible intervals of μ in km, σ in km, and γ in $10^{16}el/km^2$ scaled by a factor of 10^5, associated with the last 50,000 iterations of the PCMCMC varying βs approach.*

	$\bar{\mu}_n$	95% Cr.I.		$\bar{\sigma}_n$	95% Cr.I.		$\bar{\gamma}_n$	95% Cr.I.
μ_1	425	(412,438)	σ_1	102	(97,107)	γ_1	416	(403,430)
μ_2	425	(412,438)	σ_2	102	(97,107)	γ_2	416	(404,429)
μ_3	425	(412,437)	σ_3	102	(97,107)	γ_3	416	(406,427)
μ_4	423	(411,434)	σ_4	100	(96,104)	γ_4	415	(408,423)
μ_5	411	(401,421)	σ_5	101	(96,105)	γ_5	398	(393,404)
μ_6	411	(401,420)	σ_6	106	(101,109)	γ_6	369	(364,374)
μ_7	410	(402,418)	σ_7	103	(99,106)	γ_7	339	(335,344)
μ_8	396	(390,403)	σ_8	97	(94,99)	γ_8	290	(286,295)
μ_9	368	(363,373)	σ_9	96	(94,99)	γ_9	268	(266,269)
μ_{10}	348	(344,352)	σ_{10}	101	(98,104)	γ_{10}	280	(279,281)
μ_{11}	347	(343,350)	σ_{11}	110	(107,112)	γ_{11}	271	(270,272)
μ_{12}	345	(341,348)	σ_{12}	107	(104,109)	γ_{12}	255	(254,256)
μ_{13}	345	(341,349)	σ_{13}	108	(106,111)	γ_{13}	241	(240,241)
μ_{14}	340	(335,344)	σ_{14}	110	(107,113)	γ_{14}	231	(230,231)
μ_{15}	319	(314,324)	σ_{15}	116	(113,119)	γ_{15}	240	(239,241)
μ_{16}	303	(299,308)	σ_{16}	113	(110,116)	γ_{16}	243	(240,246)
μ_{17}	291	(286,296)	σ_{17}	107	(104,111)	γ_{17}	212	(208,218)
μ_{18}	286	(280,292)	σ_{18}	105	(100,109)	γ_{18}	202	(196,209)
μ_{19}	285	(279,292)	σ_{19}	104	(100,109)	γ_{19}	202	(193,211)
μ_{20}	285	(278,292)	σ_{20}	104	(99,109)	γ_{20}	202	(191,213)
μ_{21}	285	(278,293)	σ_{21}	104	(99,109)	γ_{21}	202	(189,214)
μ_{22}	285	(277,293)	σ_{22}	104	(99,110)	γ_{22}	202	(187,216)

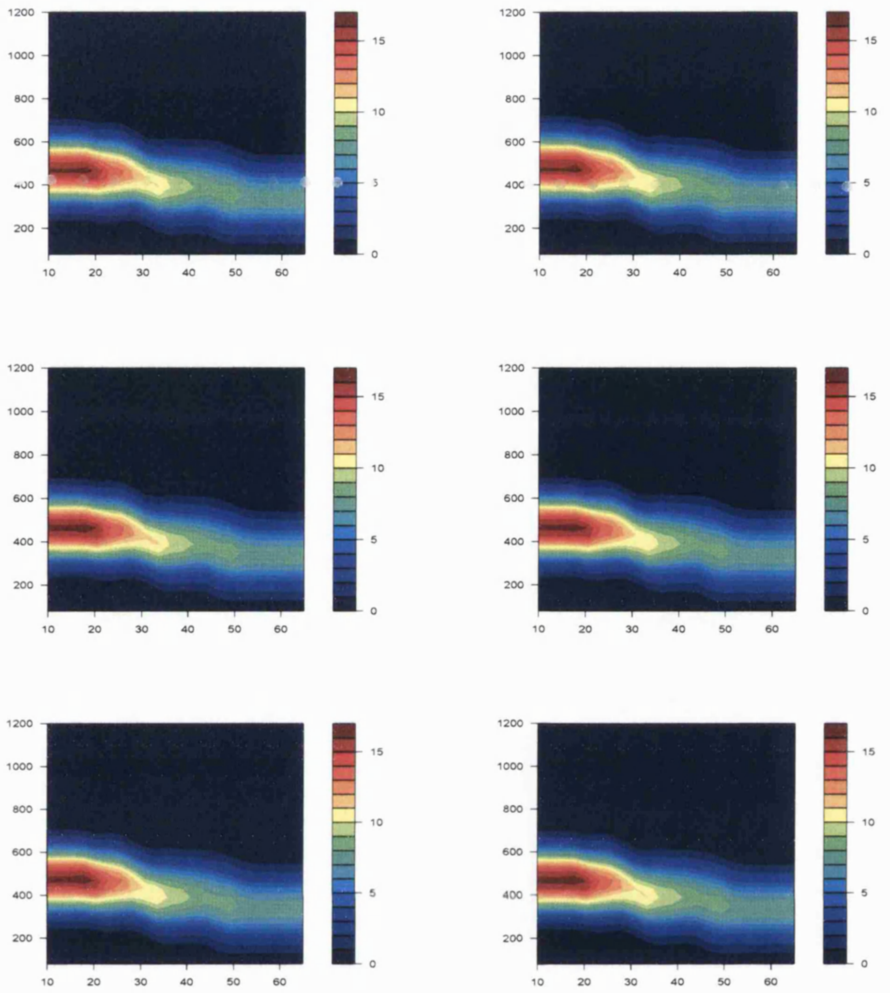


Figure 8-27: *The ionosphere maps in units of* $10^{11}\,el/m^2$ *obtained at the resulting posterior mean when all smoothing parameters are held fixed based on the following starting values and seeds: MIDAS with seed 1 (Top left), MIDAS with seed 10 (Top right), MIDAS with seed 100 (Middle left), MIDAS with seed 1000 (Middle right), random with seed 100 (Bottom left) and flat with seed 100 (Bottom right).*

The sampler is then applied to the available set of 572 TEC values. Two methods are considered. In the first method, fixed sensible values of the smoothing parameters are used, whereas in the second method these parameters are updated regularly starting from the same sensible set of smoothing values used to run the empirical Bayesian

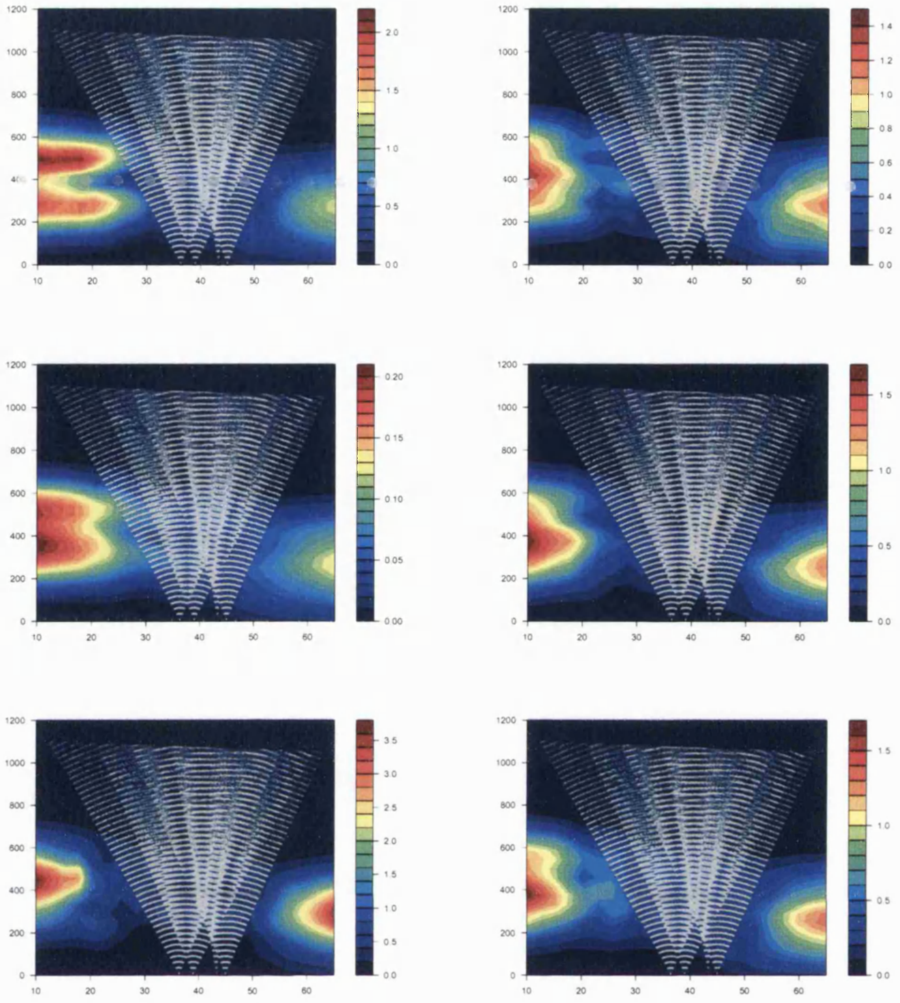


Figure 8-28: *The length of the 95% credible interval overlayed by all signal paths associated with the fixed smoothing parameters restorations obtained when all smoothing parameters are held fixed based on the following starting values and seeds: MIDAS with seed 1 (Top left), MIDAS with seed 10 (Top right), MIDAS with seed 100 (Middle left), MIDAS with seed 1000 (Middle right), random with seed 100 (Bottom left) and flat with seed 100 (Bottom right).*

approach's sampler. The acceptance rates in the first three principal components' directions are between 33% and 39%, i.e slightly far from the derived optimal acceptance rate, 44%, although step sizes of $2.4\sqrt{\lambda_i}$, are used. This indicates that the posterior model is not perfectly Gaussian as the scatter plots produced by the mixed sampler also reveal.

The results obtained by the PCMCMC are very encouraging especially when the fixed values of the model smoothing parameters are used. In this case, the integrated autocorrelation times have an average of 7.4. Even when the interaction parameters are updated regularly, despite higher integrated autocorrelation time estimates for some parameters, the algorithm performed much better than the mixed MCMC sampler. Here, we conclude that the PCMCMC is very sensitive to the smoothing parameter values used because as these estimates descend below the MPL fixed estimates, the degree of departure from normality increases. Thus, the smoothing parameters in this application, beside their main use in controlling the spatial dependence present within the ionosphere, also play a similar role to that of the normality controller parameter a of the four non Gaussian examples of section 7.2.5.

The maps of free electrons generated at the posterior means of both approaches reveal no significant difference. Further, in both approaches the TEC data are fitted very well in general, with a small estimated error of about 0.7 TECU, although the plane non-central regions suffer from a lack of information. In contrast, the error associated with MIDAS is around 61 TECU, i.e. approximately 100 times larger. This mainly indicates that our results fit the data not only much better than MIDAS but is also able to produce a very good fit as Figures 8-25 and 8-27 also reveal. In addition, our results highlight the importance and suitability of our model spatial interaction parameter values found in this application, and the high ability of our sampling process.

Unlike MIDAS approach where estimates of the unknown electron densities are considered without accounting for the uncertainty related to the model and the estimates, the Bayesian approach also incorporates this uncertainty by providing the user with ranges of all possible values with high probabilities, as those recorded in Tables 8.8 and 8.9.

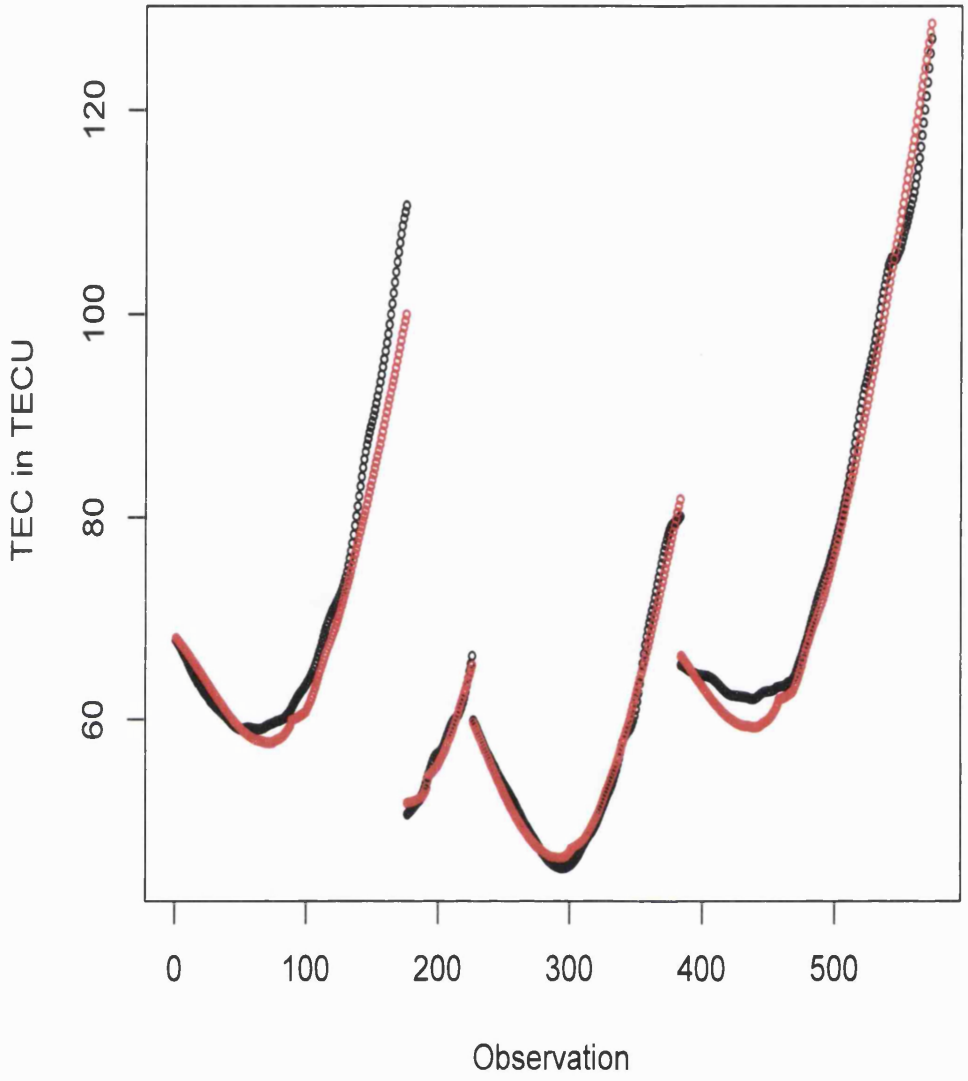


Figure 8-29: *A comparison between the 572 TEC data values Y (black) and estimates of $E(Y)$ (red) associated with MIDAS output.*

Chapter 9

Conclusions and future directions

In this thesis, we have developed a new tomographic approach which can use TEC observations to image the spatial distribution of electron concentration in the ionosphere. Our approach for tackling this problem is indirect because it is based on extracting low-level images of pixel electron-densities from high-level images of vertical electron-density profiles. The new approach consists of two main steps: building a non-linear Bayesian model that captures the expected features of the scene, and developing a reconstruction MCMC algorithm that mixes well.

We worked with a day-time data set gathered from the Navy Navigational Satellite System that consists of a single orbiting satellite and a chain of four ground-based receivers with a geographical configuration that is suitable for reconstructing two-dimensional images of the ionospheric scene. As in many previous experimental studies, the small number of monitoring receivers and their restricted geographical locations limited the ranges of the ray paths in the 2D reconstruction grid. The available prior knowledge about the true scene informs us that neighbouring areas in the ionosphere are likely to be similar. In order to address the issues of lack of information within several regions of the ionosphere and the spatial dependence among neighbouring regions, smoothing based Bayesian techniques have been considered in building the non-linear model proposed in this thesis.

The fundamental rationale behind spatial smoothing is the concept of "borrowing strength" from highly monitored neighbouring regions in the ionosphere so that stable estimates for variables of regions with little or no information can be produced. To accommodate the spatial structure of the ionosphere, GMRFs distributions are used as prior distributions for the vertical profiles. To ensure that the prior distributions used in our methodology are consistent with the nature of our problem, for example, all accepted estimates should be non-negative, we defined a sensible, broad but finite range for each parameter type. Fortunately, these constraints make our prior models proper.

Due to the high-dimensionality and complexity of our posterior model it is extremely difficult to propose estimates of the model variables of interest by non-MCMC sampling approaches. We began our investigation by implementing a standard MCMC algorithm in which both the Metropolis algorithm and the Gibbs sampler are used for a set of simulated data. We used time-series plots and multiple independent realisations of the MCMC approach to assess mixing and convergence of the Markov chain. The results indicate that this sampler may perform inefficiently especially if the reconstruction grid is discretised with few levels, for instance 3 or 4. Reparameterisation ideas are then implemented within the standard MCMC algorithm, and the modified version of this sampler is defined as the mixed algorithm. The latter sampler, when implemented for the same illustrative example, reveals some very good results even when the grid consists of only a few layers.

To avoid prolonged run times for the mixed algorithm for the real data set problem, we found it necessary to start the simulation process from a good initial state. For this purpose, we used estimates of pixel electron densities produced by another tomographic approach known as MIDAS. However, since our model variables are not pixel electron densities, we developed some algebraic formulae by which we can convert the low level image estimates into a high level valid configuration for the adopted posterior density. Further, to avoid prespecifying arbitrary values for our interaction parameters, we implemented ideas of Maximum Pseudo Likelihood (MPL) to obtain estimates for these parameters. We then applied the mixed algorithm for the real TEC measurements using the calculated initial configuration but via two different methods. In the first method, the empirical Bayesian approach is followed, i.e. the smoothing parameters are considered as fixed known constants, whereas in the second method these parameters are allowed to vary during the simulation process using the MPL approach in order to incorporate uncertainty about them. Nevertheless, the latter method is not a fully Bayesian based-approach because no prior distributions are assigned for the smoothing parameters, i.e. the new values of these parameters are not sampled from any distribution, rather they are recalculated regularly using MPL. This approach was introduced by Heikkinen and Högmander (1994). By following this approach we aim to avoid a time consuming procedure and many complicated, if not impossible, calculations that would be required if we were to apply the fully Bayesian approach instead.

The results obtained from both runs of the mixed algorithm applied to the real data reveal a slow mixing problem. However, small residual sum of squares are obtained rapidly and the empirical error associated with the resulting estimates is less than 1.0 TECU, indicating that both the developed model and the reconstruction MCMC sampler are generally good although some high integrated autocorrelation time estimates are obtained. To improve mixing and speed up convergence of the Markov chain, another more sophisticated MCMC algorithm is developed. This sampler is named the Principal Components MCMC (PCMCMC) algorithm because updates in this case are generated with appropriate sized steps and in the approximate principal component directions of the target distribution. To implement this sampler, it is necessary to have the original parameters true variance matrix, or at least an estimate of it. When this matrix is unknown then one can use the standard Metropolis sampler to generate a sample approximately from the desired distribution. Moreover, to shorten the PCMCMC run burn-in phase, the empirical expectations obtained from the standard algorithm run could be used as the desired initial values of the new sampler. In practice, the target distribution variables may have very different scales. Variables with high variances could dominate the major principal components. To avoid this and make all parameters of interest equally important, we suggested scaling all original components of the desired distribution so that each parameter has unit variance.

The PCMCMC approach was initially tested for sampling Gaussian distributions with highly correlated variables. It has been shown that this algorithm performs more efficiently if the proposal Gaussian distributions have variances of size 2.4^2 times the principal components' variances. The optimal acceptance rate and integrated autocorrelation time associated with each principal component are found to be 44% and 4.4. Further applications of this algorithm for simulating not perfectly Gaussian densities revealed that PCMCMC outperforms the standard Metropolis sampling, and although the above optimal values are not exactly met, the integrated autocorrelation times are still below 10 unless the unimodal sampled distribution is very far from normality. We conclude that PCMCMC would be a promising sampling tool as long as directions derived from the variance matrix eigenvalues make an appropriate independent decomposition.

Before implementing the new tomographic package for sampling our posterior distribution with the real data set, we tested its performance on a set of simulated TEC data. In this case, the algorithm revealed outstanding results reflected in the form of a

Markov chain that mixes extremely well and produces very accurate estimates of the retrieved parameters.

The PCMCMC algorithm is then applied to the real TEC observations. The results obtained are very encouraging especially when sampling the posterior model in the framework of the empirical Bayesian approach where carefully chosen fixed estimates of the smoothing parameters are used. The integrated autocorrelation times have an overall average of about 7.4. Even when these parameters are allowed to vary regularly, despite some higher integrated autocorrelation times for some variables, the PCMCMC algorithm outperforms the mixed sampler. It is worth mentioning that the PCMCMC algorithm is highly sensitive to the spatial smoothing parameter values involved in our model. For example, when very small arbitrary values are assigned to these parameters, the PCMCMC mixing was poor. This is because with such small interaction parameter estimates, the degree of deviation from a Normal is increased significantly.

The reconstructed images of the ionospheric scene obtained from the two PCMCMC sampling trails reveal no substantial difference. In general, these images fitted our TEC set very well with a small associated error of about 0.7 TECU. These results reveal the overall suitability of our approach for the ionospheric tomography problem. Moreover, problems caused by the generation of non-positive electron density estimates and reliance of the reconstructed image on the initial image of the ionospheric scene that many existing approaches suffer from are all eliminated.

The approach developed in this study has been designed for a 2D configuration. This is because our TEC measurements are suitable for 2D reconstructions since the satellite and receivers in the Navy Navigational Satellite System from which our data are taken lie approximately in the same plane. However, we can extend our approach to tackle 3D configurations of latitude, longitude and altitude. This is very important because not every TEC set geometry is appropriate for 2D reconstructions. For example, GPS satellites and ground-based receivers are not always in the same plane. Thus a 3D configuration should be used to recover all three spatial dimensions, although this may increase the sampling process CPU time and storage. This makes from our approach a very promising tool for imaging the ionosphere.

For any 3D future ionospheric reconstruction problem we need to consider the 2D neighbourhood structure in defining our spatial distribution rather than the present 1D structure, i.e. two vertical profiles are considered as neighbours in the grid if their corresponding columns are adjacent, so that corner profiles have two neighbours, edge profiles have three, and interior profiles have four.

The Bayesian model and MCMC algorithms developed in this thesis, although specifically constructed for TEC data inversion, could be applied more widely especially for other fields of tomography where spatial correlations about the true values of the physical quantities are to be taken into consideration. In these cases, the major advantage of using the Bayesian approach is the ability to estimate the uncertainty associated with the parameters and thus the constructed images.

Appendix A

Solving the system of the 3 electron density equations

Suppose that d_a, d_b and d_c are the electron density values within a column at heights a, b and c, where $a < b < c$. Then

$$\begin{aligned} d_a &= \frac{\gamma}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(a-\mu)^2}, & \text{(A.1)} \\ d_b &= \frac{\gamma}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(b-\mu)^2}, & \text{(A.2)} \\ d_c &= \frac{\gamma}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(c-\mu)^2}. & \text{(A.3)} \end{aligned}$$

This system can be solved for μ, σ^2 and γ via the following steps:

1. solve (A.1) for γ. This gives

$$\gamma = \sqrt{2\pi}\sigma d_a \text{exp}\left(\frac{1}{2\sigma^2}(a-\mu)^2\right). \tag{A.4}$$

2. substitute γ in (A.2) and then solve for μ. This gives

$$\mu = \frac{a^2 - b^2 - 2\sigma^2 \ln(d_b) + 2\sigma^2 \ln(d_a)}{2(a-b)}. \tag{A.5}$$

3. substitute γ and μ in (A.3) and then solve for σ^2. This gives

$$\sigma^2 = \frac{(a-c)[(a-b)(a+c) - (a^2 - b^2)]}{2[(a-b)(\ln(d_c) - \ln(d_a)) - (a-c)(\ln(d_b) - \ln(d_a))]}. \tag{A.6}$$

Appendix B

The Jacobian of the several parameters transformation

If the joint probability density of d_a, d_b and d_c is known, then the joint probability density of μ, σ^2 and γ could be determined by using the Transformation Technique. The **Jacobian** of this transformation is given by $\left|\frac{\partial(d_a,d_b,d_c)}{\partial(\mu,\sigma^2,\gamma)}\right|$, i.e.,

$$\begin{aligned}
\mathrm{J} &= \begin{vmatrix} \frac{\partial d_a}{\partial \mu} & \frac{\partial d_a}{\partial \sigma^2} & \frac{\partial d_a}{\partial \gamma} \\ \frac{\partial d_b}{\partial \mu} & \frac{\partial d_b}{\partial \sigma^2} & \frac{\partial d_b}{\partial \gamma} \\ \frac{\partial d_c}{\partial \mu} & \frac{\partial d_c}{\partial \sigma^2} & \frac{\partial d_c}{\partial \gamma} \end{vmatrix} \\
&= \frac{\partial d_a}{\partial \mu}\begin{vmatrix} \frac{\partial d_b}{\partial \sigma^2} & \frac{\partial d_b}{\partial \gamma} \\ \frac{\partial d_c}{\partial \sigma^2} & \frac{\partial d_c}{\partial \gamma} \end{vmatrix} - \frac{\partial d_a}{\partial \sigma^2}\begin{vmatrix} \frac{\partial d_b}{\partial \mu} & \frac{\partial d_b}{\partial \gamma} \\ \frac{\partial d_c}{\partial \mu} & \frac{\partial d_c}{\partial \gamma} \end{vmatrix} + \frac{\partial d_a}{\partial \gamma}\begin{vmatrix} \frac{\partial d_b}{\partial \mu} & \frac{\partial d_b}{\partial \sigma^2} \\ \frac{\partial d_c}{\partial \mu} & \frac{\partial d_c}{\partial \sigma^2} \end{vmatrix} \\
&= \frac{\partial d_a}{\partial \mu}\left(\frac{\partial d_b}{\partial \sigma^2}\frac{\partial d_c}{\partial \gamma} - \frac{\partial d_b}{\partial \gamma}\frac{\partial d_c}{\partial \sigma^2}\right) - \frac{\partial d_a}{\partial \sigma^2}\left(\frac{\partial d_b}{\partial \mu}\frac{\partial d_c}{\partial \gamma} - \frac{\partial d_b}{\partial \gamma}\frac{\partial d_c}{\partial \mu}\right) \\
&\quad + \frac{\partial d_a}{\partial \gamma}\left(\frac{\partial d_b}{\partial \mu}\frac{\partial d_c}{\partial \sigma^2} - \frac{\partial d_b}{\partial \sigma^2}\frac{\partial d_c}{\partial \mu}\right). \qquad \text{(B.1)}
\end{aligned}$$

Suppose that $j \in \{a, b, c\}$, then

$$\begin{aligned}
\frac{\partial d_j}{\partial \mu} &= \frac{\gamma}{\sqrt{2\pi}\sigma}\left(-\frac{1}{2\sigma^2}\right)(-2(j-\mu))\,e^{-\frac{1}{2\sigma^2}(j-\mu)^2} \\
&= \frac{\gamma(j-\mu)}{\sqrt{2\pi}\sigma^3}e^{-\frac{1}{2\sigma^2}(j-\mu)^2}.
\end{aligned} \qquad \text{(B.2)}$$

$$\begin{aligned}
\frac{\partial d_j}{\partial \sigma^2} &= \frac{\partial}{\partial \sigma^2}\left[\frac{\gamma}{\sqrt{2\pi}}\left(\sigma^2\right)^{\frac{-1}{2}} e^{-\frac{1}{2}(j-\mu)^2\left(\sigma^2\right)^{-1}}\right] \\
&= \frac{\gamma}{\sqrt{2\pi}}[\left(\sigma^2\right)^{\frac{-1}{2}}\left(\frac{-1}{2}\right)(j-\mu)^2(-1)\left(\sigma^2\right)^{-2} \\
&\quad \times \; e^{-\frac{1}{2}(j-\mu)^2\left(\sigma^2\right)^{-1}} - \frac{1}{2}\left(\sigma^2\right)^{-\frac{1}{2}-1} \times \; e^{-\frac{1}{2}(j-\mu)^2\left(\sigma^2\right)^{-1}}] \\
&= \frac{\gamma}{2\sqrt{2\pi}}\left[\frac{(j-\mu)^2}{(\sigma^2)^{\frac{5}{2}}} - \frac{1}{(\sigma^2)^{\frac{3}{2}}}\right] e^{-\frac{1}{2\sigma^2}(j-\mu)^2} \\
&= \frac{\gamma}{2\sqrt{2\pi}\sigma^5}[(j-\mu)^2 - \sigma^2]e^{-\frac{1}{2\sigma^2}(j-\mu)^2}. \qquad \text{(B.3)}
\end{aligned}$$

$$\frac{\partial d_j}{\partial \gamma} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(j-\mu)^2}. \qquad \text{(B.4)}$$

Now the jacobian term J could be determined as follows:

1. The first term is
$$\frac{\partial d_a}{\partial \mu}\left(\frac{\partial d_b}{\partial \sigma^2}\frac{\partial d_c}{\partial \gamma} - \frac{\partial d_b}{\partial \gamma}\frac{\partial d_c}{\partial \sigma^2}\right)$$

   where

$$\begin{aligned}
\frac{\partial d_b}{\partial \sigma^2}\frac{\partial d_c}{\partial \gamma} &= \frac{\gamma}{2\sqrt{2\pi}\sigma^5}\left((b-\mu)^2 - \sigma^2\right) e^{-\frac{1}{2\sigma^2}(b-\mu)^2} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(c-\mu)^2} \\
&= \frac{\gamma\left((b-\mu)^2 - \sigma^2\right)}{4\pi\sigma^6} e^{-\frac{1}{2\sigma^2}\left((b-\mu)^2+(c-\mu)^2\right)} \\
&= \frac{\gamma\left(b^2 - 2b\mu + \mu^2 - \sigma^2\right)}{4\pi\sigma^6} e^{-\frac{1}{2\sigma^2}\left(b^2+c^2-2b\mu-2c\mu+2\mu^2\right)}. \qquad \text{(B.5)}
\end{aligned}$$

   Similarly,

$$\begin{aligned}
\frac{\partial d_b}{\partial \gamma}\frac{\partial d_c}{\partial \sigma^2} &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(b-\mu)^2} \frac{\gamma\left((c-\mu)^2 - \sigma^2\right)}{2\sqrt{2\pi}\sigma^5} e^{-\frac{1}{2\sigma^2}(c-\mu)^2} \\
&= \frac{\gamma\left(c^2 - 2c\mu + \mu^2 - \sigma^2\right)}{4\pi\sigma^6} e^{-\frac{1}{2\sigma^2}\left(b^2+c^2-2b\mu-2c\mu+2\mu^2\right)}. \qquad \text{(B.6)}
\end{aligned}$$

   By subtracting (B.6) from (B.5) we obtain

$$\left(\frac{\partial d_b}{\partial \sigma^2}\frac{\partial d_c}{\partial \gamma} - \frac{\partial d_b}{\partial \gamma}\frac{\partial d_c}{\partial \sigma^2}\right) = \frac{\gamma\left(b^2 + c^2 - 2b\mu - 2c\mu + 2\mu^2\right)}{4\pi\sigma^6} e^{-\frac{1}{2\sigma^2}\left(b^2+c^2-2b\mu-2c\mu+2\mu^2\right)}. \qquad \text{(B.7)}$$

Hence,

$$\begin{aligned}\frac{\partial d_a}{\partial \mu}\left(\frac{\partial d_b}{\partial \sigma^2}\frac{\partial d_c}{\partial \gamma}-\frac{\partial d_b}{\partial \gamma}\frac{\partial d_c}{\partial \sigma^2}\right) &= \frac{\gamma\left(a-\mu\right)}{\sqrt{2\pi}\sigma^3}e^{-\frac{1}{2\sigma^2}(a-\mu)^2} \\ &\times \frac{\gamma\left(b^2+c^2-2b\mu-2c\mu+2\mu^2\right)}{4\pi\sigma^6} \\ &\times e^{-\frac{1}{2\sigma^2}\left(b^2+c^2-2b\mu-2c\mu+2\mu^2\right)}. \end{aligned} \tag{B.8}$$

2. The second term is $$\frac{\partial d_a}{\partial \sigma^2}\left(\frac{\partial d_b}{\partial \mu}\frac{\partial d_c}{\partial \gamma}-\frac{\partial d_b}{\partial \gamma}\frac{\partial d_c}{\partial \mu}\right)$$

   where

$$\begin{aligned}\frac{\partial d_b}{\partial \mu}\frac{\partial d_c}{\partial \gamma} &= \frac{\gamma\left(b-\mu\right)}{\sqrt{2\pi}\sigma^3}e^{-\frac{1}{2\sigma^2}(b-\mu)^2}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2\sigma^2}(c-\mu)^2} \\ &= \frac{\gamma\left(b-\mu\right)}{2\pi\sigma^4}e^{-\frac{1}{2\sigma^2}\left((b-\mu)^2+(c-\mu)^2\right)} \\ &= \frac{\gamma\left(b-\mu\right)}{2\pi\sigma^4}e^{-\frac{1}{2\sigma^2}\left(b^2+c^2-2b\mu-2c\mu+2\mu^2\right)}. \end{aligned} \tag{B.9}$$

   Similarly,

$$\begin{aligned}\frac{\partial d_b}{\partial \gamma}\frac{\partial d_c}{\partial \mu} &= \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2\sigma^2}(b-\mu)^2}\frac{\gamma\left(c-\mu\right)}{\sqrt{2\pi}\sigma^3}e^{-\frac{1}{2\sigma^2}(c-\mu)^2} \\ &= \frac{\gamma\left(c-\mu\right)}{2\pi\sigma^4}e^{-\frac{1}{2\sigma^2}\left(b^2+c^2-2b\mu-2c\mu+2\mu^2\right)}. \end{aligned} \tag{B.10}$$

   By subtracting (B.10) from (B.9) we get

$$\begin{aligned}\left(\frac{\partial d_b}{\partial \mu}\frac{\partial d_c}{\partial \gamma}-\frac{\partial d_b}{\partial \gamma}\frac{\partial d_c}{\partial \mu}\right) &= \\ &\frac{-\gamma}{2\pi\sigma^4}e^{-\frac{1}{2\sigma^2}\left(b^2+c^2-2b\mu-2c\mu+2\mu^2\right)}. \end{aligned} \tag{B.11}$$

Hence,

$$\frac{\partial d_a}{\partial \sigma^2}\left(\frac{\partial d_b}{\partial \mu}\frac{\partial d_c}{\partial \gamma} - \frac{\partial d_b}{\partial \gamma}\frac{\partial d_c}{\partial \mu}\right) = \frac{\gamma\left((a-\mu)^2-\sigma^2\right)}{2\sqrt{2\pi}\sigma^5}e^{-\frac{1}{2\sigma^2}(a-\mu)^2}\frac{(b-c)\gamma}{2\pi\sigma^4} \times \quad e^{-\frac{1}{2\sigma^2}\left(b^2+c^2-2b\mu-2c\mu+2\mu^2\right)}. \tag{B.12}$$

3. The third term is
$$\frac{\partial d_a}{\partial \gamma}\left(\frac{\partial d_b}{d\mu}\frac{\partial d_c}{\partial \sigma^2} - \frac{\partial d_b}{\partial \sigma^2}\frac{\partial d_c}{\partial \mu}\right)$$
where

$$\frac{\partial d_b}{\partial \mu}\frac{\partial d_c}{\partial \sigma^2} = \frac{\gamma\,(b-\mu)}{\sqrt{2\pi}\sigma^3}e^{-\frac{1}{2\sigma^2}(b-\mu)^2}\frac{\gamma\left((c-\mu)^2-\sigma^2\right)}{2\sqrt{2\pi}\sigma^5}e^{-\frac{1}{2\sigma^2}(c-\mu)^2}. \tag{B.13}$$

Similarly,

$$\frac{\partial d_b}{\partial \sigma^2}\frac{\partial d_c}{\partial \mu} = \frac{\gamma\left((b-\mu)^2-\sigma^2\right)}{2\sqrt{2\pi}\sigma^5}e^{-\frac{1}{2\sigma^2}(b-\mu)^2}\frac{\gamma\,(c-\mu)}{\sqrt{2\pi}\sigma^3}e^{-\frac{1}{2\sigma^2}(c-\mu)^2}. \tag{B.14}$$

By further calculations, we obtain $\frac{\partial d_a}{\partial \gamma}\left(\frac{\partial d_b}{\partial \mu}\frac{\partial d_c}{\partial \sigma^2} - \frac{\partial d_b}{\partial \sigma^2}\frac{\partial d_c}{\partial \mu}\right) =$

$$\gamma^2\frac{(bc^2-cb^2+b\mu^2-c\mu^2-c^2\mu+b^2\mu+c\sigma^2-b\sigma^2+2c\mu^2-2b\mu^2)}{4\pi\sigma^8} \times \quad e^{-\frac{1}{2\sigma^2}(a^2+b^2+c^2-2a\mu-2b\mu-2c\mu+3\mu^2)}. \tag{B.15}$$

Therefore, the Jacobian of the transformation is given by

$$\mathrm{J} = \frac{\gamma^2\left(ab^2-a^2b-ac^2+a^2c+bc^2-b^2c\right)}{4\pi\sqrt{2\pi}\sigma^9} \times \quad e^{-\frac{1}{2\sigma^2}\left(a^2+b^2+c^2-2a\mu-2b\mu-2c\mu+3\mu^2\right)}. \tag{B.16}$$

Bibliography

[1] Amit, Y., and Grenander, U. (1991), Comparing sweep strategies for stochastic relaxation, *Journal of Multivariate Analysis* 37, 197-222.

[2] Austen, J. R., Franke, and S. J., Liu, C. H. (1988), Ionospheric imaging using computerized tomography, *Radio Science* 23, 299-307.

[3] Andreeva, E. S., Galinov, A. V., Kunitsyn, V. E., Mel'nichenko, Yu. A., Tereshchenko, E. D., Filimonov, M. A., and Chernykov, S. M. (1990), Radiotomographic reconstruction of ionisation dip in the plasma near the Earth, *Journal of Experimental and Theoretical Physics Letters* 52, 145-148.

[4] Besag, J. (1974), Spatial interaction and the statistical analysis of lattice systems (with Discussion), *Journal of the Royal Statistical Society, Series B* 36, 192-225.

[5] Besag, J. (1975), Statistical analysis of non-lattice data, *The Statistician* 24(3), 179-195.

[6] Besag, J. (1989), Towards Bayesian image analysis, *Journal of Applied Statistics* 16(3), 395-407.

[7] Besag, J., and Green, P. J. (1993), Spatial statistics and Bayesian computation (with Discussion). *Journal of the Royal Statistical Society*, Series B 55, 25-37.

[8] Besag, J., York, J., and Mollié, A. (1991), Bayesian image restoration with two applications in spatial statistics (with discussion). *Annals of the Institute of Statistical Mathematics* 43(1), 1-59.

[9] Bilitza, D. (1990), The International Reference Ionosphere 1990, *National Space Science Data Center, Ref. 90-20*, Greenbelt, Maryland, USA.

[10] Brooks, S. P., and Gelman, A. (1998), General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics* 7, 434-455.

[11] Brooks, S. P., and Roberts, G. O. (1998), Assessing convergence of Markov Chain Monte Carlo algorithms. *Statistics and Computing* 8, 319-335.

[12] Censor, Y. (1983), Finite series-expansion reconstruction methods, *Proceedings of the Institute of Electrical and Electronics Engineers* 71, 409-419.

[13] Chapman, M. A. (1931), The absorption and dissociation ionisation effect of monochromatic radiation in a rotating Earth, *Proceedings of Physical Society* 43(1), 26.

[14] Chatfield, C. (2003), *The Analysis Of Time Series: An Introduction.* 6th ed., Chapman & Hall, London.

[15] Chatfield, C., and Collins, A. (1980), *Introduction to multivariate analysis*, Chapman & Hall, London.

[16] Cowles, M.K., and Carlin, B.P. (1996), Markov Chain Monte Carlo convergence diagnostics: a comparative review. *Journal of American Statistical Association* 91, 883-904.

[17] Davies, K. (1990), *Ionospheric Radio*, Peter Peregrinus Ltd., London.

[18] Dillon, W. R., and Goldstein, M. (1984), *Multivariate Analysis: Methods And Applications.* John Wiley & Sons, New York.

[19] Fremouw, E. J., Secan J. A., and Howe, B. M. (1992), Application of stochastic inverse theory to ionospheric tomography, *Radio Science* 27, 721-732.

[20] Fremouw, E. J., Secan J. A., Bussey, R. M., and Howe, B. M. (1994), A status report on applying Discrete Inverse theory to ionospheric tomography, *International Journal of Imaging Systems and Technology* 5, 97-105.

[21] Gelfand, A. E., and Smith, A. F. (1990), Sampling-based approaches to calculating marginal densities, *Journal of the American Statistical Association* 85, 398-409.

[22] Gelfand, A. E., Sahu, S. K., and Carlin, B. P. (1995), Efficient parametrization for normal linear mixed models, *Biometrika* 82, 479-488.

[23] Gelman, A. (2004), Parameterization and Bayesian modeling. *Journal of the American Statistical Association* 99, 537-545.

[24] Gelman, A. (2005), Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis* 1, 1-19.

[25] Gelman, A., Roberts, G. O., and Gilks, W. R. (1996), Efficient Metropolis jumping rules. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. (eds.), *Bayesian Statistics* 5, 599-608, New York: Oxford University Press.

[26] Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (2004), *Bayesian Data Analysis*, second edition. Chapman & Hall, London.

[27] Gelman, A., and Rubin, B.R. (1992), Inference from iterative simulation using multiple sequences, *Statistical Science* 7, 457-511.

[28] Geman, S. and Geman, D. (1984), Stochastic relaxation, Gibbs Distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 721-741.

[29] Geyer, C. J. (1992), Practical Markov Chain Monte Carlo (with Discussion), *Statistical Science* 7, 473-511.

[30] Geyer, C.J. and Thompson, E. A. (1995), Annealing Markov chain Monte Carlo with applications to ancestral inference, *Journal of American Statistical Association* 90, 909-920.

[31] Gordon, R., Bender, R., and Herman, G. T. (1970), Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography, *Journal of Theoretical Biology* 29, 471-481.

[32] Green, P. J. (1998), Penalized Likelihood, *Encyclopaedia of Statistical Sciences, update volume 3*, 578-586, Wiley, New York.

[33] Green, P. J., and Han, X. -L. (1992), Metropolis methods, Gaussian proposals, and antithetic variables, *Lecture Notes in Statistics* 74, 142-164.

[34] Green, P. J., and Silverman, B. W. (1994), *Nonparametric Regression And Generalised Linear Models: A Roughness Penalty Approach*, Chapman and Hall, London.

[35] Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (1996), *Markov Chain Monte Carlo In Practice.* Chapman & Hall, London.

[36] Gull, S., and Daniell, G. J. (1978), Image reconstruction from incomplete and noisy data. *Nature* 272, 686-690.

[37] Gustavsson, M., Ivansson, S., Moren, P., and Pihl, J. (1986), Seismic borehole tomography - measurement system and field studies. *Proceedings of the Institute of Electrical and Electronics Engineers* 74, 339-346.

[38] Hajj, G. A., Ibañez-Meier, R., Kursinski, E. R., and Romans, L. J. (1994), Imaging the ionosphere with the Global Positioning System, *International Journal of Imaging Systems Technology* 5, 174-184.

[39] Hargreaves, J. K. (1992), *The Solar-Terrestrial Environment.* Cambridge University Press, Cambridge.

[40] Hastings, W. K. (1970), Monte Carlo sampling methods using Markov Chains and their applications, *Biometrika* 57(1), 97-109.

[41] Heikkinen, J., and Högmander, H. (1994), Fully Bayesian approach to image restoration with an application in Biogeography, *Applied Statistics* 43(4), 569-582.

[42] Herman, G. T., Lent, A., Lutz, P. H. (1975), Iterative relaxation methods for image reconstruction, *Proceedings of the 1975 Annual Conference of Association for Computing Machinery*, 169-174.

[43] Higdon, D., Lee, H., and Holloman, C. (2003), Markov chain Monte-Carlo-based approaches for inference in computationally intensive inverse problems, *Bayesian Statistics*, 7, eds. J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. David, D. Heckerman, A. F. M. Smith, and M. West, 181-197. Oxford: Oxford University press.

[44] Hobert, J.P. and Casella, G. (1996), The effect of improper priors on Gibbs sampling in hierarchical linear mixed models, *Journal of the American Statistical Association* 91, 1461-1473.

[45] Jolliffe, I. T. (1986), *Principal Component Analysis.* Springer Verlag, New York.

[46] Kass, R.E., Carlin, B.P., Gelman, A. and Neal, R.M. (1998), Markov Chain Monte Carlo in practice: A roundtable discussion, *American Statistician* 52, 93-100.

[47] Kersley, L., Heaton, J. A. T., Pryse, S. E., and Raymund, T. D. (1993), Experimental ionosperic tomography with ionosonde input and EISCAT verfication, *Annales Geophysicae* 11, 1064-1074.

[48] Knorr-Held, L., and Rue, H. (2002), On block updating in Markov random field models for disease mapping, *Scandinavian Journal of Statistics* 29(4), 597-614.

[49] Kristensson, G. (1986), Inverse problems for acoustic waves using the penalised likelihood method, *Inverse problems* 2, 461-479.

[50] Lee, M., Rangarajan, A., Zubal, I. G., and Gindi, G. (1995), Using ground truth data to design priors in Bayesian SPECT reconstruction, *Information Processing in Medical Imaging*, Y. Bizais, C. Barillot, and R. DiPaola, Eds. Dordrecht, the Netherlands: Kluwer, 2738.

[51] Lent, A. (1977), A convergent algorithm for maximum entropy image restoration, with a medical X-ray application, *Image Analysis and Evaluation*, R. Shaw, Ed., Society of photographic and scientific engineers. New York, 45-57.

[52] Liu, J., Wong, W., and Kong, A. (1994), Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika* 81, 27-40.

[53] Markkanen, M., Lehtinen, M., Nygrén, T., Pirtilä, J., Henelius, P., Vilenious E., Tereshchenko, E. D., Khudokon, B. Z. (1995), Bayesian approach to satellite radiotomography with applications in the Scandinavian sector, *Annales Geophysicae* 13, 1277-1287.

[54] Marroquin, J., Mitter, S., and Poggio, T. (1987), Probabilistic solution of ill-posed problems in computational vision, *Journal of the American Statistical Association* 82(397), 76-89.

[55] 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), 1087-1-92.

[56] Mitchell, C. N., Kersley, L., Heaton, J. A., and Pryse, S. E. (1997), Determination of the vertical electron-density profile in ionospheric tomography: experimental results, *Annales Geophysicae*, 15, 747-752.

[57] Mitchell, C. N., Spencer, P. S. J. (2003), A three-dimensional time-dependent algorithm for ionospheric imaging using GPS, *Annals of Geophysics* 46(4), 687-696.

[58] Munk, W., and Wunch, C. (1979), Ocean acoustic tomography: A scheme for large-scale monitoring, *Deep Sea Research* 26A, 123-161.

[59] Poggio, T., Torre, V., and Koch, C. (1985), Computational vision and regularization theory, *Nature* 317, 314-319.

[60] Pryse, S. E., Kersley, L. (1992), A preliminary experimental test of ionospheric tomography. *Journal of Atmospheric and Terrestrial Physics* 54(7/8), 1007-1012.

[61] Pryse, S. E., Mitchell, C. N., Heaton, J. A. T., and Kersley, L. (1995), Travelling ionosperic disturbances imaged by tomographic techniques, *Annales Geophysicae* 13, 1325-1330.

[62] Raftery, A. E. and Lewis, S. M. (1992), How many iterations in the Gibbs sampler?, In *Bayesian Statistics* 4 (eds J. M. Bernardo, A. F. M. Smith, A. P. Dawid and J. O. Berger). Oxford: Oxford University Press.

[63] Raymund, T. D., Austen, J. R., Franke, S. J., Liu, C. H., Klobuchar, J. A., and Stalker, J. (1990), Applications of computerized tomography to the investigation of ionospheric structures, *Radio Science* 25, 771-789.

[64] Raymund, A. E., Franke, S. J., and Yeh, K. C. (1994), Ionospheric tomography: its limitations and reconstruction methods, *Journal of Atmospheric and Terrestrial Physics* 56(5), 637-657.

[65] Raymund, T. D., Pryse, S. E., Kersley, L., and Heaton, J. A. T. (1993), Tomographic reconstruction of ionospheric electron density with European incoherent scatter radar verification, *Radio Science* 28(5), 811-817.

[66] Ripley, B. D. (1987), *Stochastic Simulation*, John Wiley and Sons, New York.

[67] Roberts, G. O. and Sahu, S. K. (1997), Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler, *Journal of the Royal Statistical Society* Series B 59, 291-317.

[68] Rue, H.(2001), Fast sampling of Gaussian Markov random fields, *Journal of the Royal Statistical Society* Series B 63, 325-338.

[69] Rue, H. and Held, L. (2005), *Gaussian Markov Random Fields: Theory And Applications*, volume 104 of Monographs on Statistics and Applied Probability. Chapman & Hall, London.

[70] Smith, A. F. M., and Roberts, G. O. (1993), Bayesian computation via the Gibbs sampler and related Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society* Series B 55, 3-23.

[71] Spencer, PSJ., and Mitchell, C. N. (2001), Multi-instrument Data Analysis System, *Proceedings of the International Beacon Satellite Symposium*, Boston, MA, 4-6.

[72] Takauchi, Y., and Evans, J. (1995). Teleseismic tomography of the Loma Prieta aerthquake region, California: implications for strain partitioning, *Geophysical Research Letters* 22, 2203-2206.

[73] Tikhonov, A. N., and Arsenin, V. Y. (1977), *Solutions Of Ill-Posed Prolems*, DC: Winston & Sons, Washington.

[74] Wahba, G. (1990), *Spline Models For Observational Data.* Philadelphia: SIAM.

[75] Winkler, G. (2003), *Image Analysis, Random Fields And Markov Chain Monte Carlo Methods: A Mathematical Introduction*, second edition. Springer, Berlin.