BAYESIAN HIERARCHICAL MODELING AND MARKOV CHAIN SIMULATION FOR CHRONIC WASTING DISEASE

by

Christopher H. Mehl

Bachelor of Science, University of Colorado at Denver, 1998

A thesis submitted to the

University of Colorado at Denver

in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Applied Mathematics

2004

This thesis for the Doctor of Philosophy

degree by

Christopher H. Mehl

has been approved

by

Craig J. Johns

Lynn S. Bennethum

Stephan R. Sain

Douglas W. Nychka

Date

Mehl, Christopher H. (Ph.D., Applied Mathematics)

Bayesian Hierarchical Modeling and Markov Chain Simulation for Chronic Wasting Disease

Thesis directed by Assistant Professor Craig J. Johns

ABSTRACT

In this thesis, a dynamic spatial model for the spread of Chronic Wasting Disease in Colorado mule deer is derived from a system of differential equations that captures the qualitative spatial and temporal behaviour of the disease. These differential equations are incorporated into an empirical Bayesian hierarchical model through the unusual step of deterministic autoregressive updates. Spatial effects in the model are described directly in the differential equations rather than through the use of correlations in the data. The use of deterministic updates is a simplification that reduces the number of parameters that must be estimated, yet still provides a flexible model that gives reasonable predictions for the disease. The posterior distribution generated by the data model hierarchy possesses characteristics that are atypical for many Markov chain Monte Carlo simulation techniques. To address these difficulties, a new MCMC technique is developed that has qualities similar to recently introduced tempered Langevin type algorithms. The methodology is used to fit the CWD model, and posterior parameter estimates are then used to obtain predictions about Chronic Wasting Disease.

This abstract accurately represents the content of the candidate's thesis. I recommend its publication.

Signed _____
Craig J. Johns

DEDICATION

This thesis is dedicated to my lovely wife, Jennifer. She has loved and supported me for more than four years, putting up with, among other things, boring dinner conversation about deer and statistics. Without her, I would never have made it this far. This is really for you, Jen!

ACKNOWLEDGMENT

This thesis would not have been possible without the generous support of my family, friends, and those that have provided guidance over the years. Thank you Liz and Marcia, for collectively knowing everything. Thanks to my committee, Lynn Bennethum, Steve Sain, and Doug Nychka, for all of their comments and their time. Thanks to my advisor, Craig Johns, for providing his guidance and his friendship. Most of all, I would like to thank Karen Kafadar for teaching me most of what I know about statistics.

CONTENTS

FIGURES

Figure

TABLES

Table

1. Introduction

One of the most fascinating and important branches of population biology is the modeling of contagious phenomenon, or epidemics. The study of epidemics is integral to the survival of humanity. The potential impact of a disease as it moves through a society can be frightening. For example, in fourteenth century Europe, 25 million people died from the Black Death (pulmonary plague, out of a population of about 100 million. An estimated half of the 3.5 million Aztecs died from smallpox. More recently, in 1919, more than 20 million people died globally in the influenza pandemic [63, 74]. Today, there is widespread awareness and attention given to diseases such as AIDS [52, 63], as well as non-infectious illnesses such as heart disease and cancer. There are also widespread cases of less glamorous diseases such as malaria, which has an estimated 350 million cases in endemic areas [63, 74].

The potential devastation that can be caused by an infectious disease has caught the attention of both the news media and the public. Potential biological attacks aside, natural diseases still pose a grave threat to the public. Recent outbreaks of Severe Acute Respiratory Syndrome (SARS), West Nile Virus, and avian influenza have all grabbed headlines and prompted the reaction of public health officials across the world. Even illnesses that do not kill are costly to society in terms of lost productivity and reduction of the quality of life. Understanding how diseases evolve dynamically and spatially has and will continue to help public health officials combat these negative effects. Mathematical mod-

els that describe the temporal behaviour and the spatial spread of disease are important in this process.

Mathematical models of epidemics represent a simplification of the biological processes involved, but they are nonetheless important, because they enable researchers to understand disease dynamics and to predict the potential impact of an epidemic. Complicated biological systems often have features that can be reasonably modeled by relatively simple mathematical systems. Epidemics have been studied using deterministic, statistical, and stochastic modeling techniques. These three methods originated at different times and for different reasons, and at times they seem unrelated, but they have been applied jointly with great success [33, 63, 17, 52].

Deterministic models describe the spread of disease through systems of ordinary or partial differential equations [63, 33, 79]. In building a deterministic model the goal is often to give the model as much freedom as necessary to make it biologically plausible. Thus a deterministic model can have a large number of parameters, and is judged by the qualitative behaviour it exhibits and its explanatory capabilities. Little regard is given to available data or parsimony. Unfortunately, "good" but complicated models are often impossible to solve explicitly, so a modeler employs qualitative analysis or numerical approximation. Many deterministic models begin as simple population models, such as the logistic equation, and are derived as particular cases of multiple species competition models, such as the Lotka-Volterra predator-prey model. Variations on these models include time-lag differential equation models and partial differential equations that incorporate age structures. Spatial population processes can

be modeled deterministically using discrete spatial structures such as lattices. More complicated spatial models use deterministic versions of diffusion equations [51, 63, 33]. An important feature of differential equations of epidemics is that simple models with a small number of parameters often have qualitative features that are similar to those of more complicated models. This point will be exploited later in this thesis for building a model for Chronic Wasting Disease.

Stochastic models describe the spread of disease using stochastic differential equations such as contact processes, diffusion process, or continuous birth and death process [52, 63, 42]. Stochastic models consist of a dynamic model with random parameters that contain a noise term. These models are often difficult to solve explicitly. The goal is to capture, as fully as possible, the random nature of population systems, which can result in models that are mathematically rigorous but with no biological interpretation [63]. Simplifying assumptions can make the model tractable, such as the assumption that the noise term is a Brownian motion [41]. Although these assumptions are mathematically desirable and make the model both computationally feasible and easier to manipulate, they may not be biologically appropriate and do not incorporate available data. Often, however, analytical approximations lead to workable models with features that are both biologically interpretable and mathematically elegant [51, 63].

Critical to statistical models of epidemics is the direct incorporation of data to the model [52, 17, 54]. The primary considerations for a statistical model are the type and amount of data, and availability of covariate information. Limited data availability can make the use of many of the more common deterministic or stochastic models infeasible, since these models often have an abundance of

parameters. Statistical models typically have other considerations as well, such as the difficulty of fitting models with nonstandard distributions, and the relative ease and benefit of fitting models with normally distributed data. In many instances, parameters are transformed in an attempt to achieve a model with normality or linearity, simplifying the calculations for the model. One of the great benefits of statistical modeling is the ability to reduce the variability in the parameter estimates through the use of covariates and the introduction of other explanatory variables. However, transforming the parameter space and including the explanatory variables and covariates can sometimes come at the loss of "direct interpretability". For example, some of the spatial disease models employed now do not estimate transmission rates directly, but instead model the log-relative risk, since the transformed space can be modeled as a Gaussian autoregressive process [17, 54]. Through the use of Bayesian hierarchical modeling, the modeler can develop extremely complex but interpretable models. Bayesian hierarchical models relate multiple parameters that are connected by the structure of the problem. For statistical epidemic models, a priori knowledge of the qualitative behaviour of the disease is of less concern than for deterministic or stochastic models. The ease with which the data can be used to fit the model and the confidence of the parameter estimates are of greater concern. It is the data that drive the modeling process.

In the body of this thesis, these three modeling paradigms are discussed in more detail, with the emphasis on the Bayesian hierarchical modeling aspect of statistical modeling. A dynamic spatial model for Chronic Wasting Disease (CWD) in Rocky Mountain Mule deer is introduced. To date, a spatial statisti-

cal model for this disease has not been presented. This model is derived from a system of differential equations, and describes the spatial dynamics directly in the differential equations, rather than through correlations in the data. Because the posterior distributions for the CWD model have some unusual characteristics, a new Markov Chain Monte Carlo simulation technique is constructed that utilized qualities from existing techniques to efficiently sample distributions that live on bounded parameter spaces. The properties of the new MCMC technique are examined and compared to existing methods. The model is fit using the new MCMC technique. The results of the model are discussed and interpreted, and both the modeling method and the simulation method are empirically validated.

Throughout the paper, $[X]$ denotes the marginal distribution of the random variable X, $[X|Y]$ denotes the conditional distribution of X given Y. The multivariate normal distribution with mean μ and covariance matrix Σ is denoted by $N(\mu, \Sigma)$. For matrices, A^{-1} denotes the inverse matrix, and A^T denotes the transpose.

2. Models of Epidemics

Epidemic modeling is an intensely interdisciplinary subject, drawing together a variety of mathematical and statistical disciplines. Modeling approaches range from stochastic processes to deterministic models to empirical data analysis. These approaches are generally guided by biological considerations that help to construct a model and to determine its effectiveness. The approaches that are most important in the construction of a model for Chronic Wasting Disease are deterministic and statistical models.

2.1 Deterministic Models of Epidemics

Deterministic models describe the behaviour of dynamic systems through the use of ordinary or partial differential equations. The study of dynamics systems has its origins in the mid-1600's, with Sir Isaac Newton's work on gravitation and planetary motion [75]. Newton's first work on the subject involved the two-body problem of calculating the motion of the earth around the sun. Later efforts at extending Newton's methods to the three-body problem– for example, the motion of the sun, earth, and moon– led to an eventual realization that some systems of differential equations were analytically unsolvable. However, unsolvable models can still be analyzed qualitatively, and this characteristic is why these models are still used in many applications. For detailed discussions of the qualitative analysis of differential equations, see [83, 8, 75].

Perhaps the earliest use of mathematics to deterministically model a population can be traced to 1202, when Leonardo Fibonacci proposed the famous

sequence of integers to model a rabbit population [33]. The subject's first important early application did not surface until the exponential growth model introduced in 1798 by Thomas Malthus. The exponential growth model results as the solution to Malthus' constant rate of growth assumption, and is a standard example of the simple separable differential equation:

$$\frac{dN}{dt} = rN, \tag{2.1}$$

where N is the size of the population, and r is the growth rate.

Malthus' exponential model is a linear differential equation, and hence the solution predicts unbounded growth or extinction, depending on the choice of parameters and the initial conditions. Neither unbounded growth nor extinction occurs in most populations, which typically tend towards a long term stable population level, called the *carrying capacity*. Thus equation 2.1 is only useful as a short term population model.

The need for a self-limiting population model led to the logistic equation for growth, which was proposed initially by Pierre Verhulst in 1838 [63]. Verhulst's model is more realistic than the Malthus model as a long term description of population growth, and it incorporates a carrying capacity parameter K. The logistic model is now one of the most commonly used models in mathematical biology, and variations appear in many different applications. It is a relatively simple nonlinear model, and it exhibits qualitative behaviour generally observed in many populations. Verhulst suggested the model in 1838 to describe human populations, although it was not until 1920 that his work received notice, when Raymond Pearl searched for a model which has solutions characteristic of a sigmoidal curve and empirically constructed Verhulst's logistic equation [56].

However, the logistic model was not derived *analytically* until 1925 by Alfred James Lotka [45]. A nice derivation of the logistic model is provided in [63]. The logistic model with its sigmoidal curve will be the starting point for the Chronic Wasting Disease model presented in this thesis. The logistic model can be written as:

$$\frac{dN}{dt} = rN(N - K), \tag{2.2}$$

where K is the carrying capacity, $r > 0$ is the constant growth rate, and N is the size of the population. The form of equation 2.2 is relatively simple and leads to an explicit nonlinear model for population growth. The model assumes that the growth rate r is constant, as is the carrying capacity K. These assumptions are somewhat simplistic, in that growth rates typically vary with weather conditions, by regions, or due to other factors. The model can be extended in a variety of ways to include these factors. One implied property of the simple logistic model is that new members of the population immediately contribute to the growth, which is not true for populations that reproduce sexually. Members of these populations must reach a certain age before they are able to reproduce. There are several ways of incorporating this property into the differential equation 2.2. One method is by adding a time-delay term t_d to 2.2, which leads to equation 2.3.

$$\frac{dN}{dt} = rN(N(t - t_d) - K). \tag{2.3}$$

Though somewhat crude, the logistic equation 2.2 is useful in providing a qualitative understanding of simple growth of a single population under limited resources. One generalization of the logistic model that is important to epidemiology is the class of models broadly called *competition models*. These models

describe the interaction of two or more species, and are a simple extension of the one-species logistic model. The general two species model is:

$$\begin{aligned} \frac{dN_1}{dt} &= N_1(r_1 + a_{11}N_1 + a_{12}N_2) \\ \frac{dN_2}{dt} &= N_2(r_2 + a_{21}N_1 + a_{22}N_2), \end{aligned} \tag{2.4}$$

where N_1, is the size of population one, N_2 is the size of population two, a_{12} and a_{21} are interaction terms representing the competition between species for limited resources, the parameters a_{11} and a_{22} represent the within species competition, and r_1 and r_2 are the net birth/death rates for each species. The model 2.4 is extremely versatile for describing two species interactions, and can easily be extended to three or more species. Because each of the six parameters can be zero, positive, or negative, for no effect, enhancement, or impedance, respectively, there are a total $3^6 = 729$ variations for the two species model alone, and each combination leads to a different configuration with a different interpretation [63]. The focus here is on two species models, both because the numbers of variations are even larger for more species (3^{3n} for n species), and because they are of greatest importance in epidemic modeling. Various configurations for the two species model using the two by four array 2.5

$$\begin{pmatrix} sign(r_1) \mid sign(a_{11}) & sign(a_{12}) \\ sign(r_2) \mid sign(a_{21}) & sign(a_{22}). \end{pmatrix} \tag{2.5}$$

It should be noted, however, that not every mathematical model has a biological interpretation. The variations of 2.4 and 2.5 with the greatest significance are those that describe biological competition (species competing for the same limited resource), scavenging, symbiosis, and most importantly for epidemiology,

predator-prey relationships. A good examination of several competition models is provided by [63] and [79].

The predator-prey relationship is represented by the array 2.6, with N_1 being the number of prey and N_2 being the number of predators.

$$\begin{pmatrix} + \mid 0 & - \\ - \mid + & 0 \end{pmatrix} \tag{2.6}$$

Setting $a_{11} = a_{22} = 0$ removes the within species competition, so that the growth of the predators is limited only by the availability of prey, and the growth of the prey is limited only by the number of predators. The first predator-prey competition model was constructed independently by Alfred Lotka in 1925 [45] and Vito Volterra in 1926 [77]. The Lotka-Volterra model is important to epidemic models and will be developed in more detail. There are, in fact, several kinds of predation scenarios. These include parasitism and cannibalism, but the most common is that in which one species feeds upon another.

$$\begin{aligned} \frac{dN_1}{dt} &= N_1(r_1 + b_1 N_2) \\ \frac{dN_2}{dt} &= N_2(-r_2 + b_2 N_1). \end{aligned} \tag{2.7}$$

Equation 2.7 shows the Lotka-Volterra model, where all coefficients are assumed to be positive. In the absence of predators (N_2), the prey (N_1) will multiply indefinitely at rate r_1, while in the absence of prey, the predators will die off at rate r_2. The predators kill off the prey at rate b_1, while the skill of the predators in catching their prey is represented by b_2.

Qualitative analysis of the Lotka-Volterra model reveals that for any initial position $(N_1(0), N_2(0))$, the model gives a family of closed curves, which implies

cyclic behaviour [63]. This means that as the number of prey increases, the number of predators will increase, until the prey are killed off faster than they reproduce. Then the predators, lacking food, will decrease also, and the cycle will then repeat itself.

As mentioned before, most variations on the two species competition model have no biological interpretation. To construct a workable model for epidemics, however, the Lotka-Volterra model only needs to be altered slightly. Let $S = N_1$ be the number of susceptible individuals in the population, and let $I = N_2$ be the number of infected individuals (infectives). For a simple epidemic, the birth rate of susceptibles, r_1, is set to zero. Setting $b_1 = b_2 = \beta > 0$, and $r_2 = \gamma > 0$, the Lotka-Volterra epidemic model is:

$$\begin{aligned} \frac{dS}{dt} &= -\beta SI \\ \frac{dI}{dt} &= \beta SI - \gamma I. \end{aligned} \tag{2.8}$$

Equation 2.8 is a special case of the predator prey model 2.7. The fundamental difference lies in the interpretation of the parameters. Since the "death" of a susceptible results in the "birth" of an infective, for this model, β can be interpreted as the infection rate. The parameter γ is interpreted as the death rate or removal rate for the disease. The Lotka-Volterra epidemic model assumes that the population experiences *homogeneous mixing*, that is, every contact pair individuals in the population is equally likely to meet. This model does not allow for any new susceptibles to be introduced to the system, so the time frame is necessarily short. Thus the disease modeled by equation 2.8 will eventually "burn out" or run its course, since there are no new susceptibles being added to

the population. When only the general dynamics of the disease are of interest, it is not uncommon to scale both parts of the equation by the total population $N(t)$, so that both S and I represent the proportion of the total that is susceptible and infective, respectively.

Model 2.8 can also be represented concisely using a flow chart or compartmental diagram, which shows each class and indicates the rates at which individuals flow into and out of a particular class. Thus equation 2.8 can be represented as:

$$S^{-\beta SI\rightarrow}I^{\gamma I\rightarrow}.$$

The arrow between S and I indicates that individuals are flowing out of the susceptible class and into the infective class at a rate of $-\beta SI$, and out of the infective class at a rate of γI.

Equation 2.8 is a simple model of an epidemic that occurs over a relatively short period of time, such as influenza. The period over which the disease evolves is short enough that the birth rate of the susceptible population is negligible. If the disease has a longer incubation period, a net birth term α can be added for susceptibles. This yields the equation:

$$\begin{aligned} \frac{dS}{dt} &= \alpha S - \beta SI \\ \frac{dI}{dt} &= \beta SI - \gamma I. \end{aligned} \tag{2.9}$$

Using the compartmental description, we have:

$$^{\alpha S\rightarrow}S^{-\beta SI\rightarrow}I^{\gamma I\rightarrow}.$$

Equation 2.9 allows the disease to evolve over a longer period of time. This enables exploration of deeper questions about a disease, such as whether or not

the disease has long term sustainability. If the growth of the disease depends on both susceptibles and infectives, and if the infection rate is too high compared to the birth rate α, then susceptibles will be removed faster than they are replaced, causing S to approach zero, thus the number of infectives will also reach zero as there are no susceptibles left left to infect. Since the infectives die off faster than they can be replaced, the disease will burn out. Similarly, if the death rate of the disease, γ, is too high compared to the infection rate, then infectives will die off faster than they can be replaced by newly infected individuals, again causing the disease to burn out.

There are many examples of diseases that are unable to sustain themselves over long periods of time. Measles, for example, will typically run its course too quickly to sustain itself long term, although other factors such as human response to control the disease, affect this as well. More deadly diseases also fall into this category. One example is Ebola, which causes hemorrhagic fever, leading to massive internal and external bleeding in its victims. The course of infection is 2 to 21 days, ending in death for up to ninety percent of victims. So far, the disease has progressed so quickly that it has been unable to sustain itself in human populations for long periods of time.

Some infectious diseases can sustain themselves, however. One example is Acquired Immunodeficiency Syndrome (AIDS) caused by the human immunodeficiency virus (HIV). HIV can have a lengthy incubation period, and is spread only through fluid contact with mucous membranes or broken skin. The disease can take years to run its course, during which time the carrier can infect others. The relationship between the infection rate and the death rate are such

that the disease is easily able to sustain itself. Other examples of sustainable diseases include non-lethal infectious diseases such as influenza or the common cold. These diseases run their course over a period of weeks, and the removal rate is low. However, the fact that these diseases do not typically kill requires an alteration to equation 2.8.

Equation 2.8 is the simplest of the SEIR class of epidemic models. These initials stand for susceptible, exposed, infective, and removed (or quarantined). The addition of the other classes gives these models a wide range of descriptive capability, allowing the description of non-deadly diseases as well. For example, the Kermack-McKendrick model adds a quarantined class to describe diseases that are not deadly or for which victims build immunity [33]. As before, this model imposes the restriction that exposed susceptibles immediately become infected. Thus, the system of differential equations is:

$$\begin{aligned} \frac{dS}{dt} &= -\beta SI \\ \frac{dI}{dt} &= \beta SI - \gamma I \\ \frac{dQ}{dt} &= \gamma I. \end{aligned} \tag{2.10}$$

Infectives no longer necessarily die, but once they recover from the disease they are no longer susceptible and become quarantined. This model can also account for the situation where a disease management strategy is in place, and infected individuals are removed from the population and isolated. Once in the quarantined group, they are no longer considered susceptible, either because they are immune to the disease, or because they are isolated. Model 2.10 can be

represented using the diagram:

$$S^{-\beta SI\rightarrow}I^{\gamma I\rightarrow}Q.$$

Because there are no additions to the susceptible class, the time period for this model is short, and the epidemic eventually passes. The addition of a birth rate is required to describe a disease that would last a longer period of time, possibly reaching a sustainable level. Because quarantine models typically assume that an epidemic is in progress and that a control strategy is being applied, it does not include a birth rate.

A variation on this model includes an exposed class. This model was initially proposed for measles, where exposed susceptibles do not immediately become infectives. This occurs when a disease has an incubation period from the point when an individual is exposed to the disease to the point when that individual develops the disease [33]. The model can also include a birth rate for susceptibles, and allow for long term sustainability. Let τ be the length of the incubation period, and let σ be the duration of the disease. Then the model is:

$$^{A\rightarrow}S^{-\beta SI\rightarrow}E^{\tau\rightarrow}I^{\sigma\rightarrow}R.$$

Even if the incubation period is constant, the resulting differential equations are difficult to solve analytically, and must be approximated numerically. The fixed values of τ and σ lead to time-delay differential equations, which can be difficult to solve analytically. An in depth analysis of this model is given in [33].

Epidemiologists and public health officials are not only concerned with how a disease evolves temporally, but also how it evolves spatially. From the earliest times, people have been aware that there was a spatial component to disease.

During the Black Death epidemic of fourteenth century Europe, residents of urban areas where the plague had struck would flee to the country, hoping that the isolation would save them from the disease. Public health professionals are aware that it is just as important to know how a disease spreads geographically as it is to know the disease's dynamic characteristics– infection rate, etc.

Spatial models of epidemics describe the spread of the disease across a geographic region. There are several ways of modeling the spread within the context of differential equations models. Without discrete spatial units, a typical method of spatial modeling is through a stochastic diffusion process model [51, 52, 63]. Non-discrete spatial systems can also be modeled deterministically, using partial differential equations similar to those used in modeling reactive contaminant transport in porous media [18]. These equations are deterministic analogs to stochastic diffusion models.

When the spatial regions are discretized, a model can be built by first constructing an epidemic model such as equation 2.9 within each spatial unit, and then adding a term in each equation that models the interaction between that unit and every other unit. If there are $i = 1, ..., N$ discrete spatial units, then for each unit i:

$$\begin{aligned} \frac{dS_i}{dt} &= \alpha_i S_i - \beta_i S_i I_i + S_i(u_{i1} + ... + u_{iN}) \\ \frac{dI_i}{dt} &= \beta_i S_i I_i - \gamma_i I_i + I_i(v_{i1} + ... + v_{iN}) \end{aligned} \tag{2.11}$$

where u_{ij} is the net migration of susceptibles from unit j to unit i, and v_{ij} is the net migration of infectives from unit j to unit i [63]. These models are relatively easy to construct, but the addition of the migration rates, u_{ij} and v_{ij} make the

solution to the system of equations analytically intractable.

Even though an analytical solution is not available in most cases, equation 2.11 lends itself to numerical solutions or approximation by difference equations. The first-order Euler approximation is the easiest approximation to use, though higher-order approximations also exist. A discussion of most approximation methods can be found in [83]. Numerical approximations are typically designed with the assumption that the parameters in the differential equation are known, and the goal is to generate values of the process. This is the reverse of the statistical setting, where, in most cases, the data consists of discrete observations of the process, with the parameters unknown. The focus in this thesis is therefore on difference equation approaches, because these more easily fit with data types typical of epidemics.

2.2 Bayesian Hierarchical Models of Epidemics

The Bayesian hierarchical modeling strategy has proven to be extremely effective for modeling complex processes with multiple parameters [82, 78, 81]. Hierarchical models allow complex relationships between multiple parameters to be separated into several levels. The hierarchical framework helps the analyst to understand the underlying process linking to data to the model, and to develop computational strategies to simulate the desired posterior distributions. Except in rare cases, simulation is necessary. For an introduction to hierarchical modeling, see [24].

The overall strategy of the Bayesian hierarchical model has been informally but concisely outlined in [3] and [17]. The strategy provides a link– called the *process*– between the observed data and the unobserved parameters of interest,

denoted by $\boldsymbol{\theta}$.

$$[\text{Data}|\text{Process}, \boldsymbol{\theta}] \tag{2.12}$$
$$[\text{Process}|\boldsymbol{\theta}]$$
$$[\boldsymbol{\theta}]\,.$$

The top layer is the observed data, which is modeled by an appropriate likelihood function. It is assumed that the data is generated by an unknown process, for example, an epidemic process. The process depends on unobserved parameters, as shown in the middle layer. It is in this middle layer that the "art" of statistical modeling takes place. On the bottom layer are the prior distributions that represent a priori beliefs about the parameters.

Typically interest centers on the joint posterior distribution of the parameters, that is, the conditional distribution of the parameters, given the observed data.

$$[\boldsymbol{\theta}|\text{Data,Process}] \propto [\boldsymbol{\theta}]\,[\text{Process}|\boldsymbol{\theta}]\,[\,\text{Data}|\text{Process,}\boldsymbol{\theta}] \tag{2.13}$$

The posterior distribution, shown in equation 2.13, is found using Bayes' Theorem.

Modern disease data typically comes in the form of counts. For human diseases, the data are numbers of occurrences per unit time. If the data has a spatial component, then it is grouped by spatial unit as well. This type of data is often modeled using Poisson random variables. Disease data can also be binomial in nature, that is, number of positive cases out of a total. This type of data can also be grouped by spatial and temporal units. The groupings in each case may not be time and space. For example, the data could be grouped by

time and age [5].

Works such as [17], [54] and [25] implement the Bayesian hierarchical strategy for an epidemic model where the spatial dependency is modeled as a Markov random field. These models deal with *areal data*– that is, data aggregated over discrete spatial regions. Data consists of the number of occurrences of a disease in a spatial unit, aggregated over some unit of time as well. The count data is modeled as a Poisson random variable:

$$[y_{it}|E_i, z_{it}] \sim \text{Poisson}(E_i e^{z_{it}}), \tag{2.14}$$

where E_i is the expected number of occurrences in spatial unit i, and z_{it} is the log-relative risk, which accounts for the deviation from the expected number of cases. The expected number of occurrences per unit area, E_i, can be obtained from overall death rates or prior, expert knowledge of the disease [50].

Because Poisson data cannot easily generalized to a multivariate setting and thus cannot directly incorporate spatial effects, the spatial and temporal process is often modeled in z_{it}. Covariate information, denoted by x_{it}, is added via a linear model. This yields:

$$z_{it} = x_{it}^T \alpha + s_{it} \tag{2.15}$$

where s_{it} follows a multivariate first-order autoregressive Gaussian time series, defined by:

$$\mathbf{s}_t = H\mathbf{s}_{t-1} + \boldsymbol{\epsilon}_t. \tag{2.16}$$

The term $\boldsymbol{\epsilon}_t$ is a multivariate noise term, with $\boldsymbol{\epsilon}_t \sim N(0, \Sigma)$. The model in equations 2.14, 2.15, and 2.16, has been used successfully to model spatially

dependent count data from a variety of epidemics [54, 16, 17, 50, 78] and other sources [34].

The dynamic nature of epidemics, especially the change from growth to decline to stability, has resulted in a variation of the previous model to describe an influenza outbreak by [54, 17]. The term $\boldsymbol{\epsilon}_t$ is altered to become the epidemic forcing term, causing an increase and decrease over the course of the epidemic. This is done by introducing another parameter $\beta_{\rho(t)}$ as the mean in the distribution of $\boldsymbol{\epsilon}_t$.

Suppose that over a time interval $t_0 < t_1 < t_2$ the number of occurrences of a disease grows from a state of stability to a full fledged epidemic, and then declines down to a stable state as the epidemic burns out 2.2. As the epidemic takes off and increases in intensity, the parameter $\beta_{\rho(t)}$ moves from the value β_0 to β_1. After it reaches its peak and begins to subside, $\beta_{\rho(t)}$ changes from β_1 to β_2. Finally, as the epidemic passes, the value of $\beta_{\rho(t)}$ returns to β_0. $\rho(t)$ is an index parameter indicating the stage of the epidemic. $\rho(t) = 0, 1$, or 2 for stability, growth, or decline, respectively. Thus, $\beta_{\rho(t)}$ is:

$$\beta_{\rho(t)} = \begin{cases} \beta_0\,, & t < t_0 \\ \beta_1\,, & t_0 \leq t < t_1 \\ \beta_2\,, & t_1 \leq t < t_2 \\ \beta_0\,, & t \geq t_2 \end{cases}$$

Using $\beta_{\rho(t)}$, the error term is then $\epsilon_t \sim N(\beta_{\rho(t)}\mathbf{1}, \Sigma)$. This has the result of turning the epidemic "on" and "off".

The spatial dependencies in the data are built into the structures of the matrices H and Σ. This covariance matrix has the form $\Sigma = \sigma^2(I - \phi C)^{-1}M$,

where M is a diagonal matrix containing conditional variances, C is a matrix of partial-regression coefficients with zeros on the diagonal, and ϕ is chosen so that Σ is positive definite. The form of the covariance matrix is common to the conditional autoregressive spatial models discussed in [16, 4]. The structure of the covariance matrix is found by making use of the Markovian property of the spatial dependency. For this particular model, E_i^{-1} are the diagonal entries of M, and the entries of the matrix C are given by $c_{ij} = (\frac{E_j}{E_i})^{1/2}$, and zero for $i \neq j$ [17, 54].

Spatial structure is also built into the matrix of autoregression parameters, H. This is done by choosing a neighborhood structure for the model. For a given spatial unit i, it must be determined which of the other units j are neighbors of unit i, and whether they are first order, second, third, etc. neighbors. For those units that are not in the neighborhood of i, h_{ij} is set to zero. For the influenza model in [54, 17], the neighborhood structure is chosen to include up to second order neighbors, and $h_{ii} = \eta_0$, $h_{ij} = \eta_1$ if units i and j are first order neighbors, and $h_{ij} = \eta_2$ if units i and j are second order neighbors. The parameters η_k are then transformed so that a Gaussian prior distribution may be used. To finish the modeling process, [54, 17, 78] choose appropriate prior distributions for the remaining parameters, and simulate the resulting posterior distribution.

In the above model, the spatial dynamics of the disease are modeled through correlations in the transformed parameter space, or by spatial dependencies included in the autoregressive coefficients, again on the transformed space. This approach is convenient from a statistical point of view. The transformed parameters are normally distributed, making computation and analysis of the spatial

and temporal dependencies easier. Standard differential equation models describe the dynamics of an epidemic process in terms of numbers of susceptibles and infectives. Counts are modeled using Poisson or Binomial random variables, and there are no common multivariate generalizations of the binomial or Poisson distribution.

For binomial count data, the general strategy is the same. Suppose that N_{it} is the total number of individuals in spatial unit i at time t, and let p_{it} be the probability that a given individual has the disease. Let y_{it} be the observed number (out of N_{it}) that have the disease. Then we can use the model

$$[y_{it}|N_{it}, p_{it}] \sim Bin(N_{it}, p_{it}). \tag{2.17}$$

This type of data is commonly modeled using autologistic models. An example is a model for prostate cancer given in [5]. Using the transformation $\xi_{it} = \ln(\frac{p_{it}}{1-p_{it}})$ yields:

$$\xi_{it} = \mu + \theta_i + \theta_t + \psi_{it} + \epsilon_{it} \tag{2.18}$$

where θ_i, θ_t, and ψ_{it} are observed covariates that explain temporal, spatial, and interaction effects. The error term ϵ_{it} follows a normal distribution, with mean zero and variance ν.

The autologistic model can be generalized to include other covariate information in the equation 2.18. The error term can also be modeled as a Gaussian autoregressive process, similar to 2.16. Let $\boldsymbol{\epsilon}_t$ be the error term aggregated over spatial regions, so that:

$$\boldsymbol{\epsilon}_t = H\boldsymbol{\epsilon}_{t-1} + \nu I, \tag{2.19}$$

so that $\boldsymbol{\epsilon}_t \sim N(H\boldsymbol{\epsilon}_{t-1}, \nu I)$, and the matrix of autoregression parameters, H, includes spatial structure as in 2.16.

Both of these models share the general strategy of modeling spatial and temporal dynamics indirectly by transforming to a convenient parameter space. While this can be effective, as in [17, 54, 78], it does not directly answer the question of "how fast is the disease moving?" The Poisson conditional autoregressive models of [78], [17], and [54] use the Poisson approximation to the Binomial distribution and therefore rely on large sample sizes and relatively small infection rates. The practicality of these types of models is examined in [28].

Hierarchical Poisson models have been used to model domestic animal diseases as well. An example is a hierarchical model for clinical mastitis in herds of dairy cattle given in [64]. Let X_i be the number of cases of mastitis in herd i. The hierarchical specification is:

$$\begin{aligned} X_i &\sim Poisson(\lambda_i) \\ \lambda_i &\sim Gamma(\alpha, \beta_i) \\ \beta_i &\sim Gamma(a, b) \end{aligned} \tag{2.20}$$

where λ_i is the underlying infection rate in herd i, and β_i is the spatial (herd) explanatory variable, and a, b, and α are hyperparameters. As in the influenza and prostate cancer models described above, a conditional autoregressive model could be employed for the infection rate. A Poisson model is appropriate in this situation, because the number of cattle that are infected can be accurately determined. For epidemic models of wild animals, this is not usually the case.

Modeling epidemics in wild animal populations presents difficulties that human epidemic models do not ordinarily have. The largest difficulty is the fact that cases of a disease are not reported. Accurate data on the number of infected individuals is rarely available for wild animal populations, so researchers instead use prevalence data. Prevalence data consists of the number of animals that test positive for a disease, M, out of the total number tested, N. Thus a binomial distribution is natural.

For the Chronic Wasting Disease prevalence data, sample sizes are small, and so a Poisson approximation with its convenient transformation to normality cannot be employed. Additionally, there is not a large amount of specific data on transmission rates. What is available, in addition to the prevalence data, is a qualitative understanding of the behaviour of CWD. Using this general understanding of the disease, a differential equations model that has the same qualitative behaviour can be derived. This ODE model will be the basis for a statistical model that directly uses the available prevalence data via a Bayesian Hierarchy.

3. A Spatial Model for Chronic Wasting Disease

Chronic Wasting Disease (CWD) is a transmissible spongiform encephalopathy (TSE) that occurs in wild cervid populations in North America. It is found in both mule deer (*Cervus odocoileus spp.*) and the North American elk, also known as the wapiti (*Cervus elaphus nelsoni*) [49, 27]. CWD is related to other TSE's found in domestic animals, such as scrapie in sheep and bovine spongiform encephalopathy, commonly called mad cow disease [49]. It is also related to TSE's found in humans, such as classic Creutzfelt-Jacob disease and variant Creutzfelt-Jacob disease. The exact agent that causes CWD has not yet been positively identified, and the method of transmission is currently unknown. However, the disease is accompanied by the presence of mutant proteins called prions in the nervous tissue[14, 49, 27, 31, 32].

Chronic Wasting Disease is always fatal, causing damage to portions of the brain in infected animals. Animals with CWD show a progressive loss of body condition, behavioral changes, and eventually death [14, 27]. The course of infection in mule deer appears to include both latent and clinical periods, and spans from 18 to 36 months. Once clinical signs appear, few deer survive more than 12 months [27].

Thus far, chronic wasting disease has been far more common in deer than in elk. It has been estimated that in infected regions, between 5% to 6% of mule deer have CWD, compared to only about 1% in elk [27, 14], although recent data suggests that the infection rate in elk may be increasing [14]. While both

mule deer and white-tail deer can contract the disease, in the Rocky Mountain region the epidemic seems to be focused in those areas where mule deer (*Odocoileus hemionus*) are common and few, if any, white-tailed deer are found. Consequently, research and data collection on the dynamics of the disease in the Rocky Mountain area have focused on mule deer [27, 31]. The research that has been done on white-tailed deer suggests that the species responds similarly to CWD, but a clustering effect is more likely, since white-tailed deer are more social than mule deer.

Chronic Wasting Disease was first found near Fort Collins, Colorado, in 1967, and has been spreading over the past two decades. The disease has been making its way across Colorado and has recently crossed the continental divide [14]. The incidents of the disease in Central and Western Colorado have been isolated thus far, although they are occurring with greater frequency. Currently, the disease appears most concentrated in the region encompassed by Northeastern Colorado and Southeastern Wyoming, although there have been recent hot spots in Wisconsin, South Dakota, Nebraska, and a few reported cases in Utah and New Mexico.

Since hunting is such big business in many states, the potential consequences of the CWD epidemic are great. A recent article in the October 24, 2003, edition of the Denver Post reported that hunting added $599 million to the economy of the state of Colorado in 2002. The same article estimated the potential economic impact of CWD nationwide to be $100 billion. Thus, Chronic Wasting Disease is recognized nationally as a potentially disastrous problem, and every state except Hawaii has implemented a CWD surveillance program.

Current efforts for modeling the spread of chronic wasting disease have focused on the simulation of individual interactions between deer using standard epidemic models such as a birth-death process [27] and modeling the spread as a diffusion process [31], and these models have provided insight into the disease dynamics. The disease has been spreading throughout the Rocky Mountain Region for more than 30 years, but a statistical model with spatial components has yet to be presented.

There are several standard differential equations that are commonly used as epidemic models. These are described previously in this thesis in section 2.1. One such is the adaptation of the Lotka-Volterra predator-prey model (equation 2.9) to diseases given in [63] and [33]. Some of these variants have been applied to chronic wasting disease because they exhibit the qualitative behavior observed in many epidemics [27, 32]. The differential equations used by [27], [31], and [32], implicitly model disease dynamics based on individual interactions between deer and very small time steps. However, most available data for these models are aggregated over relatively large time steps and large areas. These authors use stochastic forward integration as the primary tool for understanding and exploring the spatio-temporal dynamics. The link between the mathematical model and the data is indirect.

The correlated Poisson model of [78], [17], and [54], discussed in section 2.2, is effective in modeling the spatial spread of human diseases such as influenza, but application to CWD is hindered by fundamental differences in data. The models discussed in section 2.2 rely on the Poisson approximation to the Binomial distribution and therefore implicitly rely on large sample sizes and relatively

small infection rates [78, 17, 54]. It is extremely difficult and expensive to obtain prevalence data on a wildlife population, so models for Chronic Wasting Disease need be to appropriate for small samples.

A Bayesian hierarchical model is constructed for chronic wasting disease that directly integrates mathematical models related to those of [27], [31] with a model for observations taken on CWD prevalence. Critical to the model is a difference equation approximation to traditional differential equation models of prevalence to model dynamics in time. A mechanism is included that allows for spatial mixing between regions of interest. Available data is incorporated through a Bayesian hierarchical framework, thus linking the data and mathematical model. This hierarchical approach integrates data and epidemic theory into a single model, is capable under the constraint of small data sets, and provides a mechanism for prediction and the study of data collection designs.

3.1 Data

Data for CWD is economically and ecologically expensive to obtain since the testing procedure, until recently, required sacrificing the animal [14, 32]. Prevalence data consists of the numbers of deer that are tested for CWD in each DAU, and the number of those tested that are found to have the disease [49] and [27]. The testing procedure used to determine whether or not an animal is infected is considered to be extremely accurate, and is performed in highly controlled conditions [14]. Consequently, the false positive and false negative rates for the test were assumed to be small enough to be neglected. Currently, the data is predominantly obtained from hunters, who submit harvested animals for testing. These submissions are required by the Colorado Division of Wildlife

in the endemic areas. For animals taken outside the endemic area, the animals may be submitted voluntarily but are subject to a $15 testing fee. Because the type of deer a hunter may take is restricted by age, and in most areas gender, harvest data may provide a biased view of the disease [15]. Determining these biases requires more detailed data than is available here. Nevertheless, the model presented here is valid for the population of huntable deer, although the definition of huntable can change from region to region.

Wildlife biologists divide the state of Colorado into geographic regions called data analysis units (DAUs). Different species of animals have different DAU definitions, so that the spatial units are specific for different species of animals. For example, deer and elk have different DAUs. Since the focus of the model presented here is mule deer, DAU will refer only to the data analysis units for deer. These DAUs can be seen in Figure 3.1. The data shows that there are 13 DAUs where chronic wasting disease is known to exist as of the 2002 hunting season. These are DAUs 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 17, 27, and 44 in Figure 3.1. Because the disease has not yet spread to much of the state, the majority of the data is from an endemic region located in the northeastern region of the state of Colorado, consisting of DAUs 4, 5, 10, 27, and 44 [14]. While there have been a number of cases from units surrounding the endemic region, the Division of Wildlife believes that the disease is still controllable in the non-endemic areas. It is thought that the disease will be a permanent presence in the endemic region [14, 32].

The prevalence data from Colorado was collected over 27 years, from 1976 to 2002. The data were obtained from the Colorado Division of Wildlife, and

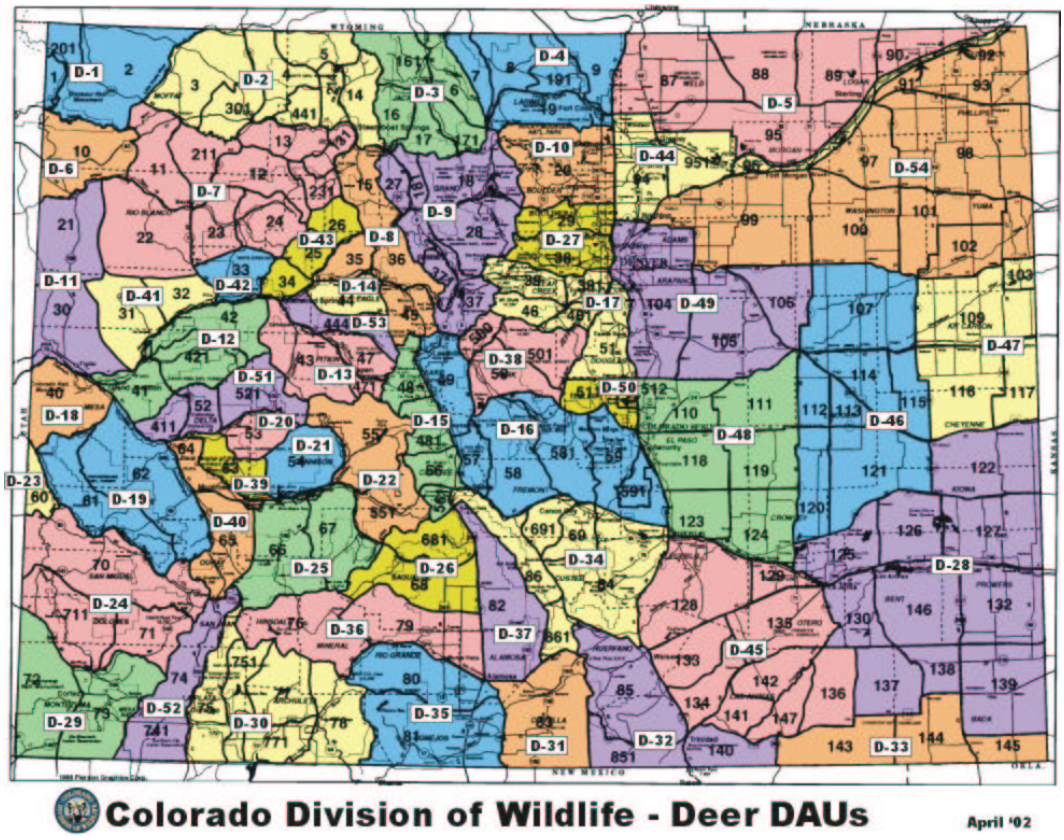

Figure 3.1: Colorado Deer Data Analysis Units across the state. The larger spatial units are the DAUs, the smaller units are game management units.

each DAU-year value consists of two numbers M and N, where N is the number of deer that were tested for CWD and M is the number that were found to be infected. Because the disease was only recently recognized as an epidemic, much of the data is from the endemic region and is most complete from the years 1996-2002. Prior to 1996 the data was collected sporadically. The Colorado Division of Wildlife has recently made the testing results for the 2002 and 2003 hunting seasons available publicly [14].

3.2 Prevalence Model for Chronic Wasting Disease

The approach presented in this thesis utilizes a system of differential equations which exhibit the appropriate qualitative temporal and spatial behaviour. Solutions to this system are approximated using a set of difference equations on a scale appropriate for the data. Through a hierarchical Bayesian construction, the data are used to estimate the unknown parameters that govern these difference equations. One fundamentally different aspect of this approach is the explicit modeling of spatial dynamics in the differential equations, rather than indirectly attempting to capture the effect using spatial correlations typical in statistical models.

3.2.1 Model Motivation

The logistic model 2.2 discussed in section 2.1 is commonly used to model biological populations [63]. The model is simple, but applicable to a surprising number of biological applications. The logistic model forms the basis for a Chronic Wasting Disease model.

For a single population, the logistic model 2.2 exhibits the desirable characteristic of having two equilibrium points, at $N = 0$ and $N = K$. A *fixed point* or *equilibrium point* for the differential equation $\frac{d\mathbf{p}_t}{dt} = f(\mathbf{p}_t)$ is a point p^* such that $\frac{dp^*}{dt} = f(p^*) = 0$. These points represent a steady state, since if the system begins at p^* it remains there for all time. Fixed points can be either stable, so that if the system is perturbed off of the fixed point it will eventually return to the steady state, or unstable, so that if the system is perturbed off of the fixed point it will not return. The carrying capacity, K, represents a maximum sustainable population. As the population size approaches this value, the growth will slow. Analysis shows that the point $N = 0$ is an unstable equilibrium point,

and $N = K$ is a stable equilibrium point. Thus, if the initial value of the population is greater than the carrying capacity, $N > K$, then the population will decrease back towards this stable point. Likewise, if $0 < N < K$, then N is increasing. This model is easily generalized to a spatial model by aggregating over discrete spatial units and including migration terms [63].

A differential equation for the prevalence of a disease based on the logistic model can be formulated by rescaling by K [33, 63]. Rescaling by the carrying capacity results in a differential equation with the same basic properties as equation 2.2, but which lives on the interval $[0, 1]$. Beginning with equation 2.2, divide by the maximum population K. This gives a logistic equation modified for prevalence, restricted to the unit interval. Let p denote the proportion of deer in a given DAU that are infected. Then,

$$\frac{dp}{dt} = \alpha p(1 - p) \tag{3.1}$$

is a model for the change in prevalence over time.

Let p_{kt} denote the proportion of infected deer in DAU_k for time period t and $\mathbf{p}_t$ be the vector of prevalences aggregated over all DAU's for a given year t. For now, the focus is on the observed qualitative behaviour within a single DAU, and the spatial dynamics will be added later. From historical observations and recent studies of CWD [49, 27], it is known that CWD takes approximately 3 years from infection to the death of the animal. For a disease with an incubation period this long, it is a reasonable supposition that a sustainable level of the disease is possible. Additionally, if the disease is in fact caused by environmental sources or carried by another animal, this would indicate that a sustainable level is likely, since an external carrier could potentially be unaffected by the

disease. A classic example of an external host is the flea that carries the plague, aiding in the spread of the disease and enabling the disease to sustain itself. Furthermore, CWD has yet to show signs of elimination even in those areas where the disease has been present the longest, giving further support to the sustainability assumption. Another reasonable assumption is that if the disease is not present in a given area, i.e. if $p_{kt} = 0$, then the proportion of deer that are infected will stay at zero, unless infected deer from another area migrate in (although this implies that deer are the vector for the infection, and not an environmental source, and it is not known for certain that this is the case). These two assumptions seem, on the surface, to be supported by the current understanding of the disease. They provide support for the choice of the logistic model 3.1, which has these qualitative characteristics.

The parameter α in equation 3.1 represents a kind of acceleration for the disease, and can be interpreted as an infection rate. Fixing α as a constant restricts the dynamics of the model by fixing the infection rate. Instead, by choosing $\alpha = \alpha(p,t)$ and following the basic tenet of the correlated Poisson model discussed in section 2.2, more freedom is obtained for the model. This type of freedom can be extremely useful in describing different types of disease dynamics. One such dynamic is the observed behaviour in many epidemics of fast growth in the initial stages of the epidemic, followed by a decline in the latter stages of the epidemic. The parameter $\alpha(p,t)$ plays a role similar to that of the control parameter $\beta_{\rho(t)}$ in the influenza model of [17, 54] discussed in section 2.2.

The form of $\alpha = \alpha(p,t)$ must be chosen to provide the appropriate behaviour. It can be designed to account for a variety of effects, such as spatial and seasonal effects. With appropriate data, explanatory covariates can also be included in the expression for $\alpha(p,t)$. In this case, due to the small amount of data, the desire for appropriate qualitative behaviour guides the form of $\alpha(p,t)$. Making the assumption that there is a non-zero, long-term sustainable level for the disease, set $\alpha^* = \alpha(\delta - p_t)$. This acts as a varying acceleration. The acceleration slows down the closer the proportion gets to δ. When $p_t = \delta$, the acceleration term is zero, and the internal rate of change of the DAU is also zero. The parameter δ is the long term proportion of infected deer that the DAU is able to sustain, and plays a role similar to that of the carrying capacity in the logistic equation 2.2.

For a single DAU, the differential equation becomes:

$$\frac{dp}{dt} = \alpha p(\delta - p)(1 - p). \tag{3.2}$$

With this modification, prevalence is increasing for levels below δ, and decreasing for levels above δ. If $\delta = 0$, then the disease is killing off the existing infectives faster than it is creating new infectives and there will be an eventual burn out of the disease. For $\delta = 1$, then the disease will eventually infect all the deer. When $0 < \delta < 1$, the model has a long-run stable proportion of infectives at level δ.

Figure 3.2 shows the graph of the derivative, p', versus p for equation 3.2, with $\delta = 0.25$. The equilibrium points, p^* are on the p axis, where $p' = 0$. To graphically determine the nature of the equilibrium point, arrows are drawn to the right for values $p' > 0$, and to the left for values $p' < 0$. When the arrows flow

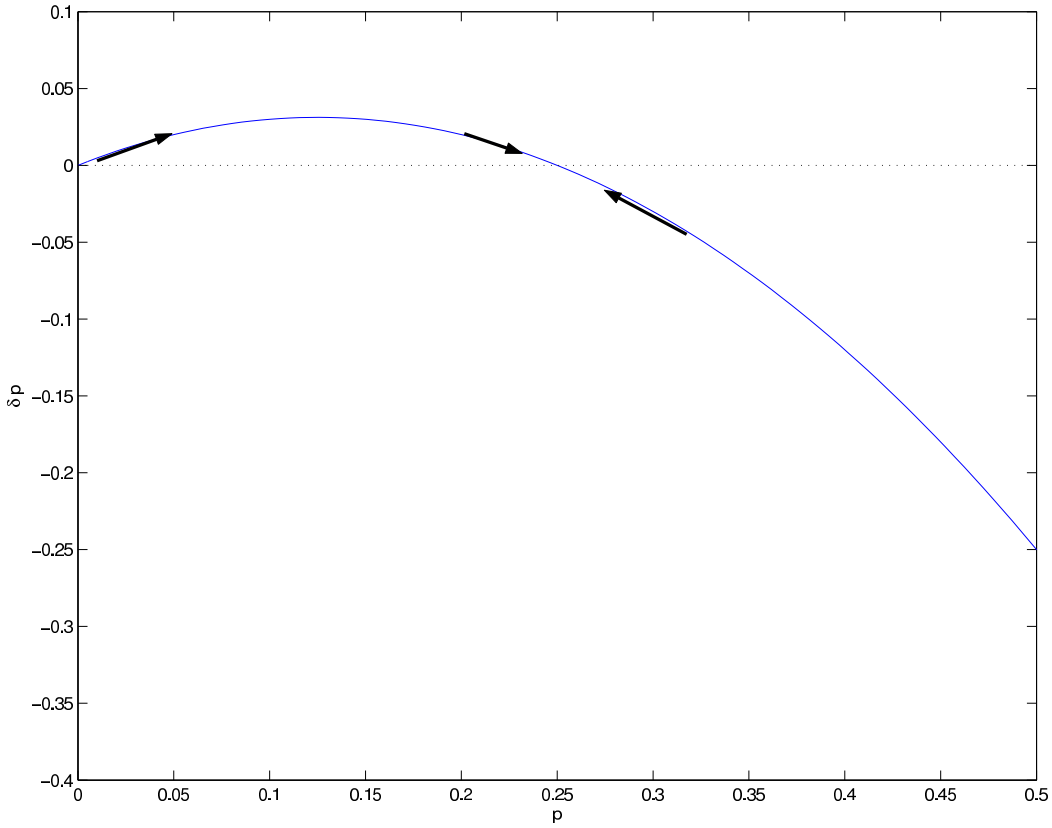

Figure 3.2: Graph of the derivative of p versus p. Stable points (0,1/4) are on the p axis.

away from each other the point, p^*, is unstable, when the arrows flow towards each other p^* is stable. The graph shows that the equilibrium point $p^* = 0$ is unstable, while the other equilibrium point, $p^* = \delta$, is stable, supporting the interpretation of the parameter δ as a long term sustainable level. The idea can be extended to several dimensions as well.

Having developed an appropriate model for the dynamics of prevalence in a single DAU, the model is extended to multiple DAU's. This is done most directly

by replicating over the DAU's to produce the system of differential equations:

$$\frac{d\mathbf{p}_t}{dt} = \alpha(\delta\mathbf{1} - \mathbf{p}_t) \odot \mathbf{p}_t \odot (\mathbf{1} - \mathbf{p}_t) \tag{3.3}$$

where $\mathbf{p}_t$ is the vector of prevalences, so that $\mathbf{p}_t = (p_{1,t}, p_{2,t}, ..., p_{54,t})$. The symbol $\odot$ represents the Hadamard (element by element) matrix product. Importantly, this system of differential equations allows for different prevalence values in each DAU.

3.2.2 Incorporating Spatial Mixing

Although equation 3.3 is a relatively easy extension that has the appropriate general temporal dynamics, it treats each DAU separately and does not take into consideration any spatial dynamics. In equation 3.3, neighboring DAU's do not affect each other, which is an unlikely scenario. It is known that the disease is spreading across the state, so a spatial component for the model is needed.

The spatial dynamics are incorporated into the model by adding a spatial mixing term to equation 3.3, yielding:

$$\frac{d\mathbf{p}_t}{dt} = \alpha(\delta\mathbf{1} - \mathbf{p}_t) \odot \mathbf{p}_t \odot (\mathbf{1} - \mathbf{p}_t) + Q\mathbf{p}_t, \tag{3.4}$$

where Q is a matrix such that the values q_{ij} represent the instantaneous net effect that the prevalence in DAU j has on the prevalence of DAU i. Thus the rate of change in the prevalence over DAUs consists of an linear combination of the internal dynamic processes governed by logistic equations, and the external contributions by the other DAUs.

The multi-DAU spatial model is extremely difficult to solve analytically, so a first-order Euler discretization is used to approximate the differential equation

by a difference equation. This discretization is a simple method of approximating a derivative, and has the form $f'(t) \approx \frac{f(t+\Delta t)-f(t)}{\Delta t}$.

Since the data is aggregated by year, the time step is assumed to be $\Delta t = 1$. The difference equation for (3.4) is therefore:

$$\mathbf{p}_{t+1} = \alpha(\delta\mathbf{1} - \mathbf{p}_t) \odot \mathbf{p}_t \odot (\mathbf{1} - \mathbf{p}_t) + W\mathbf{p}_t. \tag{3.5}$$

where W is a spatial mixing matrix that indirectly represents deer migration, and is a discrete version of the matrix Q in equation 3.4.

Equation 3.5 is the dynamic spatial model for Chronic Wasting Disease, and the prevalence data is used to find the parameters for this equation. The migration of deer changes prevalence in that a single infected deer migrating from one DAU to another changes the makeup of prevalence for both the "from" and "to" DAUs. However, the matrix W does not directly describe this movement of deer, but instead describes the movement of the disease prevalence. The change in prevalence for a given DAU consists of a mixture of the internal dynamic processes modeled by the logistic equations, and the external contributions from migration to and from neighboring DAUs. Including the migration matrix W neither implies nor removes the possibility that deer are the transmission method for the disease in the model.

The fixed points for the difference equation will be the same as those for the corresponding differential equation, provided that $W\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is a unitary vector [75, 83], and therefore the parameters in the difference equation 3.5 have the same interpretation as the parameters in the differential equation 3.4.

Various forms of the spatial mixing matrix W could be designed to incorporate elaborate schemes for disease transport. However, because the mechanism

by which the disease moves is currently unknown, two simple cases were considered. Assuming that the disease can only migrate between adjoining DAUs, so that if DAU_i and DAU_j are neighbors, and DAU_j and DAU_k are neighbors, but DAU_i and DAU_k are not, then in a single time step (one year in this case) migration occurs between i and j, and between j and k, but not between i and k. This is a reasonable assumption supported by biological studies of the behavior of deer in the wild and the spread of CWD in Colorado, and the manner in which the state of Colorado has constructed the DAU's [14, 15, 49, 27, 31, 32].

Let W_{ij} denote an element of the stochastic transition matrix W. For parsimony, fix $W_{ij} = \gamma$ if DAU_i and DAU_j are first-order neighbors with a "significant" proportion of boundary that touch. If DAU_i and DAU_j have less of boundary that touch, or are almost touching, then $W_{ij} = \omega < \gamma$. For DAUs that are separated and hence are not neighbors, $W_{ij} = 0$. The diagonal values of W are fixed so that row sums of W are equal to 1. Thus, the W_{ij} can be loosely interpreted as the average proportion of deer that migrate from DAU_i to DAU_j each year. The classification of first-order neighbors and second-order neighbors was admittedly subjective. Tables 3.1 and 3.2 show the relationships between the DAUs that were used.

In order to ensure that equation 3.5 produces values for $\mathbf{p}_t$ that are interpretable, certain restrictions are made on the parameters α, δ, γ, and ω. For simplicity, let $F(\mathbf{p}, \boldsymbol{\eta}) = \alpha(\delta\mathbf{1}-\mathbf{p})\odot\mathbf{p}\odot(\mathbf{1}-\mathbf{p})+W\mathbf{p}$, where $\boldsymbol{\eta} = (\alpha, \delta, \gamma, \omega)$. The parameters γ and ω are the nonzero elements of the matrix W. The relationship in equation 3.5 is therefore written as $\mathbf{p}_{t+1} = F(\mathbf{p}_t, \boldsymbol{\eta})$.

DAU	Near	Secondary	DAU	Near	Secondary
1	2,6	7	28	33,45,46,47	48
2	1,3,4	8,9	29	24,52	-
3	2,4,9	8,10	30	35,36,52	25
4	2,3,5,10	9,44	31	32,35,36	37
5	4,44,54	-	32	31,34,45	33
6	1,7,11	-	33	28,45	32
7	6,8,11,41,42,43	1	34	16,32,37,45	15,26,48
8	7,9,14,43	2,3,15,16,53	35	30,31,36	37
9	3,8,10,17,27	2,4,16,38	36	25,26,30,31,35,37	-
10	4,9,27,44	3	37	26,34,36	15,31,35
11	6,7,18,41	12	38	16,17	9,50
12	13,41,51	11,18,19,42,43	39	20,21,40	25,51
13	12,15,22,51	42,43	40	19,25,39	24,51

Table 3.1: Neighborhood relationships between DAUs. Digits refer to DAU number in Figure 3.1.

DAU	Near	Secondary	DAU	Near	Secondary
14	8,43,53	-	41	7,11,12	42
15	13,16,22	8,26,34,37,53	42	7,43	12,13,41
16	15,34,38,48,50	8,9,45	43	7,8,14,42	12,13,53
17	9,38,44,49,50	48	44	5,10,17,54	4,27
18	11,19	12,23	45	28,32,33,34,48	16,46
19	18,23,24,40	12,51	46	28,47,48,49,54	45
20	21,39,51	22	47	28,46,54	-
21	20,22,25,39	-	48	16,45,46,49,50	17,28,34
22	13,15,21,25,26	20	49	17,27,46,48,54	50
23	19	18,24	50	16,17,48	38,49
24	19,29,52	23,40	51	12,13,20	19,39,40
25	21,22,26,36,40	30,39,52	52	24,29,30	25
26	22,25,36,37	15,34	53	14	8,15,43
27	9,10,49	44	54	5,44,46,47,49	-

Table 3.2: Neighborhood relationships between DAUs. Digits refer to DAU number in Figure 3.1.

The parameters α, δ, γ, and ω are restricted so that $0 \leq \alpha \leq \min\{4/(1-\delta)^2, 4/\delta^2\}$, $0 \leq \delta \leq 1$ and $0 \leq \omega \leq \gamma \leq 1/\max(\sum_{j \neq i} W_{ij})$. With these restrictions and W defined as above, there is the following theorem.

Theorem 3.2.1 *Let α, δ, γ, and ω be restricted as above. Then the update function F has the following properties:*

$$F(\mathbf{p}, \boldsymbol{\eta}) \in [0,1]^K \quad \forall\, \mathbf{p} \in [0,1]^K \tag{3.6}$$

$$F(\mathbf{0}, \boldsymbol{\eta}) = \mathbf{0} \tag{3.7}$$

$$F(\delta\mathbf{1}, \boldsymbol{\eta}) = \delta\mathbf{1} \tag{3.8}$$

The first property (3.6) guarantees that the dynamic system will always return prevalences between zero and one. The second property (3.7) implies that if there is no disease in the system, the disease cannot propagate. The third (3.8) implies that the system has a steady state where the disease maintains a stable relationship with the deer.

PROOF: To show that these properties hold, first note that the form of W is such that $W\mathbf{1} = \mathbf{1}$, $W^T\mathbf{1} = \mathbf{1}$ and $W_{ij} \geq 0$. Thus, $W\delta\mathbf{1} = \delta\mathbf{1}$, and $W\mathbf{p} \in [0,1]^K$ whenever $\mathbf{p} \in [0,1]^K$. It remains to be shown that $g(x) = x + \alpha x(1-x)(\delta - x) \in [0,1]$ whenever $x \in [0,1]$. Note that $g(x) = x(1 + \alpha\delta - \alpha(1+\delta)x + \alpha x^2) = xq(x)$ where q denotes the quadratic form. Now g has a root at $x = 0$ and may have up to two more roots at the roots of q. If the roots of q are complex, then $g(x) = 0$ if and only if $x = 0$. It is easy to show that the roots of q are complex for $0 < \alpha < 4/(1-\delta)^2$. Thus $g(x) \geq 0$ on $[0,1]$ when $0 < \alpha < 4/(1-\delta)^2$. Let $h(x) = 1 - g(x)$ and note that $g(x) \leq 1$ whenever $h(x) \geq 0$. Factoring out $(1-x)$ in h, argue as before to show that $g(x) \leq 1$ for $x \in [0,1]$ whenever $\alpha \leq 4/\delta^2$. $\square$

In what follows, let $\alpha \in (0,1)$ be the proportional acceleration and the acceleration to be $4\alpha \min(1/\delta^2, 1/(1-\delta)^2)$. Likewise, re-scale ω and γ to represent the proportion of complete interchange, so that ω and γ are the probabilities used in the spatial transition matrix W.

3.2.3 The Hierarchical Statistical Model

Because there are different sampling schemes for different stages of the knowledge of disease prevalence, a simple modification is required. The change amounts to modeling (location,time) observations from two possible distributions. One distribution corresponds to time-periods where a DAU has a large number of deer tested for the disease. In this case, the effort behind the testing is primarily geared towards determining and understanding prevalence. The Division of Wildlife knows that the disease is present, wishes to collect as much data as possible, and has the additional goal of protecting hunters from eating infected animals. On the other hand, when the number of tested carcasses is low, that corresponds with a time when the CDOW is opportunistically searching for the disease in an area where it has not yet been found. This data comes in the form of animals found already dead, and hunters in non-infected areas that wish to pay for testing. The differences in searching for the disease add biases which are explicitly included in the model to facilitate estimation. Furthermore, model selection techniques are used to find a cutoff value which separates the two processes.

The likelihood for the observation at DAU_i at year t, denoted by Y_{it}, is modeled as one of two binomial distributions

$$\begin{aligned} Y_{it} &\sim B(N_{it}, p_{it}) \qquad \text{for } N_{it} > \mathcal{C} \\ &\sim B(N_{it}, \pi) \qquad \text{for } N_{it} \leq \mathcal{C}. \end{aligned} \tag{3.9}$$

depending on some cutoff value $\mathcal{C}$ and where N_{it} is the number of tests completed for DAU_i at year t and p_{it} is the i^{th} component of $\mathbf{p}_t$. This duality is required because of the different sampling schemes for different stages of the knowledge of disease prevalence. The first distribution corresponds to time-periods where a DAU has a large number of deer tested for the disease. In these cases, the Division of Wildlife knows that the disease is present and wishes to protect hunters from eating infected animals. On the other hand, DAU-years when the number of tested carcasses is low correspond with the CDOW searching for the disease in an area where it has not yet been found or confirmed problematic. This approach is somewhat *ad hoc*, and only moderately reflect the true nature of the sampling mechanism. Nonetheless, the modification provides sufficient flexibility to allow the model to accurately describe the changing prevalences. This modification essentially eliminates the data values below the cutoff from the spatio-temporal model, as the posterior of π given all the data is independent of the other parameters. Because of this independence, data that are used to improve knowledge regarding π cannot be used to increase information about the other parameters.

The binomial sampling scheme used here is a simplification that considers the values $N_{i,t}$ as fixed. Common practice is to consider these values as random,

perhaps modeling them as Poisson or binomial draws from the total population. There are several reasons why a second layer of hierarchy was not added. First, although populations estimates were available for most DAUs over a ten year period, these estimates were considered unreliable. Population estimates are prepared by game region, and the manner in which they are calculated varies by region. Game regions can consist of several DAUs, and information about how the game regions are structured was unavailable. Moreover, the method in which the populations are estimated has changed over time, as new estimation techniques are adopted, both because of scientific advancement and changes in game region management. Information on what population estimation methods were used was unavailable. Finally, the testing data was obtained predominantly from harvested animals. These animals form a subset of the population, because of strict regulations that govern which animals can be harvested and which cannot. In most areas of Colorado, huntable deer consists only of male deer two years or older [14]. There was no information available on what percentage or number of deer are harvested each year, although the Colorado Division of Wildlife does collect estimates on this information, grouped by game management units [14]. The simple sampling scheme that was used means that even though all of the results presented in this thesis may not be applicable to the total population of deer, they can still be reliably applied to the population of harvested deer.

Given an initial prevalence vector $\mathbf{p}_0$ and the parameters in $\boldsymbol{\eta}$, the values of $\mathbf{p}_t$ are fixed. It is important to note the difference between this approach and a standard state space model [23]. In a state space model, the value X_t at time t is equal to a function $F(X_{t-1}, \boldsymbol{\eta})$ plus error. That is, given the parameters $\boldsymbol{\eta}$

and the previous value X_{t-1}, the new value is random. In the approach followed in this thesis, the value of X_t given the parameter values $\boldsymbol{\eta}$ and the initial value X_0 are deterministic. This approach also differs from the traditional state space model by using binomial distributions for the observational equation, rather than normal distributions. By using deterministic rather than stochastic updates, the size of the parameter space is effectively reduced from $54 \times 27 + 4$ to $54 + 4$. In fact, the parameter space is smaller because many of the initial conditions (the 54 elements of the vector $\mathbf{p}_0$) are forced to zero.

The likelihood for the prevalence data is given by,

$$\begin{aligned}\mathcal{L}(\ \mathbf{M} \mid \mathbf{p}_0,\ \boldsymbol{\eta},\ \mathcal{C}\ ) = \prod_{i=1}^{54}\prod_{t=1}^{27} &\left[p_{it}^{M_{it}}(1-p_{it})^{(N_{it}-M_{it})}\right]^{1_{\{N_{it}>\mathcal{C}\}}} \\ &\times \left[\pi^{M_{it}}(1-\pi)^{(N_{it}-M_{it})}\right]^{1_{\{0<N_{it}\leq\mathcal{C}\}}} \end{aligned} \tag{3.10}$$

where $1_{\mathcal{A}}$ is 1 if condition $\mathcal{A}$ is true and zero otherwise. p_{it} is the i^{th} component of $\mathbf{p}_t$, defined by recursion in equation 3.5, N_{it} is the number of deer from DAU$_i$ that are tested in year t, M_{it} is the number of those that are infected, π is the probability that a tested deer has CWD given that a small number of deer are tested in that DAU-year, $\mathcal{C}$ is a cutoff which models the different sampling schemes relative to π and $\boldsymbol{\eta}$. The value $\mathcal{C} = 3$ provided a reasonable fit to the data, and will be discussed later. The posterior distribution of parameters is proportional to

$$\mathcal{P}(\ \mathbf{p}_0,\ \boldsymbol{\eta},\ \pi\ ) \propto \mathcal{L}(\mathbf{Y} \mid \mathbf{p}_0,\ \boldsymbol{\eta},\ \pi,\ \mathcal{C}\ )\ q(\mathbf{p}_0)\ q(\boldsymbol{\eta})\ q(\pi) \tag{3.11}$$

where $q(\cdot)$ denotes a prior distribution. For simplicity below, let $\boldsymbol{\theta} = (\boldsymbol{\eta}, \mathbf{p}_0, \pi)$ denote the complete vector of unknown parameters.

This model has qualitative characteristics, through the deterministic differential equations, that are consistent with current knowledge and expectations of CWD. It is also flexible enough to accommodate spatial mixing of prevalences due to deer migration and possibly other factors. The model is formalized with an Bayesian hierarchy, giving the added benefit of allowing for comparison of different models, testing whether or not the disease is spreading spatially (if $\gamma > \omega > 0$) and whether or not there disease can coexist with the mule deer population (if $0 < \delta < 1$). The Bayesian hierarchy also provides a method of checking sampling schemes that will maximize information from data collected in years to come [37].

Prior distributions for α, δ and π were all independent Beta distributions, since each parameter in the model is bounded on $[0, 1]$. The prior distributions for the different parameters were chosen to reflect the current understanding of CWD researchers and the statewide, large scale observations of the disease [14]. Information for the prior distributions comes from current research and expert knowledge that is indirectly based on this and other data. Consequently, the modeling strategy is not a formal Bayesian model, but instead has an empirical Bayesian flavor.

The overall level of the disease in the endemic area is currently believed to be between 8-13% [14]. While the disease still appears to be spreading spatially, in those areas that have been infected the longest the level of the disease seems to have stabilized. Consequently, the prior for δ was chosen to be a $Beta(8, 82)$, which has a mean of 0.09, or 9%., and a standard deviation of 0.0298.

Currently, there is little knowledge or expectations for the acceleration parameter α. Detailed study of the dynamics of Chronic Wasting Disease is relatively recent but is ongoing [31, 32, 27]. Consequently, a uniform density on $[0, 1]$ was chosen as the prior distribution to reflect this lack of information.

Because the DAUs are defined so that deer will tend to remain within their DAU, the migration parameters γ and ω are expected, a priori, to be small. However, there is a lack of information available on the migration rate of the disease [14]. Consequently, the prior distribution for γ and ω are chosen to be $Beta(1, 50)$ and $Beta(1, 100)$ respectively, subject to the constraints that $\gamma > \omega$ and that $\max_i\{\sum_{j\neq i} W_{ij}\} < 1$.

According to the Colorado Division of Wildlife, the disease was rare before 1990 [14]. CWD was unknown to non-experts in 1975 ($t = 0$), and no cases were observed in wild populations prior to 1980. Consequently, the initial prevalence values were expected to be extremely small. Additionally, because the disease prevalence has only been observed increasing, those DAUs that are not currently infected were not allowed to have the disease present at year zero. The prior distributions for $\mathbf{p}_0$ were therefore either $Beta(1, 200)$ or a point mass at zero.

3.3 Finding a Good Model

The first task in choosing a particular model is to determine a reasonable value for the cutoff $\mathcal{C}$ in the likelihood (3.10). The joint posterior distribution can be factored so that the posterior distribution of π is independent of the posterior for $\boldsymbol{\eta}$ and $\mathbf{p}_0$.

For a given N, the binomial distribution which accounts for more data variability than any other binomial model is the distribution with $p = 0.5$.

Figure 3.3 shows the proportion of deer that test positive by year and the total number tested for the three DAUs which have been infected the longest. There is a large amount of variability in the early data, where the number of deer tested is small. As the number of deer tested increases, the variability in the data appears to decrease. Thus, a search is conducted for the value of $\mathcal{C}$ that yields a cutoff level that has, as close as possible, half of the observations diseased and half not. By inspecting the data it is found that this corresponds to a cutoff value of $\mathcal{C} = 3$. Thus in DAU-years where 4 or more deer are tested for Chronic Wasting Disease, the modeled prevalence will be from the updates given by equation 3.5. Otherwise, the prevalence is modeled as $Bin(N_{it}, \pi)$.

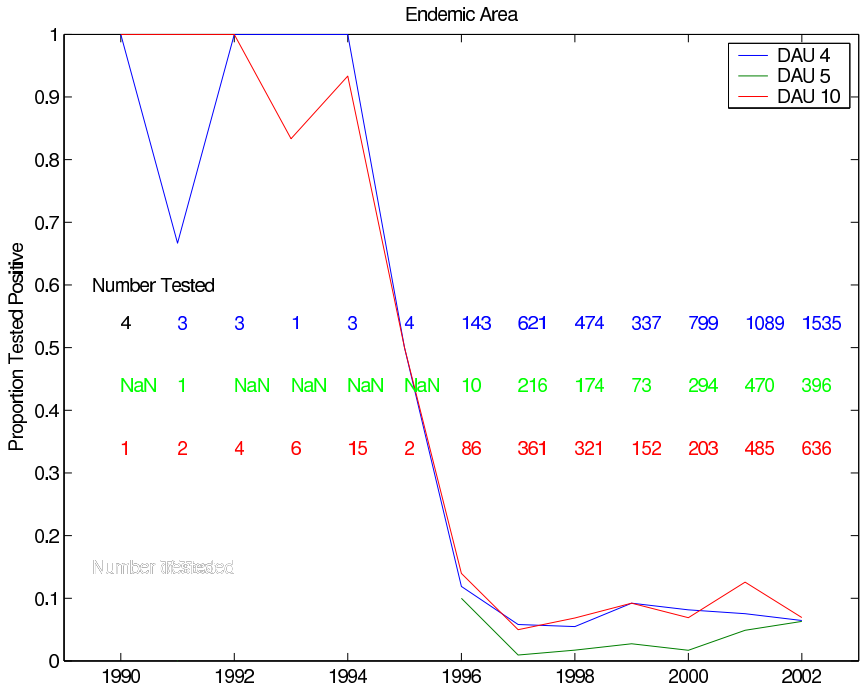

Figure 3.3: Proportion of deer which test positive by year for DAUs 4, 5, and 10. The total number tested is shown in the background, also by year. The proportion which test positive decreases as the number tested increases, as does the variability.

Given the parameters α, δ, γ, ω and π, several model variations can be formulated by determining how many, and which, of the elements of $\mathbf{p}_0$ are non-zero. Because the data begins in 1976, the year 1975 is set as year 0. The numbering scheme for the DAUs has nothing to do with Chronic Wasting Disease, so it is necessary to impose an order on them that is sensible for CWD. This is done by optimizing the posterior distribution over different arrangements of DAUs to determine which DAU is likely to have the highest infection rate of the disease down to which is least. There are a total of 54 DAUs, so to reduce the search space the focus is only on those DAUs in the endemic area where the disease is currently found. This assumption is reasonable since, at least presently, the disease is still spreading, making it unlikely that it would have disappeared from an area. The optimal ordering for the DAUs in the endemic area is 10, 4, 5, 3, 17, 9, 54, 27, and 44.

The best model given this ordering can be determined using Bayes Factors [24]. Several sets of models are tested. The first set assumes that the disease was present in only one DAU in 1975, and that all other DAUs had an initial value of 0. This is accomplished by putting the prior distribution for $p_{i,0}$ as a point mass at zero for all DAUs except for the one of interest. The second set of models assumes that two DAUs are infected at time 0 and all others are zero. The pattern continues adding one DAU at a time, up to nine, the number that are currently in the endemic region. Additionally, models were compared with $\omega > 0$ and $\omega = 0$ to determine whether or not the second order migration term was significant.

The different models are compared using Bayes factors. Bayes factors are ratios of the marginal likelihood of the data under one model to the marginal likelihood under another model. Hence:

$$BF(\mathcal{M}_i;\mathcal{M}_j) = \frac{\int \mathcal{L}(\mathbf{Y}|\boldsymbol{\eta},\mathbf{p}_0,\pi,\mathcal{M}_i)q(\mathbf{p}_0,\boldsymbol{\eta},\pi|\mathcal{M}_i)d\mathbf{p}_0 d\boldsymbol{\eta} d\pi}{\int \mathcal{L}(\mathbf{Y}|\boldsymbol{\eta},\mathbf{p}_0,\pi,\mathcal{M}_j)q(\mathbf{p}_0,\boldsymbol{\eta},\pi|\mathcal{M}_j)d\mathbf{p}_0 d\boldsymbol{\eta} d\pi} \tag{3.12}$$

where $\mathcal{M}_i$ implies that the integration is with respect to the prior and likelihood under model i [24]. Both the numerators and denominators of the Bayes factors are approximated via a random sample of the prior distributions implied by the two models $\mathcal{M}_i$ and $\mathcal{M}_j$, respectively. For $k = 1, \ldots K$, draw the appropriate collection of parameters, denoted by $\boldsymbol{\theta}_{ik}$, from the prior distribution imposed by $\mathcal{M}_i$. Let ℓ_{ik} denote the evaluation of the log-likelihood implied by (3.10) evaluated at $\boldsymbol{\theta}_{ik}$ and $w_{ik} = \exp(\ell_{ik} - \max_k\{\ell_{ik}\})$ and $\bar{w}_{i\cdot} = \sum_k w_{ik}/K$. Then $\widehat{BF}(\mathcal{M}_i;\mathcal{M}_j) = \exp\{\ell_{i\,\max} - \ell_{j\,\max}\}\bar{w}_{i\cdot}/\bar{w}_{j\cdot}$.

The Bayes factor gives a weighting of the relative skill of one model compared with another, competing, model. Under this criteria, it is found that the model with $p_{4,0}, p_{5,0}, p_{10,0} > 0$, with all other DAUs having zero starting values, and $\omega = 0$ is as or more effective than other models with the same number or more parameters. The estimated Bayes factor was 1.18 when comparing this model with a similar model that also included non-zero initial values for DAUs 3, 17, and 9. The next smallest Bayes factor (3.55) was for the comparison with the model that forced $p_{5,0} = 0$. The Bayes factors for comparison of the model with $p_{4,0},\ p_{5,0},\ p_{10,0} > 0$ and $\omega = 0$ and all remaining models were all greater than 100.

The results of the Bayes' Factor comparisons are supported by the fact that the first observed case of Chronic Wasting Disease occurred near Fort Collins,

Colorado, which is surrounded by DAUs 4, 5, and 10 [22]. These DAUs are currently the heart of the endemic area. Given this prior knowledge of the disease, it is unlikely that it would have begun anywhere else.

The particular model that is chosen is that with dynamic parameters

$$\boldsymbol{\eta} = (\alpha, \delta, \gamma),$$

and the initial conditions

$$\mathbf{p}_0 = (p_{10,0}, p_{4,0}, p_{5,0}),$$

so that $\boldsymbol{\theta} = (\boldsymbol{\eta}, \mathbf{p}_0)$. Thus, it is necessary to find the posterior distribution of $[\boldsymbol{\theta}|\mathbf{Y}]$, where $\mathbf{Y}$ denotes the data. Because this distribution does not have a convenient or known form, computational methods are needed to simulate the posterior distribution.

3.4 Accept/Reject Methods for the CWD Model

The posterior distribution for this model depends on many nonlinear relationships, which makes direct analysis difficult. Consequently, understanding the posterior is most easily achieved using a simulated random sample. The posterior distribution of the CWD model was first sampled in [37] using an acceptance sampler. The results of this sample illustrate the characteristics of the posterior distribution, and will help to determine if a more adaptable Markov chain simulation is sampling from the correct density. Developing a usable Markov chain simulation might allow for a more complete exploration of the model by future researchers.

The Accept/Reject or acceptance method of sampling results in a random sample from the desired distribution. Let $f(x)$ be the target density. Let $g(x)$

be a distribution from which it is easy to sample, called the *envelope density.* Further, suppose there exists a constant M such that $f(x) \leq Mg(x)$, everywhere on the support of the density f.

Algorithm 3.1 *The basic Accept/Reject (A/R) algorithm is:*

1. *Generate a value X from the density g, and a value U from the uniform density on* $[0, 1]$.

2. *If $U \leq \frac{f(X)}{Mg(X)}$, then accept the value X.*

An accepted X is distributed according to the density f.

The acceptance sampler can work extremely well, but its performance depends on finding an envelope density g that closely matches the target density, as well as knowledge of the constant M. Further, for properly normalized densities, the average probability of accepting the value X in the algorithm 3.1 is $1/M$, so that the expected number of iterations until a value is accepted is M [64]. Thus to make the acceptance sampler efficient, the density g must closely match the target density so that a sizable random sample can be generated in a reasonable amount of time [64, 25].

Acceptance samplers have the benefit of generating an independent sample exactly from the target distribution, and they can be applied to multivariate densities. However, the problem of finding a suitable envelope density if often prohibitive, especially in higher dimensions, although there are methods such as Adaptive Rejection Sampling that can reduce this problem for certain kinds of target densities. In addition, accept/reject samplers are difficult to modify, so that slight changes in a model can require the construction of a new sampler.

There are a total of seven parameters in the CWD model. However, the seventh parameter, π, is independent of all of the others. Its posterior distribution has a known form, $Beta(18.5, 19.5)$, so it does not need to be included as part of the simulation, since it can be simulated directly. Hence only the remaining six parameters, $p_{10,0}$, $p_{4,0}$, $p_{5,0}$, α, δ, and γ, must be sampled. Each parameter has a beta distribution as its prior, and hence the probability density function is non-zero only on the interval $(0, 1)$. Because the posterior distribution is known to be confined to the six dimensional unit cube, a beta distribution seems a natural starting point for the envelope distribution. Since there are six parameters, six independent beta distributions are used with their parameters chosen so that the modes of the beta distribution match the mode of the posterior distributions. The mode is calculated by numerically maximizing the log of the posterior function, using the Nelder-Mead non-linear optimization method [59, 55] built in to Matlab. The modes for the marginal posterior distributions in this model are shown in table 3.3. The value of the log posterior distribution at the mode was -2731.9. The specific envelope distributions for the CWD acceptance sampler were $Beta(2.7, 106)$ for the parameter $p_{10,0}$, $Beta(2.1, 164.2)$ for the parameter $p_{4,0}$, $Beta(0.7, 310.7)$ for the parameter $p_{5,0}$, $Beta(2.5, 15.6)$ for the parameter α, $Beta(9.4, 54.4)$ for the parameter δ, and $Beta(27.4, 3375.3)$ for the parameter γ. The results of the acceptance sampler are discussed in [37].

The acceptance sampler was run using Matlab, version 6, on a machine with a $2.2Ghz$ Pentium 4 processor, with $2GB$ RAM. The sampler generated a random value from the posterior roughly every 3 seconds. To generate a sample of size 25,000 took nearly 24 hours. The slowness of the acceptance sampler makes

it impractical for studying variations of the model, or for applying to future data sets. The long run time was likely due to the complicated nature of the parameter space and the independence assumed in the envelope distributions. Nevertheless, the random sample that was generated by the acceptance sampler allows for examination of the characteristics of the joint posterior, and provides a guide in designing a more efficient Markov chain sampler.

The 95% Bayesian credible intervals and the posterior estimates for the six parameters based on this sample are shown in table 3.3. These summary statistics will be used as an additional check for the Markov chain sampler.

Parameter	L.Limit	U.Limit	Mode	Mean	Median	Std.Dev.
$p_{10,0}$	1.67	5.63	2.51	3.35	3.24	1.01
$p_{4,10}$	0.82	3.17	1.27	1.77	1.69	0.61
$p_{5,0}$	0.07	0.69	0.22	0.31	0.28	0.16
α	6.67	22.13	14.08	13.01	12.51	3.98
δ	11.18	20.39	14.79	15.08	14.84	2.38
γ	0.62	1.09	0.81	0.84	0.83	0.12

Table 3.3: 95% Bayesian credible intervals, posterior estimates, and modes for the marginal posterior densities simulated using the acceptance sampler. All values have been multiplied by 100.

Scaled histograms of the variables $p_{10,0}$,$p_{4,0}$,$p_{5,0}$, α, δ, and γ from the posterior sample are given in Figure 3.4. For comparison, the prior distribution is shown with a solid line. The prior and histogram approximation of the posterior have been scaled to highlight the differences between the two distributions. This

gives some indication of both the ignorance in the prior distributions and the strength of the data in the model. Noting that the prior and posterior distributions are different for each variable, it is evident that the model is making use of the information in the available data.

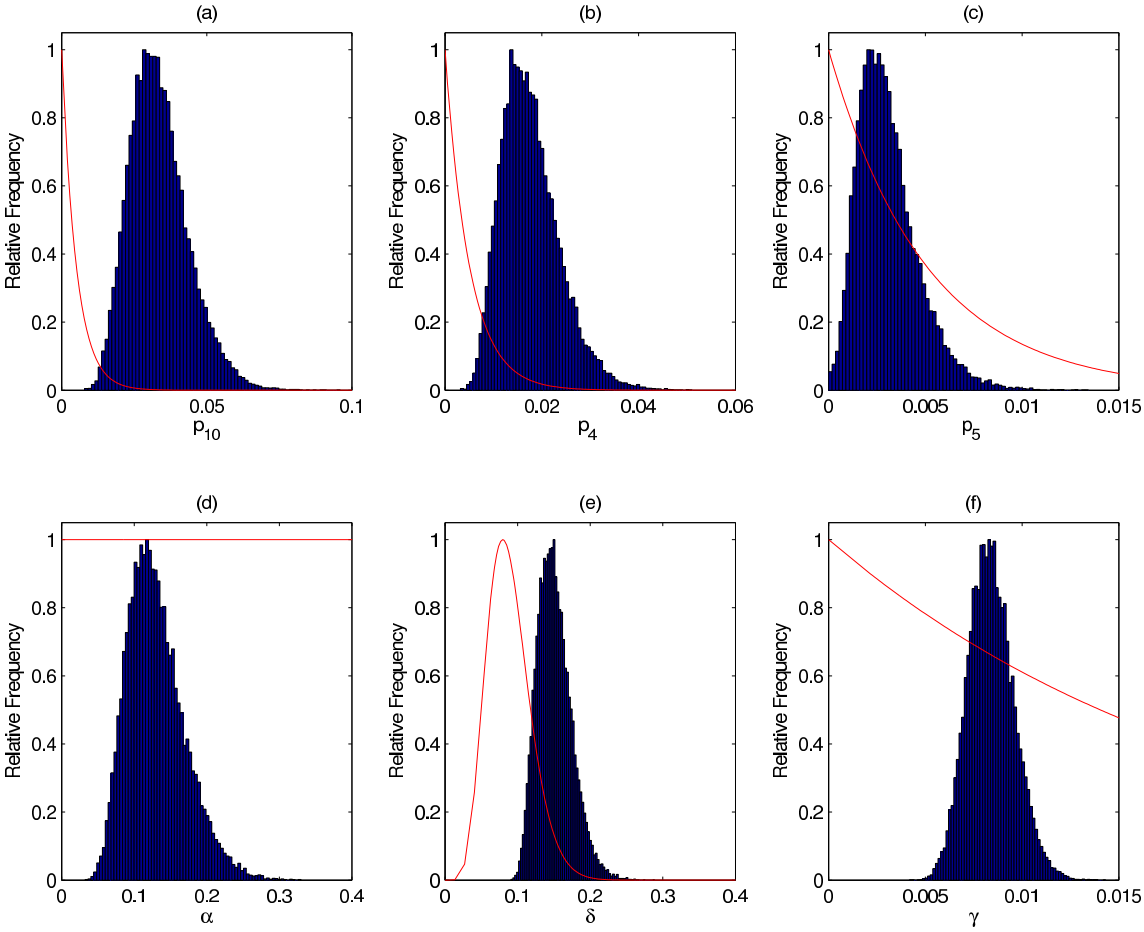

Figure 3.4: Comparisons of the posterior sample with prior distributions for parameters (a) $p_{10,0}$, (b) α the acceleration parameter, (c) δ the long run prevalence and (d) γ, the average propensity to migrate. The prior and histogram approximation of the posterior have been scaled to have maximum value of one in order to highlight the difference between the two distributions.

Examination of the posterior distribution indicates that the parameters are highly correlated. This correlation is demonstrated graphically in the pairwise

contour plots of the variables. Figure 3.5 shows the contour plot of the parameters α and δ. The points occur along a ridge of high probability. The pattern indicates a high negative interdependency between these two parameters. The ridge has a "banana" shape that is the most pronounced feature of the posterior distribution, and goes far in explaining the slowness of the acceptance sampler based on independent Beta distributions. Any successful Markov chain simulation will need to reproduce this ridge.

The pairwise contour plots of the remaining variables are shown in figures 3.6 and 3.7. Each pair plot shows a pattern that indicates a relationship between the two variables. The initial prevalence values $p_{10,0}$, $p_{4,0}$, and $p_{5,0}$ show evidence of positive correlation. The acceleration parameter α shows evidence of negative correlation with all of the other variables, while δ and γ are positively correlated.

Calculating the correlation matrix of the A/R sample confirms the graphical evidence of relationships amongst the six parameters. The correlation matrix is shown in 3.13. The initial prevalence in DAU 10 is strongly correlated with the other initial prevalences themselves, with $Corr(p_{10}, p_4) = 0.7214$ being the highest correlation within this group of parameters. The other group of variables– the dynamic parameters– also exhibits strong interdependence, although it is clear from the graphical evidence in figure 3.5 that this relationship is nonlin-

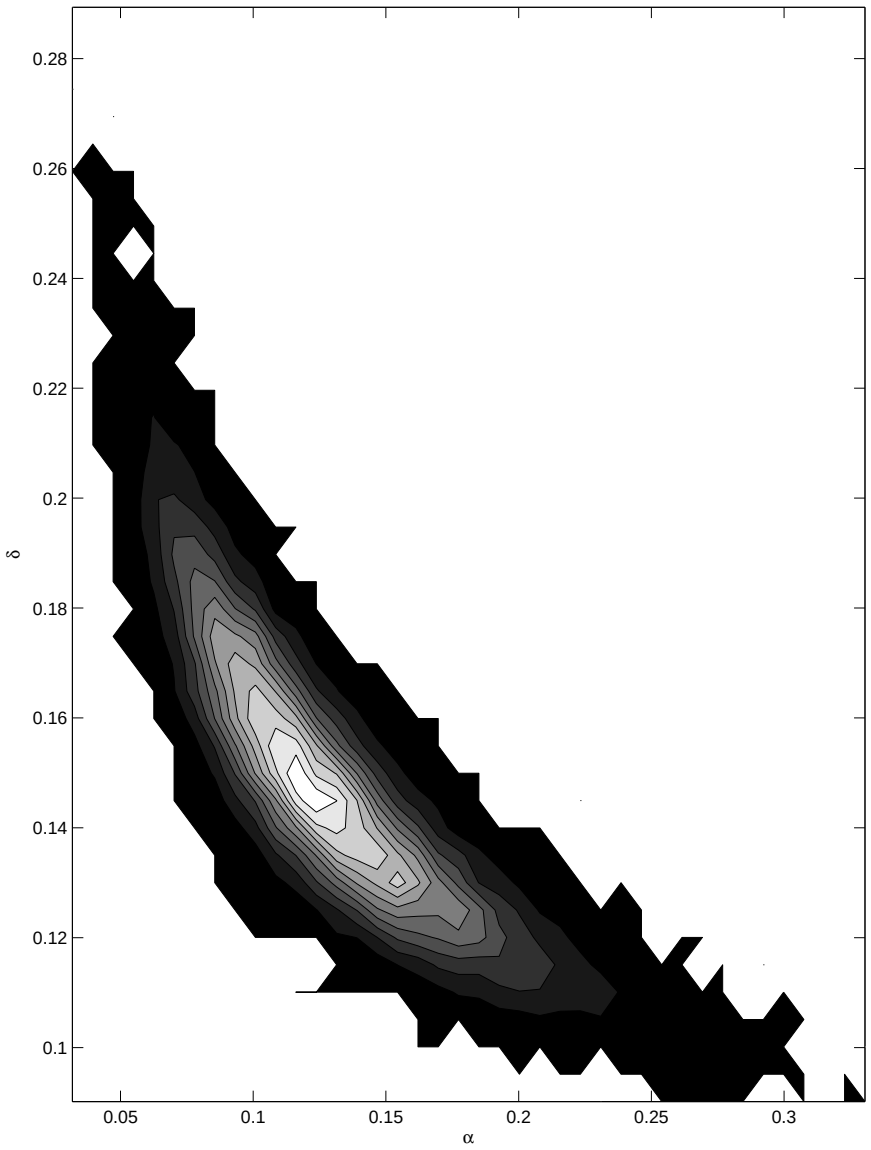

Figure 3.5: Contour plot of α by δ. Most of the values occur along a banana shaped ridge of high probability.

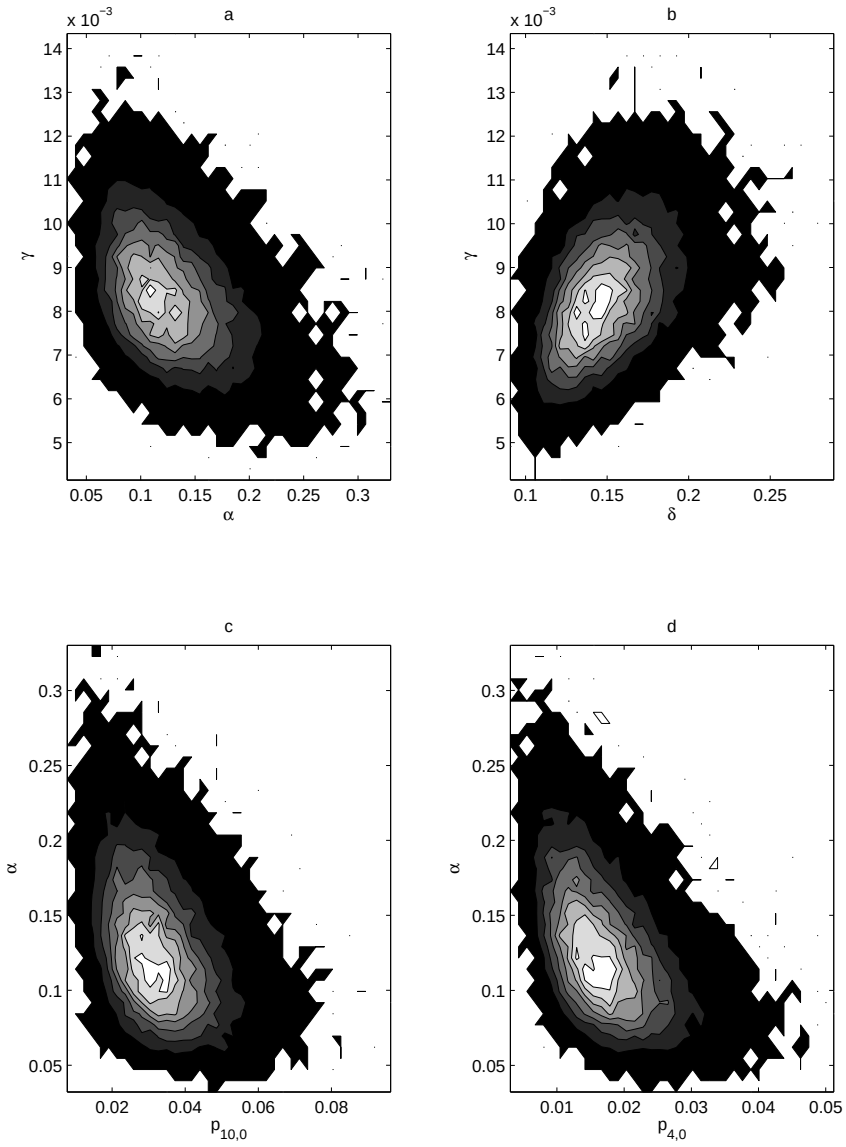

Figure 3.6: Contour plots of variable pairs for the acceptance sampler. Plot (a) shows the graph of the acceleration α versus the movement parameter γ. Plot (b) shows the graph of the long term sustainable level δ versus γ. Plot (c) shows the graph of α versus $p_{10,0}$, the initial prevalence in DAU ten. Plot (d) shows the graph of $p_{4,0}$ versus the acceleration α. These graphs indicate correlation among the parameters.

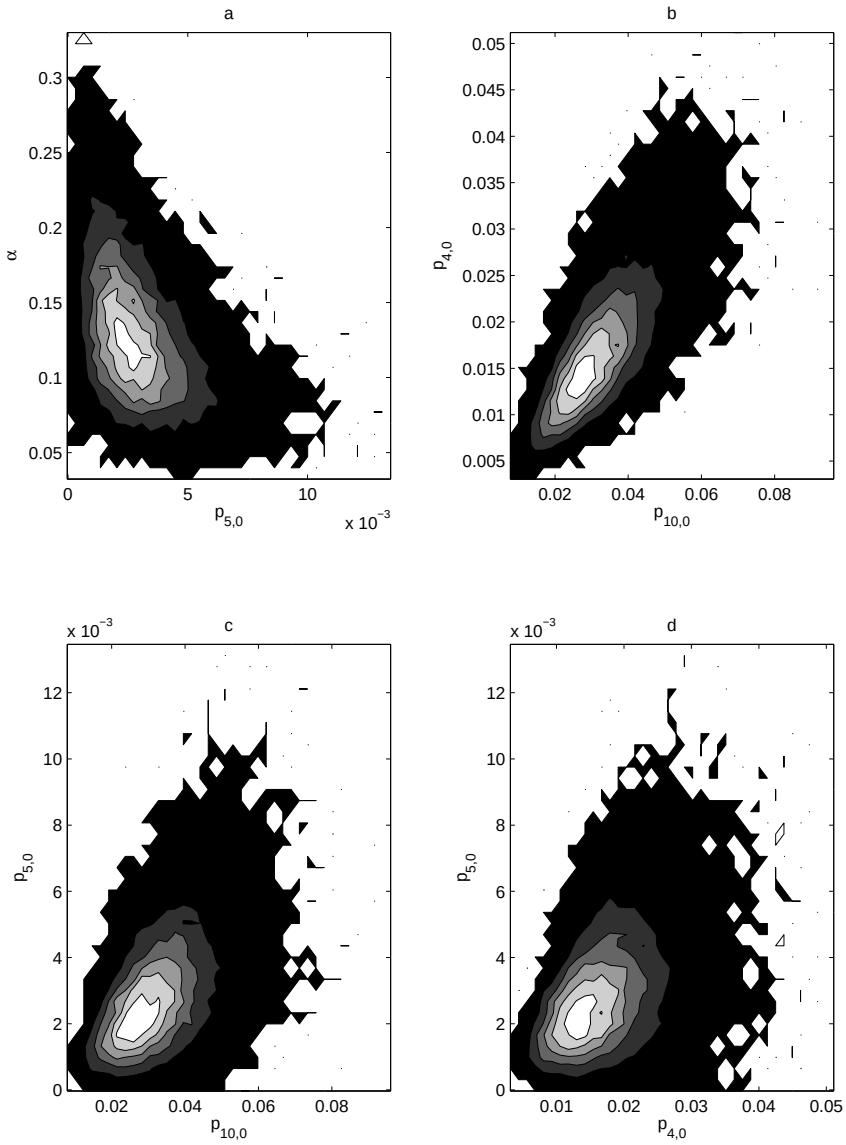

Figure 3.7: Contour pair plots of the variables for the acceptance sampler. Plot (a) shows the graph of the acceleration α versus the initial prevalence in DAU five, $p_{5,0}$. Plot (b) shows the graph of the initial prevalences $p_{10,0}$ and $p_{4,0}$. Plot (c) shows the graph of $p_{10,0}$ and $p_{5,0}$. Plot (d) shows the graph of $p_{4,0}$ versus the $p_{5,0}$.

ear.

$$\begin{pmatrix} 1.0000 & 0.7214 & 0.5135 & -0.4183 & 0.0461 & 0.0812 \\ 0.7214 & 1.0000 & 0.4060 & -0.4892 & 0.1321 & 0.2553 \\ 0.5135 & 0.4060 & 1.0000 & -0.4440 & 0.2194 & -0.0426 \\ -0.4183 & -0.4892 & -0.4440 & 1.0000 & -0.8582 & -0.4340 \\ 0.0461 & 0.1321 & 0.2194 & -0.8582 & 1.0000 & 0.4406 \\ 0.0812 & 0.2553 & -0.0426 & -0.4340 & 0.4406 & 1.0000 \end{pmatrix} \tag{3.13}$$

There appears to be two natural groups of parameters, the initial prevalence values, $p_{10,0}$, $p_{4,0}$, and $p_{5,0}$, and the dynamic parameters α, δ, and γ. There is strong interdependence within each of these groups, and some evidence, based on the pairwise graphs in figures 3.6 and 3.7, of a relationship between the two groups. However, because the model is highly nonlinear, and because the graphical evidence indicates that the relationships between the parameters are nonlinear, a more general measure is needed. A canonical correlation analysis shows the canonical correlation coefficient for the two groups to be 0.8134. This is a strong indication that the two groups of variables are interdependent. This interdependence will be an important consideration in the construction of a Markov chain sampler for the CWD model.

4. Markov Chain Monte Carlo Simulation for the CWD Model

For some distributions, qualities such as independence or conditional independence make direct sampling analytically tractable and computationally feasible. However, more complex models, such as the CWD model in chapter 3, can lead to situations where the target distribution does not have a recognizable form, or has characteristics that make direct sampling difficult or impossible to apply. When direct sampling is not possible, alternative methods such as the acceptance sampler (see section 3.4) or Markov chain Monte Carlo (MCMC) approaches must be applied. The complex form of the posterior distribution in the Chronic Wasting Disease model (equation 3.11) requires the use of computational methods to sample from the posterior distribution. This is not an uncommon situation for Bayesian hierarchical models, but in some cases the hierarchy does allow for some computational simplifications. Analytical construction and analysis of the posterior distribution is only possible in very simple hierarchical models, such as the normal model discussed in [24]. The relative advantages and disadvantages of some common simulation methods are discussed, and an efficient Markov chain sampler is constructed.

4.1 Markov Chain Methods

A *Markov Chain* is a sequence of random variables $\{X^{(t)}\}$ such that:

$$\left[X^{(n+1)}|X^{(1)},...,X^{(n)}\right] = \left[X^{(n+1)}|X^{(n)}\right]. \tag{4.1}$$

For the Markov chain $\{X^{(t)}\}$, if there exists a distribution f such that $X^{(n)} \sim f$ implies $X^{(n+1)} \sim f$, then the distribution f is called the *stationary distribution*

for the Markov chain [64, 68]. The stationary distribution is a limiting distribution, where convergence is with respect to the *total variation norm*, $\|\cdot\|_{TV}$. The total variation norm between two distributions f_1 and f_2 is defined by:

$$\|f_1 - f_2\|_{TV} = \sup_x |f_1(x) - f_2(x)|. \tag{4.2}$$

A sequence of distributions f_n *converges* to a limiting distribution f if

$$\lim_{n\to\infty} \|f_n - f\|_{TV} = 0$$

A Markov chain is called *ergodic* if the convergence to a stationary distribution does not depend on the initial state of the chain $X^{(0)}$ [64]. The possible values of the Markov chain are called the *states* of the chain. A Markov chain is called *reversible* if it has the property

$$\left[X^{(n+1)}|X^{(n)} = x\right] = \left[X^{(n+1)}|X^{(n+2)} = x\right]. \tag{4.3}$$

A *Markov chain Monte Carlo (MCMC) method* for simulating from a distribution f is any method that produces an ergodic Markov chain, $\{X^{(t)}\}$, with f as the stationary distribution. MCMC methods do not produce an independent sample exactly from the target density f. Instead, they produce a dependent sample that is approximately distributed according to f. This may seem inferior to accept/reject methods, which sample exactly from the target distribution, but Markov chain methods possess a number of benefits that make their use advantageous. First, it is usually easy to construct a Markov chain that converges, at least theoretically, to the target distribution. Markov chain methods are also easier to generalize and alter than their A/R counterparts, and therefore facilitate the exploration of more complicated models.

The *Metropolis-Hastings algorithm* is a very general method for simulating a Markov chain [64, 25]. Using virtually any distribution, a Markov chain can be constructed with the target density f as its stationary distribution.

Algorithm 4.1 *Suppose that $q(y|x)$ is a conditional density from which it is easy to simulate. Given a current value $x^{(t)}$, generate the next value in the chain as follows.*

1. *Generate a value Y from the density $q(y|x^{(t)})$.*
2. *Let $\alpha(x^{(t)}, Y) = \frac{f(Y)}{f(x^{(t)})} \frac{q(x^{(t)}|Y)}{q(Y|x^{(t)})}$.*
3. *Take $x^{(t+1)} = Y$ with probability $\min(\alpha, 1)$. Otherwise, $x^{(t+1)} = x^{(t)}$.*

Under broad regularity conditions, the algorithm 4.1 produces a Markov chain with stationary distribution f. The convergence of algorithm 4.1 depends on the relationship between the target density f and the conditional density $q(x|y)$. This relationship is given by theorem 4.1.1 taken from [64].

Theorem 4.1.1 *Let S denote the support of the density f, and suppose that S is connected. If the support of the conditional density q contains S, then the Markov chain produced by the algorithm 4.1 has f as its stationary distribution.*

There are many variations of the Metropolis Hastings algorithm. One of the most common is random walk Metropolis-Hastings, which is frequently called the *standard random walk algorithm* [25, 64, 58], and is the terminology used in this thesis. In the standard random walk algorithm, given the current state of the chain, $X^{(t)}$, the proposed value is generated by $Y = X^{(t)} + \epsilon_t$, where ϵ_t has distribution g. The conditional distribution q in algorithm 4.1 is equal to $g(y-x)$,

so that q now has the symmetric property $q(x|y) = q(y|x)$. In many applications, g is a normal distribution. If the proposal density g and the target density in the standard random walk satisfies the conditions of theorem 4.1.1, then the standard random walk will converge in the limit to the correct stationary distribution.

The conditional density q in algorithm 4.1 is called the *instrumental* or *proposal density*. Finding a proposal density is typically very easy compared to finding a suitable envelope density for the acceptance sampler. Thus the Metropolis-Hastings algorithm is in theory easy to apply. The resulting Markov chain will converge in the limit to the desired stationary distribution. Moreover, if the conditional density satisfies the conditions of theorem 4.1.1, then the Markov chain produced by algorithm 4.1 is ergodic [64]. After a suitably large number of iterations, T_0, called the *burn in period*, the chain provides a good approximation to the target distribution.

For multivariate densities, the components can be sampled individually, and the target distribution constructed by means of the *complete conditional distributions*, that is, the distribution of the component X_i given all of the other components. Suppose that the random variable $\mathbf{X}$ has the desired multivariate density f, and that $\mathbf{X}$ can be written as $\mathbf{X} = (X_1, ...X_n)$. Let $[X_i|x_1, ...x_{i-1}, x_{i+1}, ..., x_n]$ denote the complete conditional distribution of X_i given all of the other components of $\mathbf{X}$.

Algorithm 4.2 *Given* $\mathbf{X}^{(t)} = (x_1^{(t)}, ..., x_n^{(t)})$, *a Markov chain with stationary distribution* f *is produced by the following algorithm:*

1. *Generate* $X_1^{(t+1)} \sim \left[X_1 | x_2^{(t)}, ..., x_n^{(t)}\right]$.

2. *Generate* $X_2^{(t+1)} \sim \left[X_2 | x_1^{(t+1)}, x_3^{(t)}, ..., x_n^{(t)}\right]$.

3. *Repeat for each conditional until...*

4. *Generate* $X_n^{(t+1)} \sim \left[X_n | x_1^{(t+1)}, ..., x_{n-1}^{(t+1)}\right]$.

The algorithm 4.2 is called the *Gibbs sampler*, and is a classic example of a strategy called *successive substitution sampling* (SSS), which enables high dimensional problems to be broken into a series of univariate or simple block sampling problems. The Gibbs sampler is a special case of the algorithm 4.1 in which the proposed value is always accepted. The Gibbs sampler is typically used when the complete conditional densities have a known form and are easy to sample. In general, the blocks X_i may be either univariate or multivariate. For problems in which the complete conditional distributions are unknown, or do not have a simple form, the conditional densities in 4.2 may be sampled using a Metropolis Hastings step.

In practice, the burn in period for a Markov chain sampler can be prohibitively large. While building a Markov chain that theoretically converges is relatively easy, building one that works in a reasonable amount of time may be difficult. This difficulty is more common in higher dimensional problems where the straight forward application of algorithm 4.2 by means of the complete conditional distributions is impossible. Complex models may lead to situations where the target distribution has characteristics that make the chains produced by standard MCMC methods *mix* slowly, that is, the Markov Chain does not move rapidly about the support of the target distribution, which causes the

chain to converge slowly. This leads to a variety of techniques to accelerate convergence of the chain. The application of these techniques often involves careful tuning by trial and error.

Complex multivariate distributions can have a number of features that make a density difficult to simulate. If the target distribution is multi-modal, has sharp ridges of high probability, exhibits high correlation between individual variables, or exists on a bounded parameter space, then constructing a Markov chain that converges to the stationary distribution of interest in a reasonable amount of time may be difficult. For multi-modal target distributions, the chain can get stuck near one mode, oversampling near that mode and perhaps missing others entirely. When a target distribution has sharp ridges of high probability, a Markov chain can bypass these areas if the average *step size*, that is, the distance between the current state $X^{(t)}$ and the proposed future state Y, is too large. The step size is determined by both the mean and the variance of the proposal distribution. When there is high correlation between individual variables, sampling each component separately is inefficient, since the acceptance or rejection of one variable can have a large impact on the acceptance or rejection of another. With bounded parameter spaces, care must be taken to not propose outside of the parameter space too often, since too many rejected proposals will slow down convergence of the chain.

There are a variety of techniques available to alleviate some of the possible difficulties when sampling from multivariate distributions. When the variables are highly correlated, reparameterization can remove some of the correlation. Examples of regression models and random effect models where this approach

is especially effective are given in [25]. Implementation of a reparameterization technique requires the identification of an appropriate transformation, which may not be an easy task.

The technique of blocking seeks to utilize the correlation between parameters to accelerate convergence. Blocking makes use of the correlation between parameters in a multivariate setting by updating highly correlated variables together to increase the *mixing speed* of the chain, that is, the speed at which the chain moves around the support of the target density [29]. This method can be especially effective in hierarchical models with spatially structured data. A discussion of how the blocking approach can be used when dealing with *areal data*, i.e. data aggregated over discrete spatial regions, is found in [29, 6].

Hit and run samplers are a class of algorithms that have been designed to simulate target distributions that exhibit signs of multi-modality [72, 11, 25]. The hit and run samplers are based on a generalization of the Metropolis-Hastings algorithm [72, 11, 58], and use a proposal distribution that chooses a direction at random, and then moves a random distance in that direction. The difference between the hit and run method and the standard random walk is that the new move is always accepted. The hit and run sampler can be effective in sampling multi-modal target distributions, because the algorithm, with some small probability, traverses the entire parameter space in a single move, which allows mode-hopping when the proposed move is in the direction of another mode. The hit and run algorithm has problems with mixing when the target distribution has sharp ridges or spikes of high probability. The hit and run algorithm may not pick out these areas of high probability sufficiently often,

causing the resultant chain to mix slowly. The problem of slow mixing becomes even more significant in higher dimensions.

4.2 Stochastic Dynamics Methods

Stochastic dynamics methods are a class of algorithms motivated by stochastic differential equations. Stochastic dynamics methods are commonly used to simulate physical systems such as quantum molecular dynamics and heat transfer. There are a number of Markov chain methods based on stochastic differential equations, many of which are specifically designed for use in statistical physics, although some have seen broader application [58]. The Langevin algorithm is perhaps the most common, and is based on the discrete simulation of a continuous time diffusion process [58, 25, 64]. Stochastic dynamics methods suppress the random walk behaviour of a Markov chain, which has been shown to improve convergence results for high dimensional target densities [28].

A *general diffusion process* $X_t = X(t)$ is a continuous time stochastic process defined as the solution to the stochastic differential equation:

$$dX_t = \mu(X_t)dt + \sigma(X_t)dB_t, \tag{4.4}$$

where B_t is standard Brownian motion [41]. In equation 4.4, $\mu(X_t)$ is called the *drift term* or *drift coefficient*, and $\sigma(X_t)$ is the *diffusion term* or *diffusion coefficient*. For simple cases, a solution to 4.4 can be found using the Ito Calculus, which is derived in [13, 41]. If a solution to equation 4.4 does not exist or X_t tends to infinity in a finite amount of time, then the diffusion process is called *explosive* [41].

If a target density f is continuously differentiable, and if for some real numbers $N, a, b < \infty$, when $(\nabla f(x))^T x \leq a\|x\| + b$, for all $\|x\| > N$, then a diffusion

process can be constructed that has f as its stationary distribution [41, 71, 66]. The only non-explosive, reversible diffusion with this property is the Langevin diffusion [64], which has the form:

$$dX_t = \frac{1}{2}\nabla \log(f(X_t))dt + dB_t. \tag{4.5}$$

Where B_t is the standard Brownian motion.

A continuous time stochastic process cannot be directly simulated. Instead, a discretized version of equation 4.5 must be used. The simplest such discretization is the Euler discretization. If $\omega > 0$ is the size of the discretization, then the continuous time diffusion process 4.5 is approximated by a *smart random walk*, with steps given by:

$$X_{t+1} = X_t + \frac{\omega^2}{2}\nabla \log f(X_t) + \omega\epsilon_t, \tag{4.6}$$

where ϵ_t follows a multivariate normal distribution with mean 0 and covariance I. The random walk in 4.6 is called "smart" because proposed values are generally in directions of higher probability than the current state. Equation 4.6 can also be written as:

$$X_{t+1} \sim \mathbf{N}(X_t + \frac{\omega}{2}\nabla \log f(X_t), \omega^2 I). \tag{4.7}$$

The Euler discretization in 4.6 and 4.7 is a first order approximation to the stochastic differential equation 4.5. Other methods based on higher order approximations have been developed, notably the Ozaki discretization which uses a second order approximation [67]. A variety of discretization schemes for stochastic dynamics methods are discussed in [58].

Unfortunately, the behaviour of the continuous time Markov process and the discretized version may be radically different. This is because the Markov

chain based on stochastic process 4.5 samples from the correct distribution only as the discretization size goes to zero [58]. Consequently, there are many situations where the random walks given by equations 4.6 and 4.7 can be *transient*, that is, there exists a state X of the chain such that the expected number of times the Markov chain visits this state is finite. A Markov chain that is not transient is *recurrent*. Situations that lead to transience include those for which the target distribution is *insufficiently smooth*, or *erratic*, so that the gradient experiences large changes in a local area or has a large magnitude. This erratic behaviour of the gradient makes the chain unstable. Many of these problems can be corrected with a Metropolis-Hastings rejection step [5, 64, 25, 58]. This method of correcting for the discretization is sometimes called *Hybrid Monte Carlo*, *Langevin Monte Carlo* [58], or a *Metropolis Adjusted Langevin Algorithm* (MALA) [71].

Algorithm 4.3 *The MALA can be summarized as follows. Let the target distribution be denoted by* $f(x)$*, and let* $\mu(x) = \frac{\omega}{2}\nabla \log f(x)$*. Let* $\alpha(x, y)$ *be defined by:*

$$\alpha(x, y) = \frac{f(y)}{f(x)} \frac{\exp(-\frac{\|x-y-\mu(y)\|^2)}{2\omega^2})}{\exp(-\frac{\|y-x-\mu(x)\|^2)}{2\omega^2})}. \tag{4.8}$$

Given the current state X^t*, the Markov chain is simulated by:*

1. *Generate* $Y \sim \mathbf{N}(X^t + \frac{\omega}{2}\nabla \log f(X^t), \omega^2 I)$
2. *Find* $\min(\alpha(X^t, Y), 1)$.
3. *With probability* $\min(\alpha, 1)$*, set* $X^{t+1} = Y$*. Otherwise, set* $X^{t+1} = X^t$.

The proposal value is kept with probability $\min(\alpha, 1)$. The rejection step 4.8 turns the Langevin algorithm into a Metropolis-Hastings algorithm with a smart

proposal [25, 64]. The conditional density in the Metropolis-Hastings algorithm 4.1 for the MALA is $q(x|y) \propto N(y + \mu(y), \omega^2 I)$, where the form of the conditional density in 4.8 comes from the Brownian motion in the stochastic differential equations 4.4 and 4.5. The key difference of the Metropolis adjusted Langevin algorithm from the standard random walk is that the MALA suppresses the random walk behaviour of the chain by using the local properties of the target density (i.e. $\nabla \log f(X_t)$) to adjust the proposal distribution [28]. This nudges the chain "uphill" in the direction of a mode, which is a highly desirable property in high dimensional problems where the behaviour of the gradient is smooth [71, 64, 25, 67].

The chain 4.3 will tend to propose values that are in the direction of the mode. Because algorithm 4.3 is a special case of the Metropolis Hastings algorithm, convergence of the chain relies on the properties of algorithm 4.1, and not those of the Langevin diffusion process. Strategies such as reparameterization and blocking can be employed with 4.3 to increase the mixing speed. The simulation can be further adjusted for erratic behaviour of the gradient by using a truncated proposal step to achieve more robust ergodic properties [5, 71]. This truncation comes through the imposition of an upper limit on the step length in the proposal distribution [25, 65].

The MALA has been shown to perform as well as or better than the standard random walk in many settings [66, 67, 71]. As with other kinds of Metropolis-Hastings simulations, convergence properties depend largely on the smoothness of the target distribution. One situation where the MALA works especially well is when the target distribution is from the exponential family of distributions. In

situations where the target distribution is discontinuous or is not differentiable at some points, then smoothed proposals can be employed [71].

A recent variation of the Langevin algorithm is the *tempered Langevin algorithm* [67]. The standard MALA approach uses a fixed variance coefficient (ω in equations 4.6 and 4.7). A more general diffusion process can be used to get another variation of the Langevin algorithm. Note that equation 4.5 is a special case of equation 4.4, where the drift term is $\mu(X_t) = \frac{1}{2}\nabla \log(f(X_t))$, and the diffusion coefficient is $\sigma(X_t) = \omega I$. If a more general diffusion term is used in 4.4, then the tempered Langevin diffusion is obtained [67].

If the diffusion coefficient is $\sigma(x)$, then the *diffusion matrix* is given by a positive definite matrix $a(x) = \sigma(x)\sigma^T(x)$. Let $0 \leq d \leq \frac{1}{2}$. Choosing $a(x) = f^{-2d}(x)I = \frac{1}{(f(x))^{2d}}I$, and setting the drift term to be $\mu(x) = \frac{1-2d}{2}a(x)\nabla \log(f(x))$ gives the Langevin tempered diffusion [67]. The tempered diffusion is the solution to the stochastic differential equation:

$$dX_t = \frac{1-2d}{2} f^{-2d}(X_t)\nabla \log(f(X_t))dt + f^{-d}(X_t)IdB_t \tag{4.9}$$

The diffusion in equation 4.9 is a more general form of the Langevin diffusion 4.5 and is sometimes called a *heated Markov chain.* The two special cases of equation 4.9 that are important are when $d = 0$, which implies the standard Langevin diffusion, and $d = 1/2$, which gives Brownian motion [67]. Choosing different values of d gives different heated chains.

Heated Markov chains have stronger theoretical convergence properties than unheated chains, since the diffusion matrix acts as an accelerator [67]. The accelerator can alleviate some of the problems encountered with multi-modal target distributions. In areas of low probability, the variance $a(x) = f^{-2d}(x)I$

will be larger, causing the chain to take larger steps in those areas. As the chain enters areas of high probability, $a(x)$ will decrease, leading to a smaller step size.

The discretized version of the diffusion 4.9 is given by:

$$X_{t+1} \sim \mathbf{N}(X_t + \frac{1-2d}{2} a(X_t) \nabla \log f(X_t), a(X_t)\ I). \tag{4.10}$$

Using a rejection step gives a Metropolis algorithm. To adjust the probability 4.8 for the tempered diffusion, set $\omega^2 = a(X_t)$, which gives:

$$\alpha(X_t, Y_t) = \frac{f(Y_t)}{f(X_t)} \frac{\exp(-\frac{\|X_t - Y_t - \mu(Y_t)\|^2)}{2a(X_t)})}{\exp(-\frac{\|Y_t - X_t - \mu(X_t)\|^2)}{2a(X_t)})}, \tag{4.11}$$

where $\mu(x) = \frac{1-2d}{2} a(x) \nabla \log(f(x))$ and $a(X_t) = f^{-2d}(X_t)$.

The tempered Langevin algorithm, like the standard Langevin algorithm 4.3, is a random walk that makes "smart jumps". That is, the chain is likely to propose values that are in the direction of an area of high probability.

Blocking, hit and run algorithms, Langevin algorithms, and tempered Langevin algorithms, are only a subset of the many methods for sampling from complicated distributions. When more than a couple of difficulties are encountered, it can require the innovative use of several of these or other approaches to generate a sample.

The joint posterior distribution produced by the Chronic Wasting Disease model presented in chapter 3 has a number of characteristics that make the implementation of a Markov Chain simulation difficult. Firstly, the posterior distribution exists on a bounded parameter space. Secondly, calculation of the gradient at various points of the parameter space indicates that the surface of the density function contains sharp ridges of high probability.

The sharp ridges of high probability mean that a hit and run type algorithm could miss the area of high probability, thereby over-sampling regions where the density is comparatively low. Langevin algorithms also have difficulty in this situation, because the steepness of the density causes the magnitude of the gradient to be extremely large in areas of moderate probability, a common source of instability in Langevin algorithms. The difficulties are compounded with the bounded parameter space, since proposals are likely to fall outside the support of parameter space. When using a normal proposal distribution, as with the standard Langevin algorithm, too many proposals outside the parameter space will cause the chain to mix slowly.

One possible method of dealing with this problem is to use a proposal distribution in algorithm 4.3 with the same support as the parameter space. However, even using a different proposal distribution, such as a truncated normal distribution on $[0, 1]$, can slow the mixing of the Markov chain, because the potential asymmetry of the truncated normal can cause difficulties for the standard random walk or MALA. Additionally, the convergence theorems for the stochastic differential equation 4.5 no longer apply, since the noise term is no longer a Brownian motion.

4.3 A Tempered Langevin Algorithm for Bounded Densities

The posterior sample produced by the acceptance sampler in [37] and discussed in chapter 3 illustrates the complicated nature of the posterior distribution 3.11. The largest issue is the fact that the support of the posterior density exists on a bounded parameter space. All of the parameters, $p_{10,0}$, $p_{4,0}$, $p_{5,0}$, α, δ, and γ, exist only within the unit interval, whereas standard approaches

to constructing a Markov chain simulation often have target distributions that exist on the set of real numbers. The restriction of a bounded parameter space requires the choice of a sufficiently small proposal variance for a random walk sampler, otherwise, the chain may propose values outside the parameter space too often, causing it to converge slowly. On the other hand, if the variance of the proposal distribution is too small, the random walk may converge slowly since it may take too long to adequately move around the support of parameter space. Suitably scaled, a Langevin type algorithm can provide some relief for this problem, as it will nudge the chain in the direction away from areas of low probability. A truncated Langevin algorithm may also provide some relief, but the problem of scaling remains.

Another complication of the posterior distribution for the CWD model is the sharp ridges of high probability. These ridges are especially noticeable in the plot of α versus δ (figure 3.5). The step size must be carefully chosen to avoid frequently overshooting these concentrated high probability areas. Additionally, the chain could get stuck in an area of high probability, so that if the target distribution has more than one mode, it will not reach all of them.

Many of the problems presented by the bounded parameter space and sharp ridges of high probability can be overcome by using a stochastic dynamics method, such as the Langevin algorithm given by equation 4.6. However, the standard Langevin algorithm can have instability introduced by using the gradient. If the target distribution has local properties that lead to erratic behaviour in the gradient, the instability results in wildly overshooting modes or areas of high probability. This could be corrected by using an upper bound on the step.

For example, proposals of the form $Y \sim N(X_t + \min(b, \frac{\omega}{2}\nabla \log f(X_t)), \omega I)$, can retain most of the advantages of the Langevin algorithm while removing some of the inherent instability of the gradient [25]. For the bounded parameter space of the CWD model, however, the choice of the upper bound b is a difficult problem, and would likely need to depend on the current state X_t.

The use of a truncated proposal in the Langevin algorithms is called a Metropolis adjusted Langevin truncated algorithm (MALTA). These chains can have better convergence properties than the MALA [71]. By retaining the "uphill" direction while putting an upper bound on the step, many of the advantages of the Langevin algorithm are kept, while the disadvantages caused by instability in the gradient are lost. However, for target distributions with insufficiently smooth surfaces, scaling the MALTA can be difficult. As an example, if the posterior distribution possesses a narrow region of high probability, surrounded by an area of low probability, such as the Witch's Hat distribution [64], then the chain may still overshoot the mode, resulting in oversampling. Calculation of the gradient of the log of the posterior density at various points indicates that the density surface is very steep for the CWD model. The magnitude of the gradient of the log density ranges from values on the order of 10 to values greater than 10^7.

The tempered Langevin algorithm 4.10 provides a remedy for this difficulty in many cases, and has been shown to have better theoretic convergence properties [67]. With its variable step size, it will, on average, take larger steps in areas of low probability, and smaller steps in areas of high probability. For target distributions in the exponential family of densities, this technique can be highly

effective [67]. However, the tempered Langevin algorithm relies on a complicated scheme for the proposal variance, making it difficult to implement and computationally infeasible. For the Chronic Wasting Disease model, use of the tempered diffusion may require a transformation of parameters, since empirical study on the tempered diffusion has thus far been only on target densities with unbounded support [67].

Individually, the problems mentioned above can make the construction of an effective Markov chain simulation difficult. In combination, these characteristics can lead to serious problems, which are exacerbated in high dimensions. Nevertheless, a Markov chain simulation can be constructed that will approximately produce the posterior distribution 3.11. To handle the complexities of the posterior distribution and to construct an efficient simulation method, features from the truncated and tempered Langevin algorithms are used. These features are tailored to handle the particular difficulties presented by the posterior in the chronic wasting disease model.

4.3.1 Markov Chain Simulation Attempts for the CWD Model

Several attempts at constructing a Markov chain simulation for the CWD model were made using the standard random walk, the Metropolis adjusted Langevin algorithm, and the tempered Langevin algorithm methods discussed above. Each method was run for $300,000$ iterations, but graphical evidence showed that they failed to produce the correct stationary distribution.

The first method attempted was the standard random walk. Because of the correlation the parameters in the CWD posterior density, a blocking strategy was employed, and all six parameters were updated simultaneously. The proposal

distribution was a six dimensional multivariate normal distribution, with the variance of each component chosen to be proportional to the prior variance. A large number of proportionality constants for the proposal variance were tried in an attempt to construct a chain that mixed and converged quickly. Most values for the proposal variance resulted in a chain that moved very little, since many of the proposed values lay outside the parameter space. An extremely small step size was needed to prevent this from occurring. Figure 4.1 shows $300,000$ iterations of the pairwise plot of the parameters α and δ chain, for a random walk sampler with a variance on the order of 1×10^{-9}. The graphs shows little movement for these parameters, indicating that most of the proposed values are being rejected. The distinctive banana shape seen in figure 3.5 is not present in figure 4.1, although the attraction to the mode is evident. The pairwise plots for some of the other six parameters are shown in figure 4.2. These graphs clearly do not match those produced by the acceptance sampler shown in figure 3.6. Reducing the proposal variance results in more accepted values for the parameters, but the chain moves very slowly about the parameter space, and when started in an area of low probability remained there after $300,000$ iterations.

The second method attempted was the Metropolis adjusted Langevin algorithm 4.3. The MALA had problems similar to the standard random walk, but these were more pronounced. Figure 4.3 shows the pair plot of the parameters α and δ. The gradient for the posterior distribution in the CWD model causes instability in the algorithm. The log posterior density ranges from values on the order of -10^9 to -2731.9, which in turn causes large changes in the gradient,

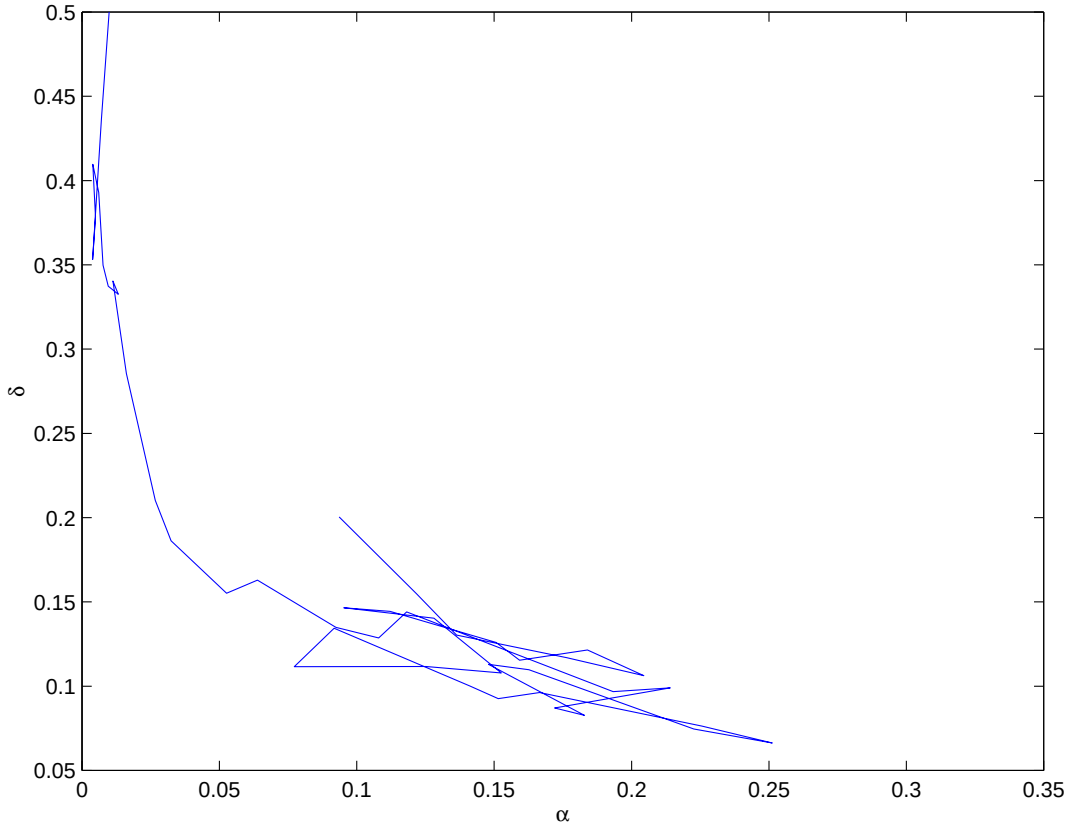

Figure 4.1: Pair plot for all $300,000$ iterations of the parameters α and δ produced by the standard random walk sampler. The distinctive banana shape seen in the acceptance sample is not present. The algorithm failed to produce the correct distribution after $300,000$ iterations.

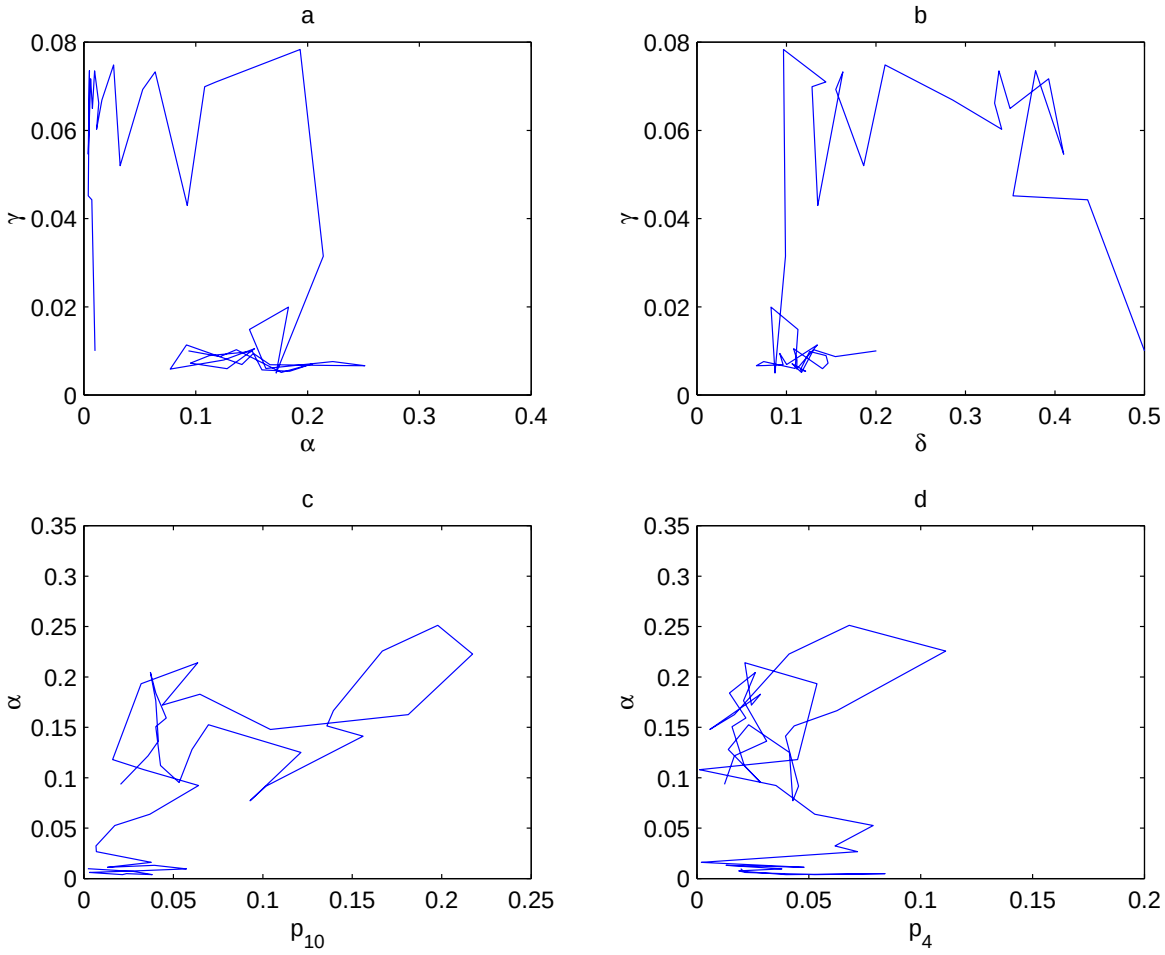

Figure 4.2: Pair plots for all iterations of the variables for the standard random walk sampler. The algorithm failed to produce the correct distribution after $300,000$ iterations. Plot (a) shows the graph of the acceleration α versus the movement parameter γ. Plot (b) shows the graph of the long term sustainable level δ versus γ. Plot (c) shows the graph of α versus $p_{10,0}$, the initial prevalence in DAU ten. Plot (d) shows the graph of $p_{4,0}$ versus the acceleration α.

making the choice for the discretization step (ω in equation 4.3) extremely difficult. Because of the differences in the posterior standard deviations shown in the sample generated by the acceptance sampler (discussed in chapter 3), the proposal variance was different for each component, and was proportional to the prior variance. Larger values for ω caused little movement in the chain. The chain would become stuck in some areas of the parameter space, since extremely large gradient values resulted in many proposals that lay outside the parameter space, and hence were rejected. Choosing extremely small values for the ω had better results, but caused the chain to move very slowly about the parameter space. Figure 4.3 shows the pairwise plot of the parameters α and δ for $300,000$ iterations of the Metropolis Adjusted Langevin Algorithm with $\omega = 1 \times 10^{-20}$ Although the chain appears to be heading towards the mode, the moves are so small that even after the large number of iterations the chain has failed to produce the distinctive ridge seen in figure 3.5.

Finally, the tempered Langevin algorithm 4.10 was attempted. This algorithm was hampered by the particular form of the diffusion matrix in 4.10, since the wide range of values for the log posterior corresponded with a wide range of values for the proposal variance. In addition, the extremely small values of the log posterior at times resulted in a nearly zero or infinite proposal variance. The problems with the proposal variance caused the algorithm to fail. Figure 4.4 shows the pairwise plot of α and δ for $300,000$ iterations for the tempered Langevin algorithm. Figure 4.5 shows four other pairwise plots. The algorithm has clearly failed to reproduce the density shown in figures 3.5 and

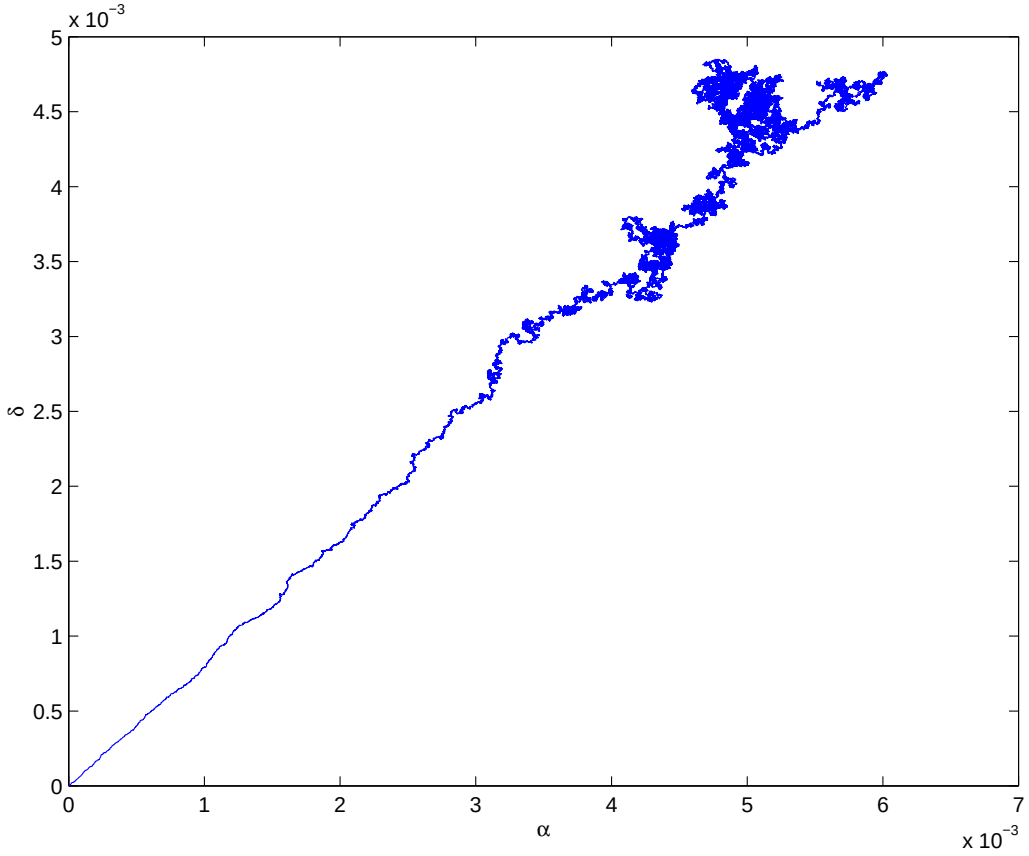

Figure 4.3: Pair plot of all iterations for the parameters α and δ produced by the Langevin sampler. The distinctive banana shape seen in the acceptance sample is not present. The step size was on the order of 1×10^{-20}. The algorithm failed to produce the correct distribution after $300,000$ iterations.

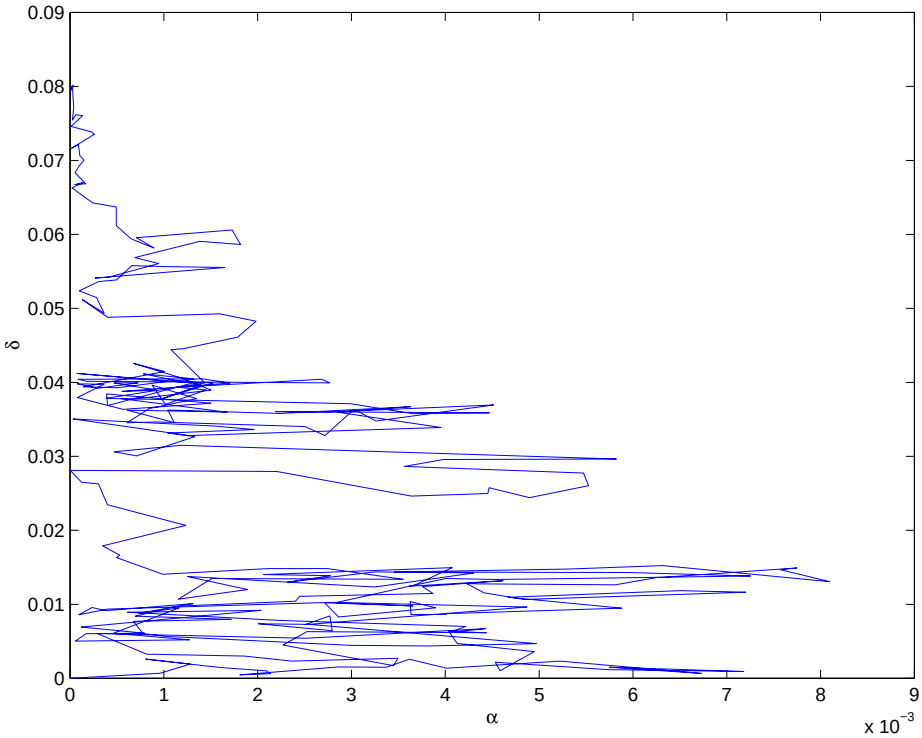

Figure 4.4: Pair plot of all iterations for the parameters α and δ produced by the tempered Langevin sampler. The distinctive banana shape seen in the acceptance sample is not present, and the chain was unable to venture far from its initial position at the origin. The algorithm failed to produce the correct distribution after $300,000$ iterations. Note from the axes that the chain is tightly concentrated near its starting value at the origin.

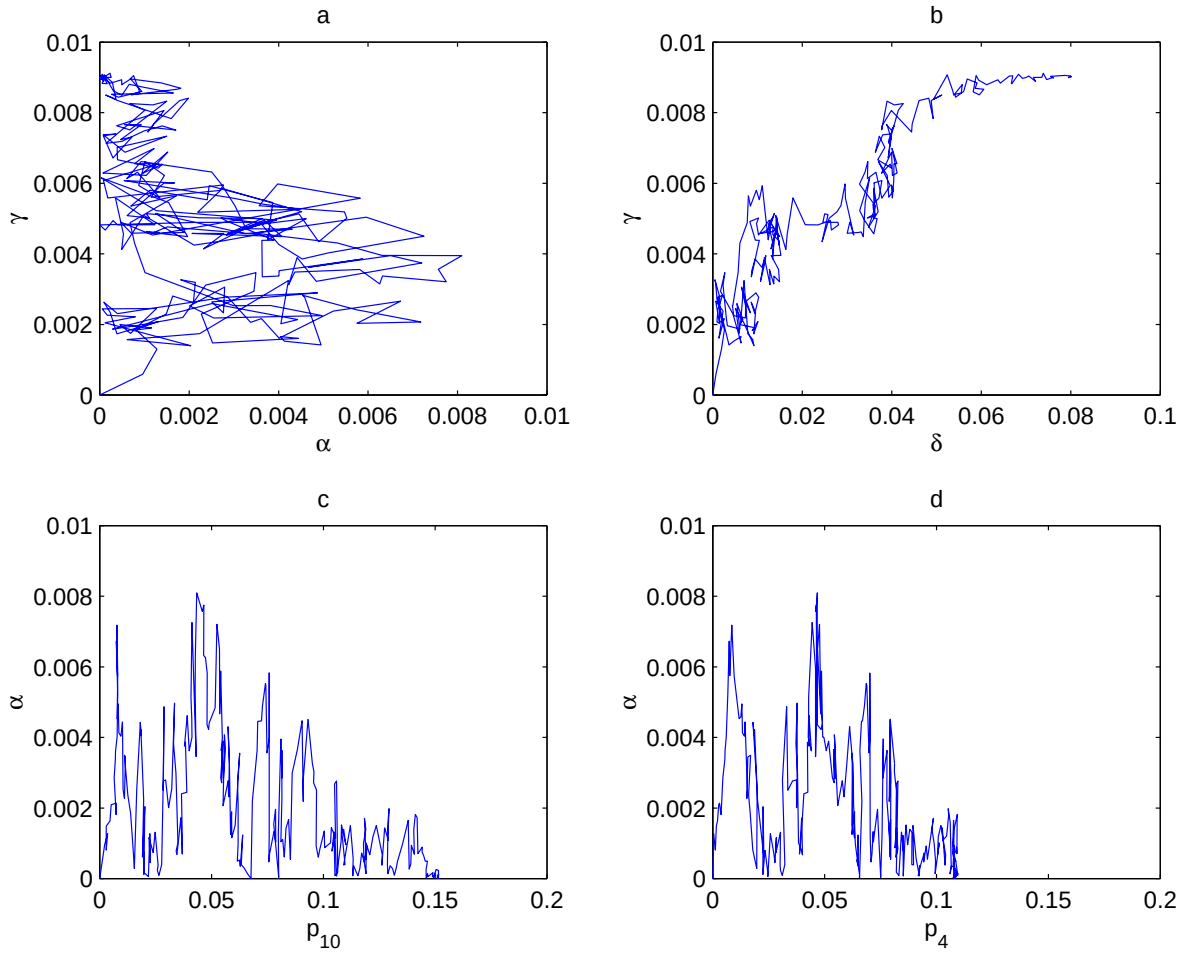

Figure 4.5: Pair plots of all iterations for the variables for the tempered Langevin algorithm. The algorithm failed to produce the correct distribution after $300,000$ iterations. Plot (a) shows the graph of the acceleration α versus the movement parameter γ. Plot (b) shows the graph of the long term sustainable level δ versus γ. Plot (c) shows the graph of α versus $p_{10,0}$, the initial prevalence in DAU ten. Plot (d) shows the graph of $p_{4,0}$ versus the acceleration α.

3.6.

4.3.2 Overcoming the Difficulties

The failure of the standard random walk, Metropolis adjusted Langevin algorithm, and tempered Langevin algorithm, to correctly and efficiently sample the posterior distribution for the Chronic Wasting Disease model prompts the construction of a new simulation method. The new method is suitable for target densities with bounded parameter spaces and concentrated areas of high probability, yet retains the qualities of the MALTA and the tempered Langevin method.

The greatest benefit of Langevin type algorithms is the use of the gradient in choosing a direction for the proposal value. However, for a target density with erratic gradient behaviour, the magnitude of the proposed value could be extremely large. Proposals with a large magnitude are a serious problem in a confined parameter space, but the gradient direction is still useful. The direction can be retained by turning the gradient into a unit vector, thus rather than using $\nabla \log f(X_t)$, it is scaled to

$$\frac{\nabla \log(f(X_t)}{\|\nabla \log(f(X_t)\|}. \tag{4.12}$$

This keeps the direction information in the gradient, but allows for more precise control over the step size. However, this requires that the gradient of the log density exists, for equation 4.12 to be defined. If the gradient of the log density is the zero vector, as it would be at the mode of the target density, then 4.12 is defined to be the zero vector, which avoids potential complications. The analytical gradient can also be replaced with a numerical approximation if necessary.

Rather than attempt to find a single step size for all parameters, a dynamic step size is constructed that is similar to the tempered Langevin algorithm. Qualitatively, the chain is "heated", so that the further the current state is from the mode of the distribution, the larger the step size that should be taken. When the current state is close to the mode, the steps should be smaller. In the case of the CWD model, the fact that the density function has small values over much of the parameter space makes the use of the posterior density function unstable, thus the log of the posterior density is used instead.

A expression must be found that is at a minimum when the current state, $X^{(t)}$, is at the mode of the target density. It should get larger the further $X^{(t)}$ gets from the mode. For the stability purposes discussed above, this expression should use the log of the posterior density. Let f denote the posterior density function. Let $\hat{x}$ denote the mode of the posterior density. Then for any $X^{(t)}$ in the support of f, we have $f(X^{(t)}) \leq f(\hat{x})$. Since the logarithm is a monotonic transformation, this implies $\log(f(X^{(t)})) \leq \log(f(\hat{x}))$. Subtracting $\log(f(X^{(t)}))$ from both sides gives $0 \leq \log(f(\hat{x})) - \log(f(X^{(t)}))$. However, since the step size in the Markov chain simulation should be nonzero, one is added to both sides. Thus,

$$1 \leq \log(f(\hat{x})) - \log(f(X^{(t)})) + 1. \tag{4.13}$$

Let $h(\hat{x}, X^{(t)}) = \log(f(\hat{x})) - \log(f(X^{(t)})) + 1$, and note that $1 = h(\hat{x}, \hat{x}) \leq h(\hat{x}, x)$ for any x. For log-concave densities, $h(\hat{x}, X^{(t)})$ increases as the value of $\log(f(X^{(t)}))$ gets farther from the log density value at the mode. The use of the difference of the log densities is useful for situations where the use of the density function would lead to numerical instability. For more stable target densities,

a ratio proportional to $\frac{1}{f(X^{(t)})}$, as in the tempered Langevin algorithm of [67], would be effective. A candidate that makes use of the mode would be the ratio $\frac{f(\hat{x})}{f(X^{(t)})}$, which is always bounded below by one. Taking the logarithm of both sides of the inequality $1 \leq \frac{f(\hat{x})}{f(X^{(t)})}$ and adding one results in the same expression as 4.13. In practice, the mode in expression 4.13 can be found using any of a host of numerical optimization procedures, such as the Nelder-Mead method used for the acceptance sampler above.

To make the step size appropriate for a given target density, both the proposal variance and the step size function $h(\hat{x}, X^{(t)})$ must be scaled correctly. In the standard Metropolis adjusted Langevin algorithm, the variance of the proposal has the form ωI, so that it is constant for each component, although this is not necessary, and varying the proposal variance for different components is possible. As noted above, the results obtained from the acceptance sampler show that the different components in the CWD posterior have widely different standard deviations, in some cases an entire order of magnitude. Consequently, the diagonal covariance matrix, Σ, is kept, but the entries are scaled differently for each component. This is a less complicated approach than the construction of the diffusion matrix used for the tempered Langevin algorithm in [67]. The entries of Σ must be chosen to ensure that values are not proposed outside of the parameter space too often. In addition to the proposal covariance, the step size k must be chosen. The tuning parameter k is similar to the discretization size ω in algorithm 4.3, and must be chosen to counteract the effects of $h(\hat{x}, X^{(t)})$, which can take on extremely large values. The parameter k can be chosen by calculating the step size for points throughout the support of the target density,

and choosing the values of k to make h as large or as small as desired. This tuning parameter gives relatively precise control over the minimum step size. Guidelines for choosing the tuning parameter k and the covariance matrix Σ are discussed in section 4.3.3. In addition to the proposal covariance, the step size k must be chosen. The tuning parameter k is similar to the discretization parameter ω in algorithm 4.3, and must be chosen to counteract the effects of $h(\hat{x}, X^{(t)})$

The resulting algorithm is a modification of the tempered Langevin algorithm, and in this thesis it will be called the Modified Adjusted Langevin Algorithm with Tempered Step size (MALTS). The MALTS will be used to simulate the CWD posterior distribution 3.11.

Algorithm 4.4 *(MALTS) Let the current state of the chain be denoted by the m-dimensional vector X_t and the target density by $f(x)$. Let $\hat{x}$ denote the mode of the distribution f. Choose the m entries of the m by m diagonal matrix Σ, and tuning parameter k. Let $h(\hat{x}, y) = kI(\log(f(\hat{x})) - \log(f(y)) + 1)$.*

1. *Set $\mu(X_t) = X_t + s_x\vec{d_x}$, where $s_x = kh(\hat{x}, X_t)$ is the step size at X_t, and $\vec{d_x} = \frac{\nabla \log(f(X_t))}{\|\nabla \log(f(X_t))\|}$, is the unit vector in the direction of the gradient at X_t. If $\nabla \log(f(X_t)) = \vec{0}$, then set $\vec{d_x} = \vec{0}$.*

2. *Generate $Y \sim N(\mu(X_t), \Sigma)$.*

3. *Let $\mu(Y) = Y + s_y\vec{d_y}$ be calculated as is step 1.*

4. *Let $\alpha(X_t, Y) = \frac{f(Y)}{f(X_t)} \frac{\exp(-0.5*(X_t-\mu(Y))^T\Sigma^{-1}(X_t-\mu(Y)))}{\exp(-0.5*(Y-\mu(X_t))^T\Sigma^{-1}(Y-\mu(X_t)))}$.*

5. *With probability $\min(\alpha, 1)$, set $X^{t+1} = Y$. Otherwise, set $X_{t+1} = X_t$.*

Convergence of algorithm 4.4 to stationarity can be established by demonstrating that it is a special case of the Metropolis-Hastings algorithm. The relationship between the target density and the conditional proposal distribution in the algorithm 4.1 must be established. Basic convergence to the target distribution is stated in the following theorem:

Theorem 4.3.1 *Let $\mu(x)$ and Σ be defined as above in algorithm 4.4. Then, if the log of target density f has a connected support contained by the set of real numbers, and continuous first partial derivatives throughout its support, the Markov chain produced by 4.4 will converge to the stationary distribution f.*

PROOF: The continuous first partial derivatives are necessary to ensure the existence of the gradient throughout the support of the target density. The gradient is used in determining the direction of the proposal value. The unnormalized conditional density in algorithm 4.1 is given by:

$$q(x|y) = \exp(-0.5 * (Y - \mu(X_t))^T \Sigma^{-1} (Y - \mu(X_t))) \tag{4.14}$$

This is the kernel of an m-dimensional multivariate normal distribution, with support $\mathbb{R}^m$.

Recall that a set A is *connected* if every two points $z_1, z_2 \in A$ can be connected by a piecewise smooth curve entirely contained within A. The connected support of the target density is not absolutely necessary, but this condition avoids many mathematical and practical difficulties, so it is assumed here [64]. Let $supp(f)$ denote the support of the target density f. If $supp(f) \in \mathbb{R}^m$, then convergence of the Markov chain produced by algorithm 4.4 to the correct stationary distribution is ensured by theorem 4.1.1. $\square$

If the target density is a mixture of both discrete and continuous components, then 4.4 could be only be applied to the continuous components, since the gradient would not exist for the discrete components. Theorem 4.3.1 ensures the basic convergence to stationarity for algorithm 4.4. For a given target density, however, actual convergence rates will depend on the values of the elements of Σ and the specific characteristics of the target density. Since the algorithm suppresses the random walk behaviour of the chain by proposing, on average, in an uphill direction, performance will in general be better than the standard random walk as the number of dimensions increases [28, 64].

The MALTS algorithm, like the MALA, uses the gradient information to pick a direction for the proposed value. These algorithms are more likely to accept proposed values than the standard random walk, because the standard random walk generates a candidate state by adding symmetrically distributed noise about the current state, which is tantamount to searching for high probability regions at random. On the other hand, Langevin type algorithms modify the search by biasing the proposal distribution in favor of candidate states that lie in directions of higher probability. This difference is clear demonstrated by a simple univariate example. Figure 4.6 shows a normal distribution with the candidate state labeled X. A standard random walk is equally likely to generate a proposal value to the right as it is to the left of the current value. However, the values slightly to the left are more likely to be accepted. This effect becomes more noticeable as the dimension of the target density increases, because the probability of the standard random walk choosing a direction with higher

density values diminishes as a result of the "curse of dimensionality".

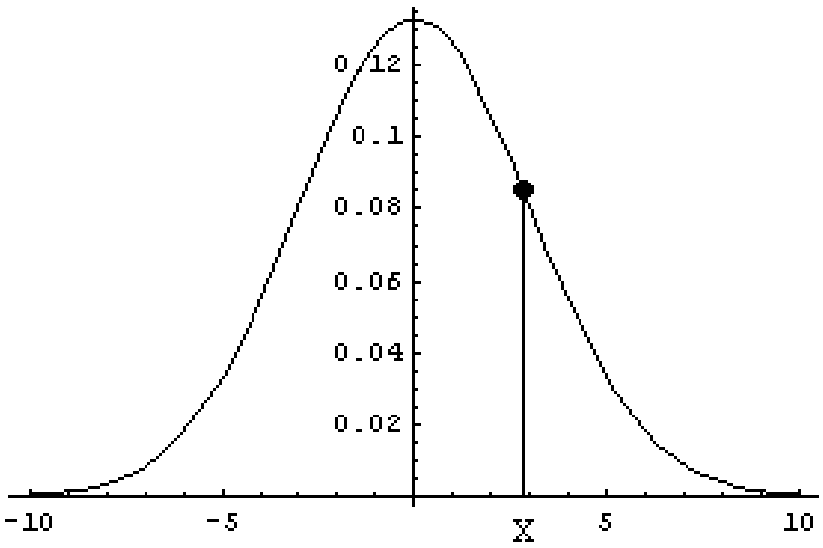

Figure 4.6: Univariate normal density with mean zero, variance three. A current state is labeled X. The higher probability area is to the left of the current state, lower probability areas are to the right.

Note that if a target density is flat, as with a uniform distribution, then the gradient vector will be zero. In such a case, the proposal direction will be the zero vector, so that the proposal mechanism in algorithm 4.4 reduces to a standard random walk. With a locally flat density the gradient contains no information about regions of higher probability, but the algorithm will still function.

Given the similarities with the tempered Langevin diffusion [67], the convergence of the chain produced by this algorithm may be better than the unheated Langevin algorithm 4.3. An empirical study of the algorithm is performed in section 4.3.4. Further results on the theoretical convergence properties for algorithm 4.4 are necessary.

MALTS is essentially a random walk with a "smart" proposal step, which makes use of the gradient information while removing potential instabilities in the gradient. Additionally, because of the properties of $h(\hat{x}, X_t)$, the chain will take larger steps in areas of low probability, and smaller steps in areas of high probability. However, the algorithm requires at least some knowledge of the mode of the target density, which will often require the use of computationally intensive optimization procedures before the method can be implemented. Finding the mode of the target distribution can become quite difficult as the number of dimensions increases. Additionally the algorithm can have a comparatively large number of parameters– the m entries of the diagonal matrix Σ and the constant k for an m-dimensional distribution. This added complexity does provide the benefit of giving the algorithm more freedom to explore complicated target densities. It should be noted, however, that like the MALA (4.3), the MALTS algorithm (4.4) can be defined to require a single tuning parameter, as Σ can be chosen to be identity matrix.

The complexity of the MALTS algorithm is comparable to the MALA. If the gradient is approximated numerically, then for each iteration, both algorithms require $2m + 1$ evaluations of the density function for an m-dimensional target distribution. By contrast, the random walk algorithm requires only one density evaluation per iteration. Each iteration of the standard random walk algorithm will therefore require less time than either the MALTS or MALA. The advantage of algorithm 4.4 or the MALA over a standard random walk is that the standard random walk can require an extremely large number of iterations to converge to stationarity [58, 9, 64].

4.3.3 Tuning the MALTS Algorithm

The considerations in choosing the tuning parameters for the MALTS algorithm are the same as those for other methods. For bounded densities, the tuning parameter k in algorithm 4.4 must be chosen to prevent the chain from frequently proposing values that lay outside the parameter space. For unbounded densities, k and Σ must be chosen to ensure sufficient exploration of the parameter space. The dynamic step size h must be also be considered, and the tuning parameter k must balance the behaviour of the step size equation h.

The variances of the prior distributions are generally easy to calculate and make convenient choices for the entries of the proposal variance matrix Σ. Another natural choice for the elements of Σ are the preliminary estimates of the marginal posterior variances. Since the MALTS algorithm requires the calculation or approximation of the mode of the target density, this information can be conveniently used in the choice of the tuning parameters. The inverse of the Hessian matrix of the log-posterior density, evaluated at the mode, provides an analytical approximation of the posterior covariance matrix. The values of the approximate covariance matrix can then be used to choose the values of the matrix Σ. If desired, Σ can be diagonal.

Using either the prior marginal variances or the posterior marginal variances alone may lead to step sizes that are too large or too small. Consequently, the matrix Σ may need to be increased or decreased. This is especially important for bounded densities, since an efficient Markov chain sampler should most often propose values that lie within the support of the target density. If $h(\hat{x}, x)$ is easy to maximize on the support of the target density, then an upper bound for the

step size can be established, to aid in choosing the tuning parameter k. A simple alternative is to calculate $h(\hat{x}, x)$ for several points in low probability areas on the support of the target density, and then scale the values of the step matrix to achieve a particular step size.

A common consideration when choosing tuning values for Markov chain simulations is the *acceptance rate*– the proportion of proposed moves that are accepted. A low acceptance rate can indicate that the step size is too large, since for many densities, a large move is extremely unlikely to be accepted [25, 64]. A smaller step size typically leads to a larger acceptance rate, since proposed moves that are extremely close to the current state are very likely to be accepted. As an example, as the step size ω in the Langevin algorithm 4.6 approaches zero, the discrete Langevin random walk approaches the continuous time Langevin diffusion, in which moves are always accepted. While a high acceptance rate indicates that the chain is making more moves, the small step size has the drawback of causing the chain to move about the parameter space very slowly.

There is some theoretical justification for an acceptance rate in the range $[0.15, 0.5]$. These values are based on limiting acceptance rates for high dimensional densities, and will be discussed in section 4.3.4. If possible, the value of the tuning parameter k should be adjusted to achieve an acceptance rate in this range. Theoretical studies of the Metropolis adjusted Langevin algorithm have indicated that the optimal acceptance rate for the MALA is near 0.5 [64, 28]. Thus the optimal acceptance rate for the MALTS algorithm 4.4, may be near 0.5 as well.

4.3.4 Test Scenarios with Known Densities

Fundamentally, a Markov chain algorithm needs to accurately simulate the correct target density, or at least a good approximation to it. This can be checked both theoretically, as in theorem 4.3.1, and empirically. Once there is confidence that a method will produce the target density, there are questions of performance. The MALTS algorithm will be tested on a simple truncated bivariate normal distribution, to empirically demonstrate the results of theorem 4.3.1 on a simple target density. An experiment will then be conducted to compare the performance of MALTS, MALA, and the standard random walk.

A simple test of accuracy can be performed by using the MALTS algorithm to sample a density with known parameters. The normal distribution truncated to $[0, 1]$, with mean $(0.5, 0.5)$ and covariance chosen so that the majority of the density is concentrated around the mode of the distribution, provides an opportunity to test algorithm 4.4 on a bounded density with known parameters. The concentration of the density 4.7 around the mode and the bounded parameter space are features similar to those possessed by the posterior distribution 3.11 in the Chronic Wasting Disease model. Figure 4.7 shows the graph of a truncated normal distribution, centered at $(0.5, 0.5)^T$, with covariance matrix $0.001 * I$.

The truncated normal distribution shown in figure 4.7 is simulated using algorithm 4.4. Four chains are run, with four initial values chosen to be dispersed about the parameter space. These starting points are: $(0, 0)$, $(0.7, 0.1)$, $(0.1, 0.7)$, and $(0.9, 0.9)$. The proposal variance for the MALTS algorithm was a diagonal matrix with the step size 1×10^{-10} for each variable, and the tuning parameter

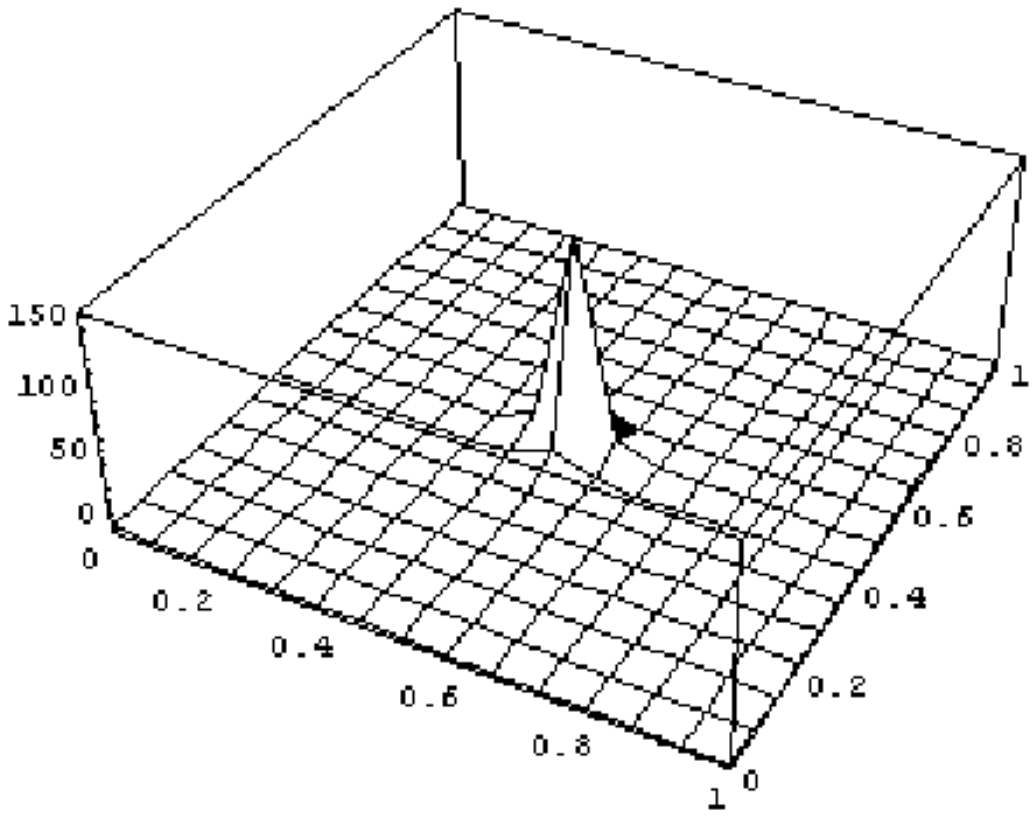

Figure 4.7: The bivariate normal distribution truncated on $[0, 1]$ with mean $(0.5, 0.5)^T$ and covariance $0.001 * I$.

was also $k = 1 \times 10^{-10}$. The results of the simulation are shown in figure 4.8. The majority of the points are concentrated around the mode of the target density, providing graphical evidence that the method has sampled the correct density.

The *potential scale reduction* is a comparison of the variance between runs and within runs, denoted by $\sqrt{\hat{R}}$. This diagnostic is used to assess the convergence of a Markov chain simulation to stationarity [24]. This measure should decrease to one as the number of iterations increases for all parameters of interest.The potential scale reduction is calculated using the fours runs of the MALTS algorithm, and the plot of the statistic $\sqrt{\hat{R}}$ versus iterations is shown in figure 4.9. The graph shows the potential scale reduction decreasing to one as the number of iterations increases, giving evidence that the simulated Markov chain is reaching approximate stationarity. A common rule of thumb for the potential scale reduction is that it should be less than 1.2 for each parameter, which occurs within 2000 iterations, indicating that this is an acceptable burn-in period. Additionally, the posterior means averaged over all four runs and adjusted for a 2000 iteration burn-in period were 0.5010 for θ_1, and 0.5001 for θ_2. The 95% Bayesian credible intervals averaged over all four simulations are $(0.4925, 0.5110)$ and $(0.4840, 0.5010)$, indicating that algorithm 4.4 was able to recover the correct parameters. Having determined the basic accuracy of algorithm 4.4, a comparison of the relative performance of the MALTS algorithm versus the standard random walk and the MALA is needed.

Having determined the basic accuracy of the MALTS algorithm, the next step is to compare the performance of the algorithm with other methods, in particular the standard random walk and the Metropolis adjusted Langevin al-

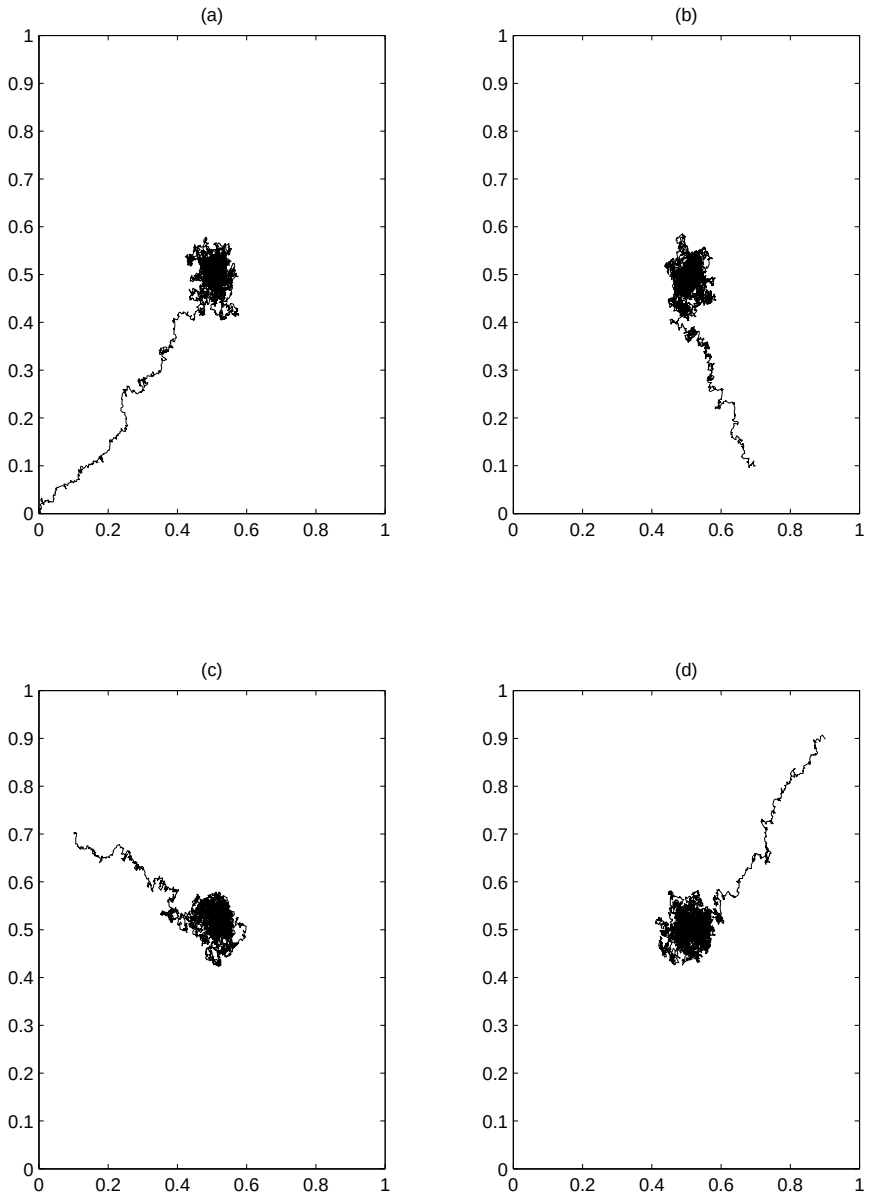

Figure 4.8: Pairwise plots of the mean parameters θ_1 and θ_2 for the truncated normal distribution with true mean $(0.5, 0.5)^T$ and true covariance $0.001 * I$. (a) shows $8,000$ iterations for the initial value $(0, 0)^T$. (b) shows $8,000$ iterations for the starting value $(0, 0.001)^T$. (c) shows $8,000$ iterations for the starting value $(0.3, 0.4)^T$. (d) shows $8,000$ iterations for the starting value $(0.1, 0.8)^T$.

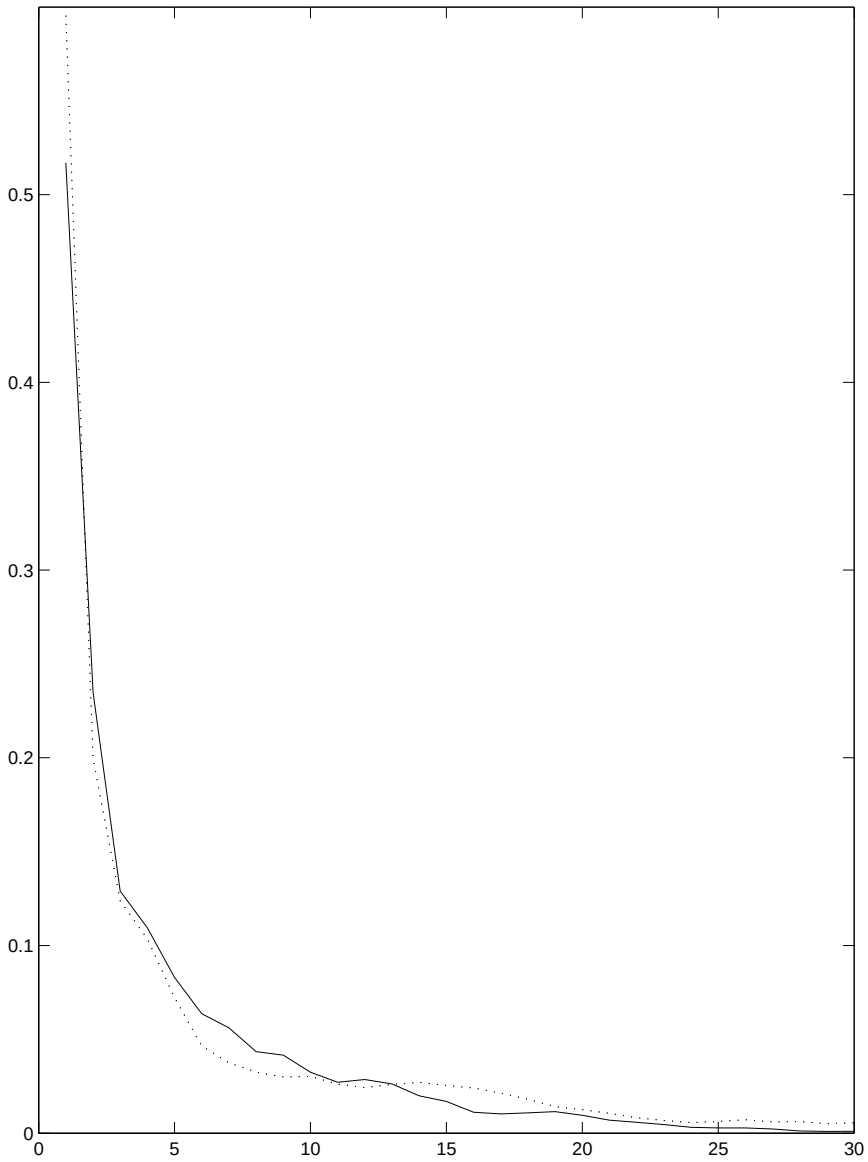

Figure 4.9: Plots of the log potential scale reductions by thousand iterations for the mean parameters θ_1 and θ_2 for the MALTS simulation of the truncated normal distribution. The solid line shows the results for θ_1, the dotted line shows the results for θ_2.

gorithm. There are a number of criteria for assessing the performance of a Markov chain simulation, including run time, convergence to stationarity, convergence to independent sampling, and mixing speed. The relative importance of the different criteria is not widely agreed upon in the literature. For this thesis, the most important quality is the number of iterations that the algorithm requires to converge to stationarity. Secondly is how well the algorithm mixes. Thirdly is the amount of time required to run the algorithm.

Theoretical results on optimal algorithm performance thus far have centered on two questions: what is the optimal step size, and how well does the algorithm perform at this step size. Results on these two questions are limited for more complex algorithms, and rely on simplifying assumptions such as independence of components or identically distributed components for an m-dimensional target density. Theoretical results examine the limiting behaviour as m diverges to infinity.

Under the assumption of independent components, the step size of the standard random walk scales with the dimension as m^{-1}, and leads to an limiting acceptance rate of 23% as $m \to \infty$ [64, 9, 28]. By contrast, the Langevin algorithm has a step size that scales as $m^{-1/3}$, and leads to a limiting acceptance rate of 57% [9, 65, 66, 28]. The higher acceptance rate for a larger step size indicates that the Langevin algorithm is moving about the parameter space more than the standard random walk.

There is no theory for more complicated scenarios, such as sampling multivariate distributions with correlated components. In such cases empirical methods are used to compare algorithms [28]. One common approach is to fix the

step size and examine the acceptance rates for different algorithms. For a given step size, a higher acceptance rate is an indication that an algorithm is making more moves about the parameter space, and consequently is mixing better than an algorithm with a lower acceptance rate. The acceptance rate is only a rough measure of how well a chain is mixing, but large differences in the acceptance rate would indicate differences in the mixing speed.

To compare the performance of the MALTS, MALA, and standard random walk, an experiment was performed to compare both the mixing speed and the convergence to stationarity of each algorithm. This experiment examined a bounded and an unbounded group of distributions. The bounded group of distributions were multivariate normal distributions truncated to the unit cube. The unbounded group of distributions were highly correlated multivariate normal distributions. These distributions form a reasonable test bed, as they have been used by many authors as prototypical target distributions (see, for example, [28, 25]). For each distribution within a group, $80,000$ iterations of a Markov chain simulation were run from five randomly chosen initial values, using the standard random walk, the MALA, and the MALTS algorithm. Each method used multivariate updating, rather than univariate, so that at each iteration, all of the components were updated. The potential scale reduction was calculated for the first and second components and for the variance of the first and second components using the five chains for each simulation method to assess the convergence to stationarity, both in terms of the number of iterations and the amount of time. Additionally, the acceptance rate was calculated to determine how well the chains produced by each simulation method were mixing.

For the truncated normal distributions, the components were independent, each with mode 0.89. Truncated normal distributions provide a classic case of a bounded density. The variance of the first component was set at 1×10^{-3}, and the variance of the remaining components was set at 1×10^{-5}. Four sets of simulations of increasing dimension were run within this group. In increasing order, the dimensions were 2, 6, 12, and 36, and the step size parameter was fixed at 1×10^{-5} for each method. The acceptance rates for each simulation are given in appendix A. The results for the simulation show the standard random walk and MALTS algorithms produced comparable acceptance rates for each dimension. This is evidence that both methods are mixing reasonably well. Both the standard random walk and the MALTS algorithm have higher acceptance rates than the MALA for all dimensions (nearly 5% higher), indicating that these methods are accepting more proposed moves, and consequently exploring more of the parameter space. While the acceptance rate for the MALTS algorithm is slightly higher the acceptance rate for the standard random walk for each dimension, the differences are small, and may not be practically significant.

Figures 4.10 and 4.12 show the plot of the log potential scale reduction versus iteration for each method for the first component, with variance 1×10^{-3}, and the second component, with variance 1×10^{-5}, for the two dimensional and 36 dimensional simulations. For both components, first order convergence to stationarity for the MALA and MALTS algorithms occurs in fewer iterations than the standard random walk. The MALA converges in fewer iterations than algorithm 4.4 for the first component, but the two algorithms performance is nearly identically for the second component. The results for second order

convergence, shown in figure 4.11 and 4.13, are similar. For the truncated normal distribution, algorithm 4.4 thus appears to provide a middle ground, with convergence to stationarity comparable to the Langevin algorithm 4.3 in fewer iterations than the standard random walk, and a mixing speed better than the Langevin algorithm and comparable to the standard random walk.

An important consideration when comparing algorithms is the amount of time it takes to run the algorithm. For the two dimensional truncated normal simulation, the random walk simulation requires an average of 5.0×10^{-4} seconds for each iteration, compared to 23.0×10^{-4} seconds for the Langevin and MALTS algorithms. Figure 4.14 shows the plot of the log potential scale reduction versus seconds for each method for the two dimensional truncated normal distribution. The longer time per iteration results for the MALTS algorithm results in a longer amount of time to reach convergence for the first component. The standard random walk and MALA reach stationarity in a much shorter amount of time. For the second component, however, the MALTS and MALA clearly reach stationarity faster than the standard random walk. Figure 4.15 shows the results for second order convergence, which are similar. The results are almost the same for the 36 dimensional simulation, shown in figures 4.16 and 4.17. The longer per iteration time of 24.0×10^{-4} for the MALTS and MALA clearly has a significant impact, but only on the order of seconds. The fewer iterations required by the derivative using methods (MALA and MALTS) will result in fewer wasted iterations, requiring less memory resources for a usable

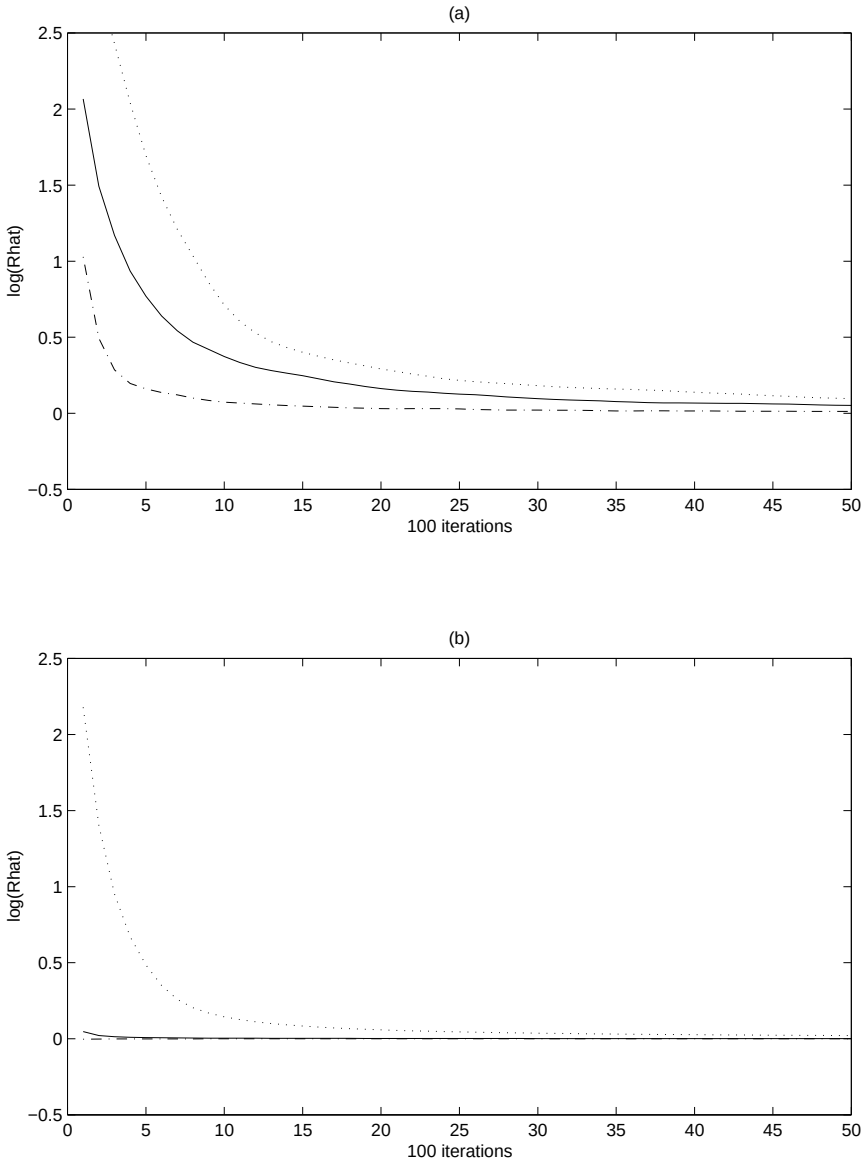

Figure 4.10: Plots of the log potential scale reductions by hundred iterations for the parameters θ_1 and θ_2 for the the simulation of the 2 dimensional truncated normal distribution. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

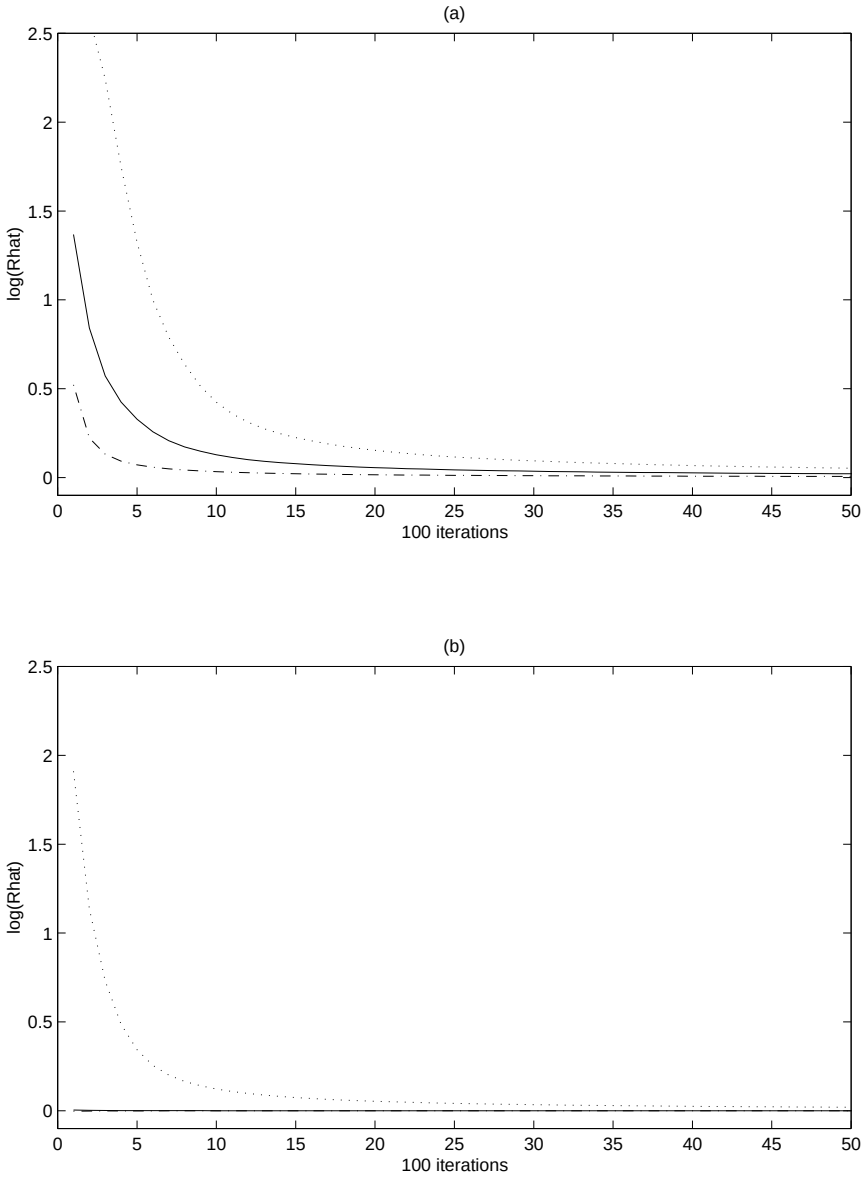

Figure 4.11: Plots of the log potential scale reductions by hundred iterations for the variance of the parameters θ_1 and θ_2 for the the simulation of the 2 dimensional truncated normal distribution. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

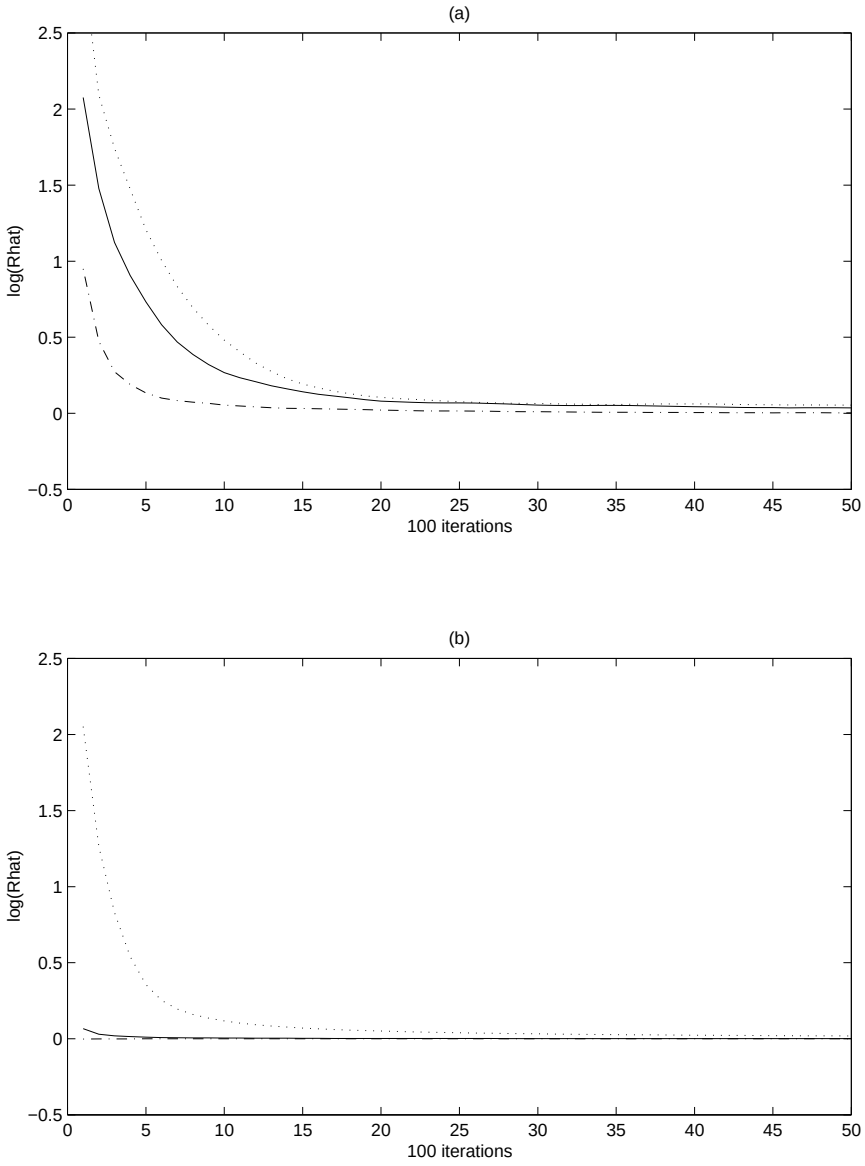

Figure 4.12: Plots of the log potential scale reductions by hundred iterations for the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional truncated normal distribution. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

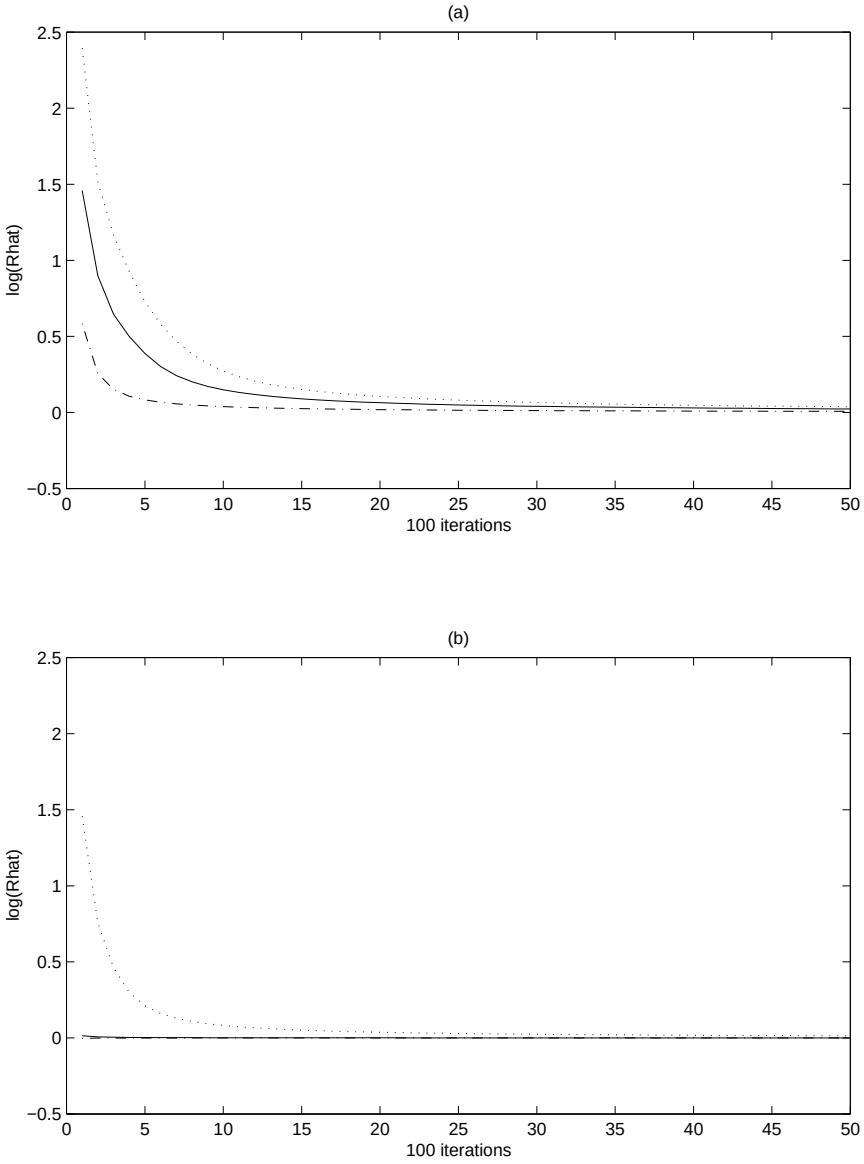

Figure 4.13: Plots of the log potential scale reductions by hundred iterations for the variance of the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional truncated normal distribution. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

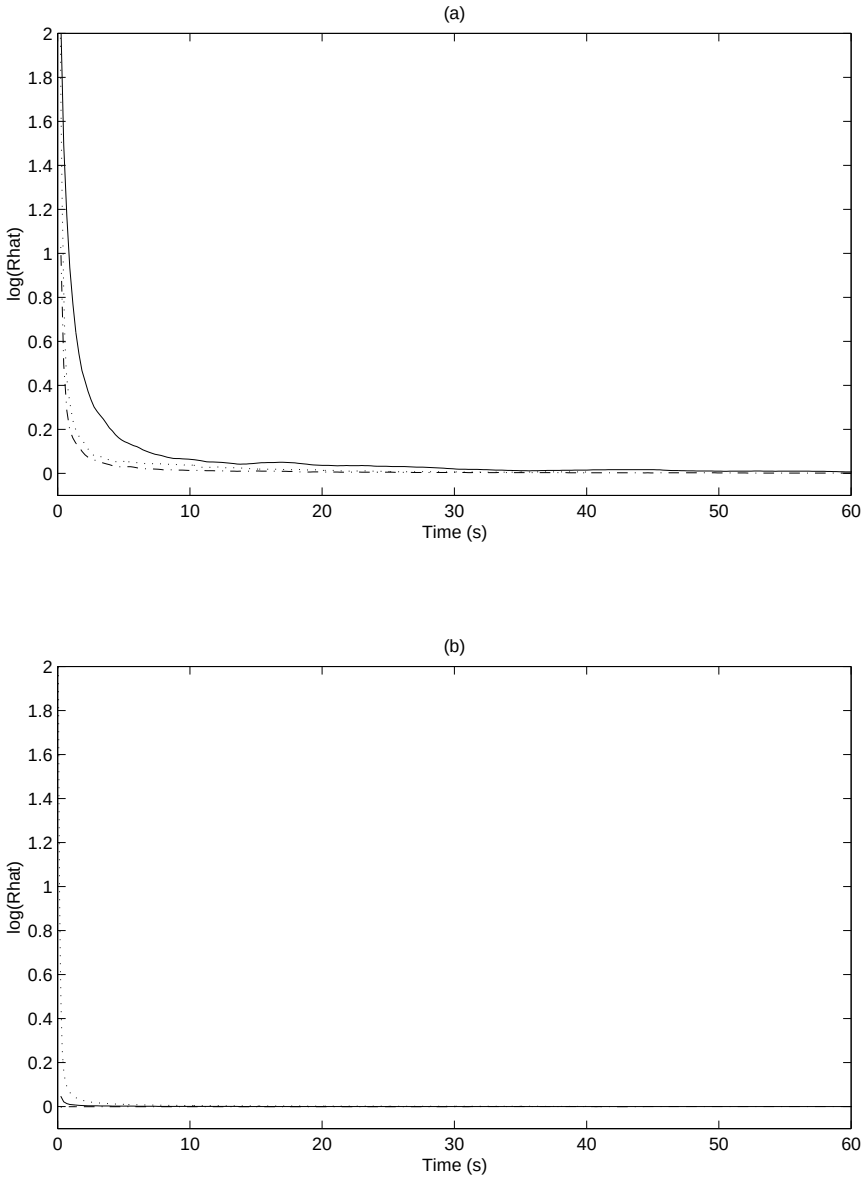

Figure 4.14: Plots of the log potential scale reductions by time for the parameters θ_1 and θ_2 for the the simulation of the 2 dimensional truncated normal distribution. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

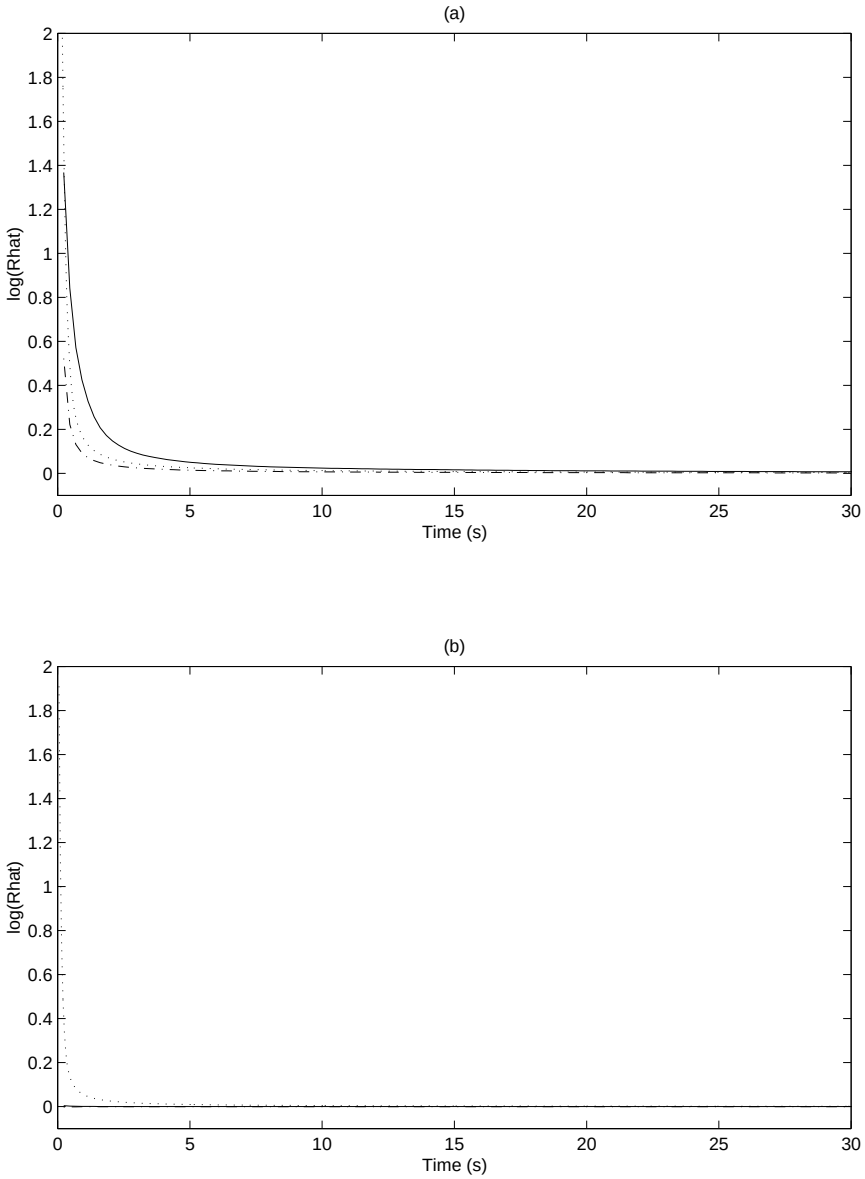

Figure 4.15: Plots of the log potential scale reductions by time for the variance of the parameters θ_1 and θ_2 for the the simulation of the 2 dimensional truncated normal distribution. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

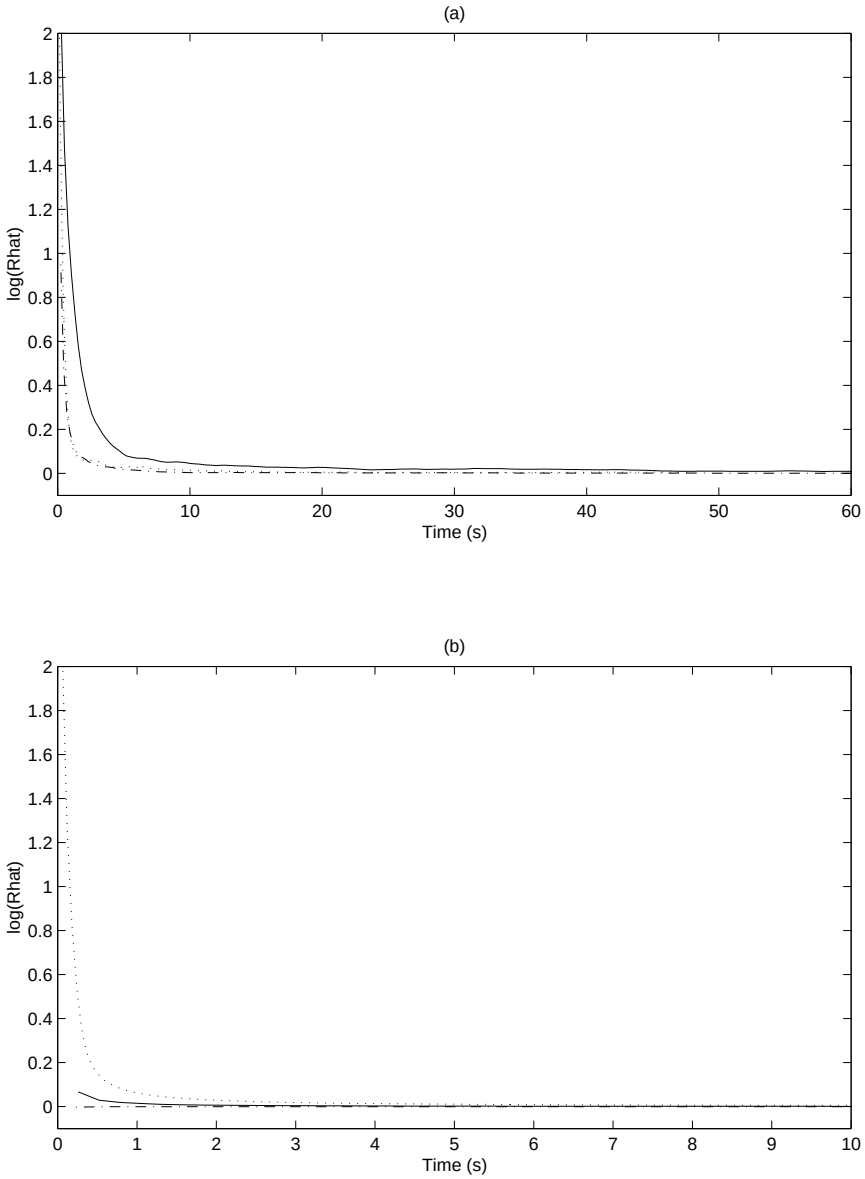

Figure 4.16: Plots of the log potential scale reductions by time for the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional truncated normal distribution. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

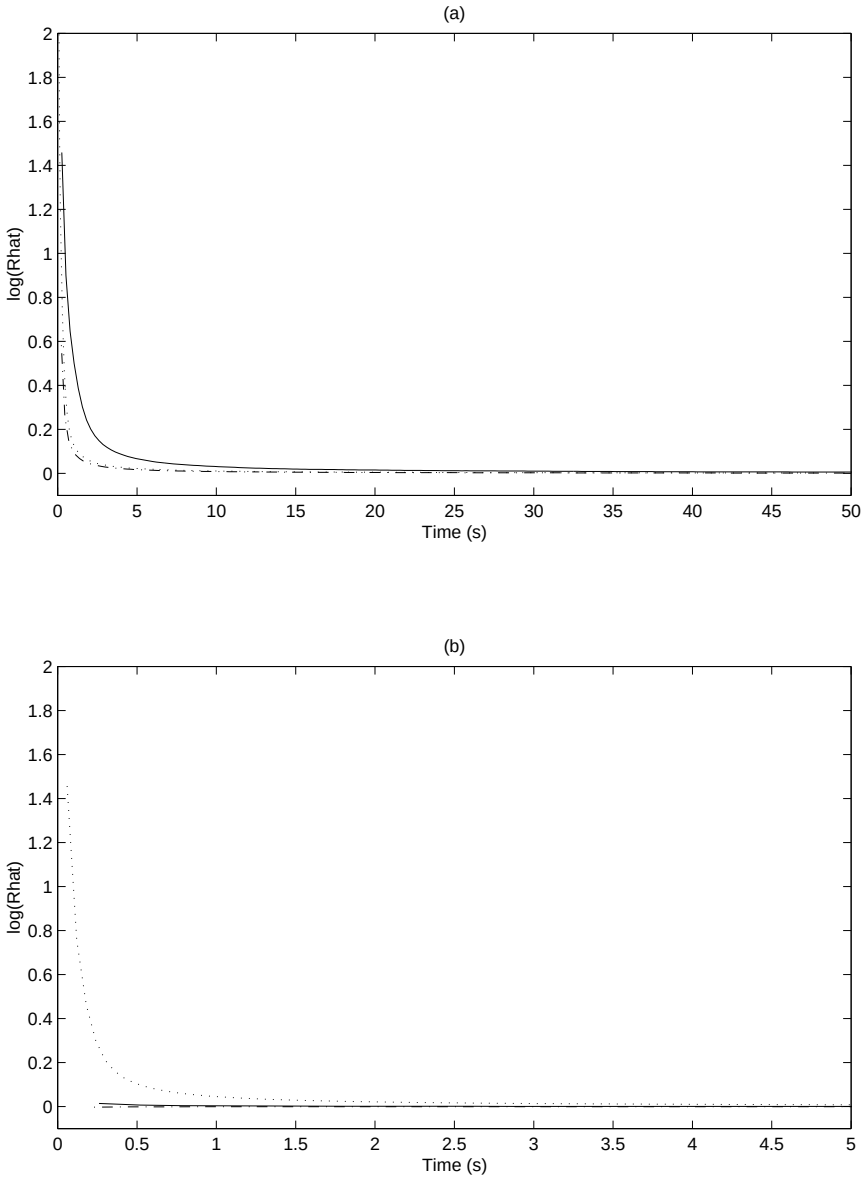

Figure 4.17: Plots of the log potential scale reductions by time for the variance of the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional truncated normal distribution. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

sample.

For the multivariate normal distribution, the components were chosen to be highly correlated. Each distribution had a mean of zero and covariance $(1-\rho)I_k + \rho J_k$, where the matrix I_k is the k-dimensional identity, J_k is a matrix of all ones, and ρ is the correlation between parameters. Three levels of correlation were examined, $\rho = 0.879$, 0.970, and 0.992. These are the same correlation levels examined in [28]. For each of these correlation values, four sets of simulations were run, with dimensions 2, 6, 12, and 36. Three step size parameters were used for each method, 1×10^{-5}, 1×10^{-4}, and 1×10^{-2}. Additionally, for the correlation level of $\rho = 0.992$ and step size of 1×10^{-5}, simulations were run with dimensions 64, 144, 200, and 300. The acceptance rates for all of the simulations are given in appendix A.

For the step size 1×10^{-5}, correlation $\rho = 0.879$, there is little difference in the acceptance rates, as shown in table A.2. At this correlation level, each of the methods has an acceptance rate above 95% for each dimension. This indicates that, at this level of comparison, all of the methods are mixing at approximately the same speed, and accepting roughly the same number of proposals. However, the graphs of the log potential scale reductions by iteration, shown in figures 4.18, 4.20, and 4.26, show differences in the number of iterations in which the methods converge to stationarity, especially for the first component. For the two dimensional simulation, with $\rho = 0.879$, the MALTS algorithm converges to stationarity in far fewer iterations than the Langevin and standard random walk, as shown in figure 4.18. As the dimension increases to 36 (figure 4.20), the MALTS algorithm still requires fewer iterations to converge than the other

two algorithms. For second order convergence, shown in figures 4.19 and 4.21, the results are similar, although the differences are less noticeable.

For the two dimensional simulation, the standard random walk requires 3.0×10^{-4} seconds per iteration, compared to 8.0×10^{-4} for the MALA and 9.0×10^{-4} for the MALTS algorithm. In spite of the longer time per iteration required by the MALA and MALTS method, the performance of the three algorithms is similar. The plot of the log potential scale reduction versus time is shown in figure 4.22. The graphs for both the first and second parameters are very similar, and show little differences between the algorithms. The graphs for second order convergence, or convergence of the second moments, shown in figure 4.23, show that the standard random walk and MALTS methods take less time to converge than the MALA. Increasing the number of dimensions to 36 produces similar results, although the shorter time per iteration should give the standard random walk a distinct advantage. The standard random walk requires 3.1×10^{-4} seconds per iteration, while a single iteration of the MALA requires 8.2×10^{-4} seconds and the MALTS algorithm requires 9.2×10^{-4} seconds per iteration. Figures 4.24 and 4.25 show the plots of the log potential scale reduction by time, which indicate that the three algorithms converge at approximately the same rate. For the second order convergence shown in figure 4.25, the differences are barely noticeable.

Increasing the level of correlation has a noticeable effect on the differences between the algorithms. This is demonstrated in figures 4.26 and 4.27, which shows that with correlation of 0.992, there is a noticeable effect on the potential scale reduction of the second component for the MALTS and MALA algorithms.

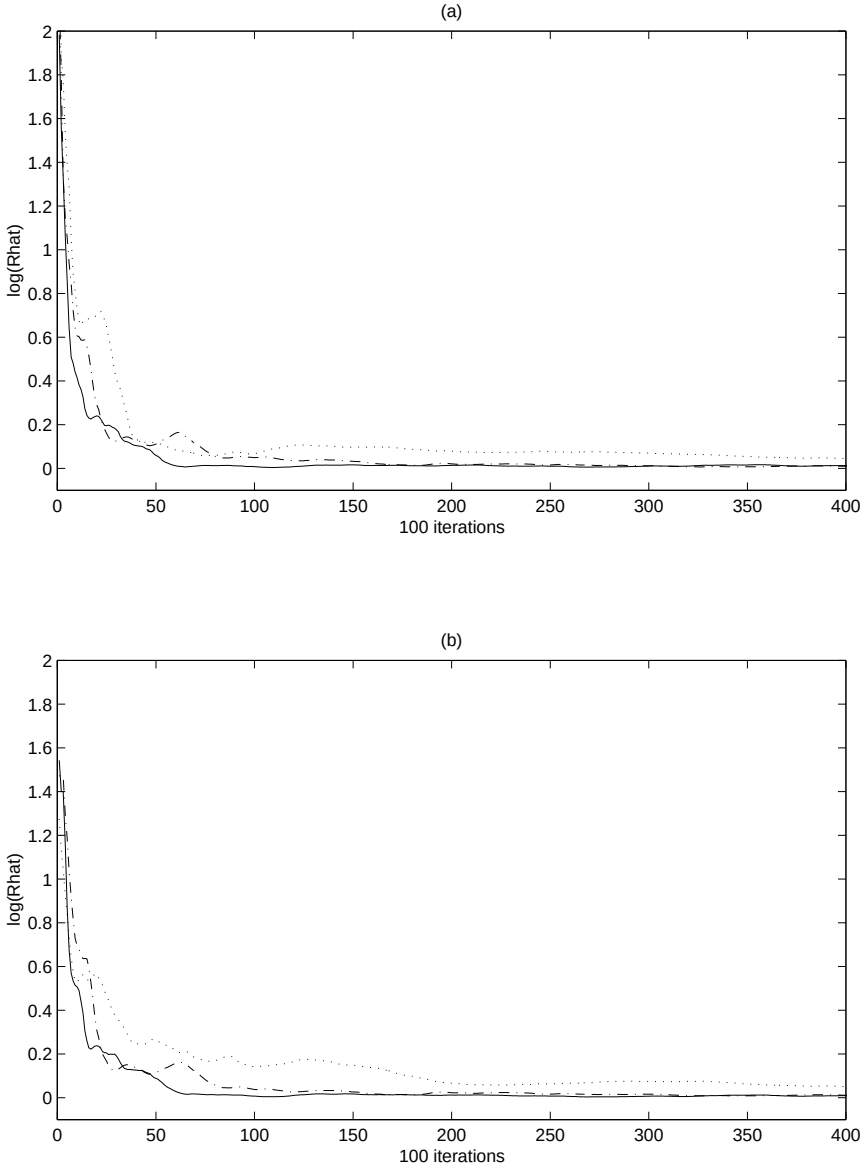

Figure 4.18: Plots of the log potential scale reductions by hundred iterations for the parameters θ_1 and θ_2 for the the simulation of the two dimensional normal distribution, with correlation $\rho = 0.879$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

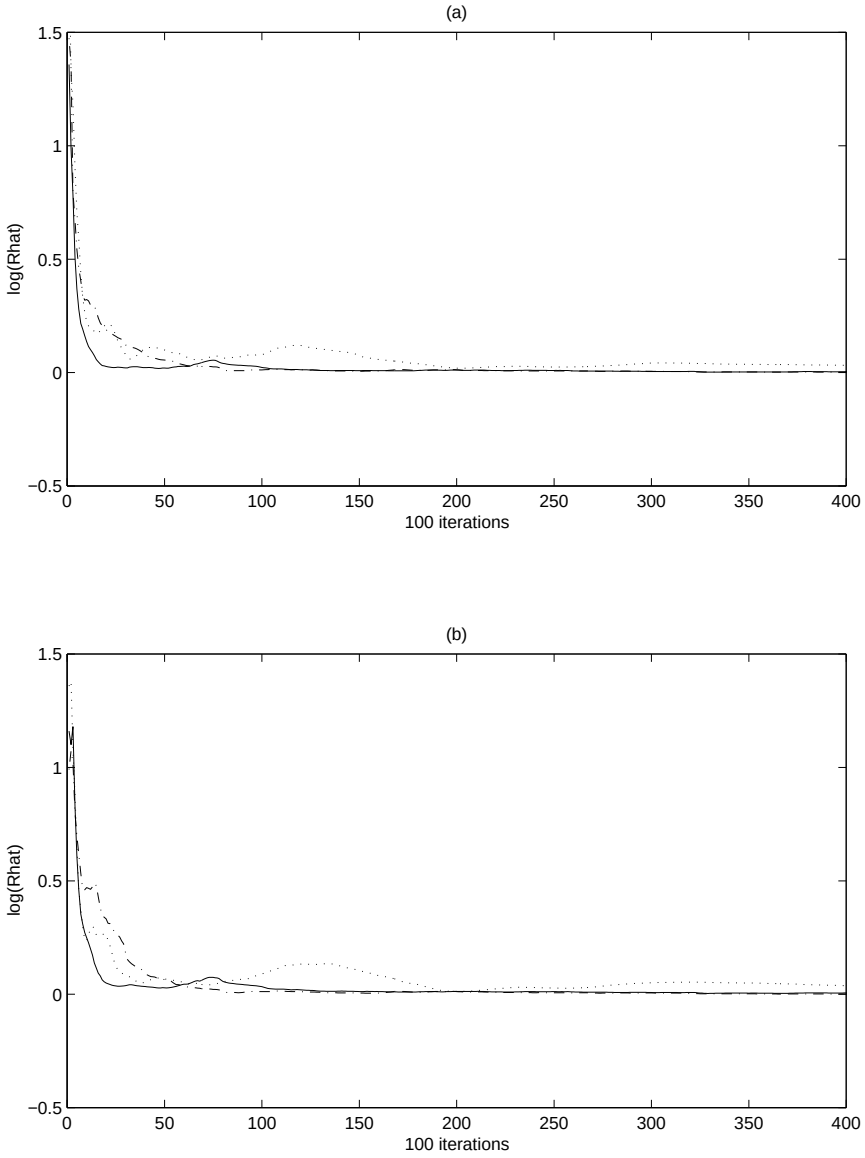

Figure 4.19: Plots of the log potential scale reductions by hundred iterations for the variance of the parameters θ_1 and θ_2 for the the simulation of the two dimensional normal distribution, with correlation $\rho = 0.879$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

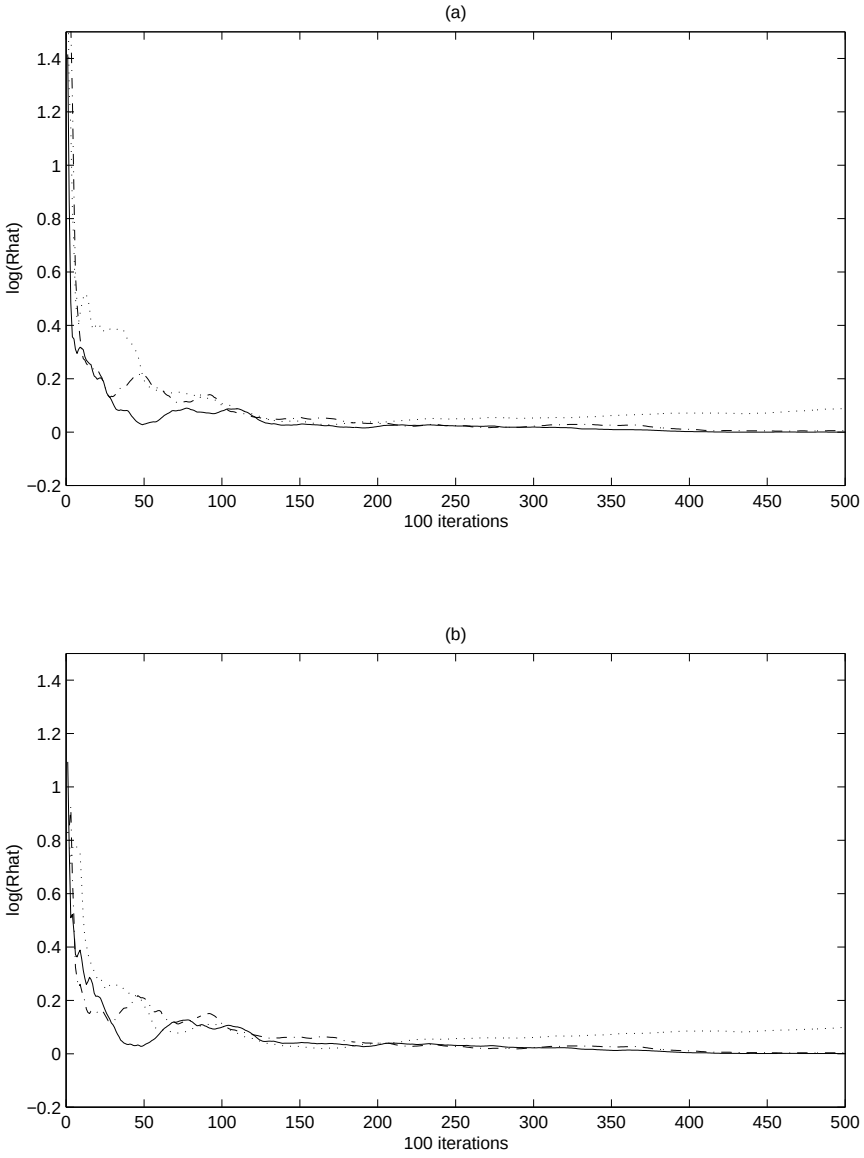

Figure 4.20: Plots of the log potential scale reductions by hundred iterations for the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.879$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

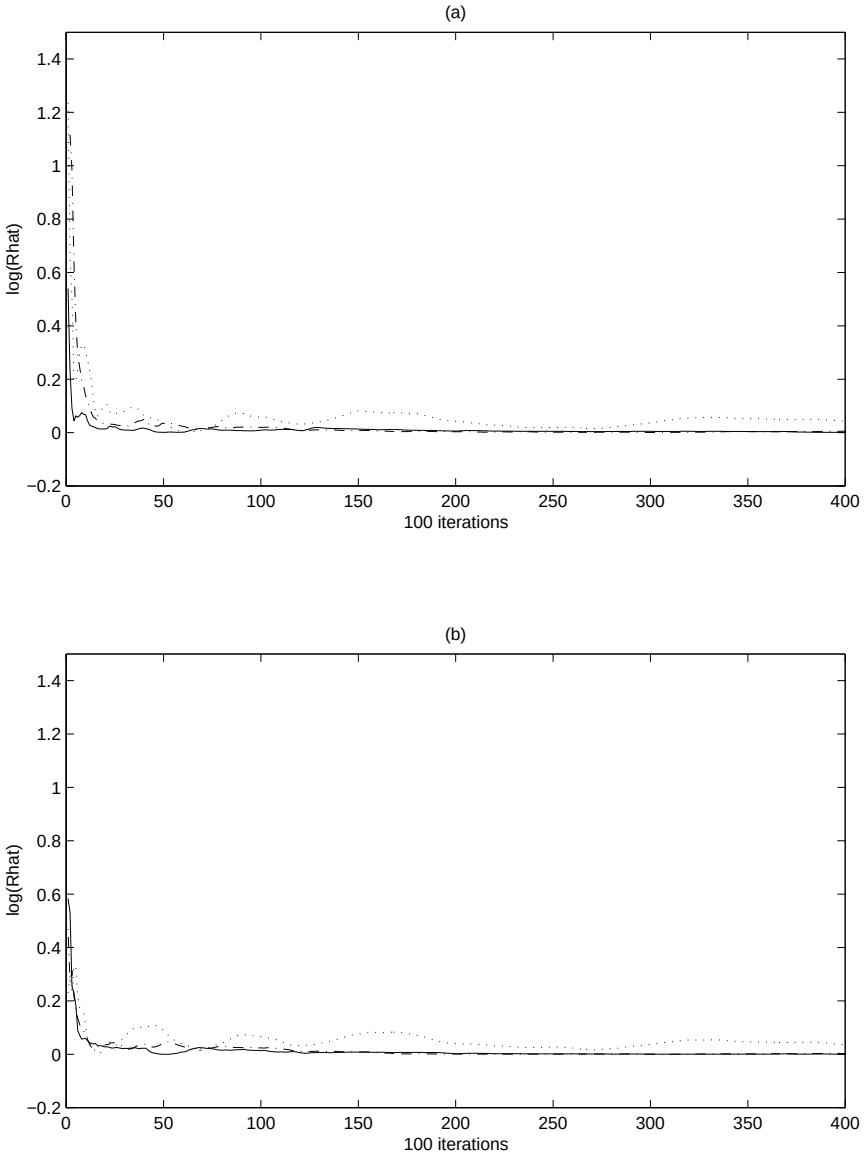

Figure 4.21: Plots of the log potential scale reductions by hundred iterations for the variance of the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.879$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

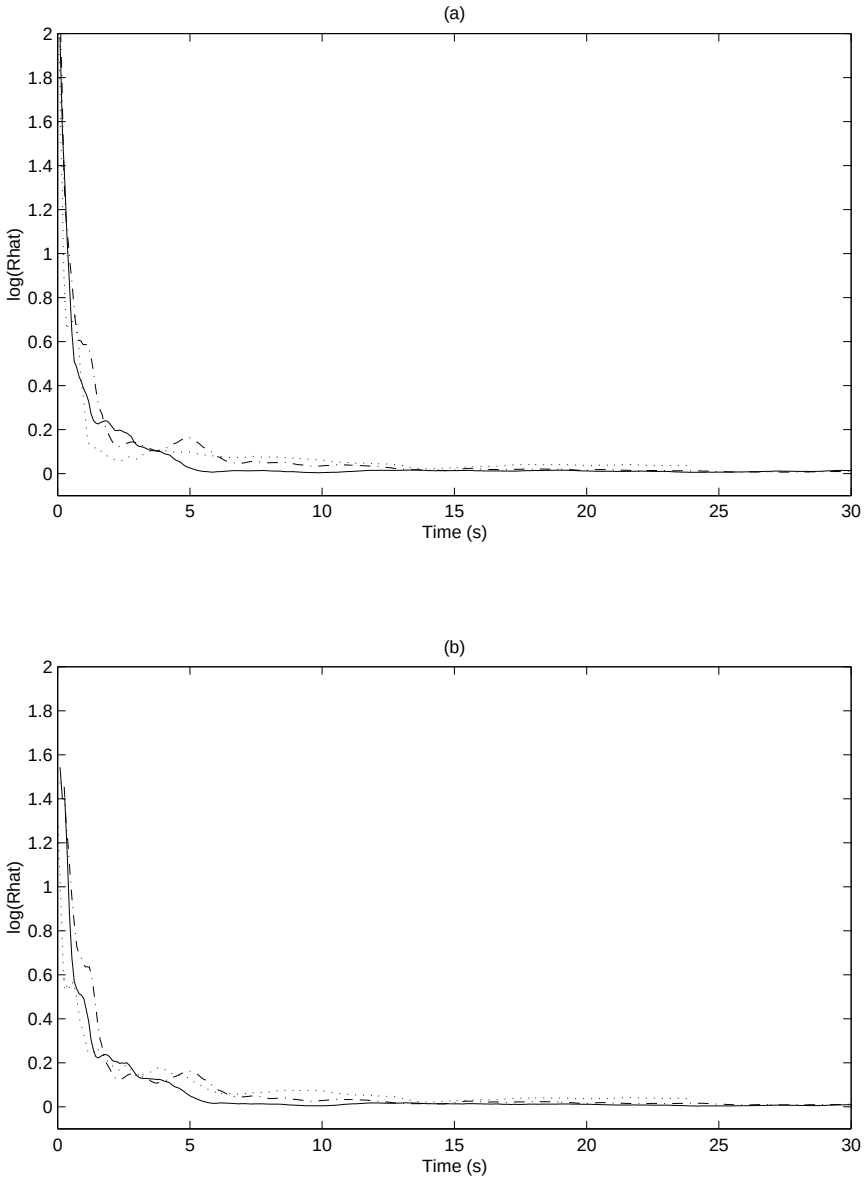

Figure 4.22: Plots of the log potential scale reductions by time for the parameters θ_1 and θ_2 for the the simulation of the two dimensional normal distribution, with correlation $\rho = 0.879$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

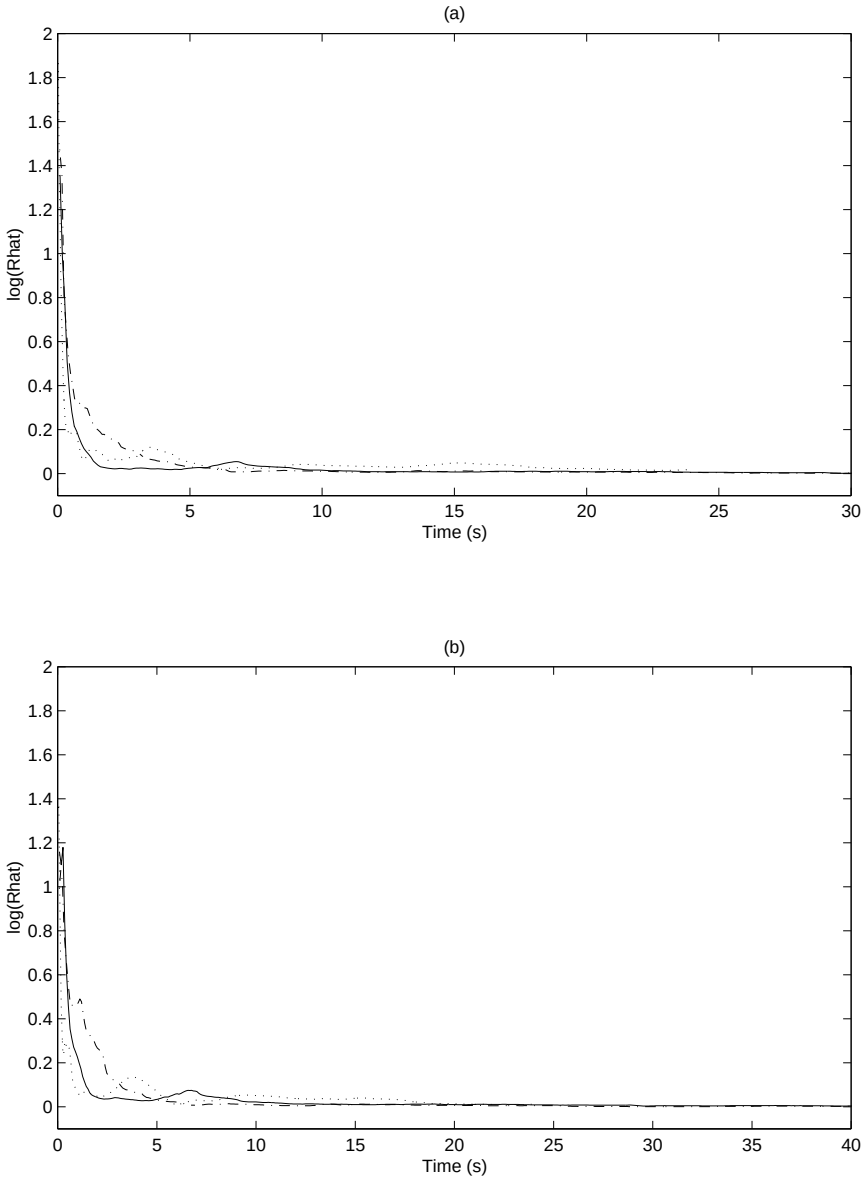

Figure 4.23: Plots of the log potential scale reductions by time for the variance of the parameters θ_1 and θ_2 for the the simulation of the two dimensional normal distribution, with correlation $\rho = 0.879$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

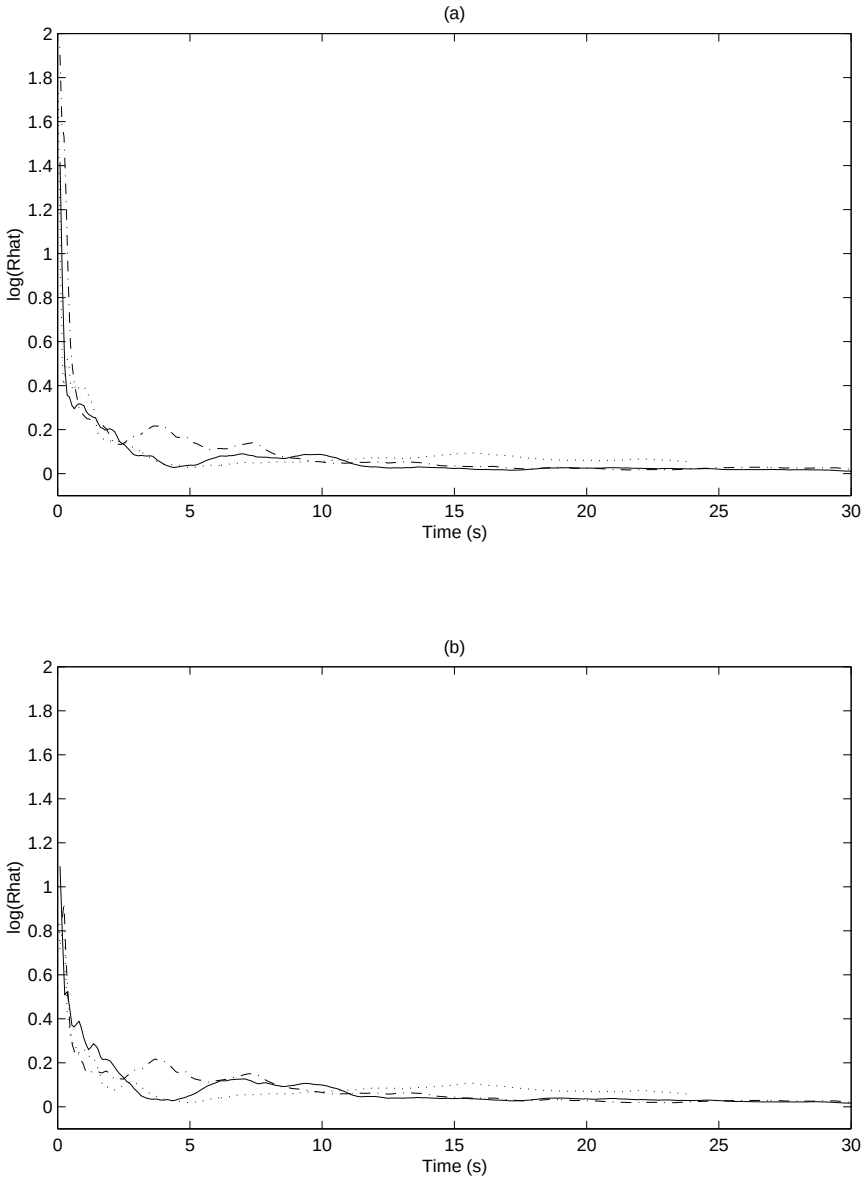

Figure 4.24: Plots of the log potential scale reductions by time for the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.879$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

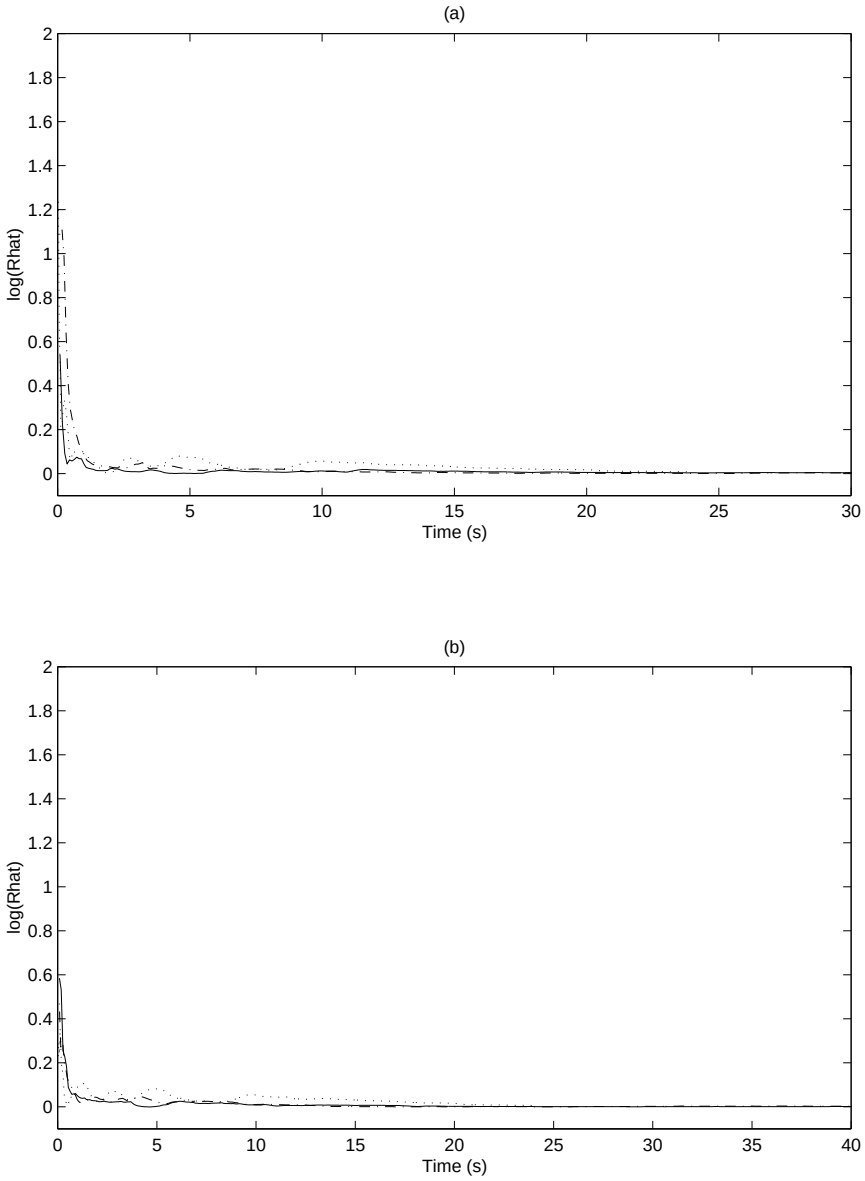

Figure 4.25: Plots of the log potential scale reductions by time for the variance of the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.879$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

The plot of the log potential scale reduction by iteration for the standard random walk does not decay to zero as completely as it does for the other methods. The standard random walk experiences difficulties with second order convergence, although the time advantage of the standard random walk by iteration becomes more apparent at the higher correlation level.

The acceptance rates for each algorithm remain quite high at this step size and correlation level $\rho = 0.992$. As the dimension increases, differences in the acceptance rates become more apparent. Simulations conducted at a correlation of 0.992 for dimensions 64, 144, 200, and 300 begin to highlight the differences between the algorithms. Figure 4.30 shows the graphs of the log potential scale reductions for the 200 dimensional normal distribution simulation. As the number of iterations increases, $\log(\sqrt{\hat{R}})$ decreases rapidly for θ_1 for both the MALA and the MALTS algorithms, compared to the standard random walk. For the parameter θ_2, the log potential scale reduction factor for the MALA decreases more rapidly than the standard random walk, and then appears to experience difficulties, flattening out as the number of iterations increases. The log potential scale reduction factor of θ_2 for the MALTS algorithm also initially decreases faster than the standard random walk, but unlike the MALA, it continues to decrease as the number of iterations increases. By this measure, the MALTS algorithm is clearly reaching stationarity in fewer iterations than the other two methods. The time advantage of the standard random walk disappears at the higher dimension, as demonstrated in figure 4.31.

For the higher dimensional distributions, there is a difference in the acceptance rates as well. As the number of dimensions is increased from 36 to 64,

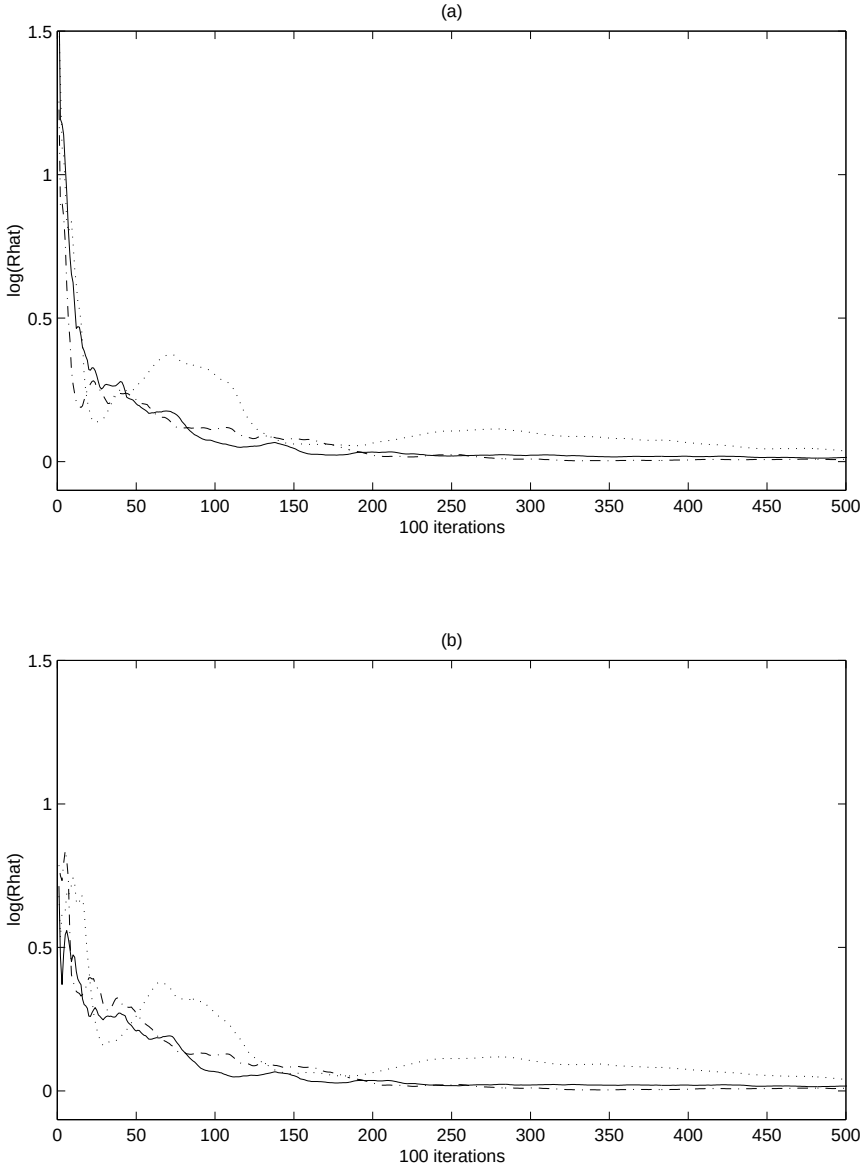

Figure 4.26: Plots of the log potential scale reductions by hundred iterations for the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.992$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

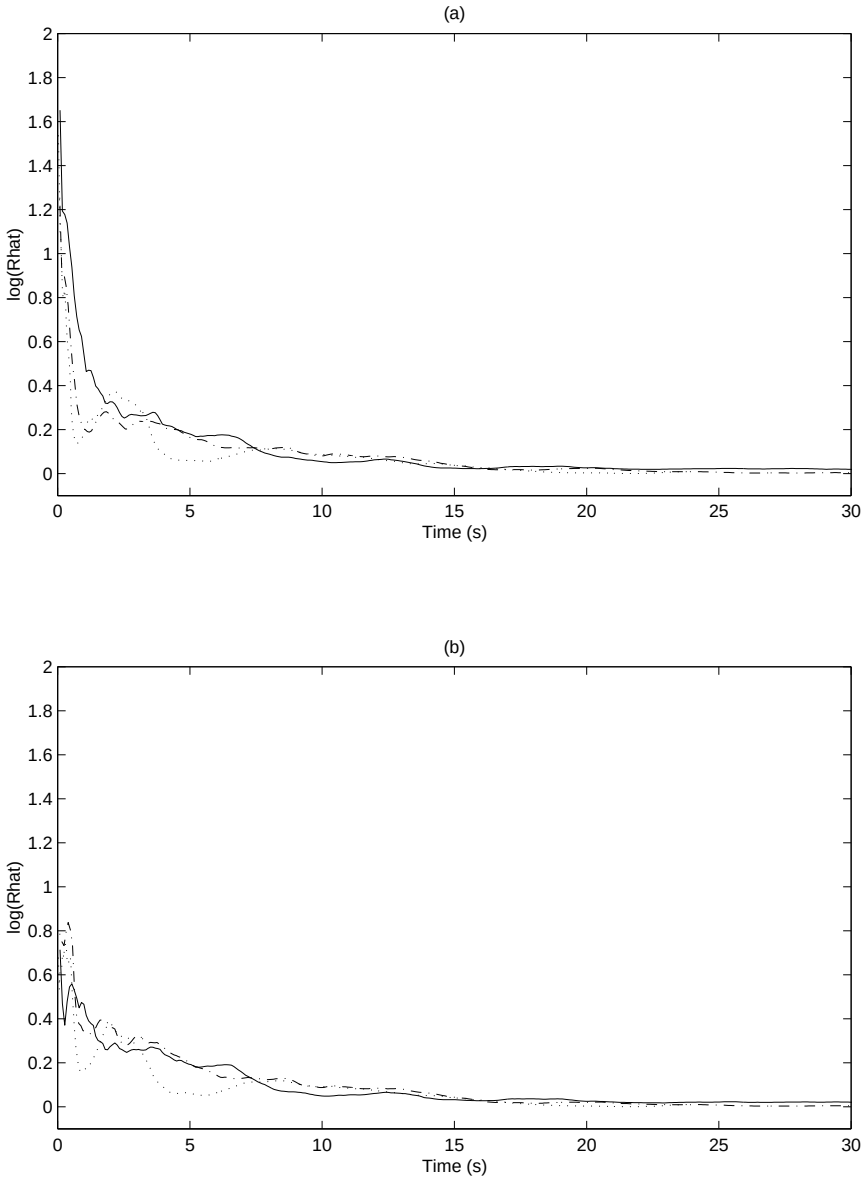

Figure 4.27: Plots of the log potential scale reductions by time for the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.992$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

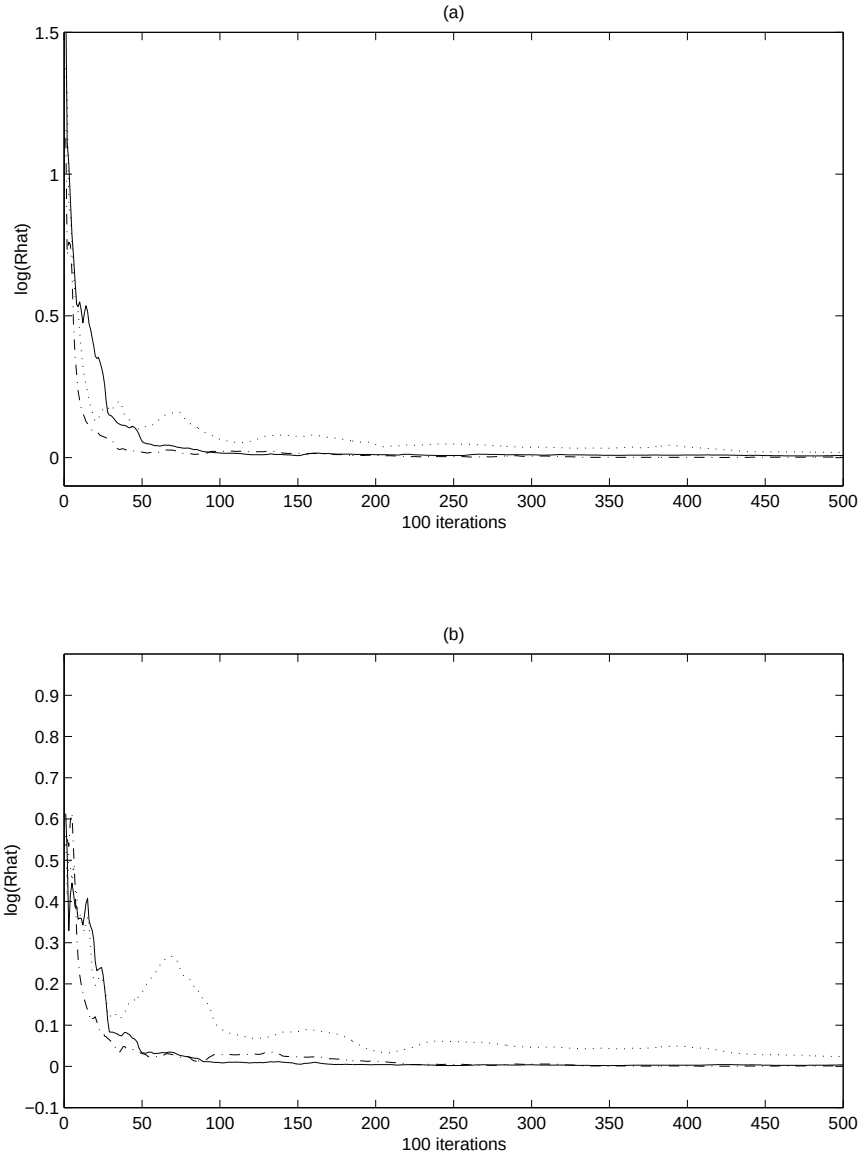

Figure 4.28: Plots of the log potential scale reductions by hundred iterations for the variance of the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.992$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

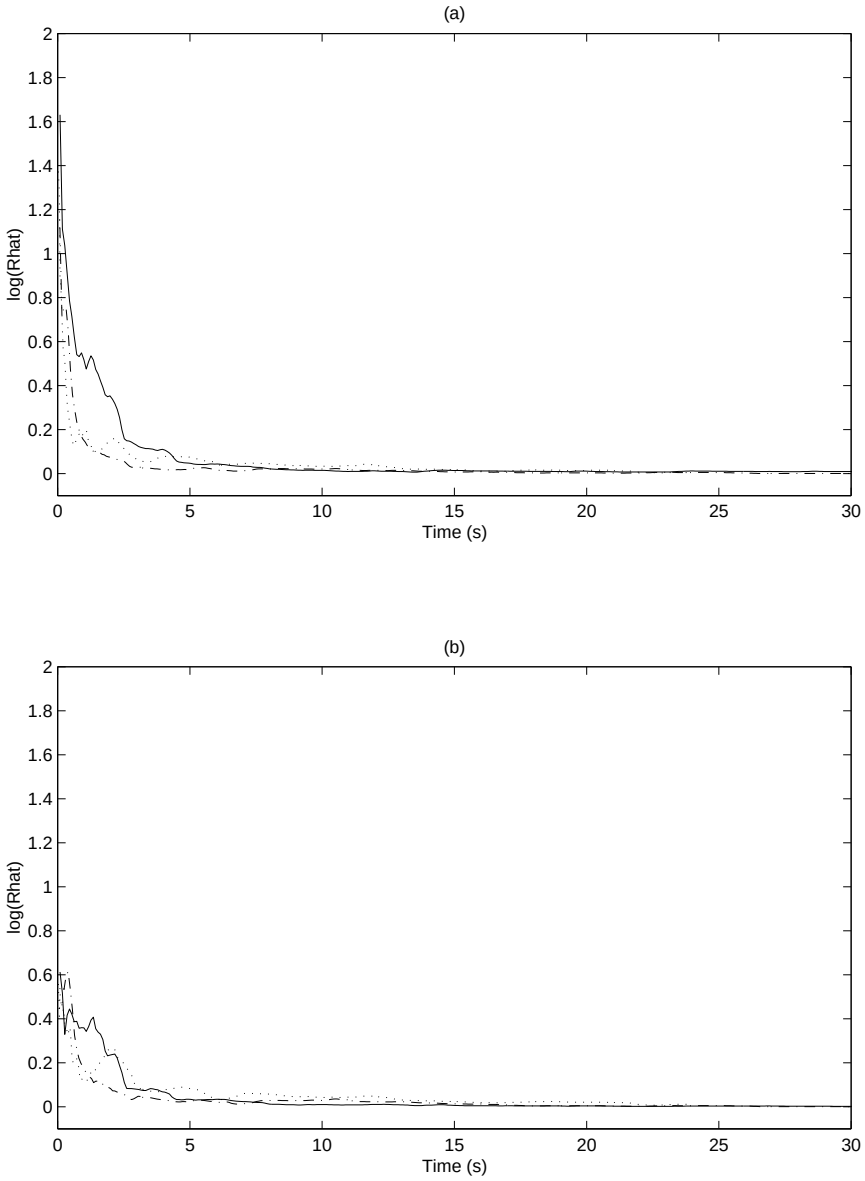

Figure 4.29: Plots of the log potential scale reductions by time for the variance of the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.992$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

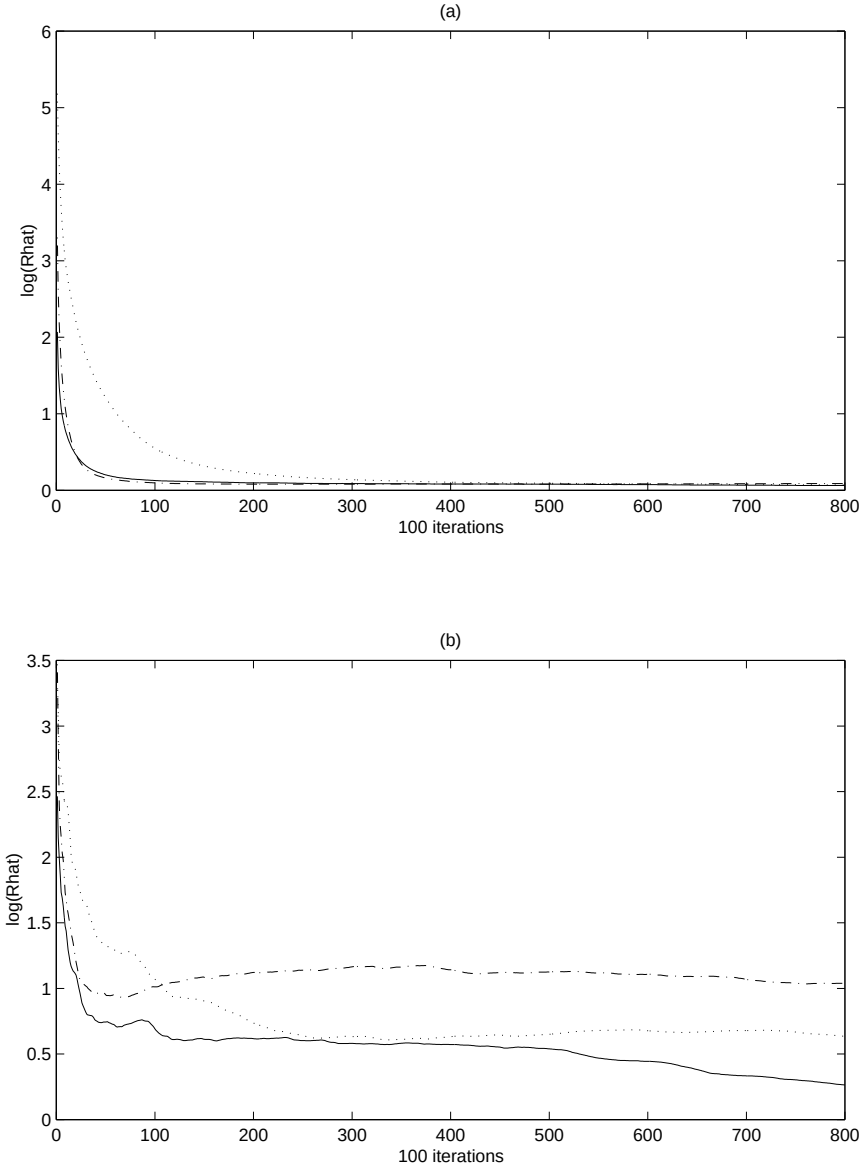

Figure 4.30: Plots of the log potential scale reductions by hundred iterations for the parameters θ_1 and θ_2 for the the simulation of the 200 dimensional normal distribution, with correlation $\rho = 0.992$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

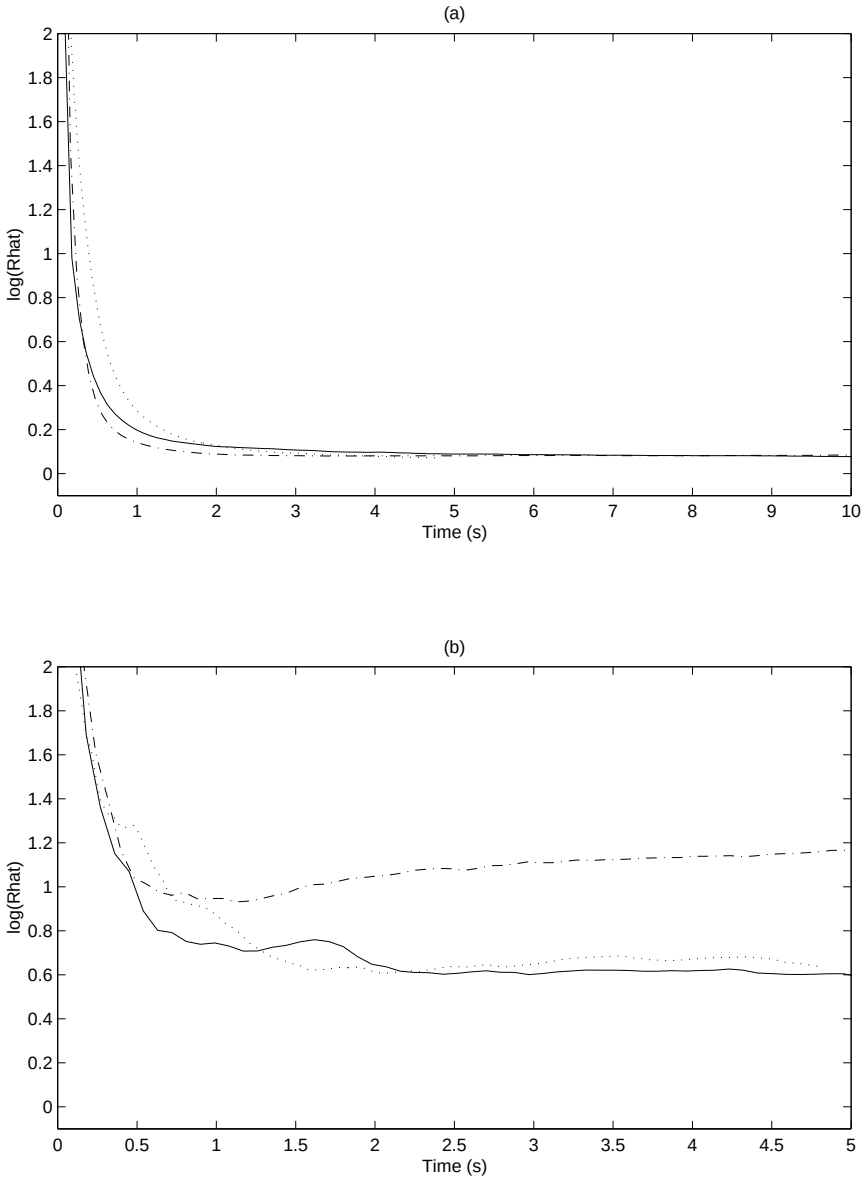

Figure 4.31: Plots of the log potential scale reductions by time for the parameters θ_1 and θ_2 for the the simulation of the 200 dimensional normal distribution, with correlation $\rho = 0.992$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

the acceptance rates drop significantly, from above 95% for each method to 88.7%, 88.0%, and 85.1% for the standard random walk, MALTS, and MALA, respectively. Importantly, as the number of dimensions is increased further, the acceptance rate for the standard random walk decreases, while the acceptance rates for the MALA and MALTS hold steady. The increased dimension has less of an impact on the Langevin type algorithms. Figure 4.32 shows the plot of the acceptance rate versus dimension. All three simulation methods experience a noticeable decline in the acceptance rate as the dimension is increased from 36 to 64, from above 95% to less than 90%. Importantly, as the dimension is increased further, the acceptance rate for the standard random walk continues to decline, while the acceptance rates for the MALA and MALTS algorithms remain steady at 85% and 88%, respectively. This indicates that the increased dimension is having less of an impact on the mixing of the MALA and MALTS algorithms.

The small step size of 1×10^{-5} causes nearly every proposal to be accepted. The slightly lower values for the acceptance rates of the MALA and MALTS algorithms are because the chains produced by these methods will on average take larger steps for a fixed step size parameter than the standard random walk. Recall that the $h(\hat{x}, x)$ in algorithm 4.4 has a minimum value of one, so that the step size parameter of 1×10^{-5} is the smallest step size that is proposed. A similar characteristic exists for the MALA, where the actual step size depends on the magnitude of the gradient at the current state. Increasing the step size by a factor of ten to 1×10^{-4} results in a decrease in the acceptance rates, given in appendix A. While the standard random walk has a higher acceptance

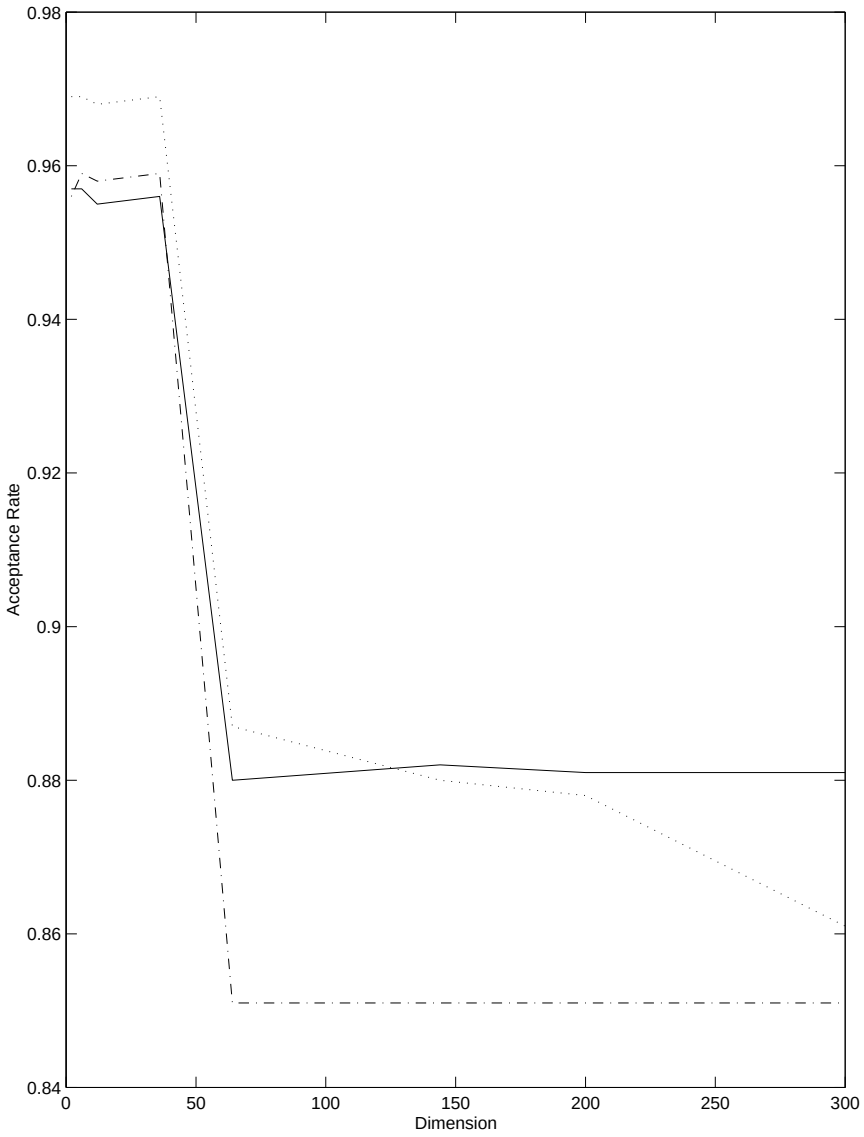

Figure 4.32: Plots of the acceptance rates by dimension for the multivariate normal simulation with correlation $\rho = 0.992$ and step size 1×10^{-5}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk.

rate than the MALA and MALTS, the three methods are within 3% of each other, indicating that they are mixing at about the same rate. The number of iterations required for convergence, shown in figure 4.33, is again less for the MALA and MALTS algorithms than for the standard random walk, although the differences are slight. The plot of the log potential scale reduction versus time in figure 4.34, shows that at this step size the three methods are equally fast. Both the mixing speed and the speed of convergence are nearly equal for the step size of 1×10^{-4}.

For the sake of comparison, the step size was increased to 1×10^{-2}. The larger step size results in a decrease in the acceptance rate of about 10% for each method. Figure 4.35 shows the plot of the log potential scale reduction versus number of iterations for the parameter θ_1 and the variance of the parameter θ_1. The graphs show that the MALTS and MALA methods achieve first and second order convergence in fewer iterations than the standard random walk. The time plots in figure 4.36 show that the three methods take about the same amount of time to reach first order convergence, while the MALA and MALTS methods achieve second order convergence faster than the standard random walk. Thus the derivative using methods (MALA and MALTS) converge to stationarity at the same rate or faster than the standard random walk, and with fewer wasted iterations.

These experiments provide a brief examination of the comparative mixing and convergence properties of the standard random walk, Metropolis adjusted Langevin algorithm, and Modified adjusted Langevin algorithm with tempered step size. A more in depth study, such as that in [28], would better quantify the

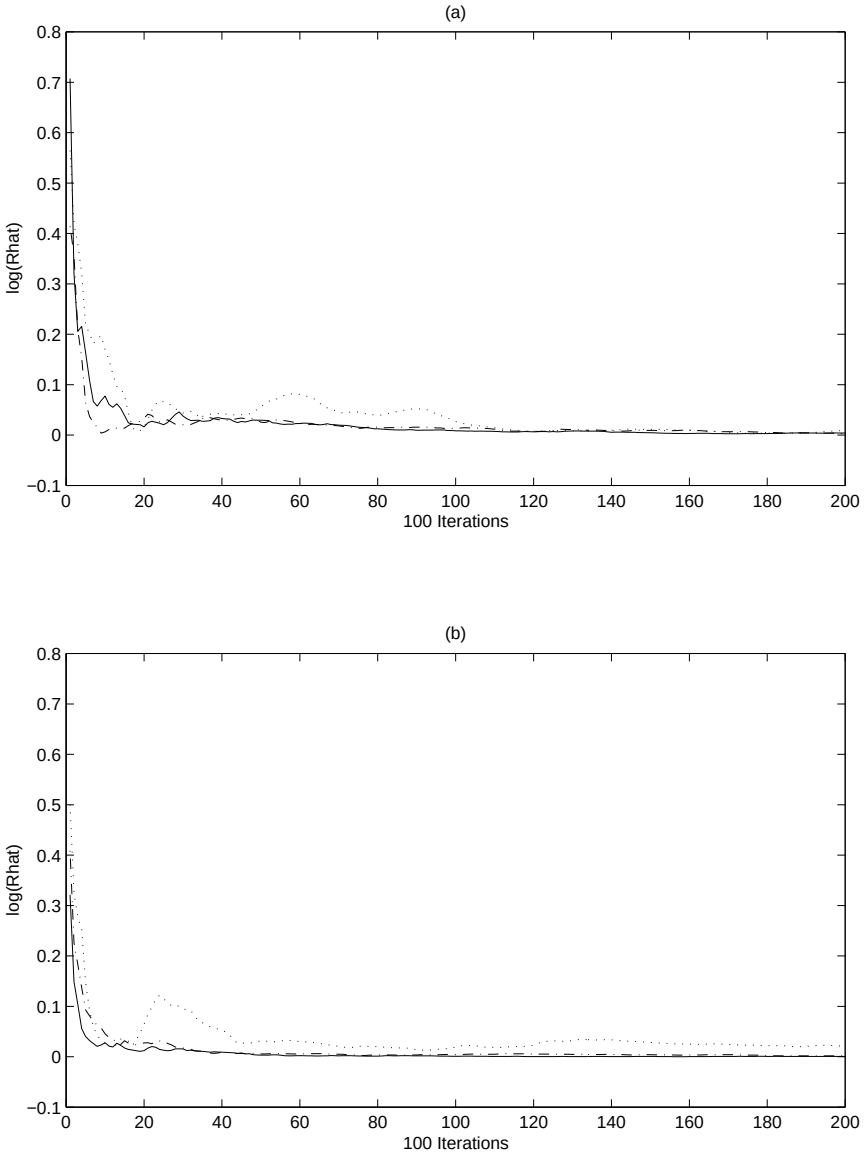

Figure 4.33: Plots of the log potential scale reductions by hundred iterations for the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.992$, and step size 1×10^{-4}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

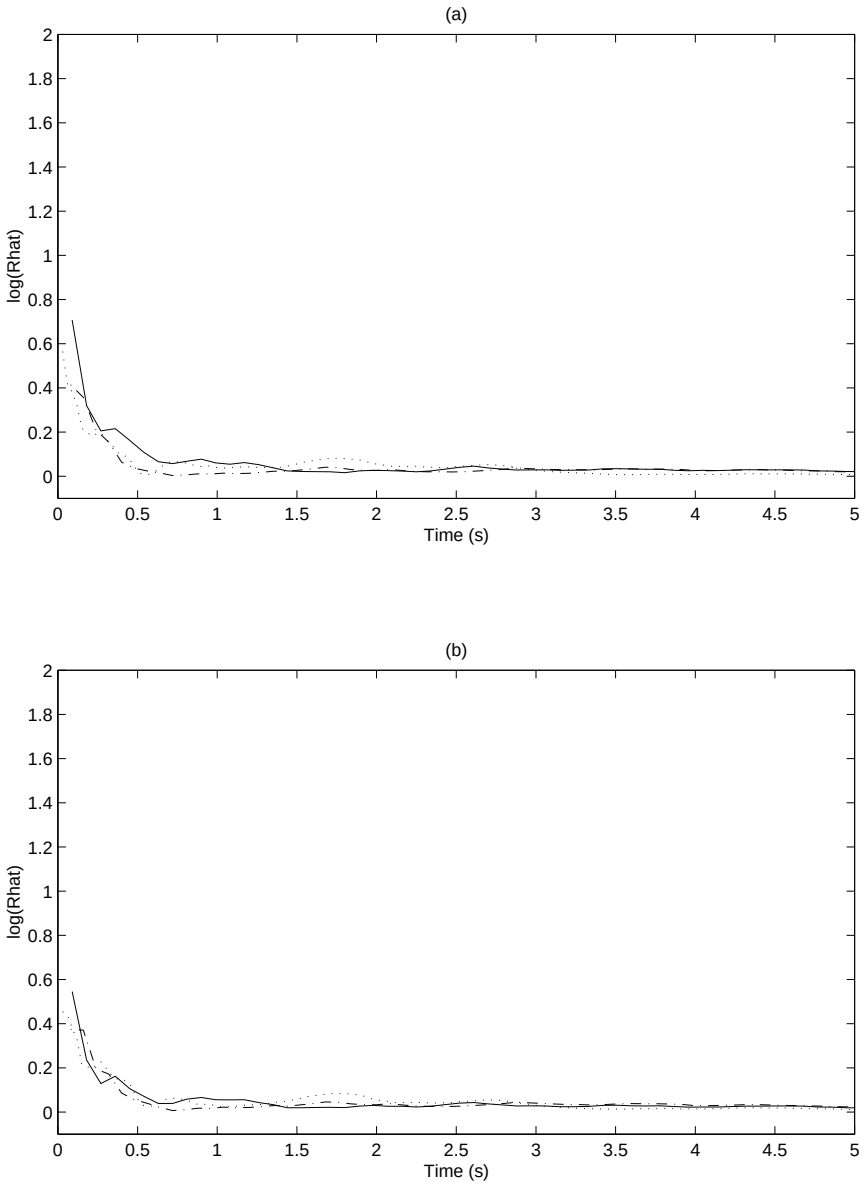

Figure 4.34: Plots of the log potential scale reductions by time for the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.992$, and step size 1×10^{-4}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for θ_2.

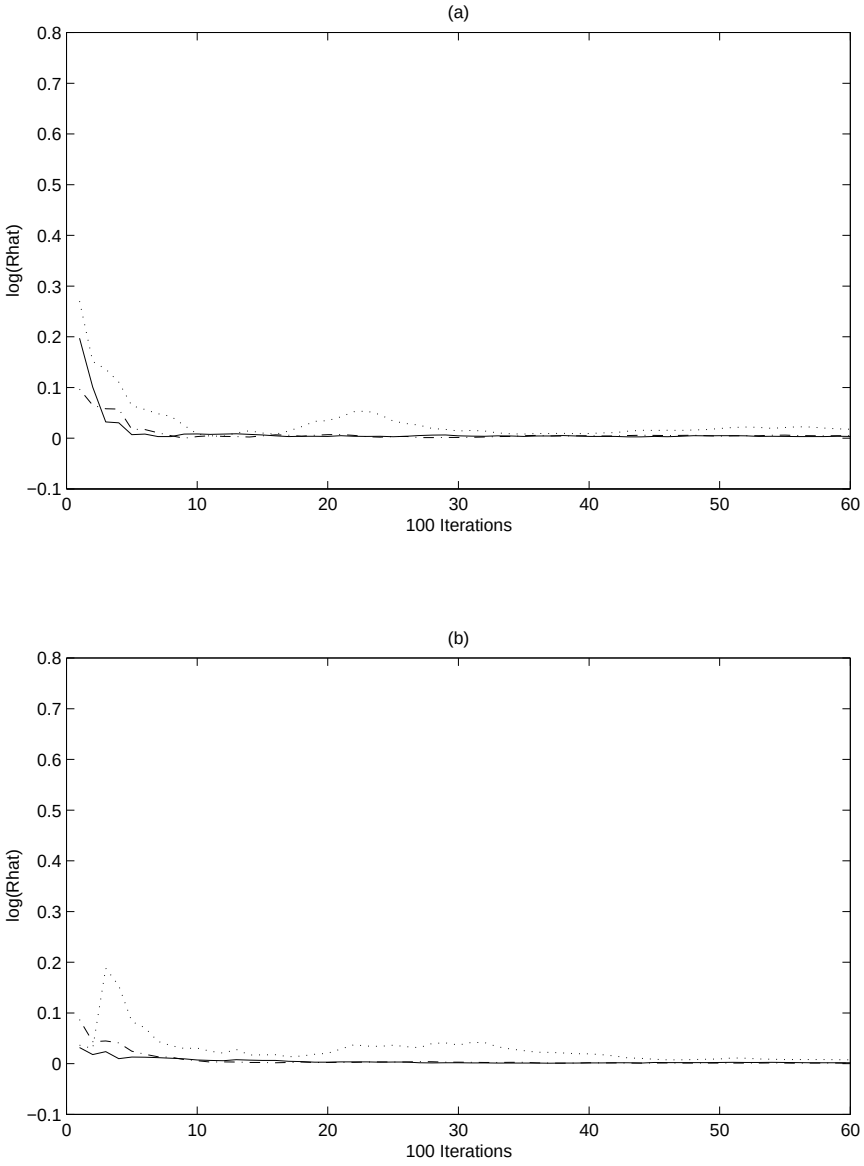

Figure 4.35: Plots of the log potential scale reductions by hundred iterations for the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.992$, and step size 1×10^{-2}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for the variance of θ_1.

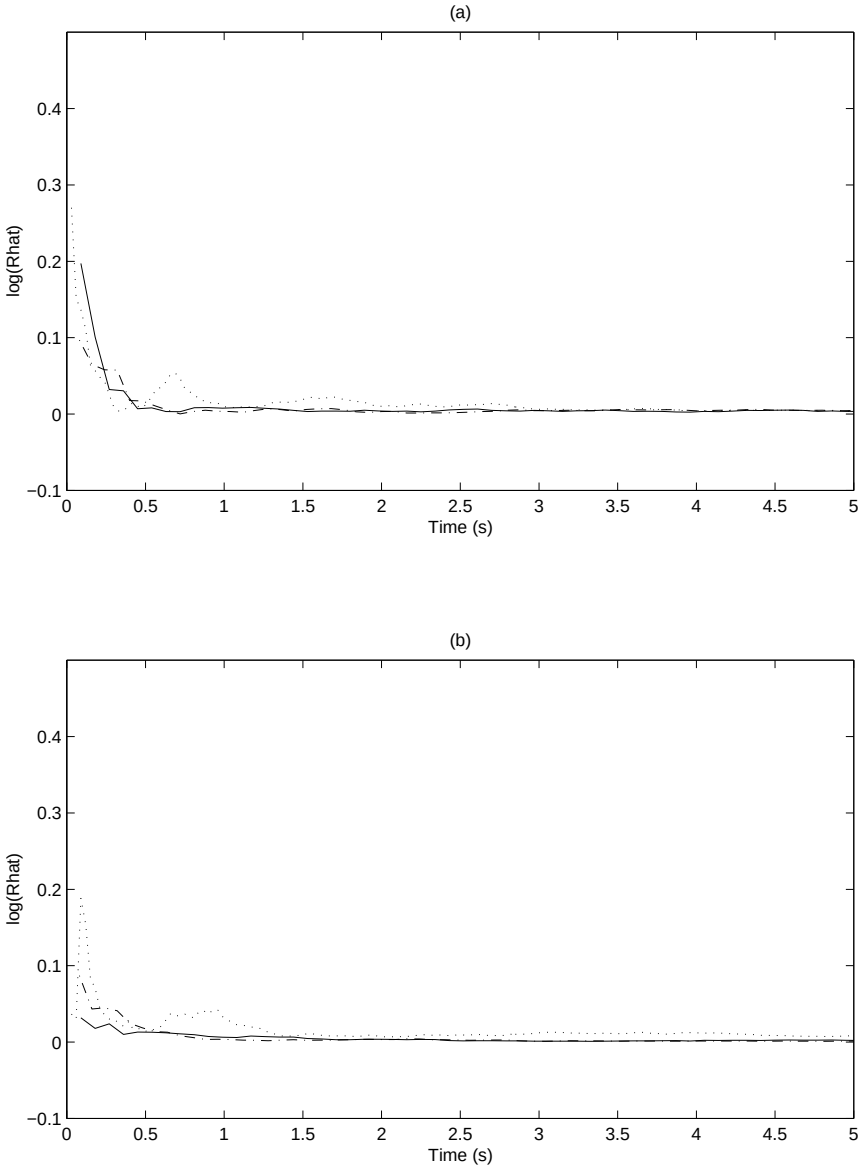

Figure 4.36: Plots of the log potential scale reductions by time for the parameters θ_1 and θ_2 for the the simulation of the 36 dimensional normal distribution, with correlation $\rho = 0.992$, and step size 1×10^{-2}. The solid line is the MALTS algorithm, the dash-dot is the Langevin algorithm, and the dotted line is the standard random walk. (a) shows the results for θ_1. (b) shows the results for the variance of θ_1.

differences between the three algorithms, as well as better explain the effect of correlation, step size, and dimension on the convergence rate, and is reserved for future work. However, the results here indicate that algorithm 4.4 will perform at least as well as the MALA and standard random walk, and for some target densities the MALTS method will outperform these competitors. It should be noted that these experiments used numerical approximations of the gradient in the MALA and MALTS algorithms, which requires log posterior evaluations and slows down the algorithm. Use of the analytical gradient will speed up the algorithms, making them closer competitors with the standard random walk.

4.3.5 Implementation for the CWD Model

Having determined the potential effectiveness of algorithm 4.4, the next step is to implement the method on the Chronic Wasting Disease model. It can easily be checked that the target density for the CWD model satisfies the requirements of theorem 4.1.1, since the posterior is continuous, and has the six dimensional unit cube for its support, which is both connected and contained in the set $\mathbb{R}^6$. Hence the Markov chain produced MALTS algorithm for the posterior density in the CWD model will theoretically converge to its stationary distribution.

The correlation between the parameters in the CWD model was exploited by employing a blocking strategy. Again, because the variable π is independent of the others, it is not included in the Markov chain simulation. Because of the strong interdependencies between the parameters, a single block of six variables was used to update the chain at each iteration.

Moving all six parameters at once can slow the mixing of a Markov chain, but the correlation among the parameters and the use of the gradient information

to guide the chain (step 1 in 4.4) compensates for the curse of dimensionality, since proposed values will tend to be in directions of higher probability than the current state. Langevin-type algorithms are known to perform better, both empirically and analytically, in higher dimensions than the standard random walk [64, 67, 71].

Algorithm 4.4 requires the calculation of the gradient. For simplicity, a first order, one-sided Euler approximation is used, rather than the analytical gradient. Because the posterior distribution for the CWD model is non-zero only within the six dimensional $[0, 1]$ cube, the algorithm rejects values outside of this space, since the acceptance probability will be zero. The use of a one-sided approximation presents a slight problem, since the approximation perturbs each component to the right, and will produce a value that must be rejected if any of the parameters have a current value greater than $1 - \epsilon$, where $\epsilon > 0$. This difficulty is overcome by restricting the parameter space to the six dimensional $[0, 1)$ cube, which results in the loss of a set of measure zero, and does not noticeably affect the simulation.

The step size chosen in step 2 of 4.4 requires the comparison of the posterior density at the current point X_t to the density at the mode. If $\ell(x)$ is the log of the posterior function evaluated at x, then the step vector at X_t is $k(\ell(\hat{x}) - \ell(X_t) + 1)$, where $\hat{x}$ denotes the mode of the posterior. The diagonal elements of the proposal covariance matrix Σ were chosen to independently scale the variance for each component of the parameter vector $\boldsymbol{\theta}$. Both k and Σ are chosen to prevent the frequent proposal of values outside the support of the parameter space. This was done by calculating the step size at several points in the parameter space

and adjusting k and Σ as necessary. For the CWD model, the non-zero diagonal components of Σ are 1×10^{-9} time the variances of the prior distributions, with a step size parameter of $k = 1 \times 10^{-9}$. The small step size was used because of the erratic behaviour of the gradient, and it is not known if these are the optimal values. The code is given in appendix C section C.4.4. Tuning guidelines are discussed in section 4.3.3.

4.4 Performance of the Markov Chain Sampler

The Markov chain sampler was run using Matlab, version 6, on a machine with a $2.2Ghz$ Pentium 4 processor, with $2GB$ RAM. The machine was the same machine as that used to run the acceptance sampler. The Markov chain simulation was significantly faster than the acceptance sampler, completing 300,000 iterations in one hour and 15 minutes. To assess the performance of the Markov chain sampler, the generated sample is compared to the sample generated by the acceptance sampler. As a performance benchmark, the results of algorithm 4.4 are compared to the standard random walk implementation for the CWD model. The standard random walk simulation took 40 minutes to run on the same machine on which algorithm 4.4 was run.

There are a number of criteria for determining a suitable stopping value for a Markov chain simulation [64]. One of the most important is convergence of the chain to the stationary distribution. Currently, methods such as *perfect sampling* rigorously achieve stationarity from the beginning, but these methods are currently impractical and difficult to apply in most settings [64, 60]. For the CWD model, convergence will be determined by starting the chain from multiple starting points and calculating the potential scale reduction discussed and used

in section 4.3.4, and by the application of graphical methods. Graphical methods are the easiest to apply, and allow for a qualitative decision on whether or not a Markov chain has reached stationarity, but the results can be misleading [64].

Figure 4.37 shows the plots of the initial prevalence in DAUs 10 and 4 versus the iteration. While there does appear to be a small amount of structure in the graphs, overall they are oscillating around a stable value. The same is true of the parameters $p_{5,0}$ and γ, shown in figure 4.38. In each case, the marginal chain seems to have settled into a nearly stationary distribution.

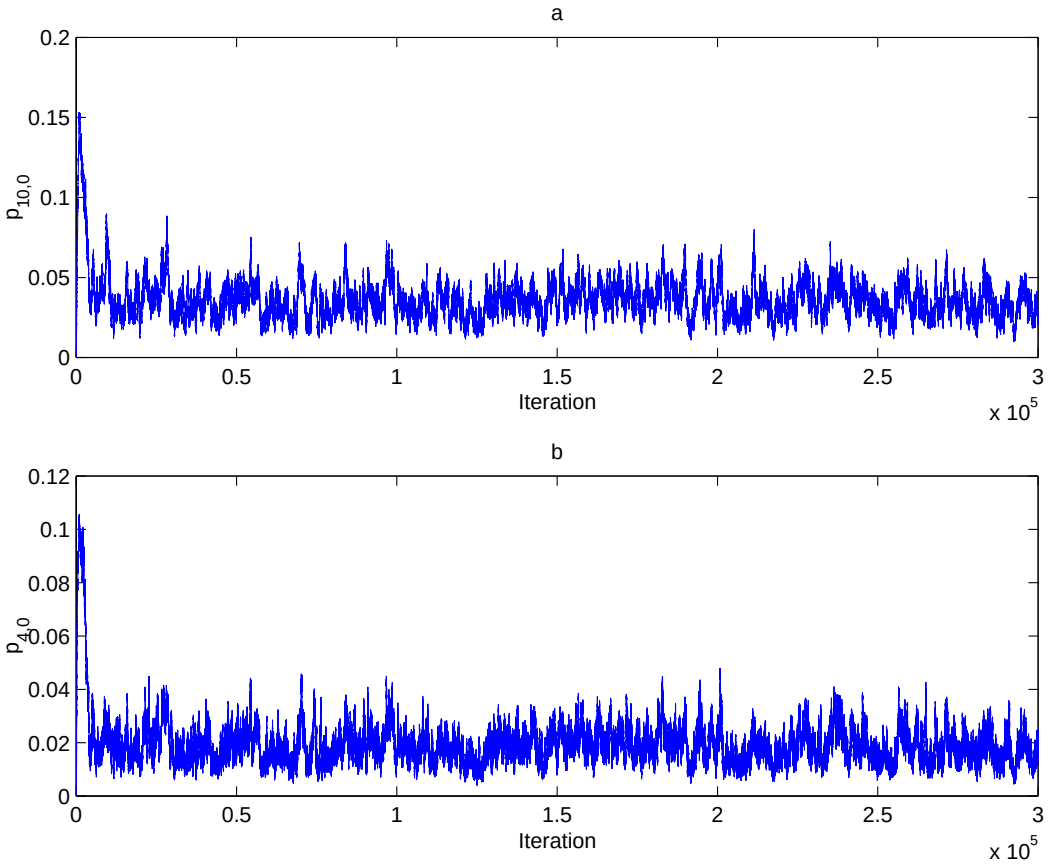

Figure 4.37: Time plots of the parameters $p_{10,0}$ and $p_{4,0}$ for the Markov chain simulation. (a) shows the plot of $p_{10,0}$ by iteration. (b) shows the plot of $p_{4,0}$ by iteration.

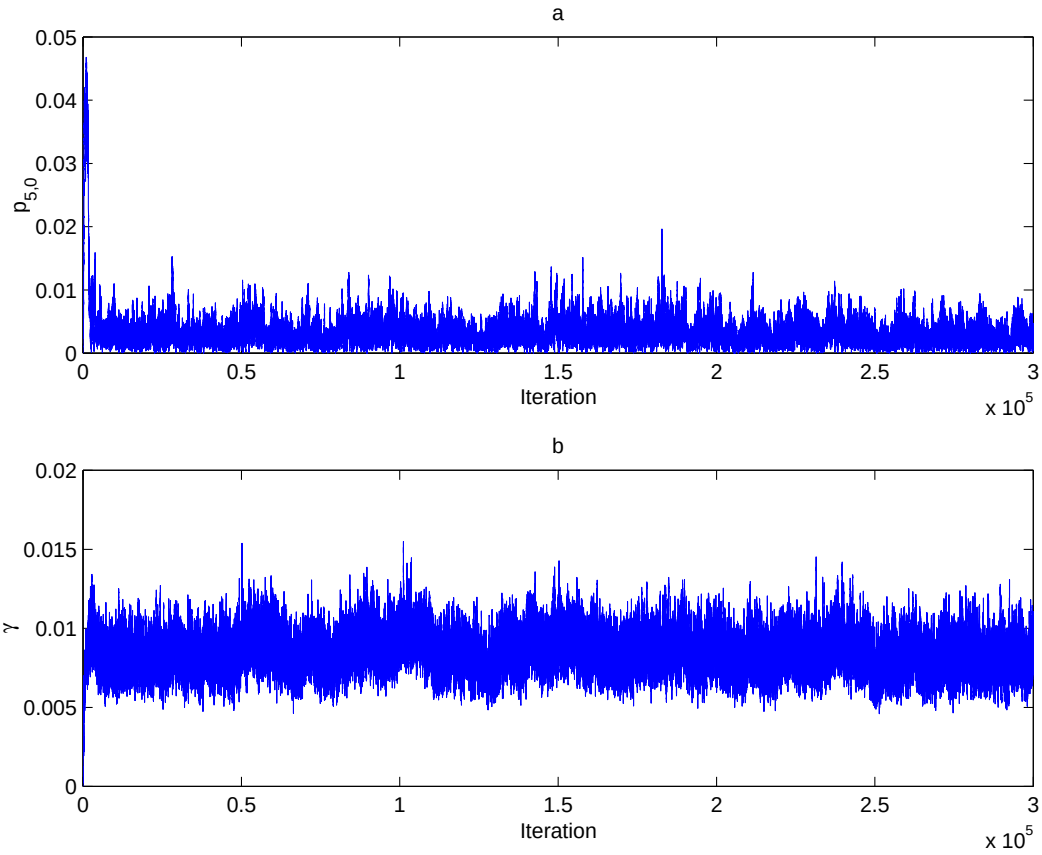

Figure 4.38: Time plots of the parameters $p_{5,0}$ and γ for the Markov chain simulation. (a) shows the parameter $p_{5,0}$ by iteration. (b) shows the parameter γ by iteration.

The plots of the parameters α and δ versus the iteration, shown in figure 4.39, are not as comforting as those for the other parameters. There is not convincing evidence one way or the other to show that the marginal distributions have reached their stationary state, even though they have run for a total of $300,000$ iterations. Neither of the plots in figure 4.39 appears to have settled around a single value, and there appears to be large scale oscillations. However, these parameters are highly interdependent, as shown by the pairwise contour plot in figure 3.5 generated by the acceptance sampler. Careful examination of the step plots of α and δ shows that the large scale behaviour of one parameter is opposite that of the other. That is, as α trends up, δ trends down. It is likely that the correlation between these two parameters is the reason for the undesirable behaviour. The pairwise contour plot of the two, shown in figure 4.40, is perhaps a better indication of whether or not the chain is close to its stationary. The pairs plot shows the "banana"– the ridge of high probability in figure 3.5– that was expected for the posterior sample.

The other pairwise plots show the same structure and range as the plots generated by the acceptance sampler. Figures 4.41 and 4.42 show the contour plots for the remaining pairs of parameters. These plots show strong similarities to those produced by the acceptance sampler (figures 3.6 and 3.7), which gives evidence that the Markov chain is approximating the correct distribution.

The potential scale reduction can alleviate the concerns about the behaviour of the parameters α and δ, as well as provide a quantified assessment of convergence. A Markov chain simulation was run for $300,000$ iterations from each of six initial values using both algorithm 4.4 and the standard random walk.

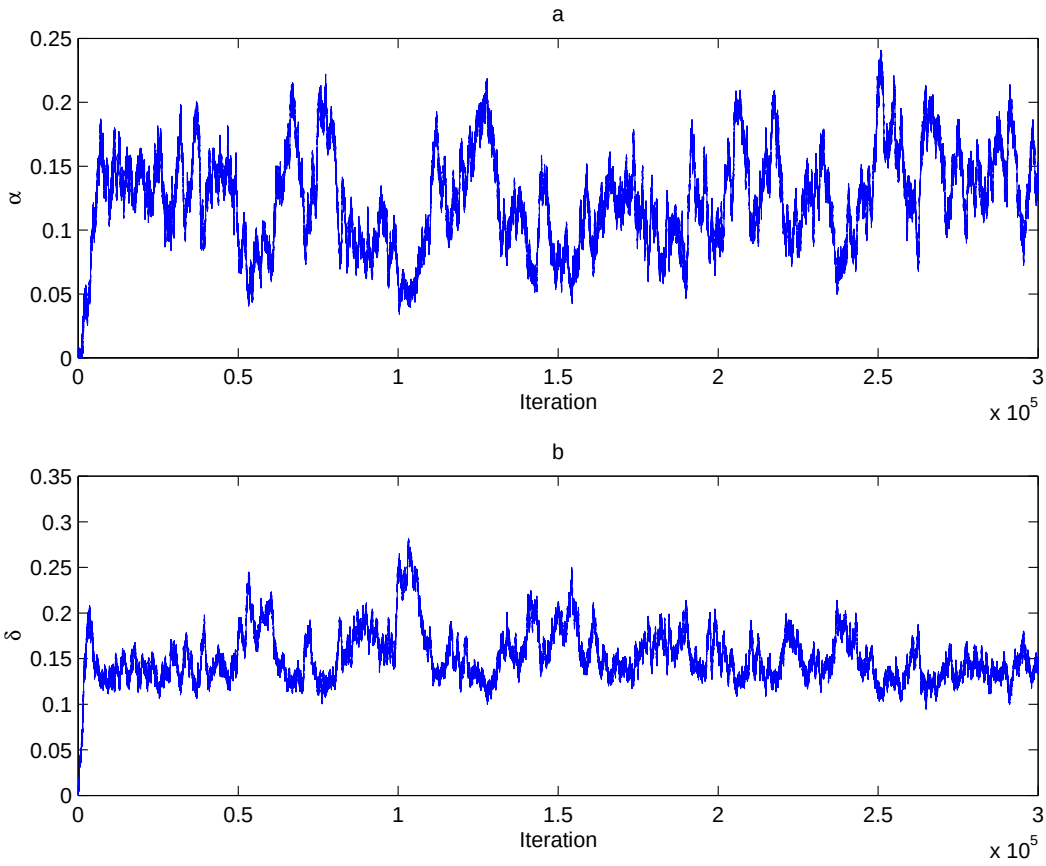

Figure 4.39: Time plots of the parameters α and δ for the Markov chain simulation. It is not clear from these plots that the chain has reach stationarity. (a) shows the parameter α versus the iteration. (b) shows the parameter δ versus the iteration.

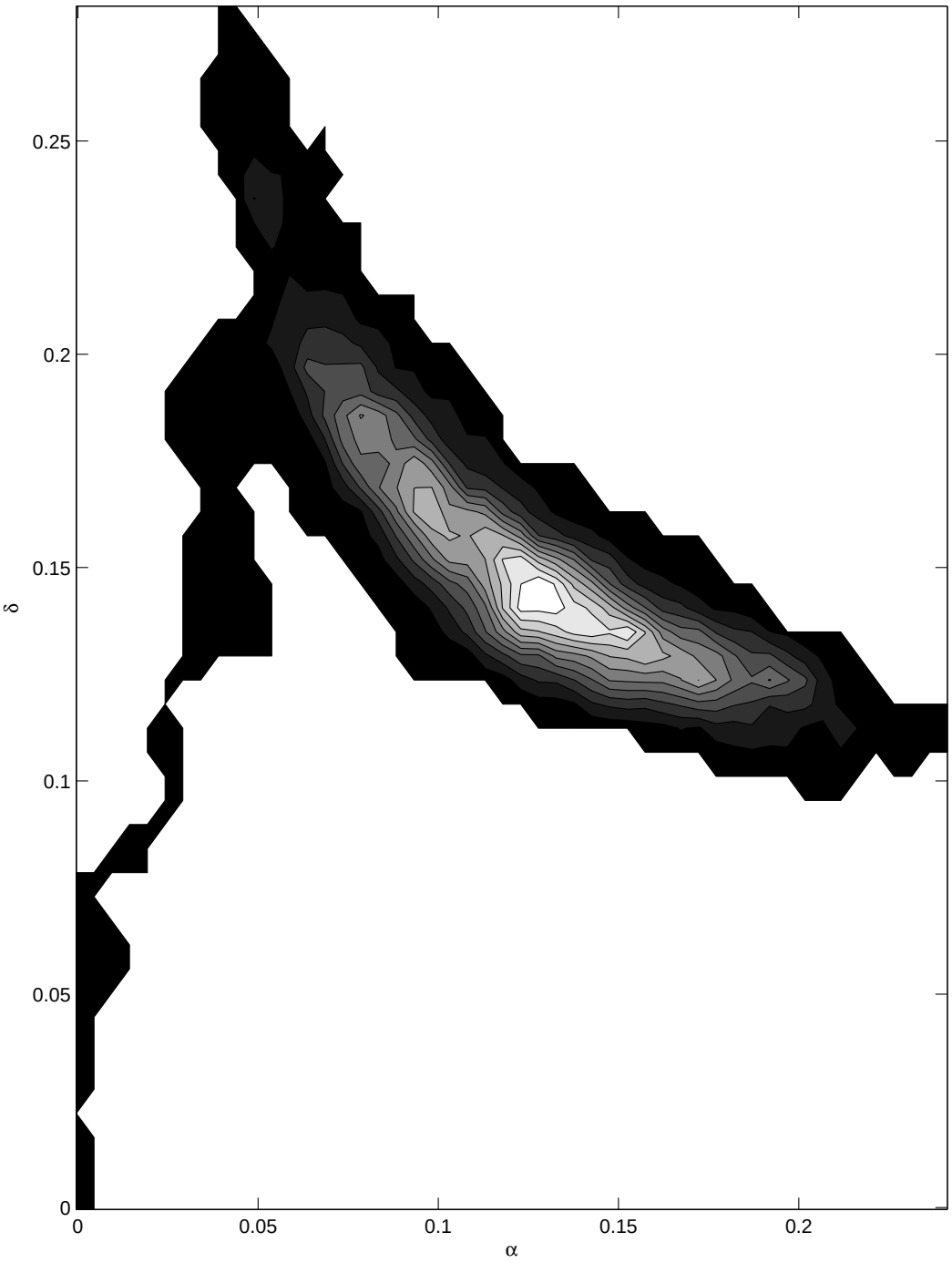

Figure 4.40: Contour plot of the dynamic parameters α by δ for $250,000$ iterations of the Markov chain simulation. The contour plot is extremely close to the acceptance sampler plot in figure 3.5 and produces the distinctive banana shaped ridge.

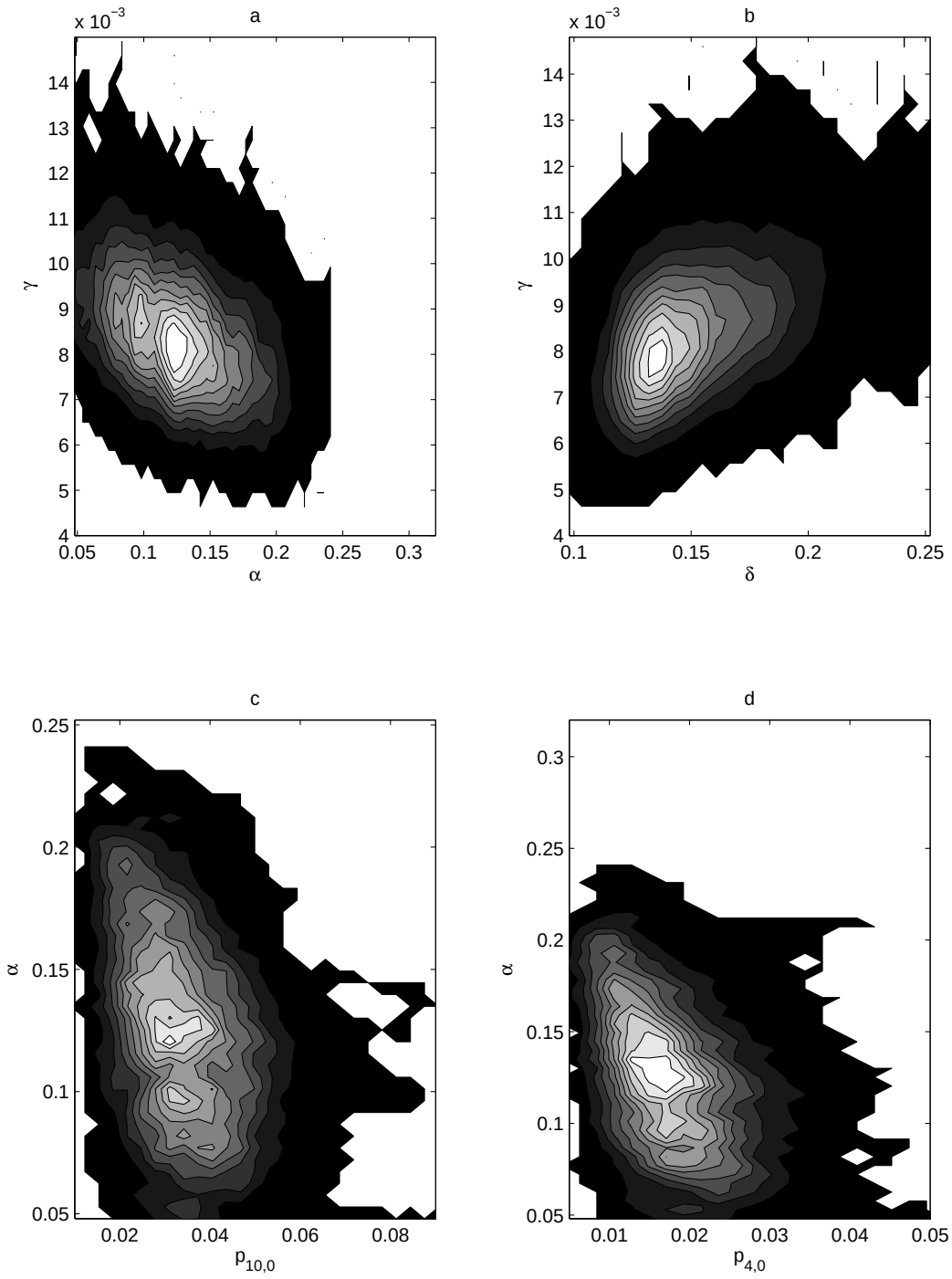

Figure 4.41: Contour plots produced by 250, 000 iterations of the Markov chain simulation. The large number of points produces a smoother image than the contour image produced by the acceptance sampler. (a) shows the parameters γ versus α. (b) shows the parameters γ versus δ. (c) shows the parameters α versus $p_{10,0}$. (d) shows the parameters α versus $p_{4,0}$.

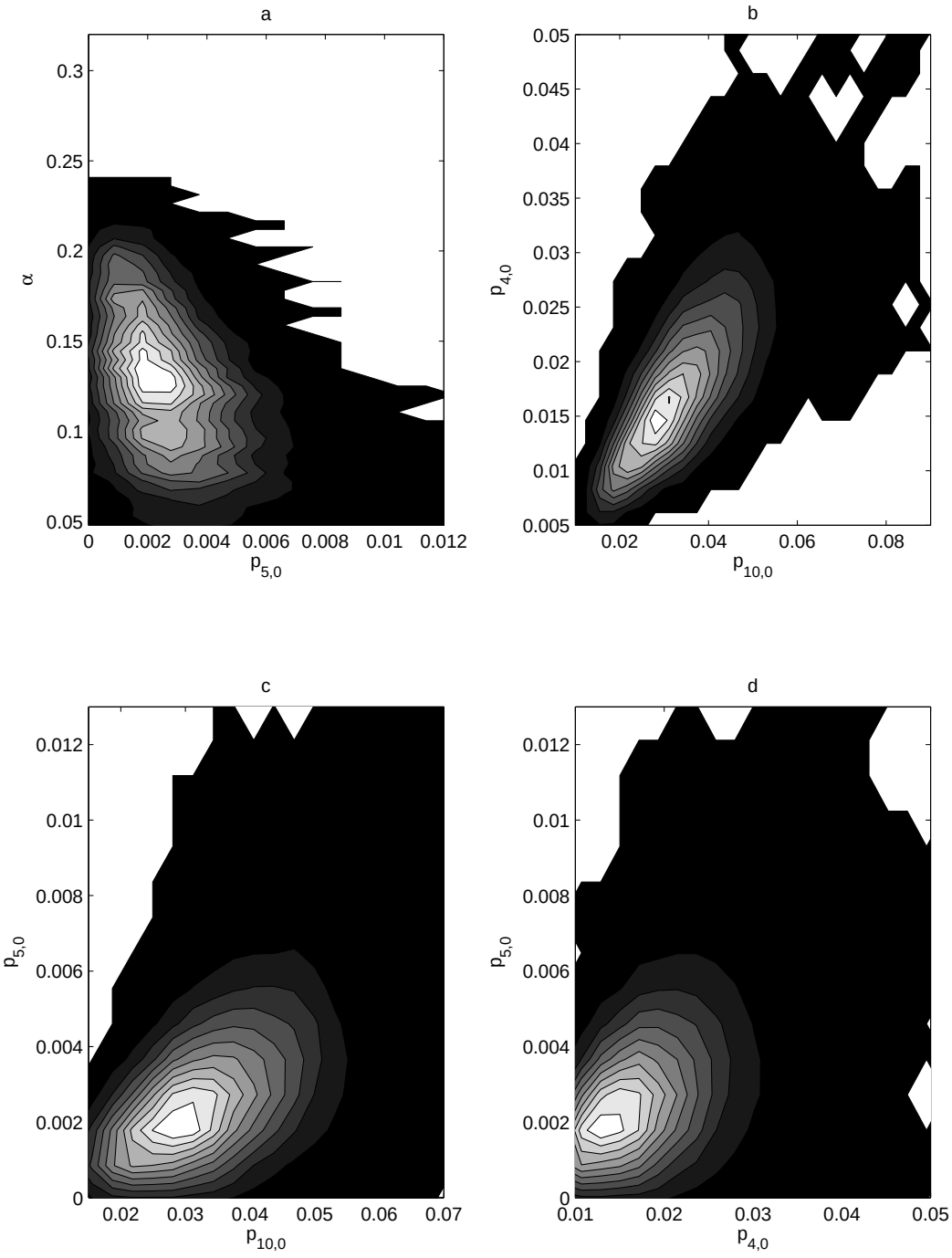

Figure 4.42: Contour plots produced by 250, 000 iterations of the Markov chain simulation. The large number of points produces a smoother image than the contour image produced by the acceptance sampler. (a) shows the parameters α versus $p_{5,0}$. (b) shows the parameters $p_{4,0}$ versus $p_{10,0}$. (c) shows the parameters $p_{5,0}$ versus $p_{10,0}$. (d) shows the parameters $p_{5,0}$ versus $p_{4,0}$.

The potential scale reduction was calculated every $1,000$ iterations and the results, shown in figure 4.43, alleviate any concerns about convergence raised by figures 4.39. The rule of thumb for convergence to approximate stationarity is that the statistic $\sqrt{\hat{R}}$ is below 1.2 for each parameter [24], which occurs by 5000 iterations for algorithm 4.4, and does not occur at all for some of the parameters with the standard random walk. Although the standard random walk produced $300,000$ iterations faster than algorithm 4.4, the potential scale reduction clearly indicates that this method has not reached stationarity.

Calculation of the potential scale reduction gives evidence that the Markov chain produced by algorithm 4.4 has converged to its stationary distribution. To ensure that the chain has converged to the correct distribution, the marginal distributions produced by the Markov chain sampler are compared to the marginal distributions produced by the acceptance sampler. The comparisons are done by means of the non-parametric Kolmogorov-Smirnov Test on the marginal distributions [2, 53]. For both the acceptance sample and the Markov chain sample, 1000 values were drawn at random. For each parameter, the null hypothesis is that the sample produced by the acceptance sampler and the sample produced by the Markov chain simulation were drawn from the same continuous distribution. This is tested against the alternative hypothesis that the two samples were drawn from different continuous distributions. The results of the Kolmogorov-Smirnov tests are shown in table 4.1. The smallest p-value is 0.432 for the parameter $p_{5,0}$, so the null hypotheses are clearly supported by these results. Any differences in the marginal distributions are not statistically significant.

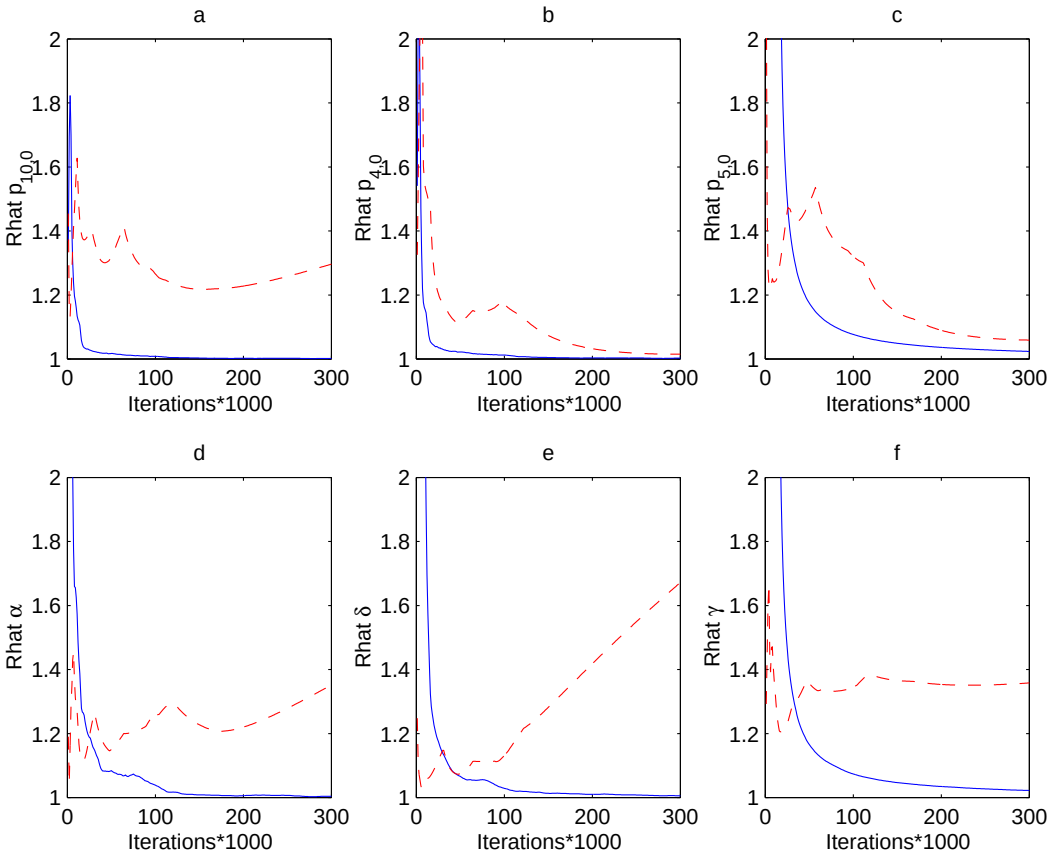

Figure 4.43: Plot of the potential scale reduction by iteration for each parameter. The dashed line shows the results of the standard random walk. The solid line shows the results for algorithm 4.4. The graphs illustrate both the poor performance of the standard random walk and the convergence to stationarity of 4.4. (a) shows the results for $p_{10,0}$. (b) shows the results for $p_{4,0}$. (c) shows the results for $p_{5,0}$. (d) shows the results for α. (e) shows the results for δ. (f) shows the results for γ.

Parameter	K-S Statistic	p-value
$p_{10,0}$	0.026	0.888
$p_{4,10}$	0.029	0.794
$p_{5,0}$	0.039	0.432
α	0.022	0.969
δ	0.029	0.794
γ	0.021	0.980

Table 4.1: Values of the Kolmogorov-Smirnov test statistic and the corresponding p-values comparing the acceptance sample with the Markov chain sample for each parameter.

The potential scale reductions and the results of the Kolmogorov-Smirnov Tests indicate that the Markov chain produced by algorithm 4.4 has converged to its stationary distribution, and the marginal distributions produced by the Markov chain sampler are statistically identical to the marginal distributions produced by the acceptance sampler discussed in chapter 3. This conclusion is further supported by comparing the contour plots from the Markov chain simulation with those obtained by the acceptance sampler. The results of the Markov chain simulation can therefore be confidently used to analyze the posterior distribution of the Chronic Wasting Disease model.

The 95% credible intervals for the Markov chain simulation are shown in table 4.2. These intervals are almost identical to those produced by the acceptance sampler, shown in table 3.3. This is further evidence that the Markov chain is producing a good approximation to the target distribution. The mean, median, and standard deviation of each parameter can also be examined to

compare them with those obtained by the acceptance sampler. The posterior estimates and credible intervals are shown in table 4.2. The standard deviations for the Markov chain sampler are slightly smaller than those for the acceptance sampler. Visual inspection of the graphs of the parameters versus the iteration indicates that approximate stationarity occurs at about the 8000th iteration, so the first 8000 values can be removed from a single run before the posterior Monte Carlo Estimates are constructed. To better approximate an independent sample, however, the last $50,000$ iterations of each of the six runs are combined and the results used as the posterior sample. This is an accepted method of achieving a nearly independent sample [24, 25].

Parameter	Lower Limit	Upper Limit	Mean	Median	StdDev
$p_{10,0}$	1.73	6.34	3.58	3.41	1.28
$p_{4,0}$	0.86	3.71	1.92	1.78	0.88
$p_{5,0}$	0.08	0.78	0.34	0.30	0.28
α	5.90	20.76	12.27	11.96	3.76
δ	11.32	21.00	15.24	14.96	2.55
γ	0.63	1.10	0.85	0.84	0.12

Table 4.2: 95% Bayesian credible intervals and Monte Carlo estimates for posterior density from the Markov chain simulation. All values have been multiplied by 100.

Algorithm 4.4 works exceptionally well for the Chronic Wasting Disease model. The MALTS algorithm is significantly faster than the acceptance sampler that was used, and analysis of multiple chains indicates that the chain

reaches stationarity relatively quickly compared to the standard random walk. Using the sample generated by the acceptance sampler as a reference, a goodness of fit test indicates that the dependent sample produced by the Markov chain is closely approximating the target distribution. Having built a model for Chronic Wasting Disease, and designed an efficient sampling mechanism, the next stage is using the results of the Markov chain sample to analyze the posterior distribution, and then interpreting the results of the simulation.

5. Results

5.1 Analysis of the Posterior Estimates

Having generated two samples from the posterior distribution (using both an acceptance sampler and a Markov chain sampler), the posterior parameter estimates must be interpreted. This is done using the sample generated by the Markov chain simulation. Although there is some Monte Carlo error inherent in using the approximate sample generated by a Markov chain, the large sample size of $300,000$ should mitigate this error. Moreover, since six runs were made using different initial values, the last $50,000$ iterations of each run were combined and the results used as the posterior sample. Combining the results from each run results in a nearly independent sample [24]. The results of the model will be compared to reasonable expectations for the future behaviour of Chronic Wasting Disease.

Firstly it must be determined whether or not the model provides a reasonable fit for the data. This is done by comparing the predicted values with the actual values from the data. Figures 5.1, 5.2, and 5.3 show fitted prevalence values, based on the median of posterior sample (depicted by dashed line and ∇,) with corresponding 95% frequentist confidence intervals for the prevalences (dotted line, $\square$). The figure shows that the model provides a reasonable fit to the data (solid line, $\circ$). The five DAUs shown here are those at the heart of the endemic area, and the data corresponding to the five panels are primarily responsible for the model form as they carry most of the information in the

likelihood and hence posterior distribution. The intervals capture nearly all of the data, and provide assurance that the model is not completely off base.

Figure 5.4 shows the predicted prevalence curve for the DAUs based on the year by year medians of the prevalence tracks based on the posterior sample. The curves maintain the sigmoidal shape expected by the logistic equation, increasing with time toward the median long-term stable state, $\delta = 0.149$. This behaviour is expected from the logistic differential equation model constructed for prevalence (3.4). This result indicates that the posterior distributions are retaining the qualitative behaviour of the differential equations. This retention quality is crucial to the performance of the model. Those DAUs that were first infected approach δ first. Because the spatial mixing term γ is greater than zero, and because every DAU is connected with at least one other DAU via the relationships in Table 1, all DAUs eventually become infected, and thus eventually approach the stable state. By design, non-infected DAUs can become infected only through spatial mixing, an assumption that is in-line with the current understanding of this and other infectious diseases. Because the disease is seen over time to be spreading to the non-infected DAUs, the spatial mixing is effective. Figure 5.4 demonstrates that incorporating spatial dynamics directly into the model is an effective mechanism.

Figure 5.5 shows the medians (solid line) and 95% credible intervals (dashed lines) for the prevalence tracks over time for the five endemic DAUs. The credible intervals are calculated by drawing a random sample of $25,000$ from the simulated joint posterior, and using each of these parameter values to deterministically calculated future prevalence values using equation 3.5. For each year, a

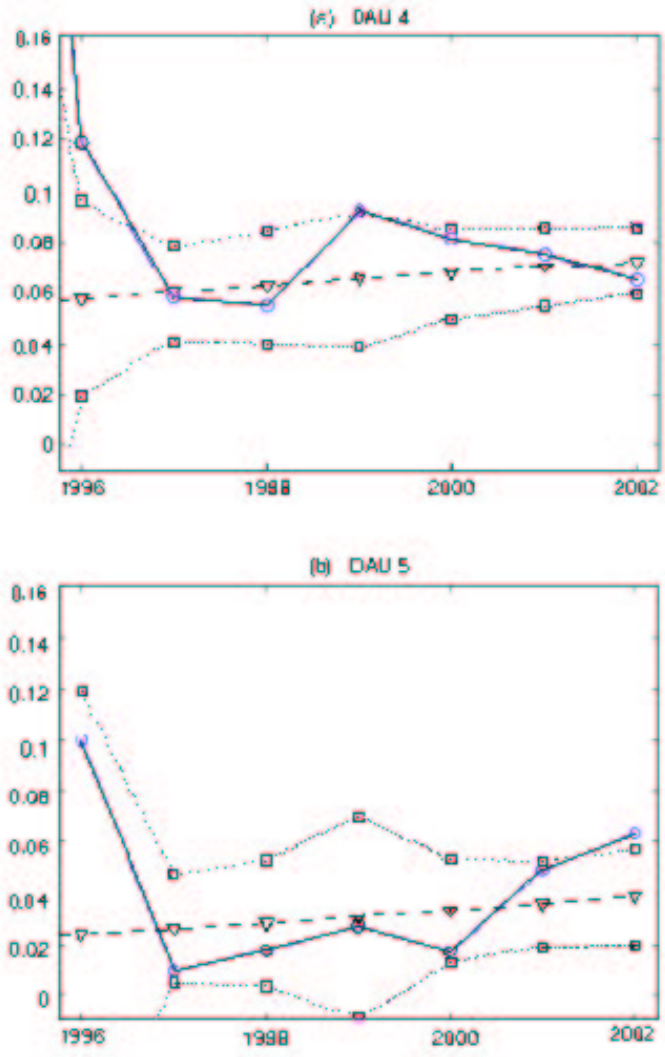

Figure 5.1: Fitted values and corresponding frequentist confidence intervals for the proportion of infected deer in the endemic region, based on the component-wise median of the posterior sample. Depicted are (a) DAU 4, and (b) DAU 5. Observed prevalences are denoted with $\circ$, 95% confidence intervals are denoted by $\square$ and the predicted value is denoted by ∇.

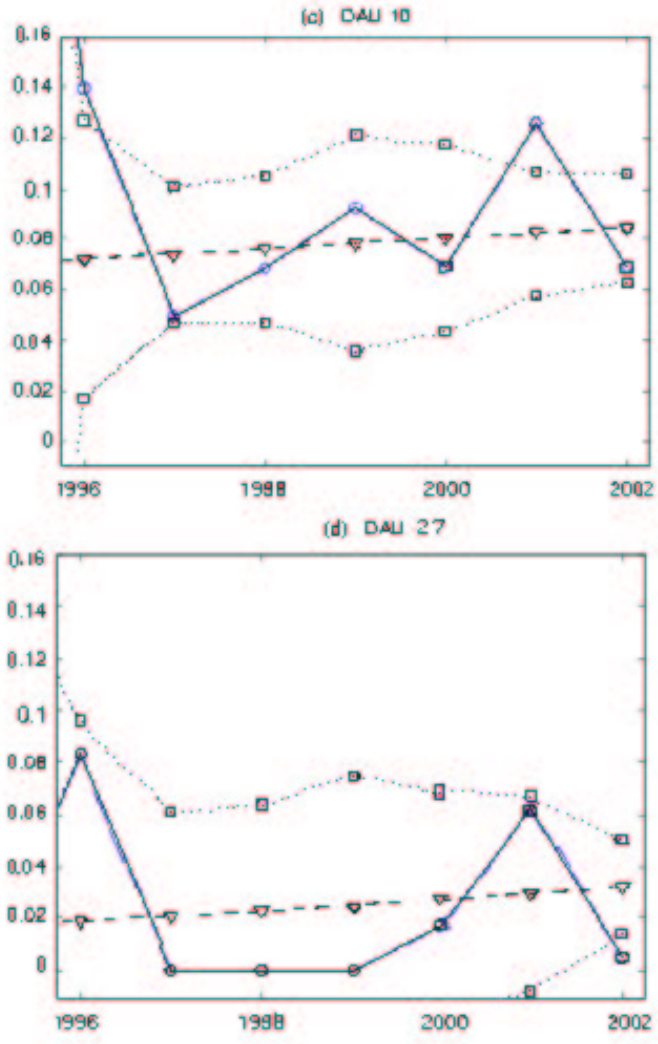

Figure 5.2: Fitted values and corresponding frequentist confidence intervals for the proportion of infected deer in the endemic region, based on the component wise median of the posterior sample. Depicted are (c) DAU 10, and (d) DAU 27. Observed prevalences are denoted with $\circ$, 95% confidence intervals are denoted by $\square$ and the predicted value is denoted by ∇.

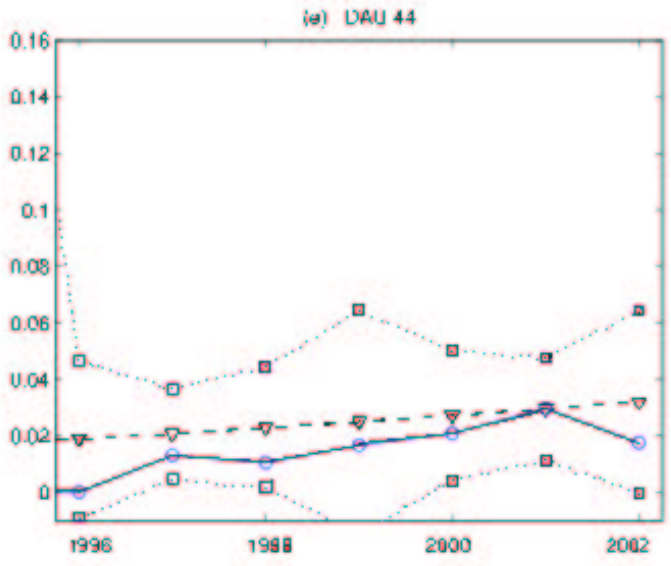

Figure 5.3: Fitted values and corresponding frequentist confidence intervals for the proportion of infected deer in the endemic region, based on the component wise median of the posterior sample. Depicted are (e) DAU 44. Observed prevalences are denoted with $\circ$, 95% confidence intervals are denoted by $\square$ and the predicted value is denoted by ∇.

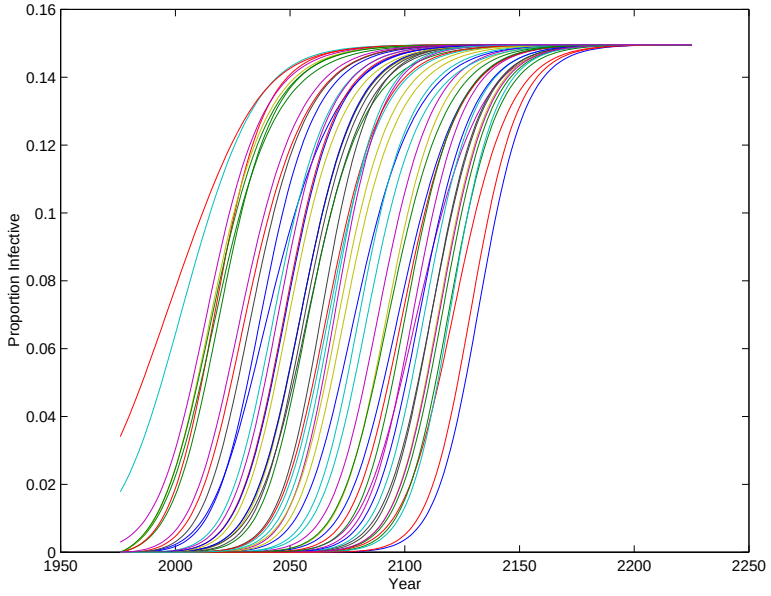

Figure 5.4: Predicted prevalences from 1976 to 2250, for all DAUs based on the median curve based on the posterior Markov chain sample.

95% credible interval is calculated. These intervals are narrowest in that period where the majority of the data lies (1996-2002) and grow wider as the model projects further into the future. The model is a simplified representation of the actual dynamic process, but it provides both a reasonable fit to the data and predictions for future prevalence that are congruent with current scientific knowledge of CWD. The steadily increasing trend shown in the graph indicates that the disease will spread across the state and increase to a maximum sustainable level of about 14%. This prediction may seem pessimistic, but considering that a period of decrease for Chronic Wasting Disease has not yet been observed in nearly 30 years of monitoring, it is neither unreasonable nor surprising. These

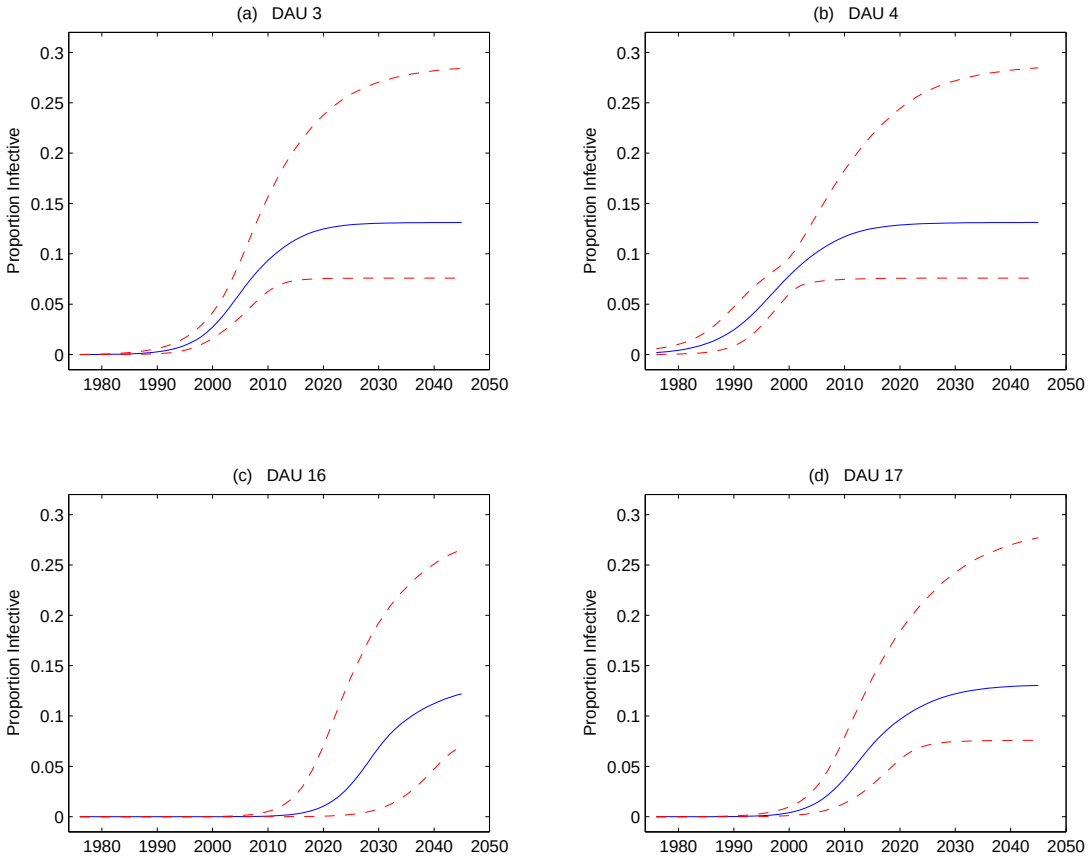

Figure 5.5: Solid line is the median track for prevalence based on the posterior Markov chain sample ($N = 300,000$). Dashed lines represent pointwise 95% credible intervals based on Markov chain sample. (a) shows DAU 3. (b) shows DAU 4. (c) shows DAU 16, which borders the endemic region. (d) shows DAU 17.

predictions do not account for any control strategies.

Examination of the posterior credible intervals gives more insight into the behaviour of the disease as predicted by the model. From a wildlife management perspective, the most important parameter is probably the long run disease prevalence δ. The posterior distribution on δ has the median and mode at approximately 0.148. The estimated 95% credible interval from the Markov chain sample is (11.3%,21.0%), compared with (4.0%,15.5%) for the prior distribution. This credible interval is evidence of a non-zero long-run sustainable level for the disease, and therefore is strong evidence that CWD is unlikely to disappear naturally from the deer population. This conclusion is graphically depicted in figures 5.4 and 5.5. Thus, carefully crafted strategies are of critical importance in order to manage CWD. From what is currently known, the steady increase to a sustainable level is reasonable, perhaps expected, over the given time frame. Thus far, the disease has only been observed to increase.

The posterior estimates for the initial prevalence values for DAUs 10, 4, and 5 provide another qualitative check for the model. The posterior mode for the marginal distribution for the initial prevalence for DAU 10, denoted by $p_{10,0}$, is 2.51%, with mean and median 3.58% and 3.41%, respectively, and a corresponding credible interval of (1.68%,5.63%) from the Markov chain sample. The initial prevalences for DAUs 4 and 5 are given in chapter 4. These results may seem somewhat high for an initial prevalence value, but it must be considered that they represent prevalence in 1975, the year before the first data values. Chronic Wasting Disease had first been diagnosed near Fort Collins, Colorado in 1967 [22], and the disease had eight years to propagate before the first data record.

Thus a value of 3.41% is not unreasonable, especially considering the proximity of Fort Collins with DAUs 10, 4, and 5. The prior standard deviation, 0.0298, is larger than the posterior standard deviation, 0.0238, indicating that the data is improving the knowledge of the parameter δ.

The main prediction of the model, in terms of the future evolution of Chronic Wasting Disease, is that, unless drastic and effective control measures are taken, the disease will continue to march across Colorado. This model corroborates recent research that indicates current control strategies have been ineffective [69, 14]. This could potentially have a devastating impact on hunting and tourism in the state.

5.2 Model Validation

The problem of recovering the parameters for a differential equation is in general ill-posed. For most differential equations, small changes in the data can lead to large changes in the parameter estimates. Theorems and techniques for inverse problems are very specialized and applicable only to very specific problems [83]. Consequently, while the modeling procedure outlined in chapter 3 appears to have worked for the specific CWD model and data, it is unlikely that it would be applicable for a more general case. Nevertheless, the effectiveness of the method can be checked by generating data for a simpler variation of the model, and determining whether or not the parameters can be recaptured. The MCMC method introduced in chapter 4 can be used to simulate the posterior distributions for these simpler versions of the CWD model.

A simulation was performed on nine variations of the CWD model. The spatial structure was a simple three by three rectangular grid, with nine test

DAU's, numbered one through nine in the usual left to right, top to bottom order. Regions were considered neighbors if they shared an edge. The values of α and γ were fixed, with $\alpha = 0.10$ and $\gamma = 0.01$. Three sets of initial conditions were chosen, the first $p_{1,0} = 0.05$, the second $p_{1,0} = 0.05$ and $p_{2,0} = 0.01$, and the third $p_{1,0} = 0.01$. There were also three values of the parameter δ for each initial condition, 0.05, 0.15, and 0.50. These combine to yield nine test cases. These nine sets of initial conditions were used to deterministically generate twenty years of prevalence values for each DAU. The number tested, N, was fixed in each region at 15 for years one to four, 100 for years five to fifteen, and 1000 for years sixteen to twenty. These prevalence values and the values for N were then used to generate binomial data for each year and each region. This data was then used with the same likelihood form and prior distributions that were presented in chapter 3 to construct a posterior distribution. The prior distributions were kept the same throughout all 450 simulations for convenience.

For each combination of initial conditions, fifty pseudo-data sets were generated. For each individual pseudo-data set, algorithm 4.4 was used to simulate $200,000$ iterations of a Markov chain. The fourth group of simulations (see section B.4) was used to test the code, and an example from this group was run for $200,000$ iterations. The graphical evidence from this test indicated that approximate stationarity was reached within $50,000$ iterations. Consequently, the first $50,000$ iterations were used as the burn-in period for each of the 450 simulations. The fourth group of simulations successfully captured the true parameter values, and will be discussed below.

Algorithm 4.4 requires the value of the log-posterior density at the mode, and for the CWD implementation, several minutes had been spent numerically maximizing the log-posterior density to obtain a good approximation for the mode. With nine model variations and fifty simulations, there were a total of 450 different posterior functions that would need to be maximized, which was impractical. Instead, for each of the nine model variations, one of the fifty data sets was used to numerically maximize the resulting log-posterior density. This was thought to be a reasonable compromise strategy since the characteristics of the posterior density within a group were expected to be similar, although the ill-posed nature of the inverse problem suggests this may not be the case.

The use of algorithm 4.4 for these simulations also requires careful choice of the tuning parameters and proposal variance (depending on the simulation group) to appropriately scale the step size, which in turn requires some empirical exploration of the log posterior at several points. This process is not possible for each of the 450 simulations, so the values for the tuning parameters were kept the same throughout the simulations, and were fixed as the same values used in the CWD implementation in chapter 4, and given in appendix C, section C.5. However, the inability to correctly tune the algorithm for each simulation and the lack of a good approximation to the mode will negatively affect the convergence of the chain. If a poor approximation of the mode is used in the MALTS algorithm, the dynamic step size 4.13 can bias the proposal in the wrong direction. Let $\tilde{x}$ be a poor approximation to the mode of the target density f, and suppose the current state of the chain is $X^{(t)}$ is closer to the true mode, than $\log(f(\tilde{x})) - \log(f(X^{(t)})) + 1$ may be negative. This would have the effect

of pulling the chain in the wrong direction, away from the areas of higher probability. A corrective measure to this potential problem, which was not applied here, is to update the mode approximation at every iteration by checking if $\log(f(\tilde{x})) < \log(f(X^{(t)}))$, and setting $\tilde{x} = X^{(t)}$ as the new approximation to the mode if this is true.

Most of the results are listed in appendix B. The results that are listed here are the groups with the best and worst performances. The method was a complete success for one group of parameters, and a complete failure for one group. The method had varying degrees of success on the remaining seven groups of simulations, capturing at least some of the true parameter values. All of the tables show the lower and upper limits of the 95% Bayesian credible intervals for each parameter averaged over the fifty simulations. The tables also show the mean, median, and standard deviations for each parameter, averaged over all fifty runs. Most of the simulation groups met with mixed success. The results ranged from one parameter captured, to all but one captured. The results of the simulations are given in appendix B.

Table 5.1 shows the results of the third group of simulations. The true parameters for this group were $p_{1,0} = 0.05$, $\alpha = 0.10$, $\delta = 0.50$, and $\gamma = 0.01$. This time, none of the parameters were captured by the credible intervals. This group of simulations had the poorest performance, although the true value of the parameter $p_{1,0}$ lay just outside the calculated credible interval. The poor performance is likely due to the prior distributions. In particular, the prior for δ, with mean 0.0889, is an informative prior centered far from true value of $\delta = 0.5$.

Parameter	L.Limit	U.Limit	Avg. Mean	Avg. Median	Std.Dev.
$p_{1,0} = 0.05$	0.0606	0.0797	0.0772	0.0791	0.0057
$\alpha = 0.10$	0.1374	0.2453	0.2335	0.2362	0.0425
$\delta = 0.50$	0.1243	0.1898	0.1854	0.1873	0.0314
$\gamma = 0.10$	0.0305	0.0411	0.0407	0.0425	0.0037

Table 5.1: Results of the group 3 simulation, with $p_{1,0} = 0.05$, $\alpha = 0.10$, $\delta = 0.50$, and $\gamma = 0.01$. Values are the average result of 50 pseudo data sets.

The results of the fourth group of simulations, with true parameters $p_{1,0} = 0.05$, $p_{2,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.05$, and $\gamma = 0.01$, are shown below in table 5.2. The results are a clear success for this group of models, since all of the parameters are captured by the credible intervals, although the credible interval for α is quite large. This group of simulations is more similar to the true data than the previous groups, both in terms of the number of parameters and the values for the parameters. The true values of the parameters are close to the values of the CWD posterior shown in table 4.2, which may account for the success of the method. Moreover, this group of simulations was used to determine the burn-in period.

The results of these simulations point to a number of difficulties with the method discussed in chapter 3. There are a number of constraints that were imposed to insure that the deterministic update equation 3.5 produced prevalence values on the unit interval. Additionally, algorithm 4.4, introduced in

Parameter	L.Limit	U.Limit	Avg. Mean	Avg. Median	Std.Dev.
$p_{1,0} = 0.05$	0.0210	0.0515	0.0350	0.0347	0.0067
$p_{2,0} = 0.01$	0.0035	0.0106	0.0066	0.0064	0.0021
$\alpha = 0.10$	0.0462	0.4388	0.1661	0.1482	0.0911
$\delta = 0.05$	0.0477	0.0992	0.0773	0.0735	0.0228
$\gamma = 0.05$	0.0064	0.0131	0.0093	0.0093	0.0014

Table 5.2: Results of the group 4 simulation, with $p_{1,0} = 0.05$, $p_{2,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.05$, and $\gamma = 0.01$. All of the true parameter values are captured by the credible intervals. Values are averaged over 50 pseudo data sets.

chapter 4, has several characteristics that make it difficult to apply. First, the tuning parameter k and proposal variance Σ must be chosen carefully to achieve convergence for a given posterior distribution. These parameters were not adjusted for each new posterior distribution, or even within a particular group of runs. Because the dynamic step size is dependent upon the mode, and because the tuning parameters are chosen to compensate for large values of 4.13, moderate changes in the model could potentially have a drastic impact on the performance of the MALTS algorithm. Changes made to the model that result in moderate or large changes in the value of the log posterior density at the mode will require significant re-tuning of algorithm 4.4.

The changes made to the model for the simulations discussed in this section almost certainly had a negative impact on the convergence of the chains to the stationary distribution. For each of the 450 simulations, a cursory statistic was used as a rough convergence measure, the difference in the means for each parameter between the first and last half of the chain. For some of the runs,

this difference is greater than a standard deviation, indicating that these runs probably did not reach the stationary distribution, calling the results obtained by the simulations into question. The CWD simulation in chapter 4 had a distinct advantage that these simulations did not, namely, that the characteristics and shape of the posterior density were known ahead of time.

Perhaps the most serious difficulty is the strong interrelationships among the parameters in equation 3.5, especially between α and δ. Some of this interdependency comes from the restrictions on the parameters imposed to guarantee that the difference equation 3.5 always produces prevalence values in $(0, 1)$, and from biological considerations through the data, as discussed in chapter 3. It is likely that some combinations of parameter values in these simulations were unrealistic and hence impossible to capture.

Another source of complications were the prior distributions that were used in both the simulations and the analysis of the Chronic Wasting Disease data. Some of these prior distributions were informative, based on prior information about the large scale behaviour of the disease. The sensitivity of the model to these prior distributions is unknown. However, the simulations in groups six and nine, with $\delta = 0.50$, were able to capture the true value within the credible interval for the this parameter. The prior distribution is informative with a mean of 0.09, which is far from the true value of δ. The value $\delta = 0.50$ was chosen partly to determine if the prior distribution for this parameter had too great an effect on the posterior, and it was expected that the method would fail to capture the true value for all of these instances. Surprisingly, the method was still able to recover the true value for δ in two of the three cases with $\delta = 0.50$,

in spite of the misguided prior distribution.

The results of the nine groups of tests show that the modeling approach used for Chronic Wasting Disease in chapter 3 has limitations that make it difficult to apply in a more general setting. Still, the success of the method for the fourth group of models, shown in table 5.2, indicates that the method is effective in some situations. Moreover, the interpretation of the posterior estimates in section 5.1 gives results that are in-line with experts predictions for CWD [69, 14, 22].

6. Discussion

Chronic Wasting Disease presents a large number of possible avenues of research, especially in biological areas, many of which are being pursued [31, 32]. However, they are outside the mathematical and statistical arena, and are not directly related to this thesis. There are several avenues for future research springing from this work, namely, improvements for the model including the incorporation of covariate information, using the model to examine different strategies for controlling the spread of the disease, a more complete exploration of the methodology introduced by the model, and a detailed analysis of the MALTS method used for the MCMC posterior exploration.

Prevalence is an important aspect of studying CWD, but it is not the only one. Of equal importance in understanding the disease is knowing how the number of deer that are in a given region affects the ability of the disease to "burn out," coexist, or kill off the host. The model in chapter 3 does not explicitly describe the interplay between total population size and prevalence, but there are many differential equations models designed for such interactions. Models that include the interaction of population size with prevalence are beyond the scope of this thesis, but certainly are avenues worth pursuing. The exact nature of this relationship is an important avenue of future work. This lack notwithstanding, the model presented here does give some evidence that the disease and the deer population can coexist.

Another possible avenue of future research is to determine the impact different control strategies imposed by a state wildlife division might have on the posterior distribution of δ, and whether the controls could be used to lower the long run prevalence to a level close to zero. The Bayesian hierarchy allows for different schemes to be tested via computational simulation prior to implementation, which has been done elsewhere [37]. In particular, the sampling designs in [37] suggest that the CDOW should increase testing of deer harvested in non-endemic areas to gain the most information about the transmission of the disease. This corresponds with conventional wisdom that one learns more by asking questions for which the answer is unknown than by asking questions for which the answer is known. Part of the effect may be due to the binomial assumption and the relationship between the probability of infection (prevalence) and variance of observations. Namely, the variance of binomial distributions is smaller for distributions with smaller probabilities of "success".

The model can be modified to include added information as more is learned about the disease. In particular, information on how the disease is spread could be used to improve the disease migration effects modeled by elements of the matrix W in equation 3.5. This could be perhaps be done through the use of covariate information. For example, the parameter γ could be dependent on DAU, weather, the total deer population, and other ecological or environmental factors. These could be included in the expression for γ, so that, for example:

$$\gamma_t = \mathbf{X}_t \boldsymbol{\beta} \tag{6.1}$$

where $\mathbf{X}_t$ is a vector of covariate variables and $\boldsymbol{\beta}$ is a vector of regression parameters. Additional information through covariates would give more information

on the migration effects. This change in the form of γ could affect the structure of the matrix W.

Changing W by adding covariate information may have some effect on the estimation of δ because of the high correlation between parameters, but these differences are likely to be quantitative rather than qualitative. Likewise, as more is learned about the transmission mechanism, modifications to the form of acceleration, currently, $\alpha(\delta\mathbf{1} - \mathbf{p})$, can be considered to reflect that knowledge. In particular, it is possible to let the acceleration parameter, α, depend on DAU and covariates, such as deer density, range type and condition, and incorporated into the model via an autologistic model for the level of α as a function of the covariates. Recent efforts by the Colorado Division of Wildlife, such as the recent moratorium on transport of domesticated deer and elk from ranches or the increase in hunting season lengths and number of licenses issued in endemic areas, may have an impact on the spread of the disease. The effects of these kinds of policies could also be incorporated into the model through modifications to the acceleration parameter α.

The Chronic Wasting Disease model in chapter 3 has some similarities with random coefficient autoregressive models (RCAR) [23, 1, 40]. There are, however, fundamental differences. RCAR models were introduced as a tool for handling the nonlinear features commonly seen in real-life data, without resorting to a nonlinear model. This strategy is not unreasonable, given that there seems to be a closer correspondence between linear stochastic models and nonlinear deterministic models than between nonlinear deterministic models and nonlinear stochastic models [51].

The random coefficient autoregressive model of order p, denoted by $RCAR(p)$ is given by:

$$X_t = \sum_{k=1}^{p} (a_k + u_{k;t}) X_{t-k} + \epsilon_t \tag{6.2}$$

where $t \in \mathbb{Z}$, $(a_1, ..., a_p)^T$ is a vector of autoregressive parameters, $\{\mathbf{u}_t = (u_{1;t}, ..., u_{p;t})^T; t \in \mathbb{Z}\}$ is an unobservable zero mean i.i.d. sequence of p-dimensional random variables, mutually independent of $\{\boldsymbol{\epsilon}_t = \epsilon_t; t \in \mathbb{Z}\}$, another unobservable sequence of zero mean i.i.d. random variables. When $\mathbf{u}_t$ is almost surely zero, the $RCAR(p)$ reduces to the ordinary autoregressive model, $AR(p)$.

Random coefficient autoregressive models still possess a linear form, even though they were introduced to describe nonlinear data [1, 40]. The strong adherence to linear models is perhaps due to the desire to avoid the difficulties associated with non-Gaussian noise terms, since many $RCAR(p)$ models have asymptotic normality properties. The property of normality gives rise to estimators and a variety of parametric and semiparametric methods [1]. Many of these methods still rely on a Gaussian noise term, at least in part, which is frequently an inappropriate assumption for nonlinear data [1, 40].

The CWD model in chapter 3 is fundamentally different in that deterministic updates are used for prevalence. That is, given the values of the parameters α, δ, γ, and the initial prevalences $\mathbf{p}_0$, equation 3.5 generates the prevalence value for each DAU-year in a deterministic manner. These deterministic updates for prevalence are quite strict and may not accurately reflect the true nature of disease spread, but they do rest on a firm mathematical foundation. The leap of faith occurs when the error term in the update equation is neglected.

This is not to say that the process is truly deterministic. Rather, all of the uncertainty in the model is pushed into the likelihood and the distributions on the parameters. This line of thinking has been used in stochastic processes [41]. However, it is not common in a statistical setting. The greatest benefit of using deterministic rather than stochastic updates is that deterministic updates reduces the dimension of the parameter space. Rather than estimating a density function for prevalence at each DAU-time period combination, only the density on the prior for the initial conditions, $\mathbf{p}_0$, and the parameters is needed. This was an important consideration considering the limited amount of data that was available.

Another critical difference is that the model is inherently nonlinear. Constructing a suitable differential equation model leads to a nonlinear difference equation for the updates. Rather than beating the model into a linear form, such as in the $RCAR$ models, the nonlinear model is retained. This nonlinearity makes the details of the model difficult, and in particular is likely to be the primary reason for the high interdependencies amongst the parameters. The nonlinearity arises naturally from the differential equations, and it is necessary to provide the desired qualitative behaviour, such as the bounded growth of the sigmoidal curve. Nonlinearity also enables the direct modeling of the spatial dynamics of the disease.

The method used for describing the spatial dynamic is drastically different from the common approach. In most statistical epidemic models, spatial effects are modeled as dependencies in the data through covariance terms. With binomial count data, this is often done via an autologistic model with an attendant

transformation of the parameter space to achieve normality. The CWD model is different and more direct in that spatial effects are modeled as rates of movement, just as they are in the differential equations that are used to model the prevalence. One of the key questions that biologists ask, i.e. how fast is the disease moving across the state, is modeled directly by the parameter γ, which can be interpreted as the average proportion of infected deer that migrate from DAU_i to DAU_j. This approach flows directly from the interpretation of parameters in the differential equations. Fitting the difference equations is in essence fitting a first order approximation of the spatial differential equation model to the data. Conditional autoregressive models for diseases such as those presented in chapter 2 rely on both large sample sizes and small infection rates to apply a Poisson approximation to the Binomial distribution. This in turn allows for the use of transformations to achieve normality, where the spatial effects are again modeled through the covariance in a Gaussian multivariate autoregressive process. Some spatial structure can be built into the coefficient matrix for the transformed log-relative risk autoregressive process, but again, this approach is indirect. The results of the posterior analysis discussed in section 5.1 and the model validation results in section 5.2 show that building the spatial mixing term directly into the update equation is an effective way of describing the spatial dynamics.

The deterministic updates are a simplification, but they do provide the benefit of parsimony. Moreover, the model attempts to directly describes the dynamics. The resulting model, while perhaps not a detailed description of the spatial and temporal process, does possess the desired qualitative behaviour.

The model provides some predictive capability that can be checked against future observations. The use of deterministic updates seems to contradict the approach of statisticians since 1777. Consequently, the lack of a noise term is a difficult pill to swallow, especially in a model of a biological process. However, the simulations conducted in section 5.2 indicate that the estimation process is effective in some situations. More data would allow for the addition of a noise term to the update equation 3.5, which may make the model more flexible. Exploring the connection between the model presented here and random coefficient autoregressive models may be the way to do this.

The posterior in the CWD model possesses a number of characteristics that make it difficult to simulate. In general, a practical solution to a problem should be accurate, expedient, and flexible. Algorithm 4.4 used for the Markov chain simulation for this model was designed with these qualities in mind to deal with the specific difficulties of the complex parameter space in the CWD model. Sampling from multivariate distributions is in general difficult, especially when the target density is bounded and contains steep ridges of high probability surrounded by a relatively large area of low probability.

The MALTS algorithm is not the only possible method of simulating the CWD posterior. Applying a transformation to the model to reduce the correlation and to obtain an unbounded parameter space in a common approach to mitigate the difficulties presented by the CWD posterior. However, identifying an appropriate transformation is not necessarily obvious, and none were attempted here. Additionally, the failures of the methods discussed in section 4.3.1 to converge to stationarity within $300,000$ iterations do not mean that these methods

are inappropriate. In particular, the standard random walk will converge to the correct stationary distribution, given enough time. A straightforward solution to the simulation difficulties would be to simply run the standard random walk for an extended period of time. However, while this approach would be accurate, it would not be expedient, nor is it a flexible solution, as any new simulations would require an extremely large number of iterations to produce a usable sample. Changes in the model that do not drastically alter the value of the log posterior density function at the mode will not require retuning of the MALTS algorithm, which can produce a large number of simulated draws from the posterior in a relatively short amount of time. The standard random walk, on the other hand, would still require a large amount of time to produce a usable sample, and could require retuning.

An empirical comparison of the performance of standard random walk Metropolis-Hastings algorithms to Langevin type algorithms, which use derivative information to suppress the random walk behaviour of the Markov chain, is performed in [28], and shows that the Langevin algorithms perform better on average than the standard random walk. Langevin algorithms often require fewer burn-in iterations and thus converge to the stationary distribution faster. Other researchers have examined Langevin algorithms analytically, and reached the same conclusion [66, 71]. With this in mind, it was expected that a Langevin algorithm to perform better for the CWD model than a random walk. However, the methods studied by [28], [66] and [71] do not consider models with bounded parameter space. The difficulties in sampling from such a space in higher dimensions is perhaps one of the reasons so many models make use of transformations

to achieve normality, and with it an unbounded parameter space. Such transformations may have been useful here, but were not attempted. Additionally, the model was derived from a system of differential equations, and it was the parameters of these equations that were of direct interest. The inability of the standard random walk and the Langevin algorithms to quickly produce a usable sample from the CWD posterior prompted creation of a new technique (algorithm 4.4).

The difficulties of the target distribution 3.11 were overcome by adapting the qualities, and not the details, of several MCMC methods. From the Metropolis Adjusted Langevin Truncated Algorithm (MALTA) the use of the gradient to direct the proposal and suppress the random walk was taken [71, 5, 25]. The idea of truncating the proposal to control the instability in the gradient commonly seen in Langevin algorithms for a non-normal target density was also used. The step was truncated without losing the information in the gradient by finding the unit vector with the same direction. This allowed for more precise control over the step size than the use of a single dicretization parameter as in the MALA.

Methods for choosing the appropriate step size for Langevin algorithms have received a lot of attention [28, 66, 9, 65]. For the posterior distribution in the CWD model, there is a sharp ridge of high probability surrounded by areas of low probability. This is especially noticeable in figure 3.5. Too large a step, and the high probability area is missed, too small, and oversampling occurs in low probability areas. Rather than seeking a single optimal step size in this difficult situation, a more dynamic approach is chosen, inspired by recent work on tempered Langevin algorithms. Tempered Langevin diffusions are examined

in [67], in which both the step size and the variance of the proposal is adjusted by the value of the density function at the current value in the chain. The form used in [67] was chosen both to ensure convergence of the continuous time process and to achieve a generalization of the Langevin diffusion, and it is a bit cumbersome for practical use.

A dynamic step size effect similar to that of the tempered Langevin diffusion is achieved by adjusting the value of the step size with respect to the mode. The form used in algorithm 4.4 is bounded below by one, allowing the minimum step size for each parameter being simulated to be chosen through the use of tuning parameters. Unlike the tempered diffusion, the variance is not adjusted at each step. In addition to a varying step size, a diagonal matrix is used rather than an identity matrix in the proposal covariance, allowing the variance of the proposal to be scaled differently for each parameter. This addition was useful because the parameters in the target distribution differ in their standard deviations by up to an order of magnitude.

The experiments performed in section 4.3.4 show that, for a bounded target density, the MALTS algorithm 4.4 can be as or more effective than the corresponding standard random walk or MALA. Algorithm 4.4 has a number of tuning parameters, which could make it difficult to implement. Guidelines are provided in this thesis for choosing values of the tuning parameter and proposal covariance, but a more rigorous method for choosing the optimal values for these tuning parameters is necessary. Results for choosing optimal discretization sizes for the Metropolis Adjusted Langevin Algorithm for uncorrelated parameters are given in [66]. Applying these results to the algorithm introduced in this

thesis, and to correlated parameters, would be helpful in making algorithm 4.4 easier to apply. Clearly, poor choices for the tuning parameters lead to poor or uncertain results, as shown in section 5.2.

The efficiency of algorithm 4.4 was assessed empirically, and for the CWD model was judged a success in part because the run time was significantly shorter than for the acceptance sampler. Use of algorithm 4.4 to simulate a simple test case demonstrated that the method could correctly sample the target distribution, a truncated bivariate normal. A structured comparison of the performance of the MALTS algorithm to the standard random walk and Langevin algorithms on several test cases showed that the convergence and mixing properties of the new method were comparable to those of the other methods. Calculation of the potential scale reduction showed the MALTS algorithm converges at least as fast as the standard random walk and MALA. The acceptance rates for the bounded densities indicate a noticeable improvement in the mixing speed of the MALTS algorithm over the MALA for a fixed step size.

The convergence analysis and comparison of the standard random walk and algorithm 4.4 on the CWD posterior distribution showed dramatic differences. Although the standard random walk had a shorter run time than algorithm 4.4, both the graphical evidence and the potential scale reduction showed the failure of the standard random walk to reach the stationary distribution after $300,000$ iterations. By contrast, at the same proposal variance of 1×10^{-9}, the MALTS algorithm quickly and successfully converged to the stationary distribution, and is likely to be useful in other situations where the distribution is highly correlated and the gradient experiences large changes. Statistical comparison of the pos-

terior sample obtained using the new Markov chain method with that obtained by the acceptance sampler indicated that the marginal distributions of the two samples were the same. The new Markov chain simulation algorithm was a clear success for CWD posterior distribution. A more detailed theoretical analysis of the algorithm's performance is necessary and is reserved for future work. In particular, [67] introduce the notion of *exponential ergodicity* for the tempered Langevin diffusion. Whether algorithm 4.4 has this property is unknown.

Blocking– updating highly correlated parameters together– is a generally accepted strategy of accelerating the convergence of a multi-dimensional Markov chain when the component variables are highly correlated, and seems to have worked for the CWD example where the joint distribution has a total of seven parameters. Because the seventh, π, is independent of the other six, it is sampled separately. The remaining six parameters show a high level of interdependence, so they are updated together. However, if the parameters α, δ, or γ are allowed to vary by DAU, a simple blocking strategy may not work, and more sophisticated methods of exploiting the spatial structure may be needed. The structured MCMC approach in [29], which exploits the spatial structure inherent in the model and the data, is one possible avenue that could be explored. If the spatial structure has the properties of a Markov Random Field then the structured approach can be especially effective [4].

Perhaps the most important area for further study is the precise relationship between the differential equations used in the model and the distributions in the statistical hierarchy. Previous work has indicated that some of the properties of the differential equations cannot be overcome by data, so that if the ODEs are

inappropriate, then attempting to fit a model will be a failure [35]. The results in section 5.2 indicate that the relationship between the prior, data, statistical hierarchy, and differential equations is complex. A detailed understanding of this relationship is necessary to fully explore this method.

7. Conclusion

This thesis introduced a spatial statistical model for chronic wasting disease that was derived from and directly incorporated a system of differential equations. Importantly, the spatial dynamics were not modeled through instantaneous correlations in the data, but as rates of movement in the differential equations. Thus the model provided a more direct answer for the question of "how fast is the disease spreading?" in the form of the matrix W in equation 3.5 and the parameter γ. The use of deterministic updates, rather than stochastic, reduced the number of parameters in the model, and allowed for better use of the limited amount of data that was available. The use of deterministic updates is different from the standard statistical way of thinking, but appears to be effective in this setting. Model validation calculations with simulated data indicated that the approach can, in some settings, successfully recover the true parameter values. The conditions in which the use of deterministic updates in a statistical model is appropriate need to be fully explored.

The Chronic Wasting Disease model posterior distribution had a number of qualities that make it difficult to sample. An acceptance sampler was used previously to generate a random sample from the posterior distribution [37], but it was both slow and difficult to modify. Using existing Markov chain simulation methods based on stochastic dynamic processes as a starting point, a strategy was developed and a simulation approach designed that has similarities with tempered Langevin algorithms, and was effective for simulating the posterior

despite the restrictions of a bounded parameter space and ridges of high probability. A series of simulation experiments indicate that the new method, the MALTS algorithm, is competitive with the MALA and standard random walk. The MALTS algorithm proved to be significantly faster than the acceptance sampler, and was successful in simulating the CWD density, while attempts with a standard random walk failed to converge within a large number of iterations. The convergence and mixing properties of the MALTS algorithm were examined empirically, but its theoretical properties need to be rigorously studied.

In summary, this thesis has introduced a spatial statistical model for Chronic Wasting Disease, and a new Markov chain simulation method was designed to overcome the difficulties of the posterior distribution produced by that model. The new simulation method provided an accurate, expedient, and flexible method of producing a sample from the posterior density, and the posterior parameter estimates were used to obtain predictions about Chronic Wasting Disease. The modeling methodology was tested on simulated data, with mixed results, indicating that the method used here has limitations. However, the Markov chain simulation technique introduced here will allow the methodology of the CWD model to be applied to other disease modeling problems. It is hoped that the simulation method introduced here overcomes some of the problems of other Langevin type algorithms, and will be an effective tool for simulating target densities with bounded parameter spaces, as well as those with erratic gradient behaviour.

Appendix A. Acceptance Rates for Simulation Comparisons

This appendix contains the acceptance rate results of the experiments conducted on the truncated normal distributions and multivariate normal distributions discussed in section 4.3.4.

A.1 Acceptance Rates for the Truncated Normal Simulations

The standard random walk, Metropolis adjusted Langevin algorithm (MALA), and Modified adjusted Langevin algorithm with Tempered step size (MALTS) were used to simulate multivariate normal distributions of increasing dimension, truncated to the unit cube. The dimensions were 2, 6, 12, and 36. The step size was fixed for each method at 1×10^{-5}.

Method	Dim=2	Dim=6	Dim=12	Dim=36
Std. RW	0.699	0.699	0.699	0.698
MALA	0.661	0.659	0.660	0.658
MALTS	0.700	0.700	0.701	0.697

Table A.1: Acceptance rates for the multivariate truncated normal distribution.

A.2 Acceptance Rates for the Multivariate Normal Simulations

The standard random walk, Metropolis adjusted Langevin algorithm (MALA), and Modified adjusted Langevin algorithm with Tempered step size (MALTS)

were used to simulate multivariate normal distributions of increasing dimension and correlation. The dimensions were 2, 6, 12, and 36, and the levels of correlation were 0.879, 0.970, and 0.992. Additional simulations were run at correlation 0.992 with dimensions 64, 144, 200, and 300. The step size was fixed for each method at 1×10^{-5}, 1×10^{-4}, and 1×10^{-2}.

Method	Dim=2	Dim=6	Dim=12	Dim=36
Std. RW	0.967	0.969	0.969	0.968
MALA	0.959	0.959	0.958	0.958
MALTS	0.955	0.955	0.956	0.956

Table A.2: Acceptance rates for each dimension for the multivariate normal distribution with correlation $\rho = 0.879$. The step size is 1×10^{-5}.

Method	Dim=2	Dim=6	Dim=12	Dim=36
Std. RW	0.904	0.902	0.902	0.903
MALA	0.871	0.871	0.871	0.871
MALTS	0.873	0.872	0.872	0.873

Table A.3: Acceptance rates for each dimension for the multivariate normal distribution with correlation $\rho = 0.879$. The step size is 1×10^{-4}.

Method	Dim=2	Dim=6	Dim=12	Dim=36
Std. RW	0.809	0.809	0.808	0.809
MALA	0.758	0.758	0.759	0.756
MALTS	0.775	0.774	0.773	0.774

Table A.4: Acceptance rates for each dimension for the multivariate normal distribution with correlation $\rho = 0.879$. The step size is 1×10^{-2}.

Method	Dim=2	Dim=6	Dim=12	Dim=36
Std. RW	0.997	0.969	0.969	0.969
MALA	0.995	0.958	0.958	0.959
MALTS	0.995	0.956	0.958	0.958

Table A.5: Acceptance rates for each dimension for the multivariate normal distribution with correlation $\rho = 0.970$. The step size is 1×10^{-5}.

Method	Dim=2	Dim=6	Dim=12	Dim=36
Std. RW	0.902	0.901	0.901	0.902
MALA	0.870	0.871	0.871	0.870
MALTS	0.872	0.873	0.872	0.873

Table A.6: Acceptance rates for each dimension for the multivariate normal distribution with correlation $\rho = 0.970$. The step size is 1×10^{-4}.

Method	Dim=2	Dim=6	Dim=12	Dim=36
Std. RW	0.809	0.807	0.809	0.808
MALA	0.756	0.758	0.758	0.757
MALTS	0.773	0.774	0.774	0.774

Table A.7: Acceptance rates for each dimension for the multivariate normal distribution with correlation $\rho = 0.970$. The step size is 1×10^{-2}.

Method	Dim=2	Dim=6	Dim=12	Dim=36
Std. RW	0.969	0.969	0.968	0.969
MALA	0.956	0.959	0.958	0.959
MALTS	0.957	0.957	0.955	0.956

Table A.8: Acceptance rates for each dimension for the multivariate normal distribution with correlation $\rho = 0.992$. The steps size is 1×10^{-5}.

Method	Dim=64	Dim=144	Dim=200	Dim=300
Std. RW	0.887	0.880	0.878	0.861
MALA	0.851	0.851	0.851	0.851
MALTS	0.880	0.882	0.881	0.881

Table A.9: Acceptance rates for each dimension for the multivariate normal distribution with correlation $\rho = 0.992$. The steps size is 1×10^{-5}.

Method	Dim=2	Dim=6	Dim=12	Dim=36
Std. RW	0.903	0.902	0.901	0.902
MALA	0.873	0.871	0.871	0.871
MALTS	0.872	0.873	0.873	0.873

Table A.10: Acceptance rates for each dimension for the multivariate normal distribution with correlation $\rho = 0.992$. The step size is 1×10^{-4}.

Method	Dim=2	Dim=6	Dim=12	Dim=36
Std. RW	0.810	0.808	0.808	0.808
MALA	0.759	0.758	0.758	0.757
MALTS	0.774	0.774	0.774	0.774

Table A.11: Acceptance rates for each dimension for the multivariate normal distribution with correlation $\rho = 0.992$. The step size is 1×10^{-2}.

Appendix B. Tables for the Model Validation Simulations

This appendix contains the tables for the remaining model validation simulations discussed in section 5.2. The tables for groups three and four are in section 5.2.

B.1 Group One Simulation Results

The group one simulation had the true parameter values $p_{1,0} = 0.05$, $\alpha = 0.10$, $\delta = 0.05$, and $\gamma = 0.01$.

Parameter	Lower Limit	Upper Limit	Mean	Median	StdDev
$p_{1,0}$	0.0273	0.0320	0.0305	0.0305	0.0005
α	0.0737	0.0825	0.0805	0.0804	0.0012
δ	0.0966	0.1025	0.1003	0.1003	0.0006
γ	0.0075	0.0108	0.0100	0.0100	0.0003

Table B.1: Results of the group 1 simulation, with $p_{1,0} = 0.05$, $\alpha = 0.10$, $\delta = 0.05$, and $\gamma = 0.01$.

B.2 Group Two Simulation Results

The second group of simulations had true parameters values $p_{1,0} = 0.05$, $\alpha = 0.10$, $\delta = 0.15$, and $\gamma = 0.01$.

Parameter	Lower Limit	Upper Limit	Mean	Median	StdDev
$p_{1,0}$	0.0311	0.0370	0.0352	0.0352	0.0016
α	0.0783	0.0871	0.0860	0.0868	0.0023
δ	0.0987	0.1033	0.1026	0.1026	0.0010
γ	0.0088	0.0107	0.0104	0.0104	0.0003

Table B.2: Results of the group 2 simulation, with $p_{1,0} = 0.05$, $\alpha = 0.10$, $\delta = 0.15$, and $\gamma = 0.01$.

B.3 Group Three Simulation Results

The true parameter values for the third group of simulations were $p_{1,0} = 0.05$, $\alpha = 0.10$, $\delta = 0.50$, and $\gamma = 0.01$. This table (5.1) is shown in section 5.2.

B.4 Group Four Simulation Results

The fourth group of simulations used the true parameters $p_{1,0} = 0.05$, $p_{2,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.05$, and $\gamma = 0.01$. This table (5.2) is shown in section 5.2.

B.5 Group Five Simulation Results

The group 5 true parameter values are $p_{1,0} = 0.05$, $p_{2,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.15$, and $\gamma = 0.01$.

B.6 Group Six Simulation Results

The group 6 simulations used the true parameter values are $p_{1,0} = 0.05$, $p_{2,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.50$, and $\gamma = 0.01$.

B.7 Group Seven Simulation Results

Parameter	Lower Limit	Upper Limit	Mean	Median	StdDev
$p_{1,0}$	0.0205	0.0521	0.0360	0.0355	0.0072
$p_{2,0}$	0.0033	0.0091	0.0059	0.0058	0.0016
α	0.1271	0.3090	0.2075	0.2032	0.0572
δ	0.0907	0.1453	0.1209	0.1183	0.0179
γ	0.0061	0.0100	0.0081	0.0081	0.0010

Table B.3: Results of the group 5 simulation, with $p_{1,0} = 0.05$, $p_{2,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.15$, and $\gamma = 0.01$.

Parameter	Lower Limit	Upper Limit	Mean	Median	StdDev
$p_{1,0}$	0.0173	0.0300	0.0244	0.0242	0.0034
$p_{2,0}$	0.0040	0.0064	0.0053	0.0052	0.0007
α	0.0976	0.1128	0.1048	0.1046	0.0040
δ	0.4894	0.5048	0.4964	0.4964	0.0041
γ	0.0081	0.0100	0.0090	0.0090	0.0005

Table B.4: Results of the group 6 simulation, with $p_{1,0} = 0.05$, $p_{2,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.50$, and $\gamma = 0.01$.

The group 7 simulations used the true parameter values $p_{1,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.05$, and $\gamma = 0.01$.

B.8 Group Eight Simulation Results

The group 8 simulations have the true values for the parameters were $p_{1,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.15$, and $\gamma = 0.01$.

Parameter	Lower Limit	Upper Limit	Mean	Median	StdDev
$p_{1,0}$	0.0026	0.0107	0.0072	0.0071	0.0025
α	0.0443	0.3985	0.1500	0.1320	0.0971
δ	0.0522	0.0888	0.0759	0.0725	0.0264
γ	0.0055	0.0143	0.0097	0.0095	0.0022

Table B.5: Results of the group 7 simulation, with $p_{1,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.05$, and $\gamma = 0.01$.

Parameter	Lower Limit	Upper Limit	Mean	Median	StdDev
$p_{1,0}$	0.0020	0.0107	0.0087	0.0082	0.0033
α	0.0664	0.6720	0.3521	0.3380	0.1449
δ	0.0418	0.1024	0.0762	0.0720	0.0231
γ	0.0061	0.0121	0.0086	0.0085	0.0014

Table B.6: Results of the group 8 simulation, with $p_{1,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.15$, and $\gamma = 0.01$.

B.9 Group Nine Simulation Results

The true parameter values for the group 9 simulations are $p_{1,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.50$, and $\gamma = 0.01$.

Parameter	Lower Limit	Upper Limit	Mean	Median	StdDev
$p_{1,0}$	0.0039	0.0076	0.0056	0.0056	0.0009
α	0.0971	0.1156	0.1050	0.1046	0.0046
δ	0.4836	0.5069	0.4965	0.4945	0.0056
γ	0.0081	0.0100	0.0091	0.0091	0.0004

Table B.7: Results of the group 9 simulation, with $p_{1,0} = 0.01$, $\alpha = 0.10$, $\delta = 0.50$, and $\gamma = 0.01$.

Appendix C. Matlab Code for the Markov Chain Simulations

This appendix contains the code for the Markov chain simulation of the truncated normal distribution and the posterior distribution in the Chronic Wasting disease model. The code was written and run with Matlab, version 6, on a $2.2Ghz$ Pentium 4 processor, with $2GB$ RAM.

C.1 Miscellaneous Probability Functions

These are functions that are used throughout the simulations. They are the log density kernels of the multivariate normal distribution and the normal distribution truncated to $[0, 1]$.

C.1.1 Function: logmvnkern.m

This function calculates the kernel of the log of the multivariate normal density function.

```
function YY=logmvnkern(X,mu,Sig)

%This function calculates the kernel of the log of the multivariate
%normal density function with mean vector mu and covariance matrix
%Sig at the values given in the vector X.
%

%Christopher H. Mehl, 2003.

YY=-0.5*log(det(Sig))-0.5*((X-mu)'/Sig)*(X-mu);
```

C.1.2 Function: logtruncmvn.m

This function calculates the kernel of the log of the normal density function, truncated on $[0, 1]$.

```
function YY=logtruncmvn(X,mu,Sig)

%This function calculates the kernel of the log of the truncated
%multivariate normal density function with mean vector mu and
%covariance matrix Sig at the values given in the vector X. The
%truncation is on [0,1] in each dimension. The covariance matrix
%must be diagonal for this to work.
%Christopher H. Mehl, 2004.

mu=mu(:); X=X(:);

mvnkern=logmvnkern(X,mu,Sig);

% use the erf to get the normal cdf at 0 and 1 for each dimension
%
arg1=(1-mu)./sqrt(diag(Sig)*2);
arg0=(0-mu)./sqrt(diag(Sig)*2);
cdf1=0.5*erf(arg1)+0.5;
cdf0=0.5*erf(arg0)+0.5;
```

```
%Truncation part
if (0>min(X)) | (max(X)>1)
   YY=-inf;
else
   YY=mvnkern-sum(log(cdf1-cdf0));
end

%%
```

C.2 Code for Simulating the Truncated Normal Distributions

This code was used to simulate a normal distribution truncated to the interval $[0, 1]$. The simulation methods were the algorithm 4.4, the MALA, and the standard random walk, all discussed in chapter 4.

C.2.1 Program: truncnormcomps.m

This program performs a MALTS step, a MALA step, and a standard random walk step for the truncated normal distribution. The program calculates the acceptance rate and the potential scale reduction using five runs which are started from random initial values.

```
warning off

global Mu CoSig cnt1 cnt2 cnt3 modes dim rho;

global arrw armehl arlang;
```

```
arrw=zeros(5,1);
armehl=zeros(5,1);
arlang=zeros(5,1);
rwvar1=zeros(5,80000);
langvar1=zeros(5,80000);
mehlvar1=zeros(5,80000);
rwvar2=zeros(5,80000);
langvar2=zeros(5,80000);
mehlvar2=zeros(5,80000);
mehlrhat1=zeros(1,800);
mehlrhat2=zeros(1,800);
langrhat1=zeros(1,800);
langrhat2=zeros(1,800);
rwrhat1=zeros(1,800);
rwrhat2=zeros(1,800);

for runs=1:5
   Mu=0.89*ones(dim,1);
   modes=Mu;
   CoSig=0.00001*eye(dim);
   CoSig(1,1)=0.001;
   cnt1=0;
   cnt2=0;
```

```
cnt3=0;
NChain=80000;
theta0a=zeros(dim,NChain);
theta0b=zeros(dim,NChain);
theta0c=zeros(dim,NChain);
%Runs started at random points.
theta0a(:,1)=rand(dim,1);
theta0b(:,1)=theta0a(:,1);
theta0c(:,1)=theta0a(:,1);
tempsamp1=zeros(dim,1);
tempsamp2=zeros(dim,1);
tempsamp3=zeros(dim,1);
logpost1=logtruncmvn(theta0a(:,1),Mu,CoSig);
logpost2=logtruncmvn(theta0b(:,1),Mu,CoSig);
logpost3=logtruncmvn(theta0c(:,1),Mu,CoSig);
for k=2:NChain
   tempsamp1=theta0a(:,k-1);
   tempsamp2=theta0b(:,k-1);
   tempsamp3=theta0c(:,k-1);
   [tempsamp1,logpost1]=mehlwitch(tempsamp1,logpost1);
   [tempsamp2,logpost2]=langwitch(tempsamp2,logpost2);
   [tempsamp3,logpost3]=rwwitch(tempsamp3,logpost3);
   theta0a(:,k)=tempsamp1;
   theta0b(:,k)=tempsamp2;
```

```
      theta0c(:,k)=tempsamp3;
   end
   mehlvar1(runs,:)=theta0a(1,:);
   langvar1(runs,:)=theta0b(1,:);
   rwvar1(runs,:)=theta0c(1,:);
   mehlvar2(runs,:)=theta0a(2,:);
   langvar2(runs,:)=theta0b(2,:);
   rwvar2(runs,:)=theta0c(2,:);
   armehl(runs)=cnt1/NChain;
   arlang(runs)=cnt2/NChain;
   arrw(runs)=cnt3/NChain;
end

for k=1:800
   mehlrhat1(k)=assess(mehlvar1(:,1:(100*k)));
   mehlrhat2(k)=assess(mehlvar2(:,1:(100*k)));
   langrhat1(k)=assess(langvar1(:,1:(100*k)));
   langrhat2(k)=assess(langvar2(:,1:(100*k)));
   rwrhat1(k)=assess(rwvar1(:,1:(100*k)));
   rwrhat2(k)=assess(rwvar2(:,1:(100*k)));
end
```

C.2.2 Function: mehlwitch.m

This function performs the tempered, modified Langevin step for algorithm 4.4 on the truncated normal distribution discussed in section 4.3.4.

```
function [newpars,newpost]=mehlwitch(oldpars,oldpost)

%This function performs a modified Langevin step for
%a truncated normal distribution.
%
%Christopher Mehl, 2004.

global Mu CoSig cnt1 cnt2 cnt3 modes dim rho;

global arrw armehl arlang;

LNG=length(oldpars);
logold=oldpost;

%Approximation of the gradient.
h=0.000000005;
%
SIG=0.00001*eye(LNG);
delpar=zeros(LNG,1);
shift=zeros(LNG,1);
postfunmax=logtruncmvn(modes,Mu,CoSig);
```

```
for k=1:LNG
   shift(k)=h;
   delpar(k)=(1/h)*(logtruncmvn(oldpars+shift,Mu,CoSig)-logold);
   shift(k)=0;
end gradold=delpar; if norm(gradold,2)==0
   mutheta=oldpars;
else
   dir=gradold/norm(gradold,2);
   step=(SIG)*(postfunmax-logold+1);
   %step=(0.5*SIG.^2)*(postfunmax-logold+1);
   mutheta=oldpars+step*dir;
end thetaprop=mvnrnd(mutheta,SIG,1)';
%thetaprop=csmvrnd(mutheta,SIG,1)';
if min(thetaprop)<0 | max(thetaprop)>1
   newpars=oldpars;
   newpost=logold;
else
   lognew=logtruncmvn(thetaprop,Mu,CoSig);
   delparnew=zeros(LNG,1);
   shift=zeros(LNG,1);
   for k=1:LNG
      shift(k)=h;
      delparnew(k)=(1/h)*...
      ...(logtruncmvn(thetaprop+shift,Mu,CoSig)-lognew);
```

```
      shift(k)=0;
   end
   gradnew=delparnew;
   dirnew=gradnew/norm(gradnew,2);
   stepnew=(SIG)*(postfunmax-lognew+1);
   %stepnew=(0.5*SIG.^2)*(postfunmax-lognew+1);
   munew=thetaprop+stepnew*dirnew;
   lalpha=lognew-logold+logtruncmvn(thetaprop,mutheta,SIG)...
   ...-logtruncmvn(oldpars,munew,SIG);
   if log(rand(1,1))<lalpha
      cnt1=cnt1+1;
      newpars=thetaprop;
      newpost=lognew;
   else
      newpars=oldpars;
      newpost=logold;
   end
end

%
```

C.2.3 Function: langwitch.m

This function performs a Langevin Metropolis step for the truncated normal distribution discussed in section 4.3.4.

```
function [newpars,newpost]=langwitch(oldpars,oldpost)

%This function performs a Langevin step for
%a truncated normal distribution.
%
%Christopher Mehl, 2004.

global Mu CoSig cnt1 cnt2 cnt3 modes dim rho;

global arrw armehl arlang;

LNG=length(oldpars); logold=oldpost;

%Approximation of the gradient.
h=0.000000005;
%
SIG=0.00001*eye(LNG); delpar=zeros(LNG,1); shift=zeros(LNG,1);

for k=1:LNG
   shift(k)=h;
   delpar(k)=(1/h)*(logtruncmvn(oldpars+shift,Mu,CoSig)-logold);
   shift(k)=0;
end gradold=delpar;
mutheta=oldpars+(SIG)*gradold;
```

```
%mutheta=oldpars+(0.5*SIG.^2)*gradold;
thetaprop=mvnrnd(mutheta,SIG,1)';
%thetaprop=csmvrnd(mutheta,SIG,1)';
if min(thetaprop)<0 | max(thetaprop)>1
   newpars=oldpars;
   newpost=logold;
else
   lognew=logtruncmvn(thetaprop,Mu,CoSig);
   delparnew=zeros(LNG,1);
   shift=zeros(LNG,1);
   for k=1:LNG
      shift(k)=h;
      delparnew(k)=(1/h)*(logtruncmvn(thetaprop+shift,Mu,CoSig)...
      ...-lognew);
      shift(k)=0;
   end
   gradnew=delparnew;
   munew=thetaprop+(SIG)*gradnew;
   %munew=thetaprop+(0.5*SIG.^2)*gradnew;
   lalpha=lognew-logold+logtruncmvn(thetaprop,mutheta,SIG)...
   ...-logtruncmvn(oldpars,munew,SIG);
   if log(rand(1,1))<lalpha
      cnt2=cnt2+1;
      newpars=thetaprop;
```

```
      newpost=lognew;
   else
      newpars=oldpars;
      newpost=logold;
   end
end

%
```

C.2.4 Function: rwwitch.m

This function performs a standard random walk Metropolis step for the truncated normal distribution in section 4.3.4.

```
function [newpars,newpost]=rwwitch(oldpars,oldpost)

%This function performs a random walk MH step for
%the witch's hat distribution.
%
%Christopher Mehl, 2004.

global Mu CoSig cnt1 cnt2 cnt3 modes dim rho;

global arrw armehl arlang;

LNG=length(oldpars);
```

```
logold=oldpost;
SIG=0.00001*eye(LNG);
%SIG=0.5*SIG.^2;
tmp1=zeros(LNG,1);
thetaprop=oldpars+mvnrnd(tmp1,SIG,1)';
%thetaprop=oldpars+csmvrnd(tmp1,SIG,1)';

%Reject values outside of [0,1].
if min(thetaprop)<0 | max(thetaprop)>1
   newpars=oldpars;
   newpost=logold;
else
   lognew=logtruncmvn(thetaprop,Mu,CoSig);
   lalpha=lognew-logold;
   if log(rand(1,1))<lalpha
      cnt3=cnt3+1;
      newpars=thetaprop;
      newpost=lognew;
   else
      newpars=oldpars;
      newpost=logold;
   end
end
```

```
%
```

C.3 Code for Simulating the Normal Distributions

This code was used to simulate a multivariate normal distribution. The simulation methods were the algorithm 4.4, the MALA, and the standard random walk, all discussed in chapter 4.

C.3.1 Program: mvnormalcomps.m

This program performs a MALTS step, a MALA step, and a standard random walk step for the normal distribution. The program calculates the acceptance rate and the potential scale reduction using five runs which are started from random initial values.

```
%The true values will be
%mvrnd(zeros(dim,1),(1-rho)*eye(dim)+rho*ones(dim,dim)).

warning off

global Mu CoSig cnt1 cnt2 cnt3 modes dim rho;

global arrw armehl arlang;

arrw=zeros(5,1);
armehl=zeros(5,1);
arlang=zeros(5,1);
rwvar1=zeros(5,80000);
```

```
langvar1=zeros(5,80000);
mehlvar1=zeros(5,80000);
rwvar2=zeros(5,80000);
langvar2=zeros(5,80000);
mehlvar2=zeros(5,80000);
mehlrhat1=zeros(1,800);
mehlrhat2=zeros(1,800);
langrhat1=zeros(1,800);
langrhat2=zeros(1,800);
rwrhat1=zeros(1,800);
rwrhat2=zeros(1,800);

for runs=1:5
   Mu=zeros(dim,1);
   modes=Mu;
   CoSig=(1-rho)*eye(dim)+rho*ones(dim,dim);
   cnt1=0;
   cnt2=0;
   cnt3=0;
   NChain=80000;
   theta0a=zeros(dim,NChain);
   theta0b=zeros(dim,NChain);
   theta0c=zeros(dim,NChain);
   %Initialize runs at random points.
```

```
theta0a(:,1)=chol(CoSig)*randn(dim,1);
theta0b(:,1)=theta0a(:,1);
theta0c(:,1)=theta0a(:,1);
tempsamp1=zeros(dim,1);
tempsamp2=zeros(dim,1);
tempsamp3=zeros(dim,1);
logpost1=logmvnkern(theta0a(:,1),Mu,CoSig);
logpost2=logmvnkern(theta0b(:,1),Mu,CoSig);
logpost3=logmvnkern(theta0c(:,1),Mu,CoSig);
for k=2:NChain
   tempsamp1=theta0a(:,k-1);
   tempsamp2=theta0b(:,k-1);
   tempsamp3=theta0c(:,k-1);
   [tempsamp1,logpost1]=mehlnormal(tempsamp1,logpost1);
   [tempsamp2,logpost2]=langnormal(tempsamp2,logpost2);
   [tempsamp3,logpost3]=rwnormal(tempsamp3,logpost3);
   theta0a(:,k)=tempsamp1;
   theta0b(:,k)=tempsamp2;
   theta0c(:,k)=tempsamp3;
end
mehlvar1(runs,:)=theta0a(1,:);
langvar1(runs,:)=theta0b(1,:);
rwvar1(runs,:)=theta0c(1,:);
mehlvar2(runs,:)=theta0a(2,:);
```

```
   langvar2(runs,:)=theta0b(2,:);
   rwvar2(runs,:)=theta0c(2,:);
   armehl(runs)=cnt1/NChain;
   arlang(runs)=cnt2/NChain;
   arrw(runs)=cnt3/NChain;
end

for k=1:800
   mehlrhat1(k)=assess(mehlvar1(:,1:(100*k)));
   mehlrhat2(k)=assess(mehlvar2(:,1:(100*k)));
   langrhat1(k)=assess(langvar1(:,1:(100*k)));
   langrhat2(k)=assess(langvar2(:,1:(100*k)));
   rwrhat1(k)=assess(rwvar1(:,1:(100*k)));
   rwrhat2(k)=assess(rwvar2(:,1:(100*k)));
end

%
```

C.3.2 Function: mehlnormal.m

This function performs the tempered, modified Langevin step for algorithm 4.4 on the normal distribution discussed in section 4.3.4.

```
function [newpars,newpost]=mehlnormal(oldpars,oldpost)

%This function performs a modified Langevin step for
```

```
%a MVN distribution.
%
%Christopher Mehl, 2004.

global Mu CoSig cnt1 cnt2 cnt3 modes dim rho;

global arrw armehl arlang;

LNG=length(oldpars);
logold=oldpost;

%Approximation of the gradient.
h=0.000000005;
%
SIG=0.00001*eye(LNG);
delpar=zeros(LNG,1);
shift=zeros(LNG,1);
postfunmax=logmvnkern(modes,Mu,CoSig);

for k=1:LNG
   shift(k)=h;
   delpar(k)=(1/h)*(logmvnkern(oldpars+shift,Mu,CoSig)-logold);
   shift(k)=0;
end
```

```
gradold=delpar;

if norm(gradold,2)==0
   mutheta=oldpars;
else
   dir=gradold/norm(gradold,2);
   step=(SIG)*(postfunmax-logold+1);
   %step=(0.5*SIG.^2)*(postfunmax-logold+1);
   mutheta=oldpars+step*dir;
end

thetaprop=mvnrnd(mutheta,SIG,1)';
%thetaprop=csmvrnd(mutheta,SIG,1)';
lognew=logmvnkern(thetaprop,Mu,CoSig);
delparnew=zeros(LNG,1);
shift=zeros(LNG,1);

for k=1:LNG
   shift(k)=h;
   delparnew(k)=(1/h)*(logmvnkern(thetaprop+shift,Mu,CoSig)-lognew);
   shift(k)=0;
end
gradnew=delparnew;
```

```
dirnew=gradnew/norm(gradnew,2);
stepnew=(SIG)*(postfunmax-lognew+1);
%stepnew=(0.5*SIG.^2)*(postfunmax-lognew+1);
munew=thetaprop+stepnew*dirnew;

lalpha=lognew-logold+...
...logmvnkern(thetaprop,mutheta,SIG)-logmvnkern(oldpars,munew,SIG);
if log(rand(1,1))<lalpha
   cnt1=cnt1+1;
   newpars=thetaprop;
   newpost=lognew;
else
   newpars=oldpars;
   newpost=logold;
end

%
```

C.3.3 Function: langnormal.m

This function performs the Langevin step on the normal distribution discussed in section 4.3.4.

```
function [newpars,newpost]=langnormal(oldpars,oldpost)

%This function performs a regular Langevin step for
```

```
%a MVN distribution.
%
%Christopher Mehl, 2004.

global Mu CoSig cnt1 cnt2 cnt3 modes dim rho;

global arrw armehl arlang;

LNG=length(oldpars);
logold=oldpost;

%Approximation of the gradient.
h=0.000000005;
%
SIG=0.00001*eye(LNG);
delpar=zeros(LNG,1);
shift=zeros(LNG,1);

for k=1:LNG
   shift(k)=h;
   delpar(k)=(1/h)*(logmvnkern(oldpars+shift,Mu,CoSig)-logold);
   shift(k)=0;
end
```

```
gradold=delpar;
mutheta=oldpars+(SIG)*gradold;
%mutheta=oldpars+(0.5*SIG.^2)*gradold;
thetaprop=mvnrnd(mutheta,SIG,1)';
%thetaprop=csmvrnd(mutheta,SIG,1)';
lognew=logmvnkern(thetaprop,Mu,CoSig);
delparnew=zeros(LNG,1);
shift=zeros(LNG,1); for k=1:LNG
   shift(k)=h;
   delparnew(k)=(1/h)*(logmvnkern(thetaprop+shift,Mu,CoSig)-lognew);
   shift(k)=0;
end gradnew=delparnew; munew=thetaprop+(SIG)*gradnew;
%munew=thetaprop+(0.5*SIG.^2)*gradnew;
lalpha=lognew-logold+...
...logmvnkern(thetaprop,mutheta,SIG)-logmvnkern(oldpars,munew,SIG);
if log(rand(1,1))<lalpha
   cnt2=cnt2+1;
   newpars=thetaprop;
   newpost=lognew;
else
   newpars=oldpars;
   newpost=logold;
end
```

```
%
```

C.3.4 Function: rwnormal.m

This function performs a standard random walk Metropolis step for the multivariate normal distribution discussed in section 4.3.4.

```
function [newpars,newpost]=rwnormal(oldpars,oldpost)

%This function performs a random walk MH step for
%the MV normal distribution.
%
%Christopher Mehl, 2004.

global Mu CoSig cnt1 cnt2 cnt3 modes dim rho;

global arrw armehl arlang;

LNG=length(oldpars);
logold=oldpost;
SIG=0.00001*eye(LNG);
%SIG=0.5*SIG.^2;
tmp1=zeros(LNG,1);
thetaprop=oldpars+mvnrnd(tmp1,SIG,1)';
%thetaprop=oldpars+csmvrnd(tmp1,SIG,1)';
```

```
lognew=logmvnkern(thetaprop,Mu,CoSig);
lalpha=lognew-logold;

if log(rand(1,1))<lalpha
   cnt3=cnt3+1;
   newpars=thetaprop;
   newpost=lognew;
else
   newpars=oldpars;
   newpost=logold;
end

%
```

C.4 Code for the Chronic Wasting Disease Model

This code was used to analyze the prevalence data for the Chronic Wasting Disease model. The programs include the posterior function, and the Markov chain simulation.

C.4.1 Function: pmaker3.m

This is the deterministic update function used to calculate prevalence values, given the parameters. The update equation is discussed in section 3.2.3.

```
function pall = pmaker3(par,mdl,nn)
%%  function pall = pmakeri(par,i,nn)
```

```
%%
%%  Builds the probabilities according to
%%  the difference equation
%%  for the model with MDL non-zero probabilities and
%%  5-other parameters
%%
%%  p(:,t+1) = Q*p(:,t) +  alpha*X_t*(delta -X_t)
%%
%%  INPUTS
%%    PAR = [p1,p2,...p_i,alpha,delta,pi,omega]';
%%
%%          pp0 = [p1,p2,...,p_i, 0,0,0,0]';
%%          Each DAU has its own starting level
%%
%%          tht = [alpha,delta,pi,omega]';
%%          these parameters control the
%%          disease dynamics in the model
%%
%%    I     1 <= I <= 3
%%          the number of non-zero starting prevalences.
%%
%%    NN =  an integer > 1
%%          How many years to build?
%%          (default is set to 27 if nothing is entered)
```

```
%%
%%
%%  OUTPUTS
%%    PALL  the probabilities that correspond
%%          with each year/DAU
%%          an 16 x NN matrix
%%

global DD1 DD2 TT1 TT2 ORDR NNN;
[nr,nc] = size(NNN);
if(nargin<3)
nn = nc+1;
end

pp0=zeros(nr,1);
pp0(ORDR(1:mdl))=par(1:mdl);
alph = par(mdl+1);
delt = par(mdl+2);
gama = par(mdl+3);

accl=alph*4*min(1./delt^2,1./(1-delt)^2);
pall = zeros(nr,nn);
pall(:,1) = pp0;
```

```
WWW=gama*TT1;
WWW=eye(nr)-diag(sum(WWW'))+WWW;
for j =1:(nn-1);
    dum = WWW*pall(:,j) + ...
    ...accl*pall(:,j).*(delt-pall(:,j)).*(1-pall(:,j));
    pall(:,j+1) = dum;
end
```

C.4.2 Function: postfun3nopi.m

This is the log posterior density for the CWD model. The parameter π is not used in this posterior, since it is independent of the other parameters.

```
function pst = postfun3_nopi(par,mdl)
%% Evaluates   log(POSTERIOR DISTRIBUTION)
%% without the parameter pi.
%%  INPUTS
%%     PAR
%%
global  PRIORS PR_PROB DD1 TT1 DD2 TT2 CUTOFF MMM NNN ORDR;

[nr,nc] = size(MMM); pp0   = zeros(nr,1); ooo   = ORDR(1:mdl);
pp0(ooo) = par(1:mdl); tht   = par(mdl+(1:3)); alph  = tht(1);
delt  = tht(2); gama  = tht(3);
%pii   = tht(4);
```

```
slo=min(pp0)>=0&max(pp0)<=1&alph<1&alph>0&delt<1& delt>0;

slo = slo & max(DD1)*gama < 1 ;



if(slo)
  prpr  = sum((PR_PROB(1:mdl,1)-1).*log(pp0(ooo)));
  prpr  = prpr + sum( (PR_PROB(1:mdl,2)-1).*log(1-pp0(ooo)));
  prth  = sum((PRIORS([1:3],1)-1).*log(vec(tht)));
  prth  = prth + sum((PRIORS([1:3],2)-1).*log(1-vec(tht)));
  prior =  prpr + prth ;

  ppt = pmaker3(par,mdl);
  pp1 = ppt(1:nr,2:(nc+1));
  %pp1(NNN<=CUTOFF) = pii;
  % ensure that prevalences are between zero and 1
  tm1 = min(pp1(:))>=0 & max(pp1(:))<=1 ;
  % ensure that CWD is never observed when prevalence is 0
  % or that CWD is always observed when prevalence is 1.
  tm2 = sum(MMM(pp1==0) > 0) == 0 & ...
  ...sum((NNN(pp1==1)-MMM(pp1==1))>0)==0;
  if( tm1 & tm2 )
    tmp = ~isnan(MMM) & ~isnan(NNN) & ~(pp1==0 & MMM==0)...
    ...& ~(pp1==1 & MMM == NNN);
```

```
    tmp = tmp & NNN>CUTOFF;
    like = sum(MMM(tmp).*log(pp1(tmp)) + ...
    ...(NNN(tmp)-MMM(tmp)).*log(1-pp1(tmp)));
    like = like;
  else
    slo = 0;
  % disp('postfun3!')
    like = -(999999999+8888888*sum(abs(par)) + rand(1)*77777);
  end
end

if(slo)
 pst = prior + like;
else
% disp('postfun3!')
  pst =  -(999999999+88888888*sum(abs(par)) + rand(1)*77777);
end
%
```

C.4.3 Program: prevmehl7g.m

This is the Markov chain simulation for the CWD data, using the algorithm discussed in chapter 4.

```
%Declare global variables
global MMM NNN TT1 TT2 DD1 DD2 PRIORS PR_PROB YRS DAU UUU CUTOFF;
```

```
global ORDR;


%PR_PROB  = [ones(54,1),ones(54,1)*200];
%ORDR     = [4,10,5,27,44,3,9,17];
%PRIORS(1,:) = [1,1];       % alpha: Uniform
%PRIORS(2,:) = [8,82];      % delta:
%       mean = 8/90=0.09,  std.dev. = 0.0298
%PRIORS(3,:) = [1,50];      % gamma
%PRIORS(4,:) = [1,100];     % omega
%PRIORS(5,:) = [1.5,1.5];   % pi


warning off
%This loads the .mat file with the saved variables and data.


load matlab
load LOADME


%The Sampler
%The sampler will implement a Tempered Truncated
%Metropolis Adjusted Langevin algorithm to
%simulate the posterior distribution.
%
%The simulation will use the same functions:  postfun3_nopi.m,
%                                             pmaker3.m
```

```
%as the acceptance sampler in [Johns and Mehl, 2004].

%Length of Markov Chain, initialization
NChain=300000;

%Parameters are: theta(1,:)= p_10
%                theta(2,:)= p_4
%                theta(3,:)= p_5
%                theta(4,:)= alpha
%                theta(5,:)= delta
%                theta(6,:)= gamma

theta0=zeros(6,NChain);

%Successive Substitution Sampler
tempsamp=zeros(6,1); logpost=postfun3_nopi(theta0(:,1),3);

for k=2:NChain
   tempsamp=theta0(:,k-1);
   [tempsamp,logpost]=metpar7g(tempsamp,logpost);
   theta0(:,k)=tempsamp;
end
%%%
save LONGRUN1Ga
```

```
%
```

C.4.4 Function: metpar7g.m

This function performs a tempered, truncated Langevin Metropolis step, using the method introduced in chapter 4.

```
function [newpars,newpost]=metpar7g(oldpars,oldpost)

%This function performs a Langevin Metropolis step for
%all parameters but pi.
%
%The proposal will be in the direction of the gradient,
%using a standard normal distribution.
%
%It returns the updated parameter vector and the log
%of the posterior probability, to avoid recalculation.
%
%Arguments: oldpars= old parameter vector, should be a
%                    6 vector, we don't
%                    simulate pie
%           oldpost= value of posterior distribution

p10=oldpars(1); p4=oldpars(2); p5=oldpars(3); alph=oldpars(4);
delt=oldpars(5); gamm=oldpars(6); logold=oldpost;

%Approximation of the gradient. h= step size,
```

```
%SIG= tuning parameter.
%Because the different components of our vector have
%different SD, we want to step them different distances.
%

h=0.000000005;
SIG=(0.000000000001)*diag([0.1005,0.0781,0.0400,0.1995,0.1543,0.0346]);
modes=[0.0251;0.0127;0.0022;0.1408;0.1479;0.0081];
postfun3max=postfun3_nopi(modes,3);

delp10=(1/h)*(postfun3_nopi([p10+h;p4;p5;alph;delt;gamm],3)-logold);
delp4=(1/h)*(postfun3_nopi([p10;p4+h;p5;alph;delt;gamm],3)-logold);
delp5=(1/h)*(postfun3_nopi([p10;p4;p5+h;alph;delt;gamm],3)-logold);
delalp=(1/h)*(postfun3_nopi([p10;p4;p5;alph+h;delt;gamm],3)-logold);
deldel=(1/h)*(postfun3_nopi([p10;p4;p5;alph;delt+h;gamm],3)-logold);
delgam=(1/h)*(postfun3_nopi([p10;p4;p5;alph;delt;gamm+h],3)-logold);
gradold=[delp10;delp4;delp5;delalp;deldel;delgam];

%The quantity logmax-logold+1 is bounded below by 1.
%General idea:  the closer to the mode, the smaller the step size.

dir=gradold/norm(gradold,2);
step=(0.000000000005)*(postfun3max-logold+1);
mutheta=oldpars+step*dir;
```

```
thetaprop=mvnrnd(mutheta,SIG,1)';


%Decision rule
if min(thetaprop)<0 | max(thetaprop)>1
    newpars=oldpars;
    newpost=logold;
else
    p10new=thetaprop(1);
    p4new=thetaprop(2);
    p5new=thetaprop(3);
    alpnew=thetaprop(4);
    delnew=thetaprop(5);
    gamnew=thetaprop(6);
    lognew=postfun3_nopi(thetaprop,3);
    delp10new=(1/h)*(postfun3_nopi([p10new+h;p4new;p5new;...
    ...alpnew;delnew;gamnew],3)-lognew);
    delp4new=(1/h)*(postfun3_nopi([p10new;p4new+h;p5new;...
    ...alpnew;delnew;gamnew],3)-lognew);
    delp5new=(1/h)*(postfun3_nopi([p10new;p4new;p5new+h;...
    ...alpnew;delnew;gamnew],3)-lognew);
    delalpnew=(1/h)*(postfun3_nopi([p10new;p4new;p5new;...
    ...alpnew+h;delnew;gamnew],3)-lognew);
    deldelnew=(1/h)*(postfun3_nopi([p10new;p4new;p5new;...
```

```
    ...alpnew;delnew+h;gamnew],3)-lognew);
    delgamnew=(1/h)*(postfun3_nopi([p10new;p4new;p5new;...
    ...alpnew;delnew;gamnew+h],3)-lognew);
    gradnew=[delp10new;delp4new;delp5new;...
    ...delalpnew;deldelnew;delgamnew];
    dirnew=gradnew/norm(gradnew,2);
    stepnew=(0.000000000001)*(postfun3max-lognew+1);
    munew=thetaprop+stepnew*dirnew;

    lalpha=lognew-logold+...
    ...logmvnkern(thetaprop,mutheta,SIG)-...
    ...logmvnkern(oldpars,munew,SIG);
    %
    if log(rand(1,1))<lalpha
        newpars=thetaprop;
        newpost=lognew;
    else
        newpars=oldpars;
        newpost=logold;
    end
end

%%
```

C.4.5 Function: metpar7a.m

This function is used as the Metropolis step in the standard random walk Markov chain simulation for the CWD model.

```
function [newpars,newpost]=metpar7a(oldpars,oldpost)

%This function performs a Random Walk Metropolis step
%for all parameters but pi.
%
%
%It returns the updated parameter vector and
%the log of the posterior
%probability, to avoid recalculation.
%

p10=oldpars(1);
p4=oldpars(2);
p5=oldpars(3);
alph=oldpars(4);
delt=oldpars(5);
gamm=oldpars(6);
logold=oldpost;

SIG=0.000001*diag([0.1005,0.0781,0.0400,0.1995,0.1543,0.0346]);

thetaprop=oldpars+mvnrnd(zeros(6,1),SIG,1)';
```

```
%thetaprop=oldpars+csmvrnd(zeros(6,1),SIG,1)';

%Decision rule
if min(thetaprop)<0 | max(thetaprop)>1
    newpars=oldpars;
    newpost=logold;
else
    lalpha=lognew-logold+logmvnkern(thetaprop,oldpars,SIG)...
    ...-logmvnkern(oldpars,thetaprop,SIG);
    if log(rand(1,1))<lalpha
        newpars=thetaprop;
        newpost=lognew;
    else
        newpars=oldpars;
        newpost=logold;
    end
end

%%
```

C.5 Simulation Code for Model Validation

The code in this sections was used to generate the artificial data and run the simulations discussed in section 5.2. The code here was used for the first three simulations. The remaining simulations differed only in the initial prevalence values, as detailed in section 5.2.

C.5.1 Program: group1test.m

This ran the first three model variations. The initial prevalence value was $p_{1,0} = 0.05$, and the levels of δ were 0.05, 0.15, and 0.50. The remaining simulations varied only in the initial prevalences that were chosen. The mode was determined using the function *fminsearch.m* that is built into Matlab. For each of the nine model variations, a data set was selected randomly and used in maximizing the log posterior density. This value was used for all fifty runs as the mode.

```
%This program will generate data for a given model

global alptrue deltrue gamtrue MMM NNN TT1 DD1 PRIORS PR_PROB;
global NChain CUTOFF ORDR;



NChain=200000;

load SPATIALMATS

CUTOFF=0;
ORDR=[1,2,3,4,5,6,7,8,9];

PR_PROB  = [ones(9,1),ones(9,1)*200];
```

```
PRIORS(1,:) = [1,1];       % alpha: Uniform
PRIORS(2,:) = [8,82];      % delta:
PRIORS(3,:) = [1,50];      % gamma:

%Building the fake NNN.
yrs=20; NNN=zeros(9,yrs); NNN(:,1:4)=15; NNN(:,5:15)=100;
NNN(:,16:yrs)=1000;

%% There will be 12 sets of models tested, each for 50 runs.
%% \alpha, \gamma will be the same throughout.
%%

%%
%% The second set will have p(1,0)=0.05, p(2,0)=0.01,
%% \delta=[0.05;0.15;0.25]
%%
%% The third set will have p(1,0)=0.01,
%% \delta=[0.05;0.15;0.25]
%%
%% The fourth set will have p(1,0)=0.01, p(2,0)=0.01,
%% \delta=[0.05;0.15;0.25]
%%
```

```
%Bear in mind that \alpha and \delta are highly correlated!
alptrue=0.10; gamtrue=0.01; deltrue=[0.05;0.15;0.5];

save SIMPREP

%% Model set 1: Manufacture the data.
%% The first set will have p(1,0)=0.05,
%% \delta=[0.05;0.15;0.5]
mdl=1; p0=0.05;
%Initialize
% j is the level of \delta. j=1 is \delta=0.05.
% j=2 is \delta=0.15. j=3 is \delta=0.5.
% k is the run number (out of 50).
fakedata1=zeros(9,yrs,3,50);

for j=1:3
   tht0=[p0;alptrue;deltrue(j);gamtrue];
   pits=pmaker3(tht0,mdl,yrs);
   for k=1:50
      fakedata1(:,:,j,k)=binornd(NNN,pits);
   end
end

save GROUP1DATA
```

```
%Model set 1.
mdl=1;
grp=1;

%Stats: 4 pars, 3 delt values, 50 runs.
g1mean=zeros(4,3,50);
g1med=zeros(4,3,50);
g1std=zeros(4,3,50);
g1low=zeros(4,3,50);
g1up=zeros(4,3,50);
Dif1=zeros(4,3,50);

for j=1:3
   mode=pstmax(grp,j);
   for k=1:50
      [g1mean(:,j,k),g1med(:,j,k),g1std(:,j,k),...
      ...g1low(:,j,k),g1up(:,j,k),Dif1(:,j,k)]...
      ...=simrun(fakedata1(:,:,j,k),mdl,mode);
   end
end

clear fakedata1
```

```
save GROUP1OUTPUT g1mean g1med g1std g1low g1up Dif1

clear g1mean g1med g1std g1low g1up Dif1
```

C.5.2 Function: simrun.m

This function runs 200,000 iterations of a Markov chain sampler using algorithm 4.4 from chapter 4. The posterior estimates are calculated, using the last 150,000 iterations as a burn-in period. The posterior estimates are stored for each of the fifty runs within a particular model variation.

```
function [Pmean,Pmed,Pstd,PLowCI,PUpCI,Dif]=simrun(data,mdl,mode)

%This function will take the DATA as its argument. It will
%perform NChain iterations of my MCMC method. After computing the
%output of the MCMC method, it will calculate posterior estimates
%and credible intervals. SIMRUN calls METSIM7G.

global alptrue deltrue gamtrue MMM NNN TT1 DD1 PRIORS PR_PROB;

global CUTOFF ORDR NChain;

MMM=data;

if (mdl==1)
   tmp1=zeros(4,1);
```

```
   tmp2=zeros(4,1);
   theta0=zeros(4,NChain);
   theta0(:,1)=[0.03;0.08;0.1;0.01];
   %This loop needs to be run inside each data loop.
   %Successive Substitution Sampler
   tempsamp=zeros(4,1);
   logpost=postfun3(theta0(:,1),mdl);
   for k=2:NChain
      tempsamp=theta0(:,k-1);
      [tempsamp,logpost]=metsim7g(tempsamp,logpost,mdl,mode);
      theta0(:,k)=tempsamp;
   end
   %save CHECK theta0
elseif (mdl==2)
   tmp1=zeros(5,1);
   tmp2=zeros(5,1);
   theta0=zeros(5,NChain);
   theta0(:,1)=[0.03;0.03;0.08;0.1;0.01];
   %This loop needs to be run inside each data loop.
   %Successive Substitution Sampler
   tempsamp=zeros(5,1);
   logpost=postfun3(theta0(:,1),mdl);
   for k=2:NChain
      tempsamp=theta0(:,k-1);
```

```
      [tempsamp,logpost]=metsim7g(tempsamp,logpost,mdl,mode);
      theta0(:,k)=tempsamp;
   end
end

%Posterior Mean, Median, Std, LowCI, UpCI.
burnin=50000;
tmp1(1)=prctile(theta0(1,burnin:NChain),0.025);
tmp1(2)=prctile(theta0(2,burnin:NChain),0.025);
tmp1(3)=prctile(theta0(3,burnin:NChain),0.025);
tmp1(4)=prctile(theta0(4,burnin:NChain),0.025);
if mdl==2
   tmp1(5)=prctile(theta0(5,burnin:NChain),0.025);
end
tmp2(1)=prctile(theta0(1,burnin:NChain),0.975);
tmp2(2)=prctile(theta0(2,burnin:NChain),0.975);
tmp2(3)=prctile(theta0(3,burnin:NChain),0.975);
tmp2(4)=prctile(theta0(4,burnin:NChain),0.975);
if mdl==2
   tmp2(5)=prctile(theta0(5,burnin:NChain),0.975);
end Pmean=mean(theta0(:,burnin:NChain)');
Pmed=median(theta0(:,burnin:NChain)');
Pstd=std(theta0(:,burnin:NChain)'); PLowCI=tmp1; PUpCI=tmp2;
lngth=NChain/2;
```

```
Dif=mean(theta0(:,(lngth+1):NChain)')-...
...mean(theta0(:,burnin:lngth)');
```

C.5.3 Function: metsim7g.m

This function performs a tempered, truncated, Langevin Metropolis step for the model validation simulation, as detailed in chapter 4. The values for Σ were kept the same through all of the simulations.

```
function [newpars,newpost]=metsim7g(oldpars,oldpost,mdl,mode)

%This function performs my Langevin Metropolis step
%for all parameters of the simulated run.
%The proposal will be in the direction of the gradient,
%using a standard normal distribution.
%
%It returns the updated parameter vector and the log of
%the posterior probability, to avoid recalculation.
%
%Arguments: oldpars= old parameter vector, should be a 4
%                    or 5 vector,
%           oldpost= value of posterior distribution

global alptrue deltrue gamtrue MMM NNN TT1 DD1 PRIORS PR_PROB
```

```
global CUTOFF ORDR NChain;



logold=oldpost;
h=0.000000005;

if (length(oldpars)==4)
   postfun3max=postfun3(mode,mdl);
   p1=oldpars(1);
   alph=oldpars(2);
   delt=oldpars(3);
   gamm=oldpars(4);
   SIG=0.00001*diag([0.1005,0.5995,0.1543,0.0346]);
   %SIG=0.000000000001*eye(4);
   delp1=(1/h)*(postfun3([p1+h;alph;delt;gamm],mdl)-logold);
   delalp=(1/h)*(postfun3([p1;alph+h;delt;gamm],mdl)-logold);
   deldel=(1/h)*(postfun3([p1;alph;delt+h;gamm],mdl)-logold);
   delgam=(1/h)*(postfun3([p1;alph;delt;gamm+h],mdl)-logold);
   gradold=[delp1;delalp;deldel;delgam];
   dir=gradold/norm(gradold,2);
   step=(0.00000000000005)*(postfun3max-logold+1);
   mutheta=oldpars+step*dir;

   thetaprop=mvnrnd(mutheta,SIG,1)';
```

```
if min(thetaprop)<0 | max(thetaprop)>1
   newpars=oldpars;
   newpost=logold;
else
   p1new=thetaprop(1);
   alpnew=thetaprop(2);
   delnew=thetaprop(3);
   gamnew=thetaprop(4);
   lognew=postfun3(thetaprop,1);
   delp1new=(1/h)*(postfun3([p1new+h;...
   ...alpnew;delnew;gamnew],mdl)-lognew);
   delalpnew=(1/h)*(postfun3([p1new;...
   ...alpnew+h;delnew;gamnew],mdl)-lognew);
   deldelnew=(1/h)*(postfun3([p1new;...
   ...alpnew;delnew+h;gamnew],mdl)-lognew);
   delgamnew=(1/h)*(postfun3([p1new;...
   ...alpnew;delnew;gamnew+h],mdl)-lognew);
   gradnew=[delp1new;delalpnew;deldelnew;delgamnew];
   dirnew=gradnew/norm(gradnew,2);
   stepnew=(0.00000000000005)*(postfun3max-lognew+1);
   munew=thetaprop+stepnew*dirnew;
   lalpha=lognew-logold+...
   ...logmvnkern(thetaprop,mutheta,SIG)-...
   ...logmvnkern(oldpars,munew,SIG);
```

```
        if log(rand(1,1))<lalpha
            newpars=thetaprop;
            newpost=lognew;
        else
            newpars=oldpars;
            newpost=logold;
        end
    end
elseif (length(oldpars)==5)
    postfun3max=postfun3(mode,mdl);
    p1=oldpars(1);
    p2=oldpars(2);
    alph=oldpars(3);
    delt=oldpars(4);
    gamm=oldpars(5);
    %SIG=0.000000000001*eye(5);
    SIG=0.00001*diag([0.1005,0.0781,0.5995,0.1543,0.0346]);
    delp1=(1/h)*(postfun3([p1+h;p2;alph;delt;gamm],mdl)-logold);
    delp2=(1/h)*(postfun3([p1;p2+h;alph;delt;gamm],mdl)-logold);
    delalp=(1/h)*(postfun3([p1;p2;alph+h;delt;gamm],mdl)-logold);
    deldel=(1/h)*(postfun3([p1;p2;alph;delt+h;gamm],mdl)-logold);
    delgam=(1/h)*(postfun3([p1;p2;alph;delt;gamm+h],mdl)-logold);
    gradold=[delp1;delp2;delalp;deldel;delgam];
    dir=gradold/norm(gradold,2);
```

```
step=(0.00000000000005)*(postfun3max-logold+1);
mutheta=oldpars+step*dir;

thetaprop=mvnrnd(mutheta,SIG,1)';
if min(thetaprop)<0 | max(thetaprop)>1
   newpars=oldpars;
   newpost=logold;
else
   p1new=thetaprop(1);
   p2new=thetaprop(2);
   alpnew=thetaprop(3);
   delnew=thetaprop(4);
   gamnew=thetaprop(5);
   lognew=postfun3(thetaprop,mdl);
   delp1new=(1/h)*(postfun3([p1new+h;p2new;...
   ...alpnew;delnew;gamnew],mdl)-lognew);
   delp2new=(1/h)*(postfun3([p1new;p2new+h;...
   ...alpnew;delnew;gamnew],mdl)-lognew);
   delalpnew=(1/h)*(postfun3([p1new;p2new;...
   ...alpnew+h;delnew;gamnew],mdl)-lognew);
   deldelnew=(1/h)*(postfun3([p1new;p2new;...
   ...alpnew;delnew+h;gamnew],mdl)-lognew);
   delgamnew=(1/h)*(postfun3([p1new;p2new;...
   ...alpnew;delnew;gamnew+h],mdl)-lognew);
```

```
      gradnew=[delp1new;delp2new;delalpnew;deldelnew;delgamnew];
      dirnew=gradnew/norm(gradnew,2);
      stepnew=(0.00000000000005)*(postfun3max-lognew+1);
      munew=thetaprop+stepnew*dirnew;
      lalpha=lognew-logold+logmvnkern(thetaprop,mutheta,SIG)...
      ...-logmvnkern(oldpars,munew,SIG);
      if log(rand(1,1))<lalpha
         newpars=thetaprop;
         newpost=lognew;
      else
         newpars=oldpars;
         newpost=logold;
      end
   end
end

%%%
```

REFERENCES

[1] A. Akharif and M. Hallin. Efficient detection of random coefficients in autoregressive models. *The Annals of Statistics*, 31(2):675–704, 2004.

[2] R. Bartoszynski and M. Niewiadomsk-Bugaj. *Probability and Statistical Inference*. Wiley Series in Probability and Mathematical Statistics. Wiley, New York, NY, 1996.

[3] L. Berliner. Hierarchical bayesian time series models, 1996.

[4] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society Series B (Methodological)*, 36(2):192–236, 1974.

[5] J. Besag, P.J. Green, D. Higdon, and K. Mengerson. Bayesian computation and stochastic systems. *Statistical Science*, 10(1):3–66, 1995.

[6] J. Besag, J. york, and A. Mollie. Bayesian image restoration, with applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43:1–59, 1991.

[7] G.E.P. Box and G.C. Tiao. *Bayesian Inference in Statistical Analysis*. Wiley Classics Library. Wiley-Interscience, New York, NY, 1973.

[8] F. Brauer and J.A. Nohel. *The Qualitative Theory of Ordinary Differential Equations*. Dover Publications, Mineola, NY, 1969.

[9] L. Breyer and G. Roberts. From metropolis to diffusions: Gibbs states and optimal scaling, 1998.

[10] CDC. *Centers for Disease Control and Prevention Home Page*. World Wide Web, `http://www.cdc.gov`, 2004.

[11] M.H. Chen. Markov chain monte carlo sampling for evaluating multidimensional integrals with application to bayesian computation, 1997.

[12] O.F. Christensen, G.O. Roberts, and M. Skld. Robust mcmc methods for spatial glmm's. Centre for mathematical sciences technical report 2003:23, Lund University, Lund, Sweden, 2003.

[13] K.L. Chung and R.J. Williams. *Introduction to Stochastic Integration, 2nd Ed.* Probability and its Applications. Birkhauser, Boston, MA, 1990.

[14] Colorado. *Colorado Division of Wildlife Hunting Home Page.* World Wide Web, `http://wildlife.state.co.us/hunt/`, 2004.

[15] M.M. Conner, C.W. McCarty, and M.W. Miller. Detection of bias in harvest-based estimates of chronic wasting disease prevalence in mule deer. *Journal of Wildlife Diseases*, 36(4):661–699, 2000.

[16] N. Cressie. *Statistics for Spatial Data.* Wiley-Interscience, New York, 1993.

[17] N. Cressie and A.S. Mugglin. *Spatio-temporal Hierarchical Modeling of an Infectious Disease from (Simulated) Count Data.* COMPSTAT: Proceedings in Computational Statistics. Physica-Verlag, Heidelberg, 2000.

[18] J.H. Cushman and T.R. Ginn. Reactive contaminant transport in the saturated zone, 1998.

[19] Edinburgh Mathematical Society. *Applications of Mathematics to Medical Problems*, volume 44 of *The Proceedings of the Edinburgh Mathematical Society*, 1926.

[20] B. Eraker. Mcmc analysis of diffusion models with application to finance. *Journal of Business and Economic Statistics*, 19(2):177–191, 2001.

[21] Institute for Historical Research. *Institute for Historical Research Victorian Page.* World Wide Web, `http://www.history.ac.uk/ihr/Focus/Victorians/`, 2004.

[22] The CWD Foundation. *The CWD Foundation Home Page.* World Wide Web, `http://www.stopcwd.org`, 2004.

[23] W.A. Fuller. *Introduction to Statistical Time Series.* Wiley-Interscience, New York, NY, 1996.

[24] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis.* Texts in Statistical Science. Chapman and Hall/CRC Press, New York, NY, 1995.

[25] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice.* Interdisciplinary Statistics. Chapman and Hall/CRC Press, New York, NY, 1999.

[26] B.T. Grinfell. Chance and chaos in measles dynamics. *Journal of the Royal Statistical Society Series B (Methodological)*, 54(2):383–398, 1992.

[27] J.E. Gross and M.W. Miller. Chronic wasting disease in mule deer: Disease dynamics and control. *Journal of Wildlife Management*, 65(2):205–215, 2001.

[28] P. Gustafson, Y.C. Macnab, and S. Wen. On the value of derivative evaluations and random walk suppression in markov chain monte carlo algorithms. *Statistics and Computing*, 14:23–38, 2004.

[29] M. Haran, J.S. Hodges, and B.P. Carlin. Accelerating computation in markov random fields for spatial data via structured mcmc. *Journal of Computational and Graphical Statistics*, 12(2):249–264, 2003.

[30] L. Held. Simultaneous posterior probability statements from monte carlo output. *Journal of Computational and Graphical Statistics*, 13(1):20–35, 2004.

[31] N.T. Hobbs, M.W. Miller, E.S. Williams, K.P. Burnham, R.M. Reich, J.W. Thomas, and D.M Theobald. Spatial and temporal dynamics of prion disease in wildlife: Responses to changing land use. Natural resource ecology laboratory 2002 progress report, Colorado State University, Fort Collins, CO, 2002.

[32] N.T. Hobbs, M.W. Miller, E.S. Williams, K.P. Burnham, R.M. Reich, J.W. Thomas, and D.M Theobald. Spatial and temporal dynamics of prion disease in wildlife: Responses to changing land use. Natural resource ecology laboratory 2003 progress report, Colorado State University, Fort Collins, CO, 2003.

[33] F. Hoppensteadt. *Mathematical Theories of Populations: Demographics, Genetics, and Epidemics.* CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1975.

[34] B. Hrafnkelsson and N. Cressie. Hierarchical modeling of count data with application to nuclear fall-out. Department of statistics preprint no. 663, Ohio State University, Columbus, OH, 2000.

[35] Interface Foundation of North America. *A Spatial Model for Chronic Wasting Disease in Rocky Mountain Mule Deer*, The Proceedings of the 2003 Interface Conference, 2003.

[36] H. Jeffreys. *Theory of Probability.* Clarendon Press, Oxford, 1939.

[37] C. Johns and C.H. Mehl. A dynamic spatial model for chronic wasting disease in colorado. Submitted for publication., 2004.

[38] M.S. Kaiser, N. Cressie, and J. Lee. Spatial mixture models based on exponential family conditional distributions. Department of statistics preprint no. 655, Ohio State University, Columbus, OH, 1999.

[39] H. Knsch, S. Geman, and A. Kehagias. Hidden markov random fields. *The Annals of Probability*, 5(3):577–602, 1995.

[40] H.L. Koul and A. Schick. Adaptive estimation in a random coefficient autoregressive model. *The Annals of Statistics*, 24(3):1025–1052, 1996.

[41] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications, 5th Ed.* Universitext. Springer-Verlag, Milan, Italy, 1998.

[42] T.G. Kurtz. *Approximation of Population Processes.* CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1981.

[43] M. Lavine. Another look at conditionally gaussian markov random fields. Institute of statistics and decision sciences discussion series, Duke University, Durham, NC, 1998.

[44] J.S. Liu, W.H. Wong, and A. Kong. Covariance structure of the gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Journal of the American Statistical Association*, 89:958–966, 1994.

[45] A.J. Lotka. *Elements of Physical Biology.* Williams and Wilkins, Baltimore, 1925.

[46] A.J. Lotka. *Elements of Mathematical Biology.* Dover, New York, 1956.

[47] C.W. McCarty and M.W. Miller. A versatile model of disease transmission applied to forecasting bovine tuberculosis dynamics in white-tailed deer populations. *Journal of Wildlife Diseases*, 34(4):722–730, 1998.

[48] K.L. Mengersen and R.L. Tweedie. Rates of convergence of the hastings and metropolis algorithms, 1994.

[49] M.W. Miller, E.S. Williams, C.W. McCarty, T.R. Spraker, T.J. Kreeger, C.T. Larsen, and E.T. Thorne. Epizootiology of chronic wasting disease in free-ranging cervids in colorado and wyoming. *Journal of Wildlife Diseases*, 36(4):676–690, 2000.

[50] A. Mollie. *Bayesian mapping of disease*, chapter 20, pages 359–379. Chapman and Hall/CRC Press, 1999.

[51] D. Mollison. Spatial contact models for ecological and epidemic spread. *Journal of the Royal Statistical Society Series B (Methodological)*, 39(3):283–326, 1977.

[52] D. Mollison, V. Isham, and B. Grenfell. Epidemics: Models and data. *Journal of the Royal Statistical Society Series A*, 157(1):115–149, 1994.

[53] A.M. Mood, F.A. Graybill, and D.C. Boes. *Introduction to the Theory of Statistics, 3rd Ed.* McGraw Hill Series in Probability and Statistics. McGraw Hill, New York, NY, 1974.

[54] A.S. Mugglin, N. Cressie, and I. Gemmell. Hierarchical statistical modeling of influenza-epidemic dynamics in space and time. Department of statistics preprint no. 662, Ohio State University, Columbus, OH, 2000.

[55] S.G. Nash and A. Sofer. *Linear and Nonlinear Programming.* McGraw-Hill, New York, NY, 1996.

[56] National Academy of Sciences. *On the Rate of Growth of the Population of the United States since 1790 and its Mathematical Representation*, volume 6 of *The Proceedings of the National Academy of Sciences*, 1920.

[57] National Academy of Sciences. *The Stability of the Normal Age Distribution*, volume 8 of *The Proceedings of the National Academy of Sciences*, 1922.

[58] R.M. Neal. *Probabilistic Inference Using Markov Chain Monte Carlo Methods.* PhD thesis, University of Toronto, Toronto, Canada, September 1993.

[59] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computing Journal*, 7:308–313, 1965.

[60] P.D. O'neill. Perfect simuation for reed frost epidemic models. *Statistics and Computing*, 13:37–44, 2003.

[61] S. James Press. *Bayesian Statistics: Principles, Models, and Applications.* Wiley Series in Probability and Mathematical Statistics. Wiley, New York, NY, 1989.

[62] Imprecise Probabilities Project. *Imprecise Probabilities Project Home Page.* World Wide Web, `http://ippserv.rug.ac.be`, 2004.

[63] E. Renshaw. *Modeling Biological Populations in Space and Time.* Cambridge Studies in Mathematical Biology. Cambridge University Press, Cambridge, 1991.

[64] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer Texts in Statistics. Springer-Verlag, New York, NY, 1999.

[65] G. Roberts and J. Rosenthal. Optimal scaling of discrete approximations to langevin diffusions, 1995.

[66] G.O. Roberts and J.S. Rosenthal. Quantitative bounds for convergence rates of continuous time markov processes. *Electronic Journal of Probability*, 1(9):1–21, 1996.

[67] G.O. Roberts and O. Stramer. Tempered langevin diffusions and algorithms. Department of statistics and actuarial science technical report no. 314, University of Iowa, Iowa City, IA, 2002.

[68] S.M. Ross. *Introduction to Probability Models, 6th Ed.* Academic Press, San Diego, CA, 1997.

[69] E.M. Schauber and A. Woolf. Chronic wasting disease in deer and elk: a critique of current models and their application. *Wildlife Society Bulletin*, 31:610–616, 2003.

[70] Ø. Skare, F.E. Benth, and A. Frigessi. Weak convergence and optimal scaling of random walk metropolis algorithms. *Annals of Applied Probability*, 7:110–120, 1997.

[71] Ø. Skare, F.E. Benth, and A. Frigessi. Smoothed langevin proposals in metropolis-hastings algorithms. *Statistics and Probability Letters*, 49:345–354, 2000.

[72] R.L. Smith. The hit-and-run sampler: A globally reaching markov chain sampler for generating arbitrary multivariate distributions. *Proceedings of the 1996 Winter Simulation Conference*, pages 260–264, 1996.

[73] S.M. Stigler. *The History of Statistics: The Measurement of Uncertainty before 1900.* Belknap Press of Harvard University Press, Cambridge, MA, 1986.

[74] P.D. Stolley and T. Lasky. *Investigating Disease Patterns: The Science of Epidemiology.* Scientific American Library, New York, NY, 1995.

[75] S.H. Strogatz. *Nonlinear Dynamics and Chaos.* Addison-Wesley, Reading, MA, 1994.

[76] L. Tierney. Markov chains for exploring posterior distributions. *The Annals Statistics*, 22(4):1701–1762, 1994.

[77] V. Volterra. Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560, 1926.

[78] L.A. Waller, B.P. Carlin, H. Xia, and A.E. Gelfand. Hierarchical spatio-temporal mapping of disease rates. *Journal of the American Statistical Association*, 92(438):607–617, 1997.

[79] P. Waltman. *Competition Models in Population Biology.* CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1983.

[80] A.H. Welsh. *Aspects of Statistical Inference.* Wiley Series in Probability and Statistics. Wiley Interscience, New York, New York, 1996.

[81] C. Wikle, L. Berliner, and N. Cressie. Hierarchical bayesian space-time models. *Environmental and Ecological Statistics*, 5(2):117–154, 1998.

[82] C.K. Wikle, R.F. Milliff, D. Nychka, and L.M. Berliner. Spatiotemporal hierarchical bayesian modeling: Tropical ocean surface winds. *Journal of the American Statistical Association*, 96(454):382–397, 2001.

[83] D. Zwillinger. *The handbook of Differential Equations, 2nd ed.* Academic Press, San Diego, CA, 1992.