Bayesian Prediction and Inference in Analysis of Computer Experiments

by

Hao Chen

B.Sc., Renmin University of China, 2011

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate Studies

(Statistics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2013

Abstract

Gaussian Processes (GPs) are commonly used in the analysis of data from a computer experiment. Ideally, the analysis will provide accurate predictions with correct coverage probabilities of credible intervals. A Bayesian method can, in principle, capture all sources of uncertainty and hence give valid inference. Several implementations are available in the literature, differing in choice of priors, etc. In this thesis, we first review three popular Bayesian methods in the analysis of computer experiments. Two prediction criteria are proposed to measure both the prediction accuracy and the prediction actual coverage probability. From a simple example, we notice that the performances of the three Bayesian implementations are quite different. Motivated by the performance difference, we specify four important factors in terms of Bayesian analysis and allocate different levels for the factors based on the three existing Bayesian implementations. Full factorial experiments are then conducted on the specified factors both for real computer models and via simulation with the aim of identifying the significant factors. Emphasis is placed on the prediction accuracy, since the performances of the prediction coverage probability for most combinations are satisfactory.

Through the analyses described above, we find that among the four factors, two factors are actually significant to the prediction accuracy. The best combination for the levels of the four factors is also identified.

Preface

This thesis is original, unpublished work by the author, Hao Chen.

Table of Contents

Abstract ii

Preface iii

Table of Contents iv

List of Tables vii

List of Figures viii

Acknowledgements xi

Dedication xii

1 Introduction 1
- 1.1 The Structure of the Thesis 2
- 1.2 Related Work: Three Key Ingredients 2
  - 1.2.1 Bayesian Modelling 3
  - 1.2.2 Markov Chain Monte Carlo 4
  - 1.2.3 Stationary Gaussian Processes 8

2 Review for Existing Methods 11
- 2.1 Method of Maximum Likelihood Estimation 11
- 2.2 The General Picture for Bayesian Estimation Methods . . . 12
- 2.3 Higdon's Bayesian Method 13
- 2.4 Method of Gaussian Emulation Machine 15
- 2.5 Method of Treed Gaussian Process 17

3 Tentative Comparison of the Bayesian Methods 19
- 3.1 Two Prediction Criteria 19
- 3.2 Details for the Tentative Comparison 20
  - 3.2.1 Data 20
  - 3.2.2 Comparison Details 21

3.3 Comparison Results . . . 28
3.3.1 Prediction Accuracy . . . 28
3.3.2 Actual Coverage Probability . . . 28
3.4 Conclusion & Discussion . . . 32

4 In-Depth Comparison: Real Computer Models . . . 34
4.1 Comparison Details . . . 34
4.1.1 Four Important Factors . . . 34
4.1.2 Full Factorial Design . . . 39
4.1.3 Some Mathematics Details . . . 41
4.1.4 Other Details for the In-Depth Comparison . . . 42
4.2 Real Computer Model 1: Borehole . . . 43
4.2.1 Prediction Accuracy . . . 44
4.2.2 Actual Coverage Probability . . . 49
4.3 Real Computer Model 2: PTW . . . 53
4.3.1 Prediction Accuracy . . . 53
4.3.2 Actual Coverage Probability . . . 59
4.4 Real Computer Model 3: G-protein . . . 63
4.4.1 Prediction Accuracy . . . 63
4.4.2 Actual Coverage Probability . . . 68
4.5 Summary of the Real Computer Models . . . 72

5 In-Depth Comparison: Simulation Study . . . 73
5.1 Details for the Simulation Study . . . 73
5.1.1 Data Generating Procedures . . . 73
5.1.2 Choice of True θ_i . . . 74
5.2 Simulation Scenario 1 . . . 75
5.2.1 Prediction Accuracy . . . 75
5.2.2 Actual Coverage Probability . . . 75
5.3 Simulation Scenario 2 . . . 78
5.3.1 Prediction Accuracy . . . 78
5.3.2 Actual Coverage Probability . . . 81
5.4 Summary of the Simulation Study . . . 81

6 Conclusion & Discussion . . . 82

Bibliography . . . 85

Appendices

A Posterior Distribution of θ_i for $\pi(\sigma^2) = IG(\alpha_1, \alpha_2)$ 88

B Posterior Distribution of θ_i for $\pi(\sigma^2) \propto 1/\sigma^2$ 91

List of Tables

3.1 Acceptance rates for different choices of MCMC length. . . . 28
3.2 Average actual coverage probability, 90% true CP. 32
3.3 Average actual coverage probability, 95% true CP. 32

4.1 Full factorial design. 40
4.2 2^3 factorial design, given any level of the prior on θ_i. 41
4.3 ANOVA results for the Borehole computer model. 47
4.4 ANOVA results for the PTW computer model. 57
4.5 ANOVA results for the G-protein computer model. 66

5.1 Actual coverage probability, 90% true CP, scenario 1. 75
5.2 Actual coverage probability, 95% true CP, scenario 1. 78
5.3 Actual coverage probability, 90% true CP, scenario 2. 81
5.4 Actual coverage probability, 95% true CP, scenario 2. 81

List of Figures

1.1 Trace plot for alpha, beta and tau. 7

3.1 Traceplot for 15000 total samples. 22
3.2 Traceplot for 30000 total samples. 23
3.3 Traceplot for 60000 total samples. 24
3.4 Empirical densities for 15000 total samples. 25
3.5 Empirical densities for 30000 total samples. 26
3.6 Empirical densities for 60000 total samples. 27
3.7 Results for prediction accuracy. 29
3.8 Results for actual coverage probability, 90% true CP. 30
3.9 Results for actual coverage probability, 95% true CP. 31

4.1 Density plot for Higdon's prior. 36
4.2 Density plot for GEM's prior. 37
4.3 Density plot for TGP's prior. 38
4.4 Prediction accuracy, given Higdon's prior on θ_i, Borehole. . . 44
4.5 Prediction accuracy, given GEM's prior on θ_i, Borehole. . . . 45
4.6 Prediction accuracy, given TGP's prior on θ_i, Borehole. . . . 46
4.7 Average prediction accuracy, given Higdon's prior on θ_i, Borehole. 48
4.8 Average prediction accuracy, given GEM's prior on θ_i, Borehole. 48
4.9 Average prediction accuracy, given TGP's prior on θ_i, Borehole. 49
4.10 Prediction actual coverage probability, given Higdon's prior on θ_i, 90% true CP, Borehole. 50
4.11 Prediction actual coverage probability, given GEM's prior on θ_i, 90% true CP, Borehole. 50
4.12 Prediction actual coverage probability, given TGP's prior on θ_i, 90% true CP, Borehole. 51
4.13 Prediction actual coverage probability, given Higdon's prior on θ_i, 95% true CP, Borehole. 51
4.14 Prediction actual coverage probability, given GEM's prior on θ_i, 95% true CP, Borehole. 52

4.15 Prediction actual coverage probability, given TGP's prior on θ_i, 95% true CP, Borehole. 52
4.16 Prediction accuracy, given Higdon's prior on θ_i, PTW. 54
4.17 Prediction accuracy, given GEM's prior on θ_i, PTW. 55
4.18 Prediction accuracy, given TGP's prior on θ_i, PTW. 56
4.19 Average prediction accuracy, given Higdon's prior on θ_i, PTW. 58
4.20 Average prediction accuracy, given GEM's prior on θ_i, PTW. 58
4.21 Average prediction accuracy, given TGP's prior on θ_i, PTW. 59
4.22 Prediction actual coverage probability, given Higdon's prior on θ_i, 90% true CP, PTW. 60
4.23 Prediction actual coverage probability, given GEM's prior on θ_i, 90% true CP, PTW. 60
4.24 Prediction actual coverage probability, given TGP's prior on θ_i, 90% true CP, PTW. 61
4.25 Prediction actual coverage probability, given Higdon's prior on θ_i, 95% true CP, PTW. 61
4.26 Prediction actual coverage probability, given GEM's prior on θ_i, 95% true CP, PTW. 62
4.27 Prediction actual coverage probability, given TGP's prior on θ_i, 95% true CP, PTW. 62
4.28 Prediction accuracy, given Higdon's prior on θ_i, G-protein. . . 63
4.29 Prediction accuracy, given GEM's prior on θ_i, G-protein. . . . 64
4.30 Prediction accuracy, given TGP's prior on θ_i, G-protein. . . . 65
4.31 Average prediction accuracy, given Higdon's prior on θ_i, G-protein. 67
4.32 Average prediction accuracy, given GEM's prior on θ_i, G-protein. 67
4.33 Average prediction accuracy, given TGP's prior on θ_i, G-protein. 68
4.34 Prediction actual coverage probability, given Higdon's prior on θ_i, 90% true CP, G-protein. 69
4.35 Prediction actual coverage probability, given GEM's prior on θ_i, 90% true CP, G-protein. 69
4.36 Prediction actual coverage probability, given TGP's prior on θ_i, 90% true CP, G-protein. 70
4.37 Prediction actual coverage probability, given Higdon's prior on θ_i, 95% true CP, G-protein. 70
4.38 Prediction actual coverage probability, given GEM's prior on θ_i, 95% true CP, G-protein. 71

4.39 Prediction actual coverage probability, given TGP's prior on θ_i, 95% true CP, G-protein. 71

5.1 Norm-RMSE for TGP's prior versus Higdon's prior, scenario 1. 76
5.2 Norm-RMSE for GEM's prior versus Higdon's prior, scenario 1. 77
5.3 Norm-RMSE for TGP's prior versus Higdon's prior, scenario 2. 79
5.4 Norm-RMSE for GEM's prior versus Higdon's prior, scenario 2. 80

Acknowledgements

First, I would like to acknowledge my special debts to my supervisors, Professor William Welch and Professor Jason Loeppky, who lead me to the intriguing field of analysis of computer experiments and are always strongly supportive of my research career. It is my great pleasure to work with them. I also want to express my heartfelt gratitude to Professor James Zidek, who financially supported me during the summer of 2012 and provided lots of guidance and help to me. In addition, I owe a debt of gratitude to Professor Jerome Sacks, who accepted me as his co-author for a high quality journal paper and provided many support to me.

I would also express my gratitude to Professors Lang Wu, Jiahua Chen, Rollin Brant, Paul Gustafson, Matas Salibin-Barrera and Yew-Wei Lim for their constant support and excellent teaching. I am also grateful to Peggy Ng, Elaine Salameh, Zin Kurji and Andrea Sollberger for their hard work and kind help. Thanks to everyone for making the department such a amazing place.

Finally, I owe my special thanks to my parents for their support and understanding of my Master study.

Dedication

To my parents: Mr. Qindong Chen & Mrs. Xiuying Zhao,
who are so proud.

Chapter 1

Introduction

Many complex phenomena are extremely difficult to investigate through controlled physical experiments. Instead, computer experiments become important alternatives to provide insights into such phenomena. Nowadays, computer experiments have been successfully applied in many sciences and engineering fields, for instance climate change, where traditional laboratory-based experiments are impossible to conduct. In general, a computer experiment is a designed set of runs of computer codes, which usually have the following two distinguishing features: (1) deterministic, that is, repeating an identical input does not change the output. (2) time-consuming, a single run may take several hours to complete.

Consider a typical computer model, which was described by Gramacy and Lee (2008), Gramacy and Lee (2009). NASA was developing a new reusable rocket booster called the Langley Glide-Back Booster (LGBB). NASA has built a computer model, which is able to model the flight characteristics (lift, drag, pitch, side-force, yaw and roll) of the LGBB as a function of 3 inputs—side slip angle, mach number and angle of attack. For each input configuration triplet, the computer model will yield six response variables as described above. However, even for one set of inputs, it takes the computer model 5-20 hours on a high end workstation to solve the sets of inviscid Euler equations. A small modification of the input configuration means another 5-20 hours of run time. Hence, it will be very helpful to approximate the output of computer model with much less computation.

Suppose that we have n input configurations as well as the corresponding n outputs from a computer model. The primary goal in the analysis of computer experiments is to predict the output for untried inputs via a statistical model using the available data. The most popular method to emulate the computer code is to treat it as a realization from a Gaussian Process (GP) Sacks et al. (1989). Assuming the correlation parameters of the GP are known, the best linear unbiased predictor (BLUP) can be obtained by minimizing the Mean Square Error (MSE) of the predictor. In practice, one

must estimate the correlation parameters from the available data (n inputs and n outputs). Several estimation methods have been proposed so far and, in principle, those estimating methods can be classified into two categories based on their underlying paradigms: The first is the Frequentist-based Maximum Likelihood Estimation (MLE) and its improved versions, such as Welch et al. (1992), Ba and Joseph (2012); The second is the Bayesian-based estimation methods, such as the Treed GP proposed by Gramacy (2005) and the Bayesian method proposed by Higdon et al. (2004, 2008). In this thesis, we will emphasize the Bayesian methods but MLE will also be included for comparison.

1.1 The Structure of the Thesis

This thesis is dedicated to foundational aspects of the analysis of computer experiments. In Chapter 1, some related statistical theories, such as Bayes' theorem, will be introduced. Four popular parameters estimation methods will be reviewed in Chapter 2. Our emphasis is placed on the three methods that are based on the Bayesian framework. In Chapter 3, we will present the results of a tentative study, whose purpose is to compare the prediction performance of the three Bayesian methods. We observe that in terms of the prediction accuracy, one of the Bayesian methods performs poorly, while, the remaining two have relatively better performance.

Curiosity has driven us to explore the reasons why those Bayesian methods have such different prediction performances. Since different methods have different parameterizations as well as different prior distributions, we specify 4 important factors that are able to account for the performance difference and fix the other possible affecting factors. Each factor is assigned different levels. Full factorial experiments are then carried out on the specified factors both through real computer codes (in Chapter 4) and via simulation data (in Chapter 5). Several useful conclusions are drawn and the best combination for the levels of the factors is found. Some concluding remarks are made in Chapter 6.

1.2 Related Work: Three Key Ingredients

In this section, we will introduce three important ingredients for this recipe of comparing different Bayesian-based estimation methods. All three concepts are briefly outlined here. In some cases, in-depth analysis is post-

poned to later chapters. Readers who have already gained knowledge about Bayesian Modelling, Markov Chain Monte Carlo (MCMC) and Stationary Gaussian Processes may skip this section.

1.2.1 Bayesian Modelling

The basis of Bayesian modelling is the famous Bayes' theorem. In the simplest case, it states that for two events A and B, the probability of A happens conditional on event B, denoted as $P(A|B)$, can be expressed as $P(A \cap B)/P(B)$, provided $P(B) > 0$.

Let $\boldsymbol{\theta}$ denote the parameter(s) of a model. The *prior* distribution of $\boldsymbol{\theta}$ is $p(\boldsymbol{\theta})$. Given data Y, the *posterior* distribution for $\boldsymbol{\theta}$ is obtained by combining the prior with the likelihood $p(Y|\boldsymbol{\theta})$ by Bayes' theorem:

$$p(\boldsymbol{\theta}|Y) = \frac{p(Y|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(Y)}, \tag{1.1}$$

where $p(Y)$ is the marginal distribution for Y. In simple cases, it can be obtained by integrating out $\boldsymbol{\theta}$, i.e., $p(Y) = \int p(Y|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$.

The prior distribution on $\boldsymbol{\theta}$ contains the scientific prior knowledge about $\boldsymbol{\theta}$. However, in some circumstances, people do not have much meaningful prior information about the parameters, a vague (non-informative) prior will then be assumed. When combined with a likelihood, families of priors producing posterior distributions in the same family are called *conjugate*. Conjugate priors are very convenient in practice because they can lead to analytically tractable posterior distributions.

A significant benefit of Bayesian statistics is that it is able to fully quantify the uncertainty. The posterior distribution on $\boldsymbol{\theta}$ is a whole summary about the parameters adjusted by data Y, in contrast with the Frequentist-based MLE, which can only yield a point estimate and a standard error of the point estimate. Nevertheless, it does not mean the Bayesian approach is perfect, because it is often impossible to get an analytical expression for $p(\boldsymbol{\theta}|Y)$ for non-textbook examples. Details of Bayesian statistics, including its merits and drawbacks can be found in Hartigan (1964) and Robert (2001).

1.2.2 Markov Chain Monte Carlo

As we have pointed out, the posterior $p(\boldsymbol{\theta}|Y)$ is generally mathematically intractable. The problem largely lies in the marginal distribution of Y, which can, in principle, be obtained by $p(y) = \int p(y|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$. However, when $\boldsymbol{\theta}$ is a p-dimensional vector and p is relatively large, the high-dimensional integration becomes infeasible. A popular alternative to the intractable integration is to conduct posterior inference through simulation. Markov chain Monte Carlo(MCMC) is the standard choice(Gelman et al. (1995), Gilks et al. (1996)) for posterior inference by simulation, and is the ubiquitous tool for Bayesian inference in this thesis.

The main idea of MCMC is to establish a Markov chain whose stationary distribution is the desired posterior distribution, and then draw samples from that chain. The procedures for implementing MCMC are outlined below: Usually, we will ignore the initial stages of the chain. The initial stages are called the *burn-in* period. We then collect samples from the chain every t states, where t is the *thinning* parameter. By doing so, the independence of the samples is largely guaranteed. Currently, there are two popular MCMC algorithms: the Gibbs Sampler (Geman and Geman (1984)) and the Metropolis-Hastings (M-H) algorithm (Metropolis et al. (1953), Hastings (1970)), which is actually a particular case of the Gibbs Sampler. We will emphasize the M-H algorithm since all of the posterior inferences in Chapter 4 and Chapter 5 are done by it.

The Gibbs Sampler

The Gibbs Sampler is actually a sampler from the full conditionals. Suppose we wish to obtain a sample from the posterior distribution of $P(\theta_1, \ldots, \theta_d|Y)$. The Gibbs Sampler is able to successively and repeatedly simulate from the conditional distributions of each component (θ_i) given the other components. Under conditional conjugacy, the simulation step is usually straightforward. We outline the steps below.

- 0. Initialize with $\boldsymbol{\theta}^0 = \left(\theta_1^0, \ldots, \theta_d^0\right)$.
- 1.1 Simulate θ_1^1 from the conditional distribution of $\theta_1| \left(\theta_2^0, \ldots, \theta_d^0\right)$.
- 1.2 Simulate θ_2^1 from the conditional distribution of $\theta_2| \left(\theta_1^1, \theta_3^0, \ldots, \theta_d^0\right)$.
  ...

- 1.d Simulate θ_d^1 from the conditional distribution of $\theta_d | \left(\theta_1^1, \ldots, \theta_{d-1}^1\right)$.
- 2 Iterate the above procedures.

It is obvious that with the Gibbs Sampler, as long as the full conditional has a closed form, we can obtain a sample directly from the posterior distribution without worrying about any integrals. In practice, parameters with conditionally conjugate priors are usually sampled with Gibbs procedures.

Metropolis-Hastings Algorithm

The M-H algorithm can be viewed as a generalization of the Gibbs Sampler when the full conditional does not have a closed form. Still consider drawing samples from the posterior distribution of $p(\boldsymbol{\theta}|Y)$, which is proportional to $p(Y|\boldsymbol{\theta})p(\boldsymbol{\theta})$. Let $\boldsymbol{\theta}^i$ be the current state. The M-H algorithm proceeds by generating a new $\boldsymbol{\theta}^*$ from a transition (irreducible-aperiodic) kernel $q(\bullet|\boldsymbol{\theta}^i)$. The next state $\boldsymbol{\theta}^{i+1}$ is given by

$$\boldsymbol{\theta}^{i+1} = \begin{cases} \boldsymbol{\theta}^* & \text{with probability } \alpha \\ \boldsymbol{\theta}^i & \text{with probability } 1-\alpha \end{cases} \tag{1.2}$$

where α is

$$\alpha = \min\left\{1, \frac{p(Y|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^i|\boldsymbol{\theta}^*)}{p(Y|\boldsymbol{\theta}^i)p(\boldsymbol{\theta}^i)q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^i)}\right\} \tag{1.3}$$

and α is usually referred to as the M-H acceptance ratio. Iterating the above procedures, the constructed Markov chain converge to its stationary distribution, which is actually the desired posterior distribution. Posterior inference is conducted based on the samples collected from the chain.

In addition, there are two concepts that are important to ensure convergence of the M-H algorithm. The first one is the proposal density, which is the probability density of the transition kernel $q(\bullet|\boldsymbol{\theta}^i)$. The M-H algorithm works best if the proposal density matches the shape of the target distribution density. Since we do not know the shape of the desired posterior distribution, a uniform density has been used in the thesis. The second is the acceptance rate, which is the fraction of candidate draws that are accepted. We will cover some details of the acceptance rate in the next part.

Convergence Diagnostic

From the theory of MCMC, we expect the constructed Markov chain to eventually converge to the stationary distribution, which is also the target posterior distribution.

However, there is no guarantee that the chain has converged after H draws. Several methods have been proposed, both visual and statistical, to help judging convergence. Here we will introduce two ways to diagnose convergence: visual and statistical. The two methods have been applied to check convergence in the thesis.

Intuitively, a direct way to see if the chain has converged is to see how well the chain is moving around the parameter space. If the chain is fluctuating in the parameter space, it suggests the chain need more time to converge to the desired distribution. The **Traceplot** is a plot that draws the sample values of a parameter against the iteration number. It enables us to check whether our chain gets stuck in certain areas of the parameter space.

Here, we include a simple example to illustrate how a traceplot can help to judge convergence. This linear regression example was used by Smith (2007) to illustrate the usefulness of the R package **boa**. The example presents a regression analysis on five (x, y) observations: $(1, 1), (2, 3), (3, 3), (4, 3)$, and $(5, 5)$. The analysis was performed with the following Bayesian model:

$$y_i \sim N(\mu_i, \tau),$$

$$\mu_i = \alpha + \beta(x_i - \bar{x}),$$

where $\tau = 1/\sigma^2$ is the precision. The prior distributions for α, β and σ^2 are

$$\alpha \sim N(0, 0.0001),$$

$$\beta \sim N(0, 0.0001),$$

$$\tau \sim \text{Gamma}(0.001, 0.001).$$

Primary interest was placed on the posterior inferences for α, β and τ. In the paper of Smith (2007), two parallel chains of 200 iterations each were generated in separate runs of the MCMC sampler started at different initial values. We concentrate on the first chain. The traceplots for α, β and σ are drawn in figure 1.1. The three traceplots suggest after the first few steps,

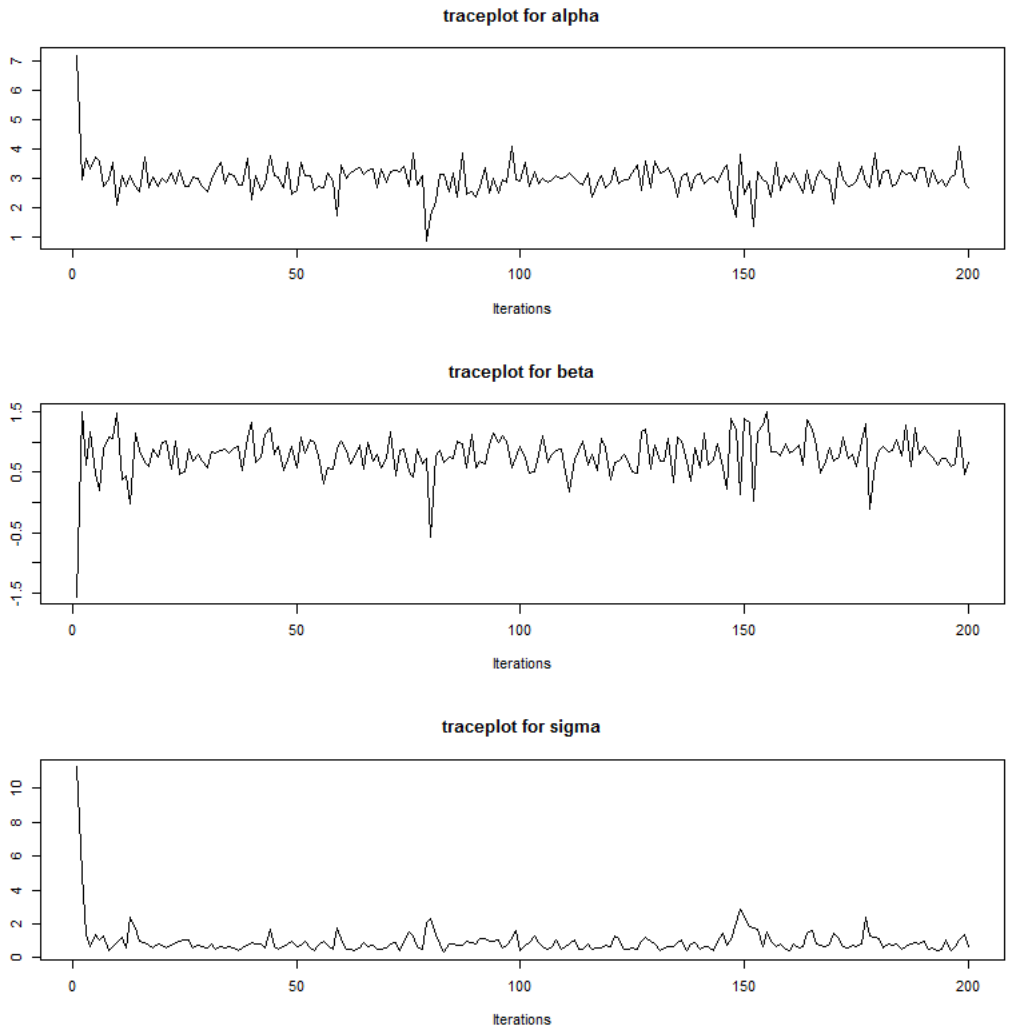

Figure 1.1: Trace plot for alpha, beta and tau.

the α, β and σ converge.

Besides the traceplot, the statistical indicator we use to help judging convergence is the acceptance rate, which was introduced before. If the acceptance rate is too high, like 80% , the successive samples will move around the space and the chain will converge very slowly to the desired distribution. If the acceptance rate is too low, like 10%, the proposals are likely to concentrate in regions of much lower probability density, and again the chain will converge very slowly. The desired acceptance rate depends on the target distribution. According to Roberts et al. (1997), the ideal acceptance rate for a one dimensional normal distribution is approximately 50% and the ideal acceptance rate for a multidimensional normal target distribution is approximately 23%. In the thesis, acceptance rate from 20%–50% will be considered sufficient to indicate convergence.

1.2.3 Stationary Gaussian Processes

Sacks et al. (1989) treated the observations of a deterministic computer code as if they were generated from the following model:

$$y(\mathbf{x}) = \boldsymbol{f}^T(\mathbf{x})\boldsymbol{\beta} + Z(\mathbf{x}). \tag{1.4}$$

Here, $\boldsymbol{x}$ is the vector of inputs to a computer model, $y(\boldsymbol{x})$ is the output, $\boldsymbol{f} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T$ contains k known regression functions, $\boldsymbol{\beta}$ is a vector of parameters with unknown values and Z(x) follows a Gaussian distribution with zero mean and unknown variance σ^2. Let $\boldsymbol{x}$ and $\boldsymbol{x'}$ be two sets of inputs. Here, we introduce 3 correlation structures between $Z(\boldsymbol{x})$ and $Z(\boldsymbol{x'})$. The key properties that we need are the correlation decreases as the distance l between $\boldsymbol{x}$ and $\boldsymbol{x'}$ increases with limiting value 1 when $l \to 0$ and limiting value 0 when $l \to \infty$.

- **Gaussian Correlation Structure**

$$R^{\text{Gauss}}(\boldsymbol{x}, \boldsymbol{x'}) = \exp\left\{-\sum_{h=1}^{p} \theta_h |x_h - x'_h|^2\right\}, \tag{1.5}$$

  where $\boldsymbol{\theta} > 0$. The correlation parameters($\boldsymbol{\theta}$) control how fast the correlation decays when distance increases. The smoothness parameters, which control the geometrical properties of the random field, are fixed at 2. The Gaussian Correlation structure is popular and is often used

in applications, see Gramacy and Lee (2008), Kennedy and O'Hagan (2001).

- **Power Exponential Correlation Structure**

$$R^{\text{PowerExp}}(\boldsymbol{x}, \boldsymbol{x'}) = \exp\left\{-\sum_{h=1}^{p} \theta_h |x_h - x'_h|^{m_h}\right\}, \tag{1.6}$$

where $\theta_h > 0$ and $m_h \in (0, 2]$. The difference between Power Exponential structure and Gaussian structure lies in the smoothness parameters. Instead of fixing the smoothness parameters, the Power Exponential structure introduces a new parameter $\boldsymbol{m} = \{m_1, m_2, \ldots, m_p\}$. In practice, m_h can either be treated as a constant or be treated as a hyperparameter.

- **Matern Correlation Structure**

$$R^{\text{Matern}}(\boldsymbol{x}, \boldsymbol{x'}) = \frac{1}{\Gamma(\mu)2^{\mu-1}} \left(\sqrt{2\mu}\frac{l}{\rho}\right)^{\mu} K_\mu \left(\sqrt{2\mu}\frac{l}{\rho}\right), \tag{1.7}$$

where l is the distance between $\boldsymbol{x}$ and $\boldsymbol{x'}$, $K_\mu()$ is the modified Bessel function and μ and ρ are non-negative parameters. As μ goes to infinity, the Matern structure will converge to the Gaussian Correlation Structure.

All of the three correlation structure mentioned above guarantee that the $n \times n$ correlation matrix R for the observations is symmetric and positive semidefinite. The diagonals of the correlation matrix R are all 1 and R is completely determined by the correlation parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_p\}$. In this thesis, we only consider the Gaussian correlation structure for its effectiveness and simplicity.

Running a computer model n times at input vectors $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$, ..., $\boldsymbol{x}^{(n)}$ produces n outputs $\boldsymbol{y} = \left(y(\boldsymbol{x}^{(1)}), y(\boldsymbol{x}^{(2)}), \ldots, y(\boldsymbol{x}^{(n)})\right)^T$. Given a new input configuration $\boldsymbol{x}^*$, we wish to predict $y(\boldsymbol{x}^*)$. According to Sacks et al. (1989) , the predictive distribution of $y(\boldsymbol{x}^*)$ conditional on $\boldsymbol{\beta}$, σ^2, $\boldsymbol{\theta}$ and $\boldsymbol{y}$ is Gaussian:

$$N\left(m(\boldsymbol{x}^*), v(\boldsymbol{x}^*)\right), \tag{1.8}$$

where

$$m(\boldsymbol{x}^*) = \boldsymbol{f}^T(\boldsymbol{x^*})\boldsymbol{\beta} + \boldsymbol{r}^T(\boldsymbol{x^*})\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F\beta}) \tag{1.9}$$

and

$$v(\boldsymbol{x}^*) = \sigma^2 \left(1 - \boldsymbol{r}^T(\boldsymbol{x^*})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x^*})\right). \tag{1.10}$$

Here, $\boldsymbol{F}$ is the $n \times k$ matrix with row i containing $\boldsymbol{f}^T(\boldsymbol{x}^i)$, the $n \times 1$ vector $\boldsymbol{r}(\boldsymbol{x^*})$ is obtained from (1.5) with element i given by $R(\boldsymbol{x}^*, \boldsymbol{x}^i)$ for all $i = \{1, \ldots, n\}$, and the $n \times n$ matrix $\boldsymbol{R}$ has element (i, j) from $R(\boldsymbol{x}^i, \boldsymbol{x}^j)$ for all $i = \{1, \ldots, n\}$ and $j = \{1, \ldots, n\}$.

We have shown that under (1.4) the predictive distribution of $y(\boldsymbol{x}^*)$ is a conditional multivariate normal distribution, if the $\boldsymbol{\beta}$, σ^2, $\boldsymbol{\theta}$ are known. In addition, the centre of the normal distribution is the usual kriging point predictor. Theoretically speaking, the $(1-\alpha)100\%$ confidence interval constructed according to (1.8) should give the correct coverage probability. In practice, however, one must estimate the model parameters, resulting in extra uncertainty that needs to be assessed.

In the next chapter, we will review four popular parameter estimation methods. Emphasis will be put on the three Bayesian methods. The Frequentist-based MLE will only be incorporated for comparison.

Chapter 2

Review for Existing Methods

It has been quite popular to model the output from a deterministic computer model as the realization from Gaussian Process. Based on (1.4), many researchers (Sacks et al. (1989), Higdon et al. (2004), Bastos and O'Hagan (2009)) assume the outputs $\boldsymbol{y}$ follow a multivariate normal distribution. The density of $\boldsymbol{y}$ given $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\theta}$ is

$$L(\boldsymbol{y}|\boldsymbol{\beta},\sigma^2,\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma^2)^{n/2}\det^{1/2}\boldsymbol{R}}\exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{F}\boldsymbol{\beta})^T\boldsymbol{R}^{-1}(\boldsymbol{y}-\boldsymbol{F}\boldsymbol{\beta})\right\}. \tag{2.1}$$

Theoretically speaking, people assume that the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\beta},\sigma^2,\boldsymbol{\theta}\}$ are known. However, one must estimate $\boldsymbol{\Theta}$ in practice. In this chapter, we will review four popular parameter estimation methods: the first one is MLE and the remaining three are Bayesian methods.

2.1 Method of Maximum Likelihood Estimation

Maximum Likelihood Estimation(MLE) is the most popular parameter estimation method within the frequentist paradigm. Based on (2.1), for any fixed $\boldsymbol{\theta}$, we take the first derivative with respect to $\boldsymbol{\beta}$ and σ^2 separately, and then set the consequent expressions to zero, the MLEs for $\boldsymbol{\beta}$ and σ^2 are obtained as

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{y} \tag{2.2}$$

and

$$\hat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}})^T\boldsymbol{R}^{-1}(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}}). \tag{2.3}$$

Please note that the MLEs for $\boldsymbol{\beta}$ and σ^2 are fully determined by $\boldsymbol{\theta}$. Plugging the MLEs of $\boldsymbol{\beta}$ and σ^2 into (2.1) gives the profile likelihood, which is only affected by $\boldsymbol{\theta}$.

$$\frac{1}{(2\pi\hat{\sigma}^2)^{n/2}\det^{1/2}(\boldsymbol{R})}\exp\left\{-\frac{n}{2}\right\}. \tag{2.4}$$

This profile likelihood need to be numerically maximized to yield that MLE for $\boldsymbol{\theta}$. There are several numerical method can achieve this goal, such as the Nelder-Mead Simplex (Nelder and Mead (1965)) and the Limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method (see, for example, Shanno and Kettler (1970)). Welch et al. (1992) also proposed an effective approach to get the MLE for $\boldsymbol{\theta}$. Since the thesis emphasizes Bayesian methods, we are not going to mention the details of the numerical maximization of $\boldsymbol{\theta}$. In addition, there are several statistical packages that are available for the MLE of $\boldsymbol{\theta}$, such as the *mlegp* package in R software and the GaSP software which was initiated by Welch.

Substituting the MLEs for $\boldsymbol{\beta}$ into (1.9), the predictive mean becomes

$$\hat{m}(\boldsymbol{x}^*) = \boldsymbol{f}^T(\boldsymbol{x^*})\hat{\boldsymbol{\beta}} + \boldsymbol{r}^T(\boldsymbol{x^*})\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}). \tag{2.5}$$

Since we use $\hat{\boldsymbol{\beta}}$ instead of the true $\boldsymbol{\beta}$, extra uncertainty has been introduced and the predictive variance in (1.10) becomes

$$\begin{aligned}\hat{v}_{\boldsymbol{\theta}}(\boldsymbol{x}^*) = \hat{\sigma}^2[1 - \boldsymbol{r}^T(\boldsymbol{x^*})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x^*})]+ \\ \hat{\sigma}^2[f(\boldsymbol{x}^*) - \boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}^*)]^T(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}[f(\boldsymbol{x}^*) - \boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}^*)].\end{aligned} \tag{2.6}$$

The extra uncertainty which is introduced by replacing $\hat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$ is nontrivial (see Abt (1999)). However, in practice, the extra uncertainty is often ignored. Consequently, the $(1-\alpha)100\%$ confidence interval based on (2.5) and (2.6) does not incorporate it, that is, the actual coverage probability shall be lower than the nominal coverage probability. The above analyses are in accordance with the results we observe in later chapters.

2.2 The General Picture for Bayesian Estimation Methods

Compared with MLE, Bayesian estimation methods are able to quantify the extra estimation uncertainty. GP parameters $\Theta = \{\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}\}$ are treated as hyperparameters within the Bayesian paradigm and different prior distributions are often assumed on them. The posterior inferences on the GP parameters are usually conducted through MCMC.

Let us begin by expressing the joint posterior distribution of the parameters $\Theta = \{\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}\}$ as

$$P(\Theta|Y) \propto P(Y|\Theta)P(\Theta). \tag{2.7}$$

This means that the joint posterior distribution of the parameters is proportional to the product of the joint prior distribution of the parameters and the likelihood. By assuming independent prior distributions, the joint prior distribution is given by

$$P(\Theta) = P(\boldsymbol{\theta})P(\sigma^2)P(\boldsymbol{\beta}). \tag{2.8}$$

Now, there are two approaches that one can take.

- The first approach is to analytically integrate $\boldsymbol{\beta}$ and σ^2 out from $p(\boldsymbol{\theta}, \boldsymbol{\beta}, \sigma^2|y)$ and sample from the resulting $P(\boldsymbol{\theta}|y)$ by the M-H algorithm.
- The second approach is to sample the full posterior using Gibbs Sampler on $\boldsymbol{\beta}$, σ^2 and M-H algorithm within Gibbs Sampler on the $\boldsymbol{\theta}$.

Although the underlying ideas are similar, the joint prior distribution $P(\Theta)$ is different for different Bayesian methods. In the next sections, we will review three popular Bayesian methods.

2.3 Higdon's Bayesian Method

The first Bayesian method we introduce here is the one that was proposed by Higdon et al. (2004, 2008). Thereafter, we will call this method **Higdon's method**. The prior distributions on the GP parameters Θ are listed below.

- An Inverse Gamma (IG (α_1, α_2)) distribution on σ^2.
- A Uniform distribution on β_i, for $i = \{1, \ldots, k\}$.
- A Beta distribution on ρ_i, where $\rho_i = \exp(-\theta_i/4)$, for $i = \{1, \ldots, d\}$.

In terms of the correlation parameters $\boldsymbol{\theta}$, we notice that Higdon's method, in fact, transforms the θ_i to a ρ_i correlation scale. After a simple transformation of variables, this is equivalent to saying that the method assumes an independent prior on each θ_i with the following density

$$\pi(\theta_i) = \frac{1}{8}\exp(-\theta_i/4)\frac{1}{\sqrt{1-\exp(-\theta_i/4)}}. \tag{2.9}$$

By integrating out $\boldsymbol{\beta}$ and σ^2, the marginal posterior distribution of $\boldsymbol{\theta}$ is given by

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{y}) &\propto \int\int p(\boldsymbol{\theta},\boldsymbol{\beta},\sigma^2|\boldsymbol{y})L(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{\beta},\sigma^2)d\boldsymbol{\beta}d\sigma^2 \\ &\propto \frac{p(\boldsymbol{\theta})}{(\alpha_2+\frac{n-k}{2}\hat{\sigma^2_{\boldsymbol{\theta}}})^{(\alpha_1+\frac{n-k}{2})}\det^{1/2}(\boldsymbol{R})\det^{1/2}(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})}, \end{aligned} \tag{2.10}$$

where $p(\boldsymbol{\theta})$ is the prior distribution on $\boldsymbol{\theta}$ and

$$\hat{\sigma}^2_{\boldsymbol{\theta}} = \frac{1}{n-k}(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}})^T\boldsymbol{R}^{-1}(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}}),$$

$$\hat{\boldsymbol{\beta}} = (F^tR^{-1}F)^{-1}F^tR^{-1}\boldsymbol{y}.$$

The Metropolis-Hastings algorithm is then applied to obtain samples from the posterior distribution of $\boldsymbol{\theta}$.

With the prior specified above, the predictive distribution of $p(\boldsymbol{y}^*|\boldsymbol{y},\boldsymbol{\theta})$ is a non-central t distribution with $n-k+2\alpha_1$ degree of freedom. That is,

$$p(y(\boldsymbol{x}^*)|\boldsymbol{\theta},\boldsymbol{y}) \sim t(\hat{m}(\boldsymbol{x}^*),\hat{v}_\theta(\boldsymbol{x}^*)),$$

where

$$\hat{m}(\boldsymbol{x}^*) = \boldsymbol{f}^T(\boldsymbol{x^*})\hat{\boldsymbol{\beta}} + \boldsymbol{r}^T(\boldsymbol{x^*})\boldsymbol{R}^{-1}(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}}) \tag{2.11}$$

and

$$\hat{v}_{\boldsymbol{\theta}}(\boldsymbol{x}^*) = \left(\frac{(n-1)\hat{\sigma}^2_{\boldsymbol{\theta}}+2\alpha_2}{n-1+2\alpha_1}\right)\left(1-\boldsymbol{r}^T(\boldsymbol{x^*})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x^*})+\boldsymbol{h}^T(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{h}\right), \tag{2.12}$$

with $\boldsymbol{h} = f(\boldsymbol{x}^*) - \boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}^*)$. In order for readers to gain a better understanding on how Higdon's method is used to make predictions, we briefly mention the procedures here.

Repeat the following steps M times, starting at $t=1$

1. For each dimension $i=1,2,\ldots,p$, at step t, one first transforms θ_i^t to ρ_i^t.

2. Using an uniform transition function, either $U(\rho_i^t-0.5w,\rho_i^t+0.5w)$ or $U(\frac{3}{4}\rho_i^t,\frac{4}{3}\rho_i^t)$, to generate a candidate ρ_i^*. w is a specified step-width.

3. Let $\boldsymbol{\rho}^* = (\rho_1^t, \rho_2^t, \ldots, \rho_{i-1}^t, \rho_i^*, \rho_{i+1}^t, \ldots, \rho_p^t)$ and $\boldsymbol{\rho}^t = (\rho_1^t, \rho_2^t, \ldots, \rho_{i-1}^t, \rho_i^t, \rho_{i+1}^t, \ldots, \rho_p^t)$.

4. Generating a number ω from $U(0,1)$.

5. If $\log(\omega) < \log(p(\boldsymbol{\rho}^*|\boldsymbol{y})) - \log(p(\boldsymbol{\rho}^t|\boldsymbol{y}))$, set $\rho_i^{t+1} = \rho_i^*$; otherwise, set $\rho_i^{t+1} = \rho_i^t$.

Here, $p(\boldsymbol{\rho}^*|\boldsymbol{y})$ and $p(\boldsymbol{\rho}^t|\boldsymbol{y})$ are computed based on (2.10). After repeating the MCMC procedure for M times, one can get a $M \times p$ matrix with the i th row containing $\boldsymbol{\rho}^i$. For each $\boldsymbol{\rho}^i$, one converts it back to $\boldsymbol{\theta}^i$ and computes $\hat{m}^i(\boldsymbol{x}^*)$ based on (2.11) and $\hat{v}^i_{\boldsymbol{\theta}}(\boldsymbol{x}^*)$ based on (2.12). The final predictor is defined as

$$\hat{m}(\boldsymbol{x}^*) = \bar{m}(\boldsymbol{x}^*) = \frac{1}{M}\sum_{i=1}^{M} \hat{m}^i(\boldsymbol{x}^*).$$

The final predictive variance is obtained through the law of total variance. That is,

$$\hat{v}_{\boldsymbol{\theta}}(\boldsymbol{x}^*) = \frac{1}{M}\sum_{i=1}^{M} \hat{v}^i_{\boldsymbol{\theta}}(\boldsymbol{x}^*) + \frac{1}{M-1}\sum_{i=1}^{M}[\hat{m}^i(\boldsymbol{x}^*) - \bar{m}(\boldsymbol{x}^*)]^2.$$

The above Bayesian MCMC approach takes the extra uncertainty into account. Therefore, the $(1-\alpha)100\%$ credible interval constructed from (2.11) and (2.12) should have a much more appropriate actual coverage probability compared with the confidence interval constructed by MLE. This also agrees with the results we obtain in Chapter 4.

2.4 Method of Gaussian Emulation Machine

Kennedy (2004) proposed another Bayesian approach to emulate real computer models. Software called Gaussian Emulation Machine (GEM) was also created according to this method, which we will refer to as **GEM** in the thesis. After examining the GEM method carefully, we notice that it is quite similar to Higdon's method. The difference is that GEM assumes different prior distributions on $\Theta = \{\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}\}$. The priors assumed are listed below.

- A Jeffreys prior on σ^2 i.e., $P(\sigma^2) \propto 1/\sigma^2$.
- A Uniform distribution on β_i.

- An Exponential distribution on θ_i.

In a quite similar way, by integrating out the $\boldsymbol{\beta}$ and σ^2, the marginal posterior distribution of $\boldsymbol{\theta}$ is given by

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{y}) &\propto \int\int p(\boldsymbol{\theta}, \boldsymbol{\beta}, \sigma^2|\boldsymbol{y}) L(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\beta}, \sigma^2) d\boldsymbol{\beta} d\sigma^2 \\ &\propto \frac{p(\boldsymbol{\theta})}{(\hat{\sigma^2_{\boldsymbol{\theta}}})^{\frac{n-k}{2}} \det^{1/2}(\boldsymbol{R}) \det^{1/2}(\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F})}, \end{aligned} \tag{2.13}$$

where $p(\boldsymbol{\theta})$ is the prior distribution on $\boldsymbol{\theta}$ and

$$\hat{\sigma}^2_{\boldsymbol{\theta}} = \frac{1}{n-k}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})^T \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}),$$

$$\hat{\boldsymbol{\beta}} = (F^t R^{-1} F)^{-1} F^t R^{-1} \boldsymbol{y}.$$

Sampling from the posterior distribution of $\boldsymbol{\theta}$ can be done through the Metropolis-Hastings algorithm.

With the prior specified above, the predictive distribution of $p(\boldsymbol{y}^*|\boldsymbol{y}, \boldsymbol{\theta})$ is a non-central t distribution with $n-k$ degree of freedom. That is,

$$p(y(\boldsymbol{x}^*)|\boldsymbol{\theta}, \boldsymbol{y}) \sim t(\hat{m}(\boldsymbol{x}^*), \hat{v}_\theta(\boldsymbol{x}^*)),$$

where

$$\hat{m}(\boldsymbol{x}^*) = \boldsymbol{f}^T(\boldsymbol{x^*})\hat{\boldsymbol{\beta}} + \boldsymbol{r}^T(\boldsymbol{x^*})\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}) \tag{2.14}$$

and

$$\hat{v}_{\boldsymbol{\theta}}(\boldsymbol{x}^*) = \hat{\sigma}^2_{\boldsymbol{\theta}} \left(1 - \boldsymbol{r}^T(\boldsymbol{x^*})\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x^*}) + \boldsymbol{h}^T(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{h}\right), \tag{2.15}$$

with $\boldsymbol{h} = f(\boldsymbol{x}^*) - \boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}^*)$. We will skip the detailed MCMC steps, since the procedures will be almost identical to the procedures mentioned in Higdon's method. After repeating the MCMC procedure M times, one can get an $M \times p$ matrix with the s th row containing $\boldsymbol{\theta}^s$. For each $\boldsymbol{\theta}^s$, one computes $\hat{m}^s(\boldsymbol{x}^*)$ based on (2.14) and $\hat{v}^s_{\boldsymbol{\theta}}(\boldsymbol{x}^*)$ based on (2.15). The final predictor is defined as

$$\hat{m}(\boldsymbol{x}^*) = \bar{m}(\boldsymbol{x}^*) = \frac{1}{M}\sum_{i=1}^{M} \hat{m}^i(\boldsymbol{x}^*).$$

The final predictive variance is

$$\hat{v}_{\boldsymbol{\theta}}(\boldsymbol{x}^*) = \frac{1}{M}\sum_{i=1}^{M}\hat{v}^i_{\boldsymbol{\theta}}(\boldsymbol{x}^*) + \frac{1}{M-1}\sum_{i=1}^{M}[\hat{m}^i(\boldsymbol{x}^*) - \bar{m}(\boldsymbol{x}^*)]^2.$$

2.5 Method of Treed Gaussian Process

Gramacy (2005) proposed a treed GP hierarchical approach for prediction in computer experiments. Generally speaking, he divides the input vectors $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_n$ into different regions and fits independent hierarchical GP model within each region. The partitioning is recursive, that is a new partition is a sub-partition of a previous partition just like a tree develops more and more branches. Predictions are made conditional on the tree structure and will be averaged out finally to yield an ultimate predictor. Here, we introduce the hierarchical GP model without doing partition, i.e., we fit one hierarchical GP model for the whole input space. Readers can imagine our "tree" has no branches at all. Hereafter, we will refer to this method as **TGP**.

The TGP method specifies a new correlation structure

$$K(\boldsymbol{x}, \boldsymbol{x'}) = R^*(\boldsymbol{x}, \boldsymbol{x'}) + g\delta_{\boldsymbol{x},\boldsymbol{x'}},$$

where,

$$R^*(\boldsymbol{x}, \boldsymbol{x'}) = \exp\{-\sum_{h=1}^{p}\frac{1}{d_h}|x_h - x'_h|^2\}.$$

Here, g is the nugget parameter and $\delta_{\boldsymbol{x},\boldsymbol{x'}}$ is the Kronecker delta function, i.e. $\delta_{\boldsymbol{x},\boldsymbol{x'}} = 1$, if $\boldsymbol{x} = \boldsymbol{x'}$; $\delta_{\boldsymbol{x},\boldsymbol{x'}} = 0$, otherwise. The TGP assumes an exponential prior on the nugget g and an independent mixture of Gamma priors on d_i, that is

$$\pi(g) \sim \exp(\lambda)$$

and

$$\pi(d_i) \sim \frac{1}{2}\left(G(1,20) + G(10,10)\right),\ i = \{1,2,\ldots,p\},$$

with λ is treated as known. We also notice that the TGP actually works on the $1/\theta_i$ scale. The equivalent prior on θ_i is

$$\pi(\theta_i) = \left\{\frac{10^{10}}{\Gamma(10)}\frac{1}{\theta_i^9}\exp(-\frac{10}{\theta_i}) + 20\exp(-\frac{20}{\theta_i})\right\}/(2\theta_i^2).$$

The two priors mentioned above are specified for the correlation matrix $K_{\boldsymbol{d},g}$. The TGP method also assumes priors on $\boldsymbol{\beta}$ and σ^2 as

- An Inverse Gamma (IG) prior on σ^2.
- A Normal distribution on β_i.

Since the Normal and Inverse Gamma are both conjugate priors, it's not hard to get the full conditional distributions for $\boldsymbol{\beta}$ and σ^2. Therefore, all of the parameters except g and $\boldsymbol{d}$ can be sampled with the Gibbs sampler. g and $\boldsymbol{d}$ will be sampled with the Metropolis-Hastings algorithm. The detailed derivations of the full conditionals along with the prediction formulas have been given by Gramacy (2005).

Although it uses a different MCMC method, the underlying idea of TGP is quite similar to that of the Higdon's method and the GEM we covered in the previous section. All the three methods are constructed within the Bayesian paradigm and are depending on numerical methods (MCMC) to conduct posterior inferences. In addition, we also observe in the next chapter that the performances of the $(1-\alpha)100\%$ credible intervals are much better than those constructed by the MLE.

In summary, we have introduced four popular parameter estimation methods for Gaussian Processes. The first method is the Frequentist-based MLE and the remaining three methods are Bayesian approaches. From now on, we will emphasize the three Bayesian methods, since, theoretically speaking, they are superior to MLE in that the Bayesian paradigm is able to quantify the extra uncertainties.

Moreover, we have done a search but failed to find any journal paper that has compared the three Bayesian methods before. Therefore, we conduct a comprehensive comparison on them. The comparison will focus on two aspects: the prediction accuracy and the prediction actual coverage probability, which will be measured by different criteria, respectively. A tentative comparison will be made in the next chapter and the formal comparison between the three methods will be covered in Chapter 4 and Chapter 5.

Chapter 3

Tentative Comparison of the Bayesian Methods

In this chapter we will conduct a tentative comparison of the three Bayesian methods. Results from MLE will also be included for comparison.

3.1 Two Prediction Criteria

As we have mentioned before, we would like to compare those Bayesian methods both on prediction accuracy and on prediction actual coverage probability. The first criterion is the "Normalized Root Mean Square Error" or "Norm-RMSE" in short, which quantifies the performance of prediction accuracy. The expression is given below.

$$\text{Norm-RMSE} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}(\boldsymbol{x}_{\text{ho}}^i) - y(\boldsymbol{x}_{\text{ho}}^i))^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\bar{y} - y(\boldsymbol{x}_{\text{ho}}^i))^2}} \tag{3.1}$$

where $\bar{y}$ is the mean of the data from the runs in the experimental design. $y(\boldsymbol{x})$ is the "true value" of the hold-out (test) set, i.e., N further runs. Besides, $\hat{y}(\boldsymbol{x})$ is the "estimated value"—predicted value from the GP. In most cases, the Norm-RMSE roughly lies between the range of 0 to 1, since the denominator is usually larger than the numerator—the usual RMSE. A 0 value in Norm-RMSE means perfect prediction and a 1 or even larger value indicates the predictor from GP model is not better than the trivial predictor ($\bar{y}$).

The second criterion is the frequentist Actual Coverage Probability (ACP) Berger et al. (2001), which measures how well a method quantifies the uncertainty. The true coverage probability is set as 90% and 95% in this chapter. In fact, the actual coverage probability is a proportion, which calculates how many of the credible intervals capture the true values among all of the

constructed intervals. We want the actual coverage probability be as close to the nominal coverage probability (90%, 95%) as possible. If the actual coverage probability of one method is apparently below the nominal one, we can conclude that this method fails to incorporate all of the uncertainties.

3.2 Details for the Tentative Comparison

We will mention some details below for the tentative comparison.

3.2.1 Data

The data we are going to use is from a G-protein compute code, one that has served as a testbed in many contexts. The G-protein computer model was proposed by Yi et al. (2005) with the purpose of modelling ligand activation of G-protein in yeast. It involves the solving a system of ordinary differential equations (ODEs) with nine parameters that can vary. The differential equations are given by

$$\begin{aligned} \frac{d\eta_1}{dx} &= -\mu_1\eta_1 x + \mu_2\eta_2 - \mu_3\eta_1 + \mu_5 \\ \frac{d\eta_2}{dx} &= \mu_1\eta_1 x - \mu_2\eta_2 - \mu_4\eta_2 \\ \frac{d\eta_3}{dx} &= -\mu_6\eta_2\eta_3 + \mu_8(G_{\text{tot}} - \eta_3 - \eta_4)(G_{\text{tot}} - \eta_3) \\ \frac{d\eta_4}{dx} &= \mu_6\eta_2\eta_3 - \mu_7\eta_4 \end{aligned} \tag{3.2}$$

where $\eta_1, \ldots, \eta_4$ are concentration of four chemical species, x is the concentration of the ligand, and $\mu_1, \ldots, \mu_8$ is a vector of eight kinetic parameters. G_{tot} is the total concentration of G-protein complex after 30 seconds and the output $y = (G_{\text{tot}} - \eta_3)/G_{\text{tot}}$ is the normalized concentration of a relevant part of the complex.

We fixed five of the kinetic parameters, allowing only μ_1, μ_6, μ_7 and x to vary. We use the GP model to construct an approximation of y as a function of the transformed variables $\log(\mu_1), \log(\mu_6), \log(\mu_7)$ and $\log(x)$, and then further transform each of these to $[0, 1]$. In this way, the G-protein data actually has 4 inputs (dimensions) and one simple output.

3.2.2 Comparison Details

We have the following set-up for this comparison.

- The training set comprises 20 runs, i.e., x points, and the design for the training set is a random Latin Hypercube Design (LHD). (McKay et al. (1979))
- The testing/hold-out set is a 1000–run random LHD.
- The regression in the GP model is fixed as Constant, i.e, $f(\boldsymbol{x}) = \mu + Z(\boldsymbol{x})$.
- Two Criteria: Norm-RMSE and Actual Coverage Probability.
- Estimation methods considered here: MLE, Higdon's method, TGP and GEM.
- Even for the same method, different designs give different outputs. Therefore, we need replications to minimize the effect of randomness. The replication number is set as 20, i.e., each method will be applied to 20 designs and will have 20 numbers for Norm-RMSE, 90% ACP and 95% ACP, respectively.

For the three Bayesian methods, in order to decide how to run the M-CMC, we do a small experiment. First of all, we specify 3 choices for the total number of MCMC samples as well as the corresponding burn-in as follows

- 15,000 total runs with the first 5,000 as burn-in.
- 30,000 total runs with the first 10,000 as burn-in.
- 60,000 total runs with the first 15,000 as burn-in.

We then generate one training set with 41 runs and use the same 1000 testing points as hold-out. The prior on the correlation parameters $(\theta_1, \ldots, \theta_4)$ are set independently as $\exp\{0.1\}$. The initial values for $(\theta_1, \ldots, \theta_4)$ are set at their MLE values, which are obtained from the *mlegp* package. The traceplot for each of the choices are given in Figures 3.1, 3.2, 3.3. The empirical densities for each of the choices are presented in Figures 3.4, 3.5 and 3.6

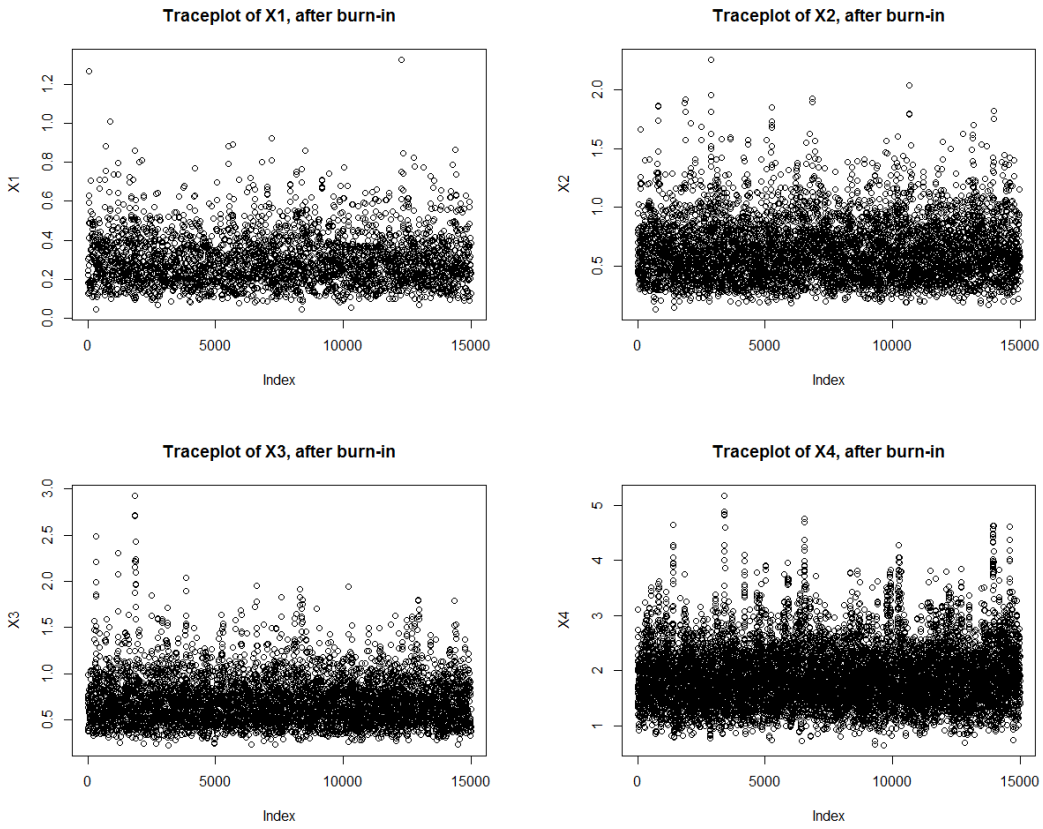

Figure 3.1: Traceplot for 15000 total samples.

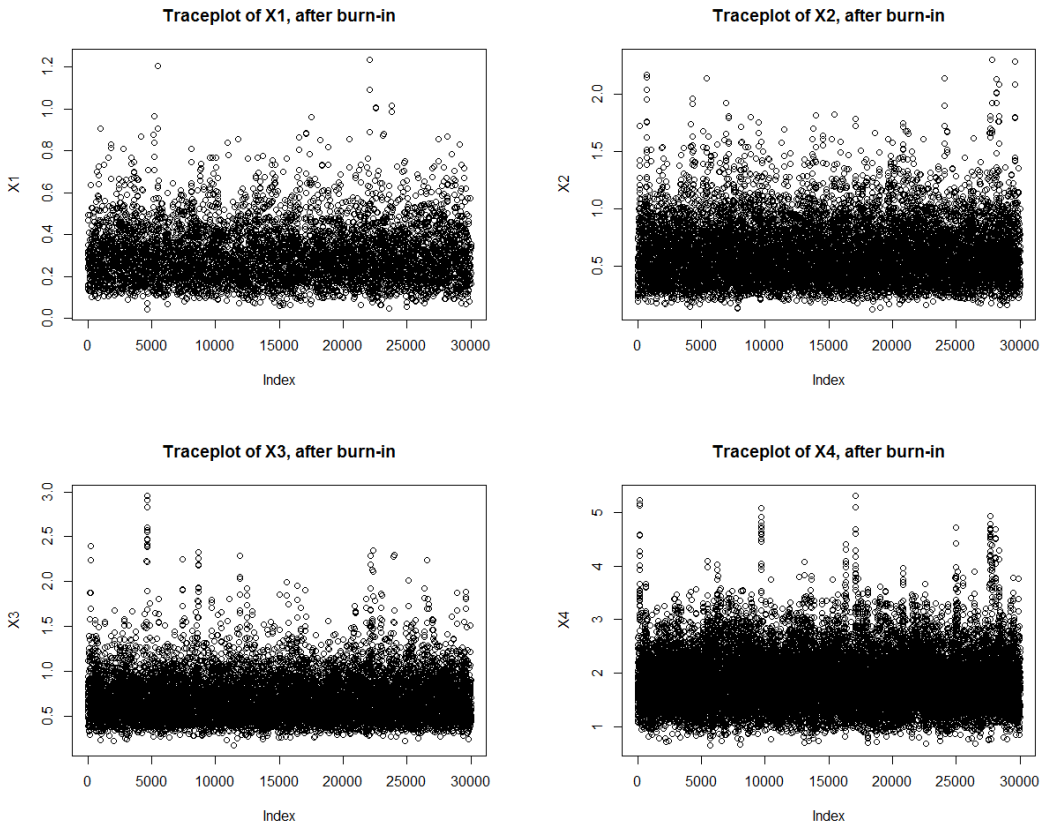

Figure 3.2: Traceplot for 30000 total samples.

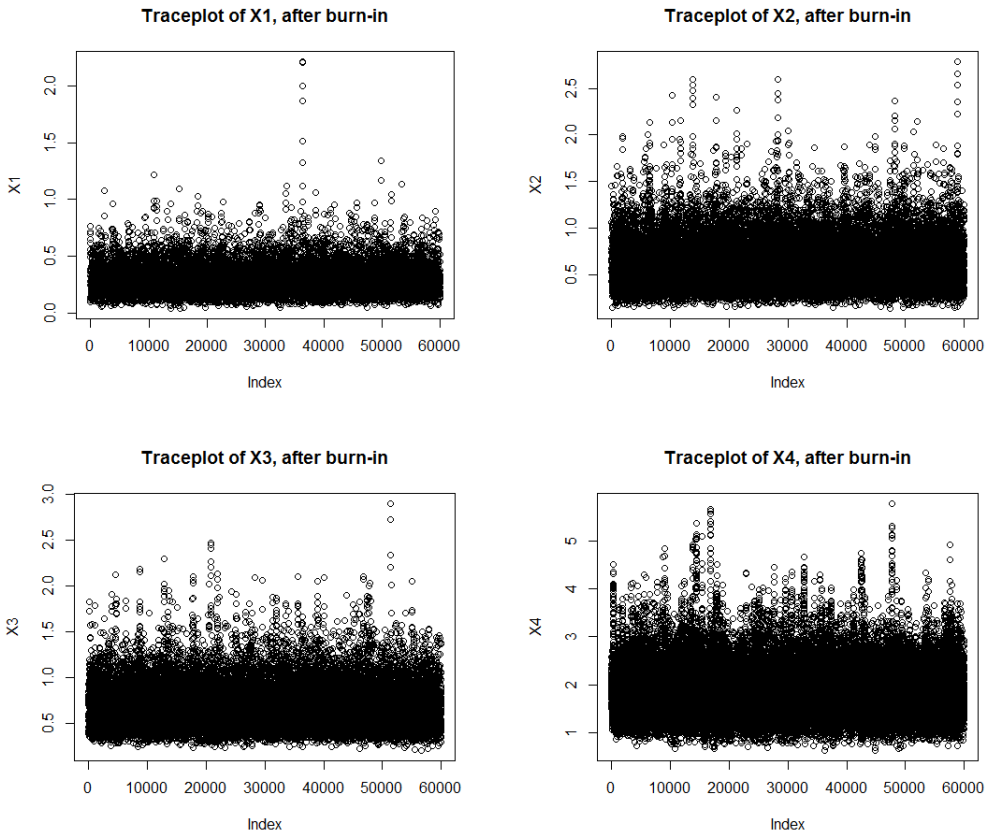

Figure 3.3: Traceplot for 60000 total samples.

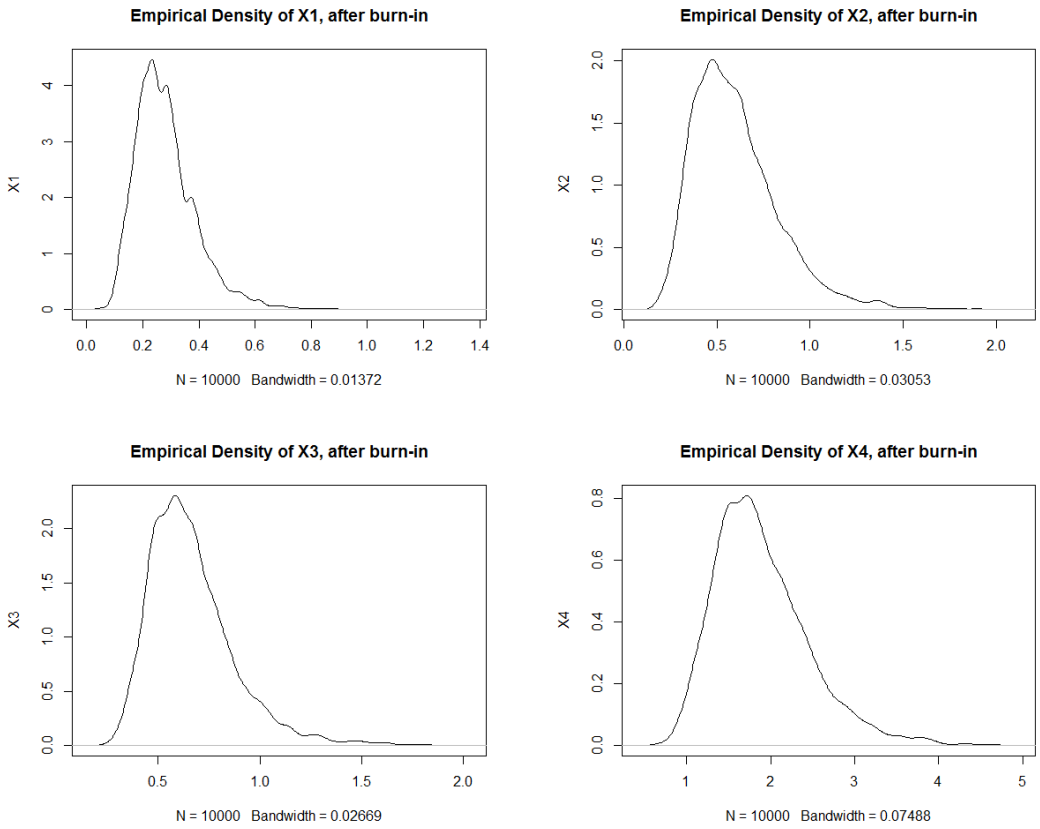

Figure 3.4: Empirical densities for 15000 total samples.

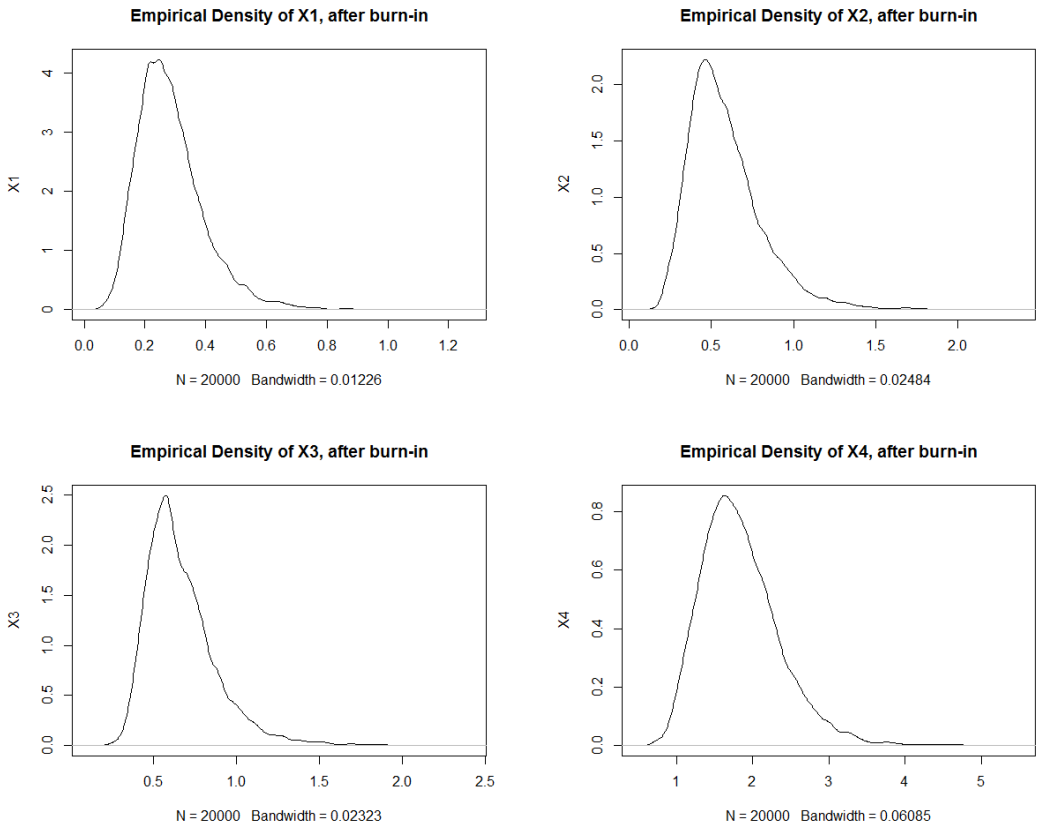

Figure 3.5: Empirical densities for 30000 total samples.

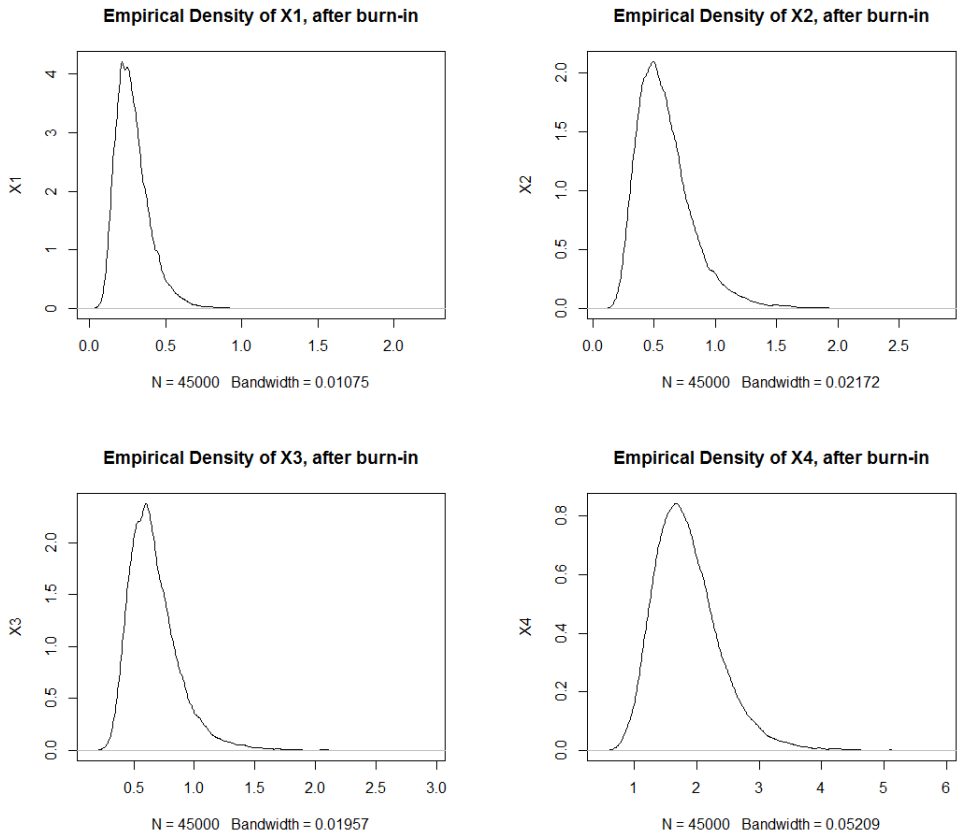

Figure 3.6: Empirical densities for 60000 total samples.

The acceptance rates for each of the choices are given in Table 3.1

Choices	θ_1	θ_2	θ_3	θ_4
15,000 with 5,000 as burn-in	0.198	0.382	0.371	0.478
30,000 with 10,000 as burn-in	0.202	0.381	0.372	0.482
60,000 with 15,000 as burn-in	0.199	0.382	0.365	0.479

Table 3.1: Acceptance rates for different choices of MCMC length.

We can see that the empirical densities do not show much difference for the different choices of number of MCMC samples. The acceptance rates for all of the three choices of MCMC are between 20% and 50%. From the traceplots, the chain appears to converge when the number of MCMC samples is set as 60,000. The 15,000 scenario or the 30,000 scenario is too small to guarantee convergence. Therefore, we decide to set the number of MCMC samples as 60,000 and the first 15,000 are deleted as burn-in. The thinning number is chosen as 10 so that the MC samples we obtain are approximately independent.

3.3 Comparison Results

3.3.1 Prediction Accuracy

First, we present the results for Norm-RMSE as follows. In Figure 3.7, the top panel contains the dotplots and the bottom panel contains the boxplots.

From Figure 3.7 , we can see that the MLE and Higdon's method are slightly better than the TGP or GEM. In general, however, the performances of the four methods are not substantially different.

3.3.2 Actual Coverage Probability

The results for the actual coverage probability are given in Figures 3.8 and 3.9. Please note that each method has 20 numbers for the ACP from 20 repeat designs. The nominal coverage probability in Figure 3.8 is 90% and 95% in Figure 3.9.

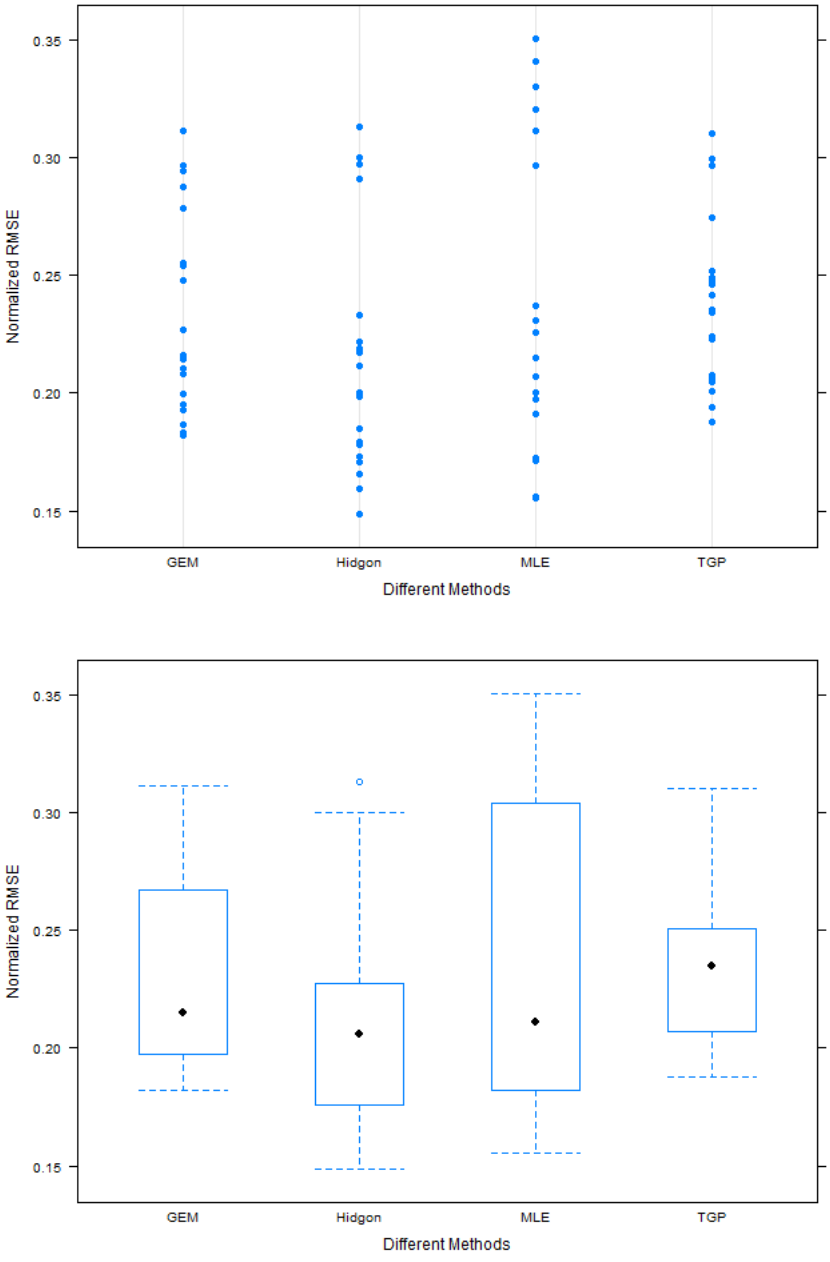

Figure 3.7: Results for prediction accuracy.

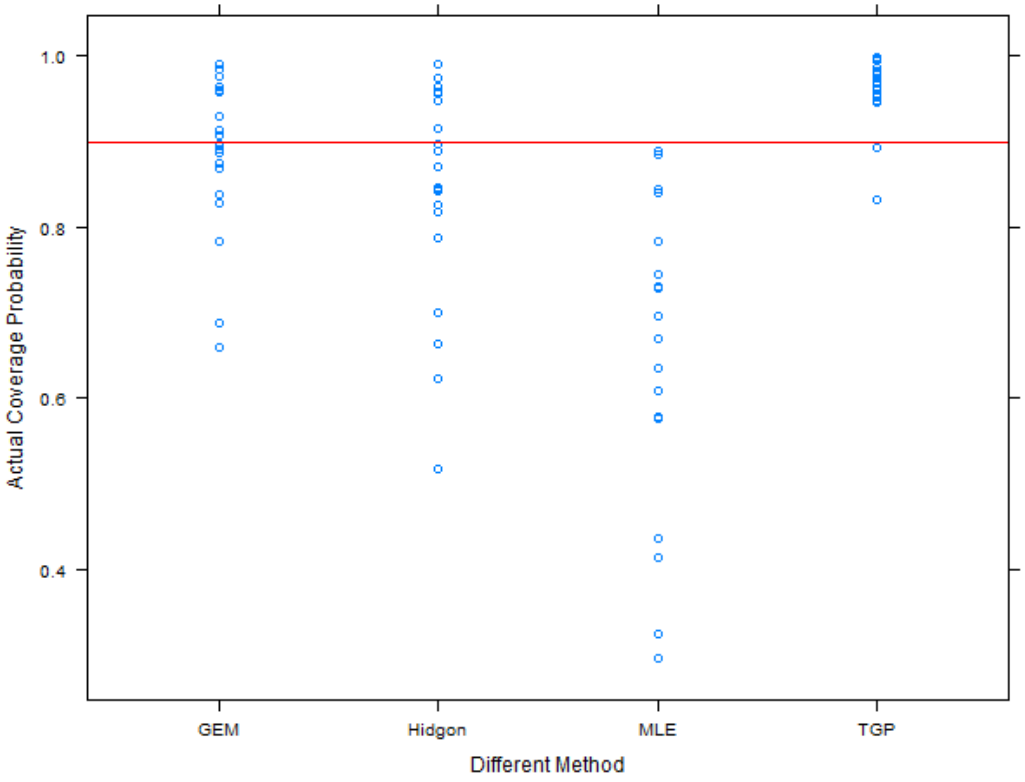

Figure 3.8: Results for actual coverage probability, 90% true CP.

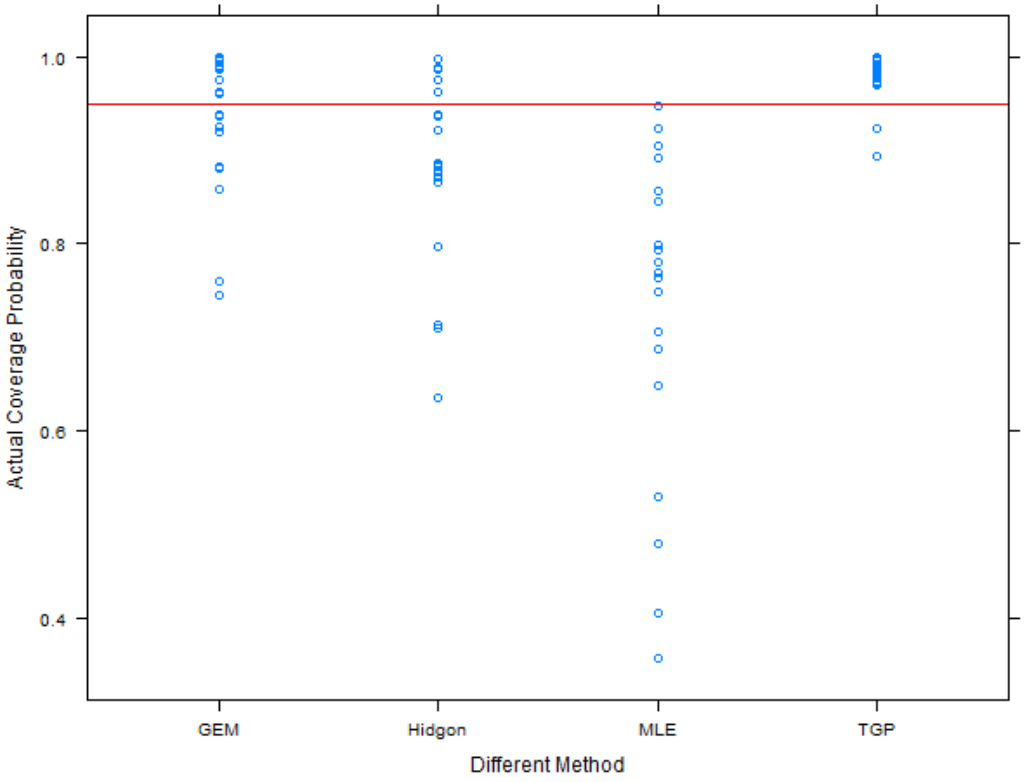

Figure 3.9: Results for actual coverage probability, 95% true CP.

The red lines in Figures 3.8 and 3.9 represent the correct 90% or 95% coverage probability, respectively. From both figures, we can see that Higdon's methods and GEM fluctuate around the correct coverage probability, while TGP largely over-covers. It is also obvious that the TGP method contains less variation than the other 2 Bayesian methods. In order to eliminate the variations, we compute the average ACP and provide the results for each method in Table 3.2 and 3.3.

MLE	**Higdon's method**	**GEM**	**TGP**
64.92%	84.16 %	88.87%	96.17%

Table 3.2: Average actual coverage probability, 90% true CP.

MLE	**Higdon's method**	**GEM**	**TGP**
72.43%	89.03%	93.03%	97.86%

Table 3.3: Average actual coverage probability, 95% true CP.

All in all, it is apparent that the MLE greatly under-covers. This result is not surprising, since from previous analyses, we already know that the MLE method fails to incorporate all of the uncertainties. In addition, Higdon's method has a little under-coverage, while TGP over-covers. The performance of GEM seems the best among the four methods we compare here.

3.4 Conclusion & Discussion

Our emphasis in this tentative comparison is put on the three Bayesian methods. In terms of prediction accuracy, we can see that the performances for all of the three methods are quite similar. In terms of the prediction actual coverage probability, GEM is better than Higdon's method and TGP.

However, this comparison is only a superficial one because each Bayesian method has its own features, such as, different parameterizations, different prior distributions on correlation parameters $\boldsymbol{\theta}$, etc. We actually do not know which factor has significant effects on the prediction results. People may ask questions that we can not answer through this tentative comparison, such as

- Does the prior on $\boldsymbol{\theta}$ affect the prediction accuracy substantially?
- Does the prior on σ^2 affect the prediction accuracy or not?
- You are using a constant model for GP. If you use a linear model instead, will it have a significant effect on the prediction performance?
- You fix the number of computer model runs as 40 in your training set. If you increase the number of runs from 40 to say 80, will it affect the prediction performance?
  ...

From the above discussion, we can see that more sophisticated comparisons/analyses are needed to gain a better understanding of the underlying factors. In the next chapters, we will specify four important factors that may affect the prediction performance. For each factor, different levels will be assigned based to the three existing Bayesian methods. We then find the important factors that have significant impacts on the prediction performance through full factorial experiments. The best combination for the levels of factors will also be identified.

Chapter 4

In-Depth Comparison: Real Computer Models

In this chapter, we will conduct an in-depth comparison of three real computer models. Conclusions drawn from the comparison can help us to answer the questions asked in Chapter 3.

4.1 Comparison Details

4.1.1 Four Important Factors

First of all, we specify 4 factors within the Bayesian paradigm that may have significant effects on the prediction performance measures. The factors are

- The prior distribution on the correlation parameter θ_i.
- The prior distribution on σ^2.
- The regression term for the underlying Gaussian Process(GP).
- The number of the runs in the training data.

(1) Prior Distribution on the Correlation Parameter θ_i

Since different Bayesian methods have different parameterizations of the correlation function, we will uniformly parameterize it as

$$R(\boldsymbol{x}, \boldsymbol{x'}) = \exp\{-\sum_{h=1}^{p} \theta_h |x_h - x'_h|^2\}. \tag{4.1}$$

Hereafter, we will refer the above parameterization as the $\boldsymbol{\theta}$ **scale**. We extract three priors from the three Bayesian methods we reviewed previously and we call these priors Higdon's prior, TGP's prior and GEM's prior. For

methods that have different parameterizations, we transform their prior distributions into the equivalent prior distributions on the $\boldsymbol{\theta}$ scale. These three priors are the three qualitative levels we specified for the first factor. We list the three priors below.

- Hidgon's prior on $\boldsymbol{\theta}$, independent and identical distributed with the density
$$\pi(\theta_i) = \frac{1}{8}\exp(-\theta_i/4)\frac{1}{\sqrt{1-\exp(-\theta_i/4)}}. \tag{4.2}$$

- TGP's prior on $\boldsymbol{\theta}$, independent and identical distributed with the density
$$\pi(\theta_i) = \{\frac{10^{10}}{\Gamma(10)}\frac{1}{\theta_i^9}\exp(-\frac{10}{\theta_i}) + 20\exp(-\frac{20}{\theta_i})\}/(2\theta_i^2). \tag{4.3}$$

- GEM's prior on $\boldsymbol{\theta}$, independent and identical exponential distribution with $\lambda = 0.01$,
$$\pi(\theta_i) = 0.01\exp\{-0.01\theta_i\}. \tag{4.4}$$

We also draw the above three densities plots with different θ_i ranges($[0,4]$ and $[0,40]$). The plots are given in Figures 4.1, 4.2 and 4.3.

From Figure 4.1, we see that Higdon's prior places extremely heavy weight on small θ_i values. This is actually the desired prior distribution for the correlation parameter, since small θ_i values mean strong correlations between points in the design space. Moreover, compared with Higdon's prior, GEM's prior assumes a flat prior distribution on θ_i (note the different y ranges), which represents its vague preference on the magnitude of correlation parameters. Moreover, in terms of TGP's prior, since the parameterization of the TGP's prior is $\theta_i = 1/d_i$, it implies if $d_i \to \infty$ $\theta_i \to 0$. However, from Figure 4.3, the TGP's prior forces d_i to stay away from ∞, hence, θ_i can not get small enough. Therefore, we conjecture that Higdon's prior and GEM's prior will have better performance than TGP's prior.

(2) Prior Distribution on σ^2

For the second factor, we notice that Higdon's method and TGP's method assume an Inverse Gamma (IG) distribution on σ^2, while, GEM's method assumes the Jeffreys' prior ($\pi(\sigma^2) \propto 1/\sigma^2$). Hence, we specify two levels for the second factor:

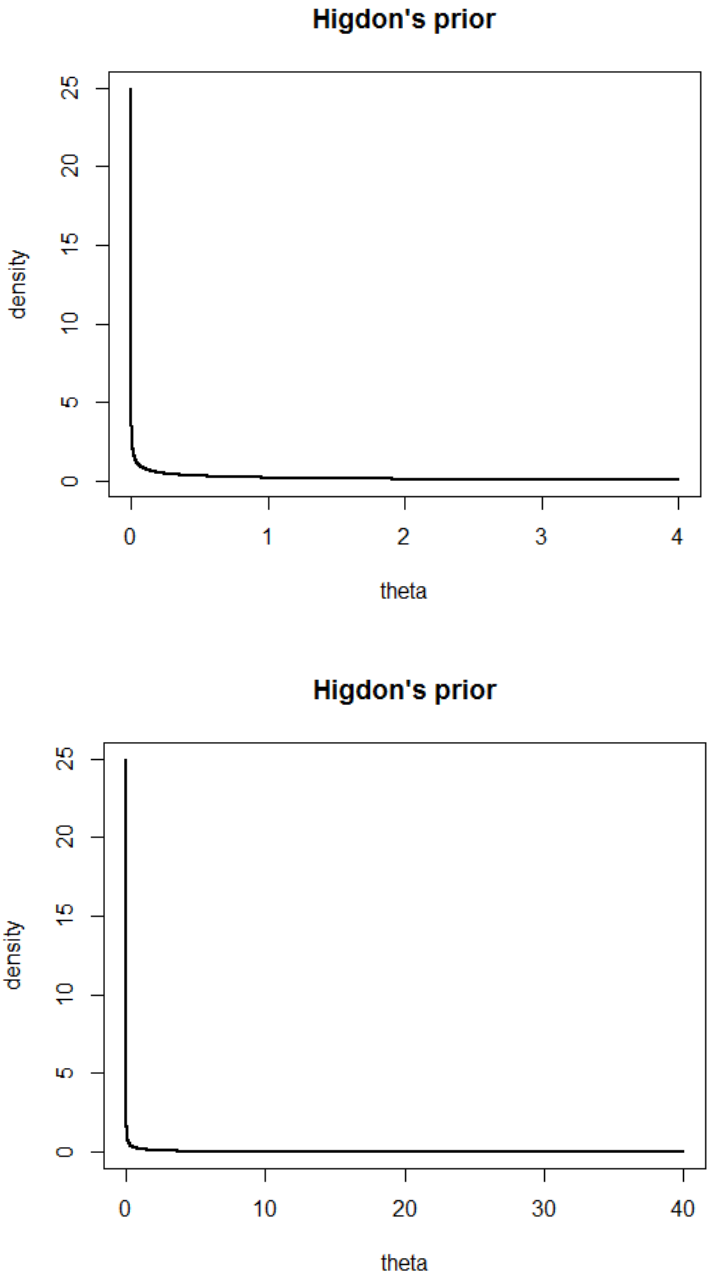

Figure 4.1: Density plot for Higdon's prior.

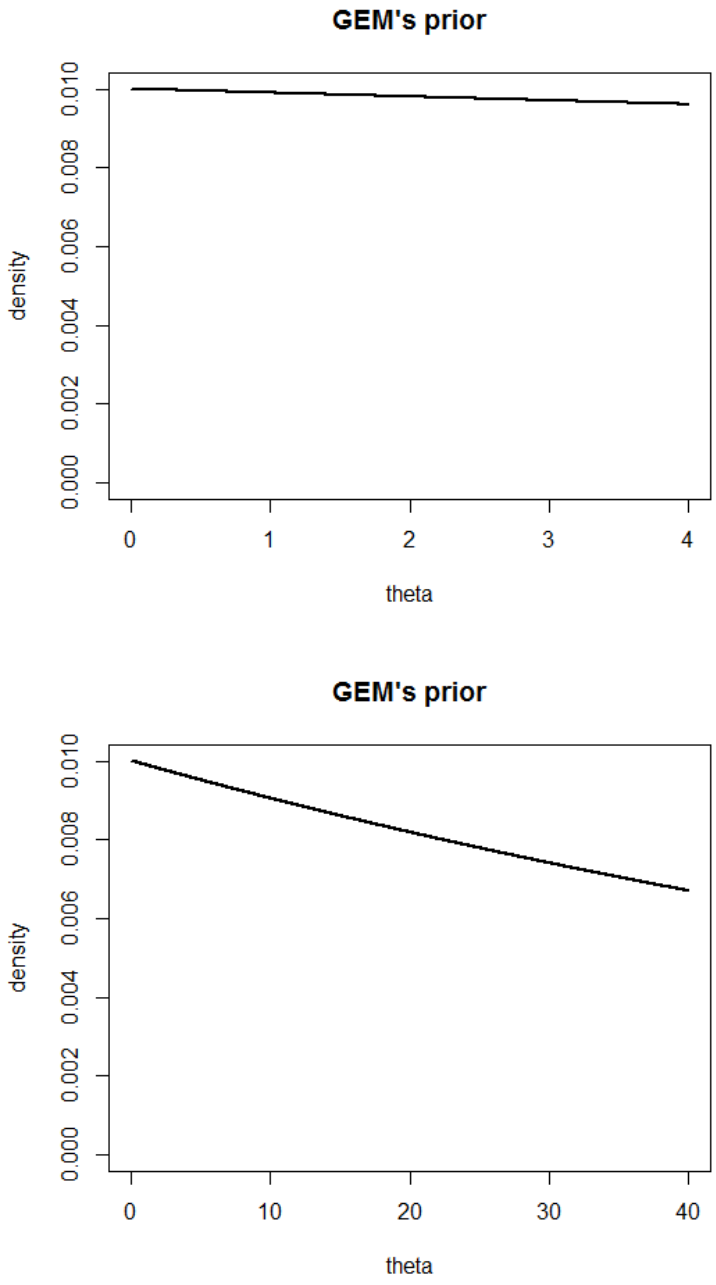

Figure 4.2: Density plot for GEM's prior.

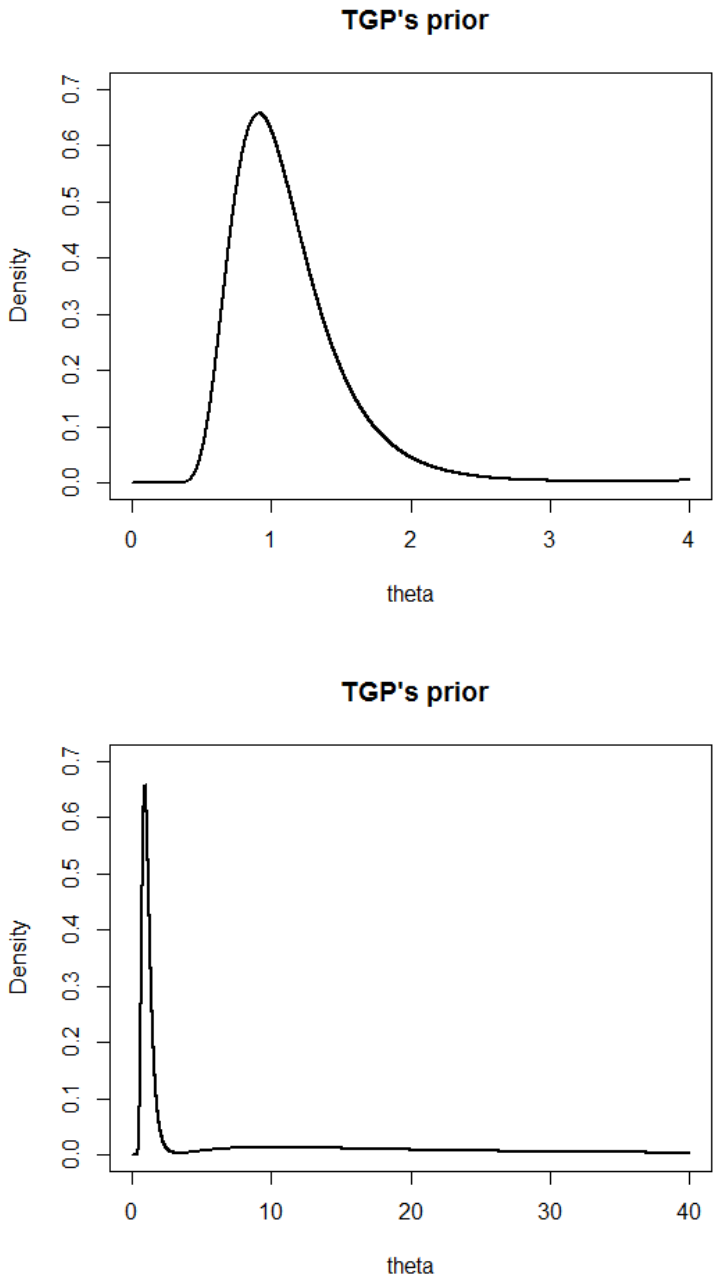

Figure 4.3: Density plot for TGP's prior.

- Inverse Gamma prior

$$\sigma^2 \sim \text{IG}(\alpha_1 = 0.1, \alpha_2 = 5),$$

  where the values that α_1, α_2 take are extracted from TPG.

- Jeffreys' prior

$$\pi(\sigma^2) \propto \frac{1}{\sigma^2}.$$

(3) Regression Terms for the Gaussian Process

We assign two levels for this factor: Constant and Full Linear. i.e.

- Constant:

$$y(\boldsymbol{x}) = \mu + Z(\boldsymbol{x}).$$

- Full Linear

$$y(\boldsymbol{x}) = \sum_{i=1}^{p} \beta_i x_i + Z(\boldsymbol{x}).$$

(4) Number of the Runs in the Training Data

Loeppky et al. (2009) provides evidence that the informal rule of $n = 10d$ is sufficient for most computer models. Let d be the input dimension, we would like to specify two levels here: $5d$ and $10d$. The $10d$ level represents the sufficient scenario for effectively emulating a computer model, while, we deliberately choose the inadequate scenario— $5d$ level to explore what will happen when the number of runs is not enough.

In addition, we fixed all of the other potentially affecting factors to make sure the comparisons are fair. We will mention the fixed factors in the subsequent sections.

4.1.2 Full Factorial Design

From previous discussion, we have specified 4 important factors and for each factor, we have assigned different levels. The following summarizes the design of the study.

- Prior on θ_i: Higdon's prior, TGP's prior and GEM's prior.
- Prior on σ^2: IG and Jeffreys' prior.

- Regression terms for the GP: Constant and Full Linear.
- Number of training runs: $5d$ and $10d$.

Based on the full factorial design, we will have $3 \times 2 \times 2 \times 2 = 24$ combinations of the 4 factors. Table 4.1 summarizes the full factorial design.

No.	**Prior on θ**	**Prior on σ^2**	**Regression**	**Runs**
1	Hig	Jeff	FL	10d
2	Hig	Jeff	FL	5d
3	Hig	Jeff	Const	10d
4	Hig	IG	FL	10d
5	Hig	Jeff	Const	5d
6	Hig	IG	Const	5d
7	Hig	IG	FL	5d
8	Hig	IG	Const	10d
9	TGP	Jeff	FL	10d
10	TGP	Jeff	FL	5d
11	TGP	Jeff	Const	10d
12	TGP	IG	FL	10d
13	TGP	Jeff	Const	5d
14	TGP	IG	Const	5d
15	TGP	IG	FL	5d
16	TGP	IG	Const	10d
17	GEM	Jeff	FL	10d
18	GEM	Jeff	FL	5d
19	GEM	Jeff	Const	10d
20	GEM	IG	FL	10d
21	GEM	Jeff	Const	5d
22	GEM	IG	Const	5d
23	GEM	IG	FL	5d
24	GEM	IG	Const	10d

Table 4.1: Full factorial design.

We can also view Table 4.1 in another way that gives the prior on θ_i, the 24 full factorial design will reduce to 2^3 factorial designs. The 2^3 full factorial design conditional on each level of the θ_i prior is presented in Table 4.2.

No.	Prior on σ^2	Regression	Runs
1	Jeff	FL	10d
2	Jeff	FL	5d
3	Jeff	Const	10d
4	IG	FL	10d
5	Jeff	Const	5d
6	IG	Const	5d
7	IG	FL	5d
8	IG	Const	10d

Table 4.2: 2^3 factorial design, given any level of the prior on θ_i.

4.1.3 Some Mathematics Details

In this section, we will mention some mathematics details. Like most non-textbook Bayesian problems, the Metropolis-Hastings algorithm is needed here to draw samples from the posterior distribution of θ_i and the posterior inferences are conducted based on the MC samples.

(1) The Marginal Posterior Distribution of θ_i

First of all, we derive the marginal posterior distribution of θ_i, which we need in the M-H algorithm to exactly sample θ_i. The $p(\boldsymbol{\theta}|\boldsymbol{y})$ is obtained by integrating out $\boldsymbol{\beta}$ and σ^2. The prior on $\boldsymbol{\beta}$ is fixed as a vague normal distribution with zero mean and large variance. We will only present the results here, the detailed derivations are given in Appendices A, B.

- For $p(\sigma^2) \sim IG(\alpha_1, \alpha_2)$,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto \frac{p(\boldsymbol{\theta})}{(\alpha_2 + \frac{n-k}{2}\hat{\sigma}^2_{\boldsymbol{\theta}})^{(\alpha_1+\frac{n-k}{2})} \det^{1/2}(\boldsymbol{R}) \det^{1/2}(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})}. \quad (4.5)$$

- For $p(\sigma^2) \propto 1/\sigma^2$,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto \frac{p(\boldsymbol{\theta})}{(\hat{\sigma}^2_{\boldsymbol{\theta}})^{(\frac{n-k}{2})} \det^{1/2}(\boldsymbol{R}) \det^{1/2}(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})}. \quad (4.6)$$

(2) The Predictive Distribution

Once we have the sampled θ_i, according to Santner et al. (2003), the predictive distribution is a non-central t distribution. That is

$$p(y(\boldsymbol{x}^*)|\boldsymbol{\theta}, \boldsymbol{y}) \sim t(\hat{m}(\boldsymbol{x}^*), \hat{v}_\theta(\boldsymbol{x}^*)),$$

where

$$\hat{m}(\boldsymbol{x}^*) = \boldsymbol{f}^T(\boldsymbol{x}^*)\hat{\boldsymbol{\beta}} + \boldsymbol{r}^T(\boldsymbol{x}^*)\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}). \tag{4.7}$$

- For $\sigma^2 \sim IG(\alpha_1, \alpha_2)$, the degrees of freedom are $n - k + 2\alpha_1$ and

$$\hat{v}_{\boldsymbol{\theta}}(\boldsymbol{x}^*) = \left(\frac{(n-1)\hat{\sigma}^2_{\boldsymbol{\theta}} + 2\alpha_2}{n - 1 + 2\alpha_1}\right)\left(1 - \boldsymbol{r}^T(\boldsymbol{x}^*)\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}^*) + \boldsymbol{h}^T(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{h}\right). \tag{4.8}$$

- For $\pi(\sigma^2) \propto 1/\sigma^2$, the degrees of freedom are $n - k$ and

$$\hat{v}_{\boldsymbol{\theta}}(\boldsymbol{x}^*) = \hat{\sigma}^2_{\boldsymbol{\theta}}\left(1 - \boldsymbol{r}^T(\boldsymbol{x}^*)\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}^*) + \boldsymbol{h}^T(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{h}\right), \tag{4.9}$$

  where h and $\hat{\sigma}^2_{\boldsymbol{\theta}}$ have been previously defined in Chapter 2.

We briefly mention the prediction procedures here. After repeating the MCMC procedure for M times, one can get a $M \times p$ matrix with the i th row containing $\boldsymbol{\theta}^i$. For each sampled $\boldsymbol{\theta}$, one computes $\hat{m}^i(\boldsymbol{x}^*)$ based on (4.7) and $\hat{v}^i_{\boldsymbol{\theta}}(\boldsymbol{x}^*)$ based on (4.8) or (4.9). The final predictor is defined as

$$\hat{m}(\boldsymbol{x}^*) = \bar{m}(\boldsymbol{x}^*) = \frac{1}{M}\sum_{i=1}^{M}\hat{m}^i(\boldsymbol{x}^*).$$

Each point-wise credible interval is constructed in a semi-parametric way: Suppose that one wants to construct the 95% credible interval. For each MC sample, one generates t samples from a non-central t distribution $t(\hat{m}^i(\boldsymbol{x}^*), \hat{v}^i_\theta(\boldsymbol{x}^*))$. Then, one combines the $M \times t$ generated samples and compute the empirical 2.5% quantile L and the empirical 97.5% quantile U from the combined samples. The 95% CI is constructed as (U, L). A 90% credible interval is similarly obtained.

4.1.4 Other Details for the In-Depth Comparison

We have the following set-up for the in-depth comparison, which are quite similar to those of the tentative comparison in Chapter 3.

- For each code, the test set is the same 1000 test points.
- The design for the input set is a random Latin Hypercube Design (random LHD). (McKay et al. (1979))
- Two Criteria: Norm-RMSE and Actual Coverage Probability. The Nominal Coverage Probability is set at 90% and 95%.

- Even for the same method, different designs give different outputs. Therefore, we need replications to minimize the effect of randomness. The replication number is 25, i.e., each method will have 25 numbers for Norm-RMSE, 90% ACP and 95% ACP, respectively.
- The total number of MCMC samples is set as 60,000 with the first 15,000 as burn-in. The thinning parameter is 10.
- We do not include nugget terms for the correlation matrix R. If a numerical problem occurs in MCMC, the generated candidate is rejected during the Metropolis step.

This completes the set-up of the in-depth comparison. The full factorial design on the 4 factors will help us explore which factors have significant impact on the prediction performances. In the next sections, we will carry out the experiments on several real outputs from computer models to identify the significant factors as well as their best levels.

4.2 Real Computer Model 1: Borehole

In this section, we will first work with the Borehole (Morris et al. (1993)) computer model, one that has served as a testbed in many contexts. The Borehole data are obtained from a computer model of the flow of water through a borehole that is drilled from the ground surface through two aquifers. The response variable from this model is y_0, the flow rate through the borehole in m^3/yr, which is determined by the equation.

$$y = \frac{2\pi T_u[H_u - H_l]}{\log(r/r_w)[1 + \frac{2LT_u}{\log(r/r_w)r_w^2 K_w} + T_u/T_l]} \tag{4.10}$$

where the 8 inputs and their respective ranges of interest and units are as follows.

- r_w = radius of boreholw, $r_w \in [0.05, 0.15]$.
- r = radius of influence, $r \in [100, 5000]$.
- T_u = transmissivity of upper aquifer, $T_u \in [63070, 115600]$.
- H_u = potentiometric head of upper aquifer, $H_u \in [990, 1110]$.
- T_l = transmissivity of lower aquifer, $T_l \in [63.1, 116]$.

- H_l = potentiometric head of lower aquifer, $H_l \in [700, 820]$.
- L = length of borehole, $L \in [1120, 1680]$.
- K_w hydraulic conductivity of borehole, $K_w \in [9855, 12045]$.

Please note that the Borehole function is not a typical computer model, since it can be expressed in a simple formula, and is used for demonstration purposes.

4.2.1 Prediction Accuracy

The Norm-RMSE results are presented in Figures 4.4, 4.5 and 4.6. We have 24 combinations and each combination has 25 Norm-RMSE numbers.

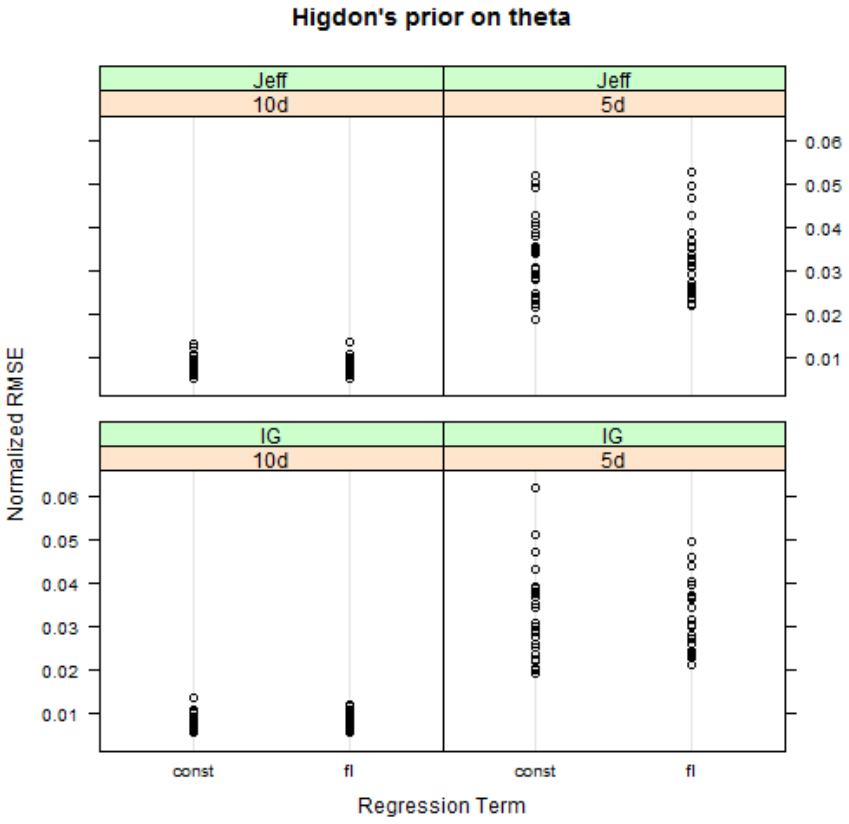

Figure 4.4: Prediction accuracy, given Higdon's prior on θ_i, Borehole.

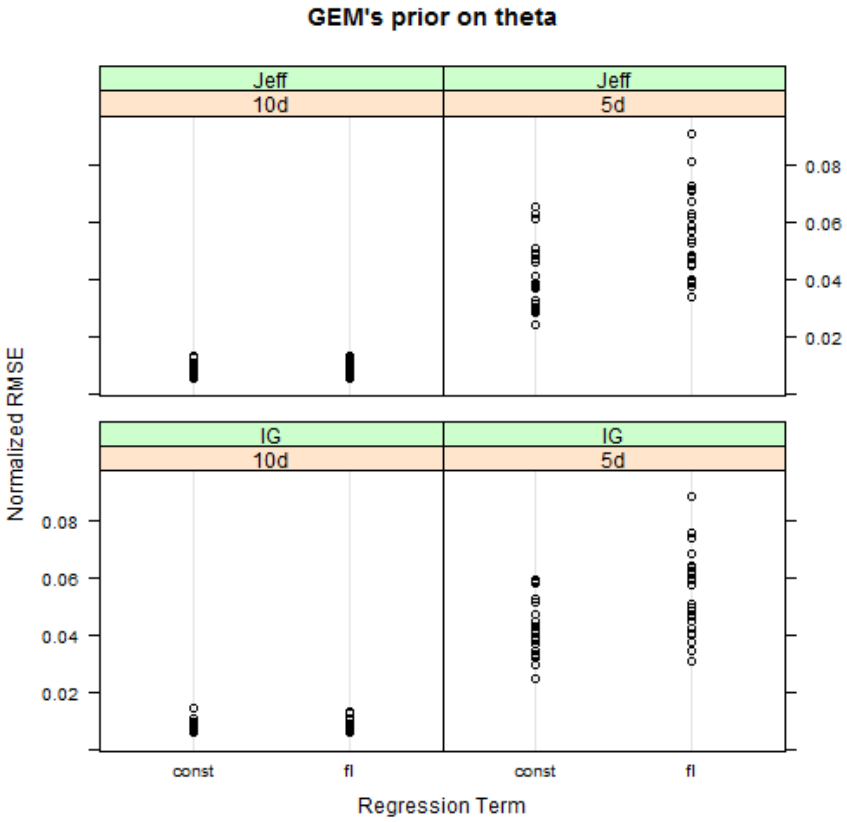

Figure 4.5: Prediction accuracy, given GEM's prior on θ_i, Borehole.

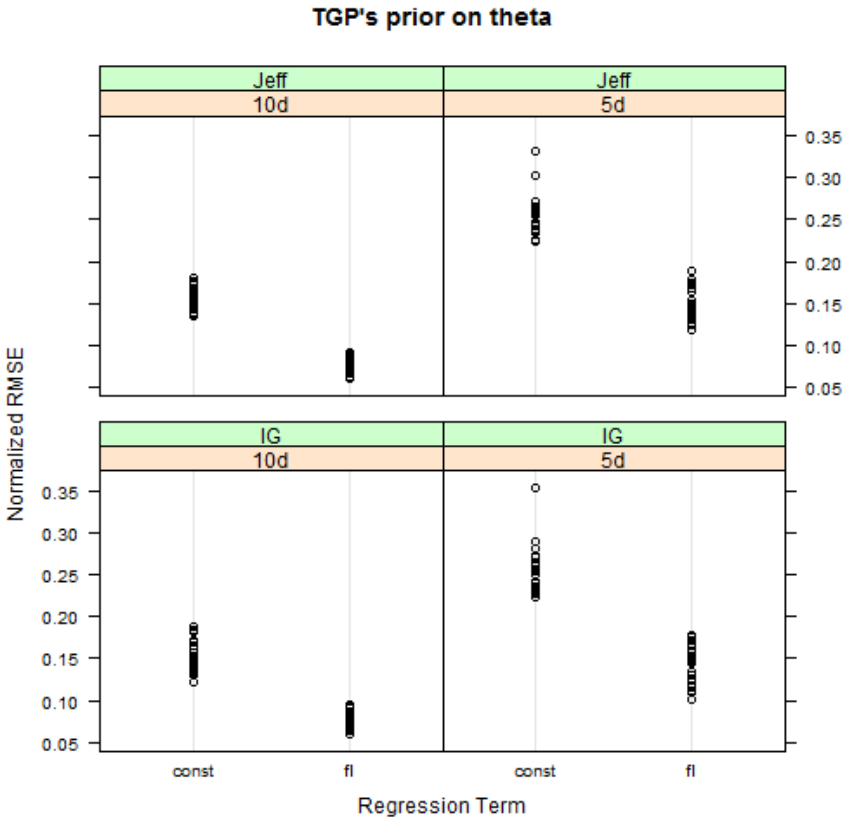

Figure 4.6: Prediction accuracy, given TGP's prior on θ_i, Borehole.

From Figures 4.4, 4.5 and 4.6, we make the following observations.

- The performance of Higon's prior is slightly better that GEM's prior on θ_i and is much better than the performance of TGP's prior on θ_i.
- A large number of runs ($10d$) gives smaller Norm-RMSE than a small number of runs ($5d$).
- With TGP's prior on θ_i, a full linear model has smaller Norm-RMSE numbers than a constant model.
- The prior on σ^2 does not seem to have much impact on the prediction accuracy.

Considering both the main effects and the interactions, an ANOVA for the Norm-RMSE numbers is given in Table 4.3.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
theta	2	2.37	1.18	7431.18	0.00	***
sigma	1	0.00	0.00	1.45	0.23	
reg	1	0.13	0.13	827.26	0.00	***
runs	1	0.37	0.37	2333.58	0.00	***
theta:sigma	2	0.00	0.00	0.60	0.55	
theta:reg	2	0.32	0.16	994.69	0.00	***
sigma:reg	1	0.00	0.00	0.42	0.52	
theta:runs	2	0.11	0.05	320.44	0.00	***
sigma:runs	1	0.00	0.00	0.42	0.51	
reg:runs	1	0.00	0.00	14.85	0.00	***
theta:sigma:reg	2	0.00	0.00	0.12	0.89	
theta:sigma:runs	2	0.00	0.00	0.04	0.96	
theta:reg:runs	2	0.01	0.01	46.21	0.00	***
sigma:reg:runs	1	0.00	0.00	2.26	0.13	
theta:sigma:reg:runs	2	0.00	0.00	1.31	0.27	
Residuals	576	0.09	0.00			

Table 4.3: ANOVA results for the Borehole computer model.

In Table 4.3, *theta* represents the prior on θ_i, *sigma* represents the prior on σ^2, *reg* represents the model for the GP and *runs* is the number of runs. We can see from the Table 4.3 that all of the three main effects are significant, except for the prior on σ^2. Then, for each combination, we take its average value of the Normalized RMSE and draw the plots for interaction

in Figures 4.7, 4.8 and 4.9.

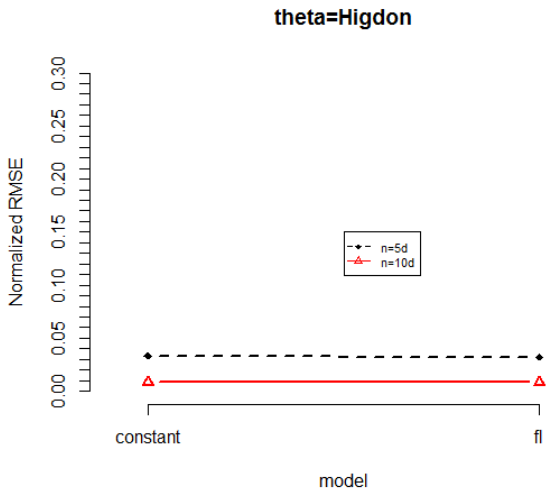

Figure 4.7: Average prediction accuracy, given Higdon's prior on θ_i, Borehole.

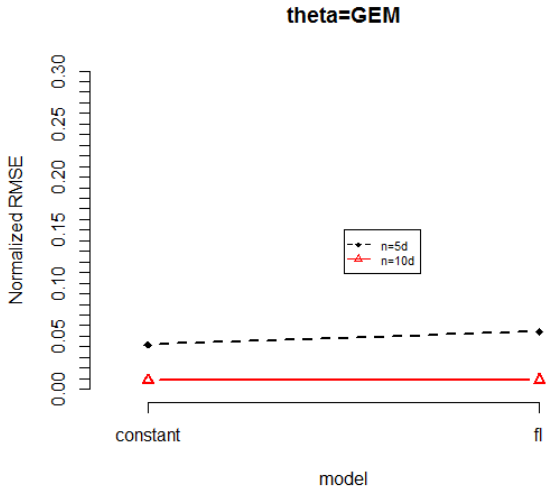

Figure 4.8: Average prediction accuracy, given GEM's prior on θ_i, Borehole.

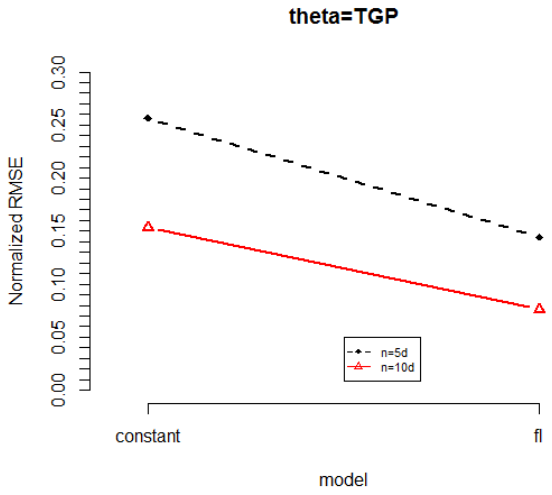

Figure 4.9: Average prediction accuracy, given TGP's prior on θ_i, Borehole.

From all of the figures and Table 4.3, we make the following conclusions.

- The prior on θ_i is important for prediction accuracy. And Higdon's prior on θ_i is preferred.
- The effect of number of runs is important and a large number of runs ($10d$) is better than a small number of runs ($5d$).
- Regression terms for the GP only matter when we assume TGP's prior on θ_i. Under TPG's prior on θ_i, a full linear regression is preferred.
- There is not enough evidence to prove that the prior on σ^2 has a statistically significant effect on prediction accuracy.

4.2.2 Actual Coverage Probability

Given a nominal coverage probability, each combination will have 25 numbers for the ACP. Since it contains non-trivial variations within the ACP values for each combination, we take the average of the ACP values and report the results in Figures 4.10, 4.11, 4.12 and Figures 4.13, 4.14, 4.15.

The red lines in Figures 4.10, 4.11, 4.12 and Figures 4.13, 4.14, 4.15 represent the correct 90% and 95% coverage probability, respectively. We

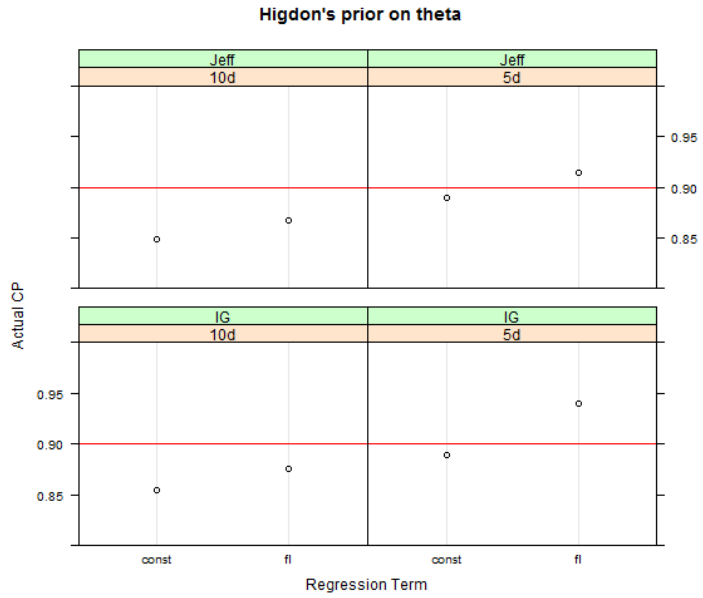

Figure 4.10: Prediction actual coverage probability, given Higdon's prior on θ_i, 90% true CP, Borehole.

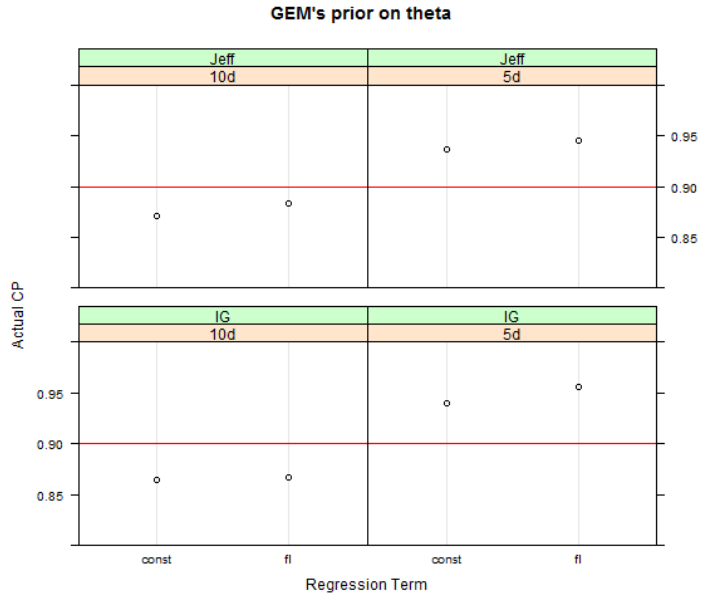

Figure 4.11: Prediction actual coverage probability, given GEM's prior on θ_i, 90% true CP, Borehole.

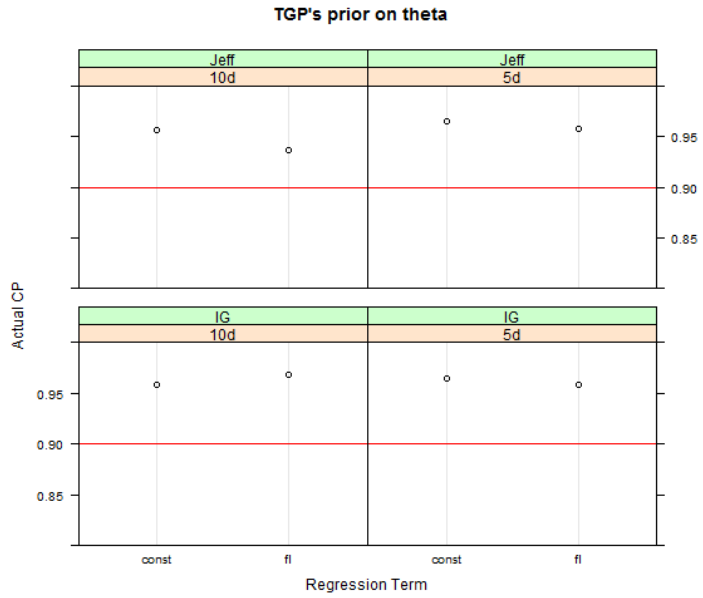

Figure 4.12: Prediction actual coverage probability, given TGP's prior on θ_i, 90% true CP, Borehole.

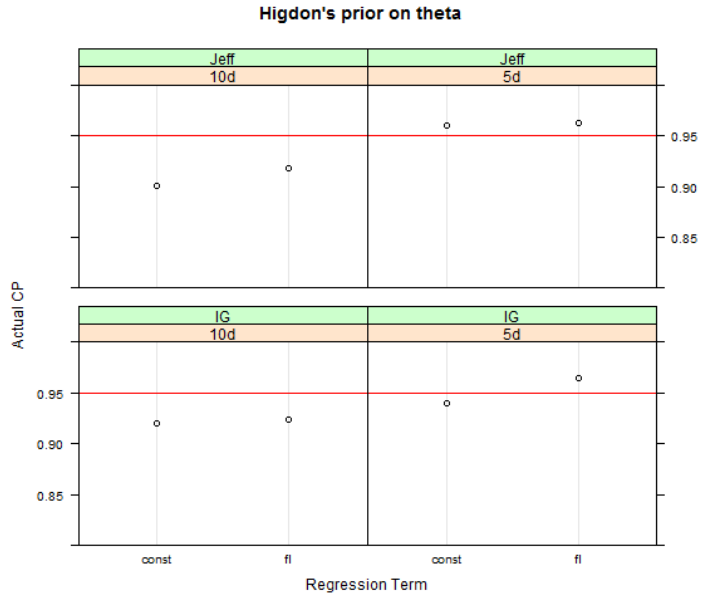

Figure 4.13: Prediction actual coverage probability, given Higdon's prior on θ_i, 95% true CP, Borehole.

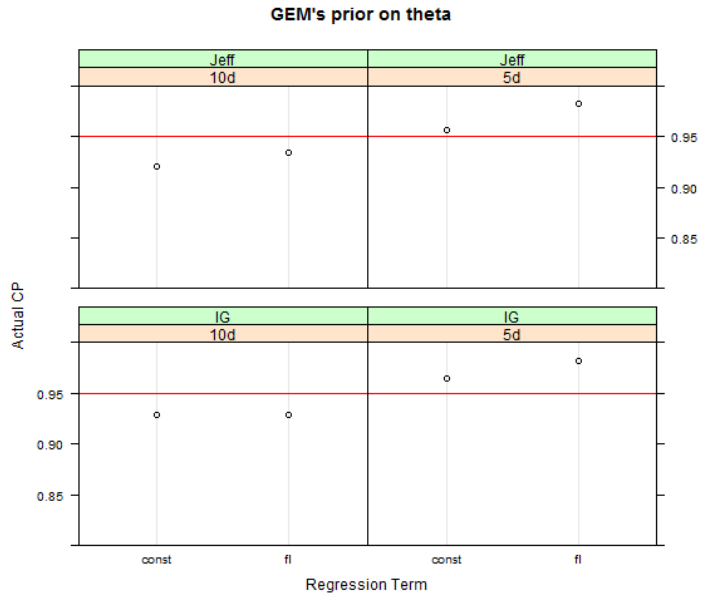

Figure 4.14: Prediction actual coverage probability, given GEM's prior on θ_i, 95% true CP, Borehole.

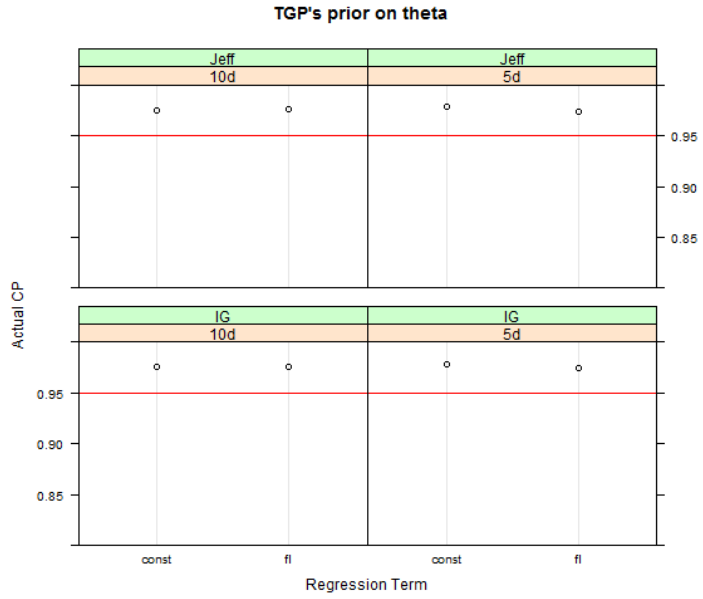

Figure 4.15: Prediction actual coverage probability, given TGP's prior on θ_i, 95% true CP, Borehole.

observe that the ACP for TGP's prior slightly over-cover. The mean ACP values for Higdon's prior and GEM's prior are satisfactory, since they are around the nominal coverage probability (90%, 95%).

We summarize what we conclude from the above analyses as follows: in terms of prediction accuracy, Higdon's prior on θ_i is highly preferred. Large number of runs ($10d$) is preferred than small number of runs ($5d$). In terms of the Actual Coverage Probability, all of the 24 combinations' performances are acceptable.

4.3 Real Computer Model 2: PTW

In this section, the real data we are going to work on is the outputs from a real physical computer model—PTW. The PTW model, which was proposed by Preston et al. (2003) in 2003, models the metallic plastic flow of metals and alloys during explosively driven deformation and high-velocity impacts. The model has 11 inputs and 1 single output measures the flow stress, the stress that is required to plastically deform the metal or alloy.

4.3.1 Prediction Accuracy

The Norm-RMSE results are presented in Figures 4.16, 4.17 and 4.18. Again there are 24 combinations and each combination has 25 Norm-RMSE numbers.

From Figures 4.16, 4.17 and 4.18, we make the following observations.

- The performance of Higdon's prior on θ_i is better than the performances of TGP's prior and GEM's prior.
- A large number of runs ($10d$) has smaller Norm-RMSE numbers than a small number of runs($5d$).
- With TGP's prior on θ_i, a full linear model has smaller Norm-RMSE numbers than a constant model.
- The prior on σ^2 does not seem to have much impact on the prediction accuracy.

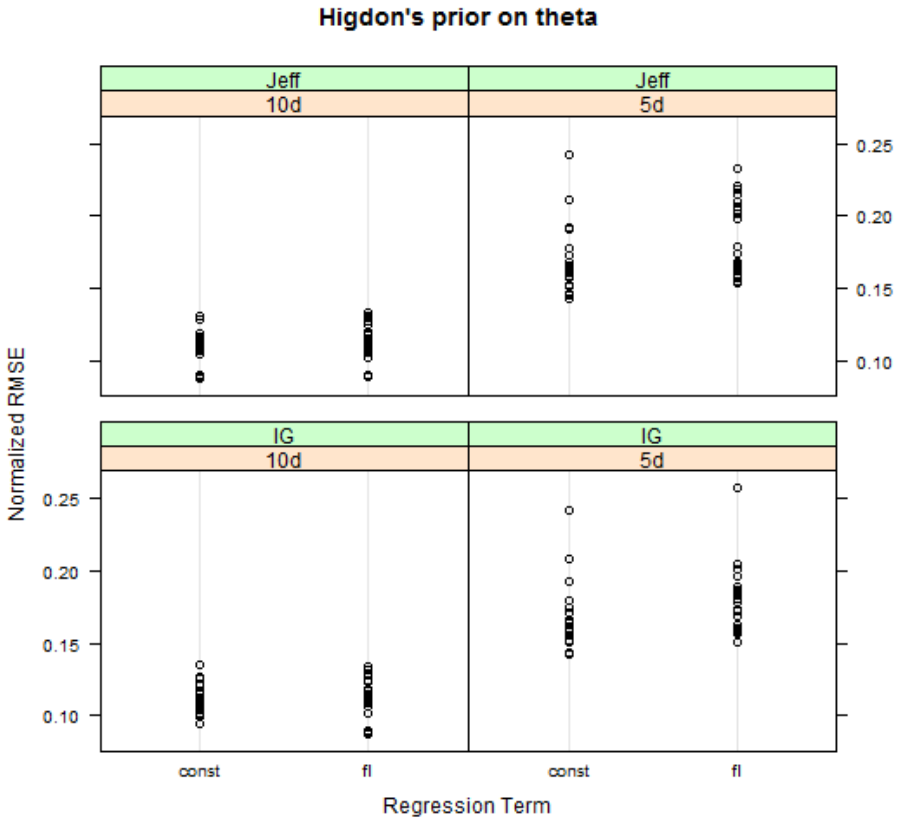

Figure 4.16: Prediction accuracy, given Higdon's prior on θ_i, PTW.

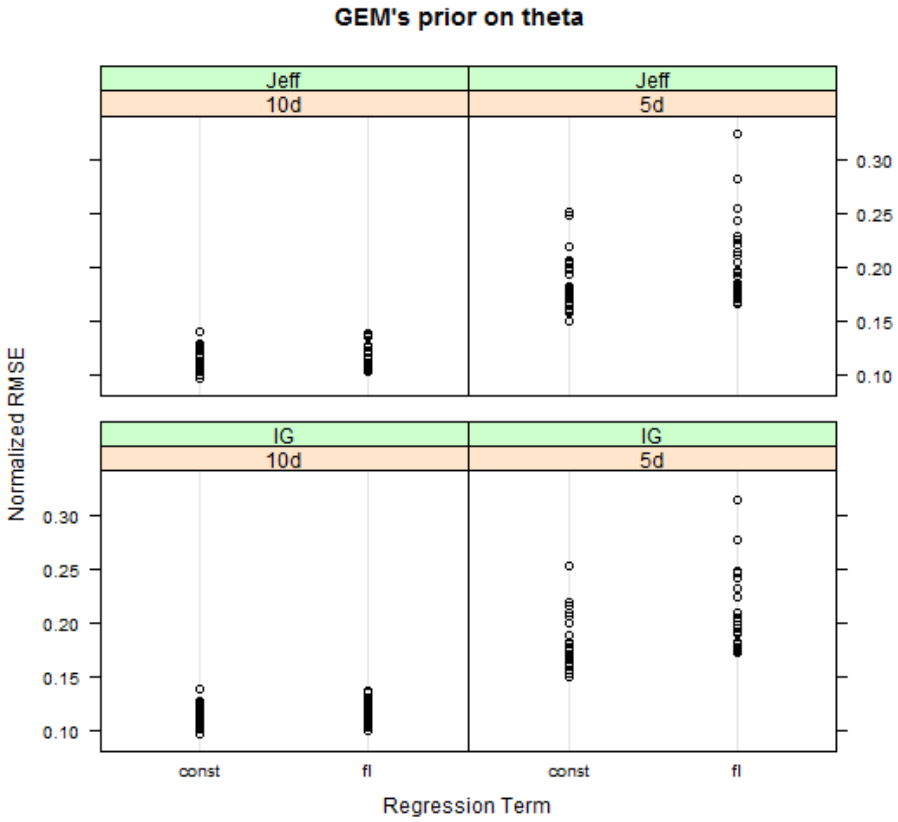

Figure 4.17: Prediction accuracy, given GEM's prior on θ_i, PTW.

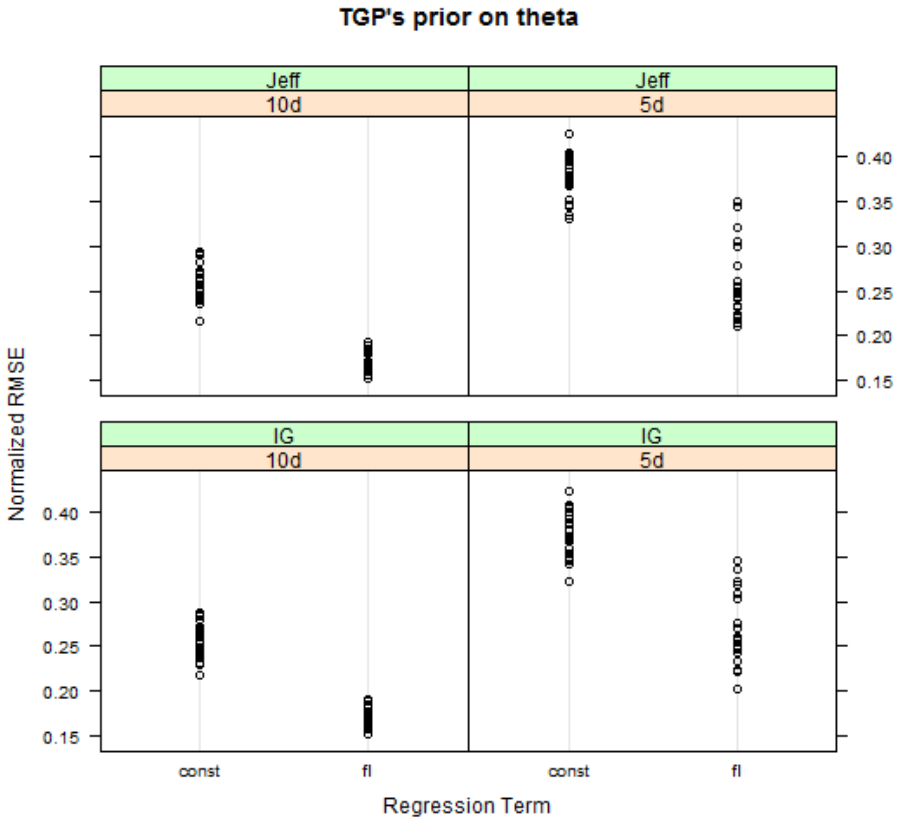

Figure 4.18: Prediction accuracy, given TGP's prior on θ_i, PTW.

Considering both the main effects and the interactions, an ANOVA for the Norm-RMSE numbers is given in Table 4.4.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
theta	2	1.84	0.92	1791.28	0.00	***
sigma	1	0.00	0.00	0.00	0.95	
reg	1	0.11	0.11	215.38	0.00	***
runs	1	1.01	1.01	1962.85	0.00	***
theta:sigma	2	0.00	0.00	0.14	0.87	
theta:reg	2	0.40	0.20	389.00	0.00	***
sigma:reg	1	0.00	0.00	0.06	0.80	
theta:runs	2	0.05	0.02	45.64	0.00	***
sigma:runs	1	0.00	0.00	0.00	0.95	
reg:ss	1	0.00	0.00	0.05	0.81	
theta:sigma:reg	2	0.00	0.00	0.36	0.70	
theta:sigma:runs	2	0.00	0.00	0.22	0.80	
theta:reg:runs	2	0.02	0.01	15.56	0.00	***
sigma:reg:runs	1	0.00	0.00	0.03	0.86	
theta:sigma:reg:runs	2	0.00	0.00	0.10	0.90	
Residuals	576	0.30	0.00			

Table 4.4: ANOVA results for the PTW computer model.

Again, we observe from Table 4.4 that all of the three main effects are significant, except for the prior on σ^2. Then, for each combination, we take its average value of the Normalized RMSE and draw the plots for interaction in Figures 4.19, 4.20 and 4.21.

From all of the figures and Table 4.4, we draw the following conclusions.

- The prior on θ_i is important for prediction accuracy and TPG's prior is least favoured.
- The effect of the number of runs is significant and a large number of runs ($10d$) is better than a small number of runs ($5d$).
- Regression terms for the GP only matter when TGP's prior is assumed on θ_i. Under TPG's prior, a full linear regression is preferred.
- There is not enough evidence to prove that the prior on σ^2 has a statistically significant effect on prediction accuracy.

Figure 4.19: Average prediction accuracy, given Higdon's prior on θ_i, PTW.

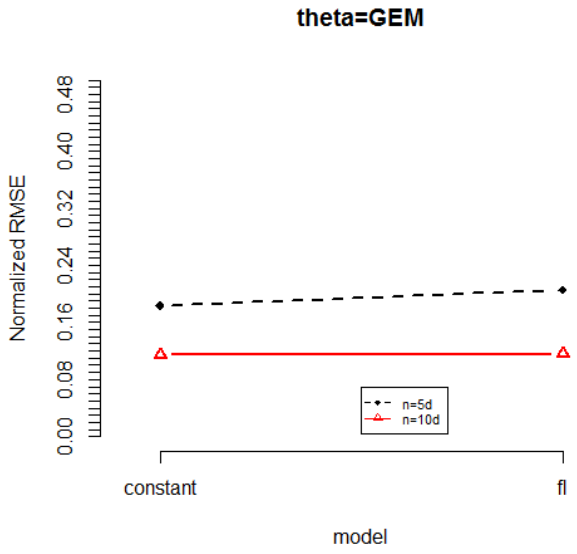

Figure 4.20: Average prediction accuracy, given GEM's prior on θ_i, PTW.

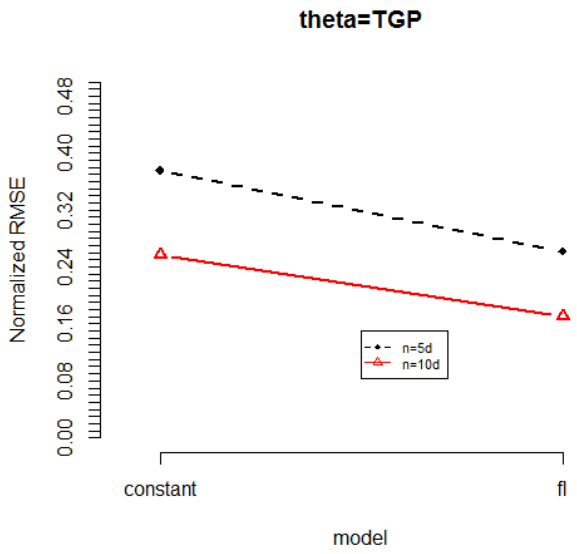

Figure 4.21: Average prediction accuracy, given TGP's prior on θ_i, PTW.

4.3.2 Actual Coverage Probability

Given a nominal coverage probability, each combination has 25 numbers for the ACP. As we did for the Borehole data, we take the average value of ACP for each combination and report the average ACP in Figures 4.22, 4.23, 4.24 and Figures 4.25, 4.26, 4.27.

The red lines in Figures 4.22, 4.23, 4.24 and Figures 4.25, 4.26, 4.27 represent the correct 90% and 95% coverage probability, respectively. We observe that the ACP for TGP's prior shows slight over coverage. The ACP for Higdon's prior and GEM's prior are satisfactory, since they are around the nominal coverage probability(90%, 95%).

The results from PTW are consistent with the results obtained from Borehole, that is, briefly speaking, in terms of the prediction accuracy, Higdon's prior on θ_i is favoured and large number of runs ($10d$) is always preferred. The Actual Coverage Probability for all of the 8 combinations under Higdon's prior on θ_i have acceptable performances.

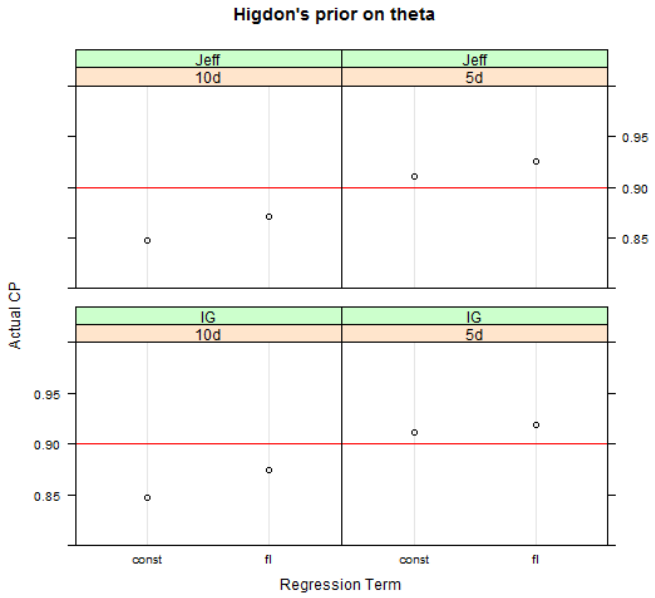

Figure 4.22: Prediction actual coverage probability, given Higdon's prior on θ_i, 90% true CP, PTW.

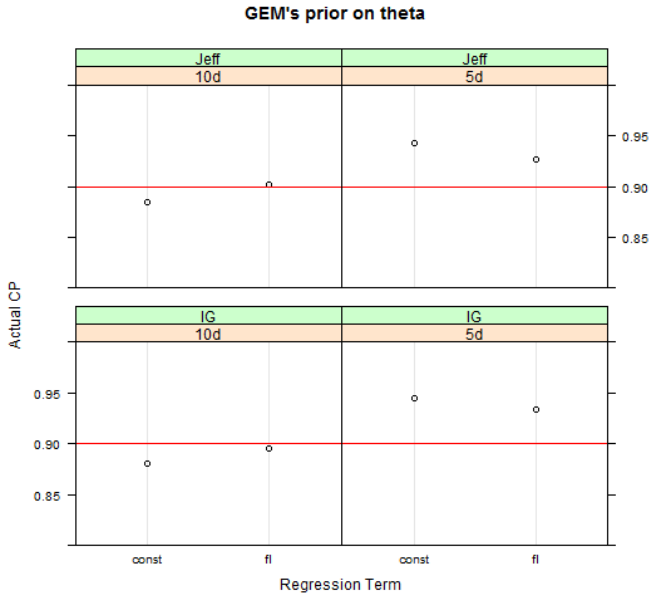

Figure 4.23: Prediction actual coverage probability, given GEM's prior on θ_i, 90% true CP, PTW.

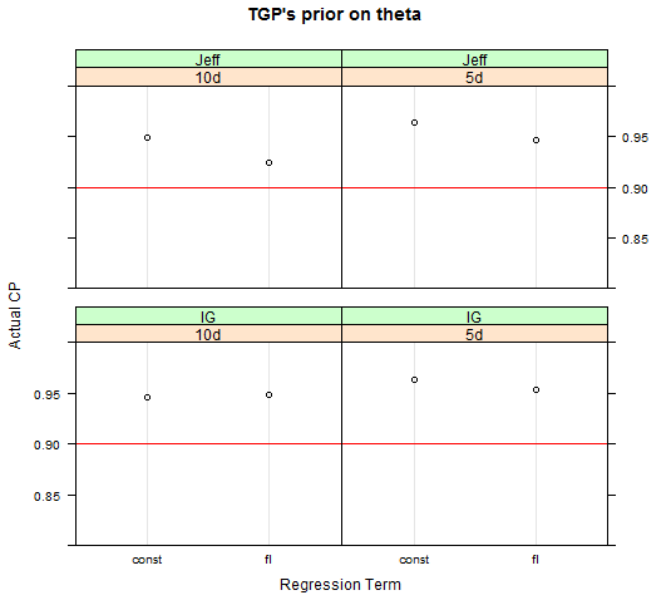

Figure 4.24: Prediction actual coverage probability, given TGP's prior on θ_i, 90% true CP, PTW.

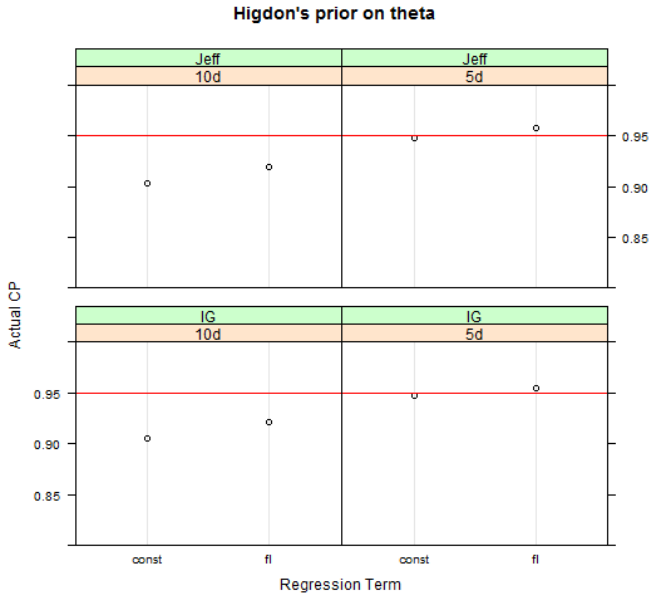

Figure 4.25: Prediction actual coverage probability, given Higdon's prior on θ_i, 95% true CP, PTW.

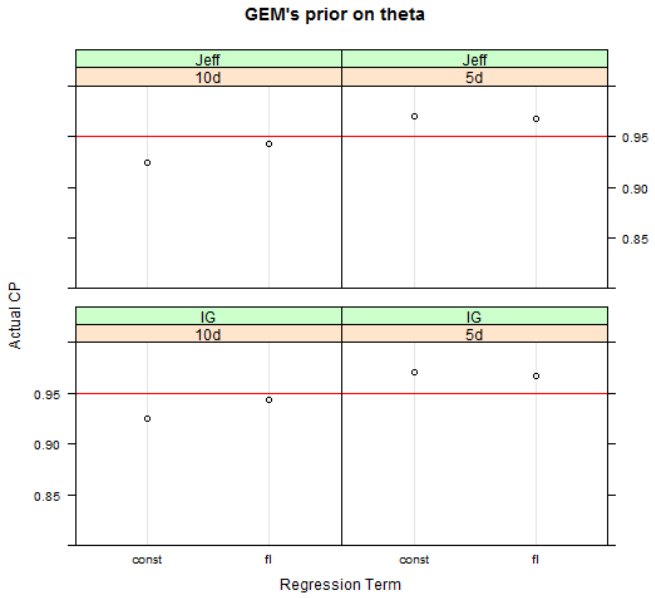

Figure 4.26: Prediction actual coverage probability, given GEM's prior on θ_i, 95% true CP, PTW.

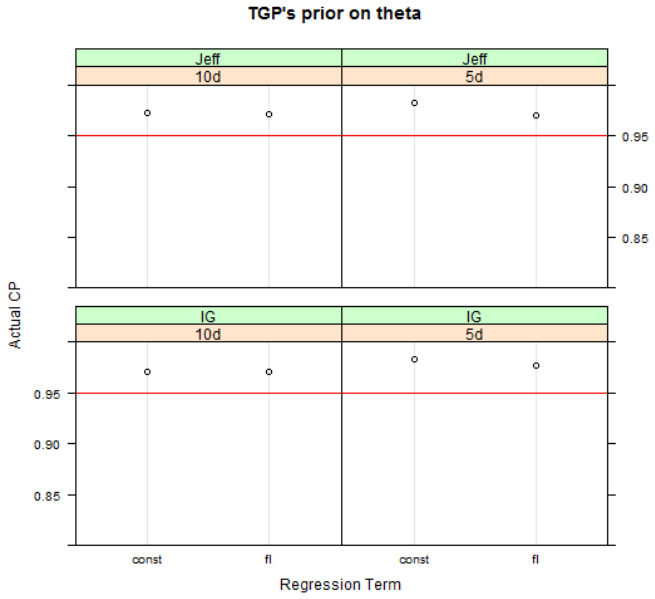

Figure 4.27: Prediction actual coverage probability, given TGP's prior on θ_i, 95% true CP, PTW.

4.4 Real Computer Model 3: G-protein

In this section, the real outputs are the G-protein dataset, which was used as a testbed in Chapter 3.

4.4.1 Prediction Accuracy

The Norm–RMSE results are presented in Figures 4.28, 4.29 and 4.30. Each combination has 25 Norm-RMSE numbers.

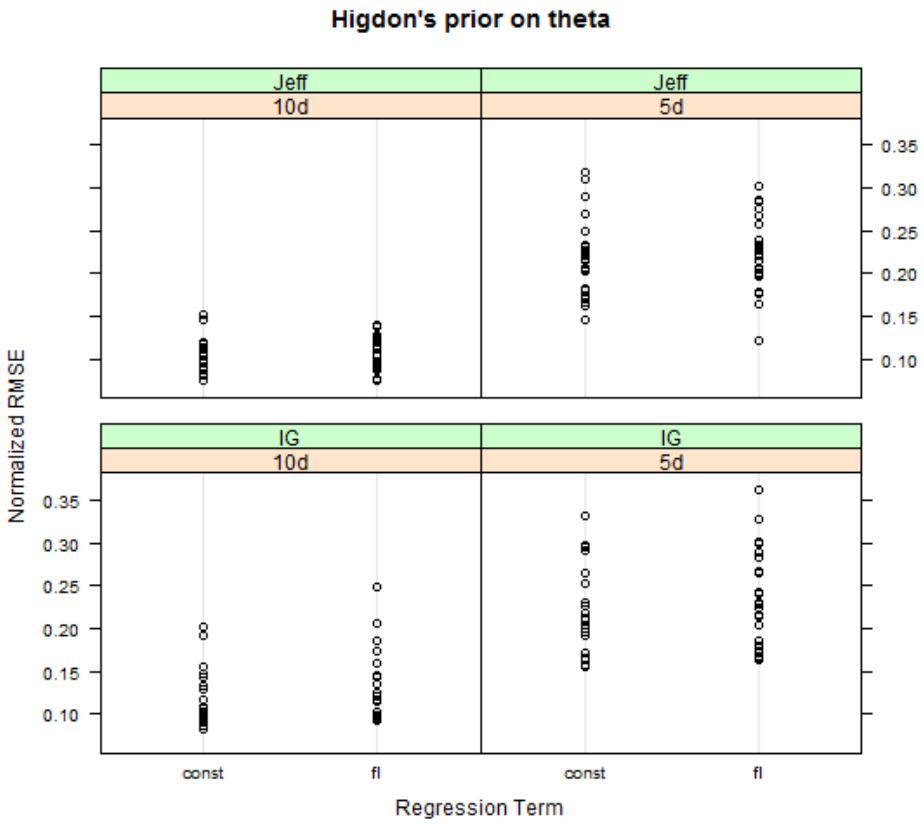

Figure 4.28: Prediction accuracy, given Higdon's prior on θ_i, G-protein.

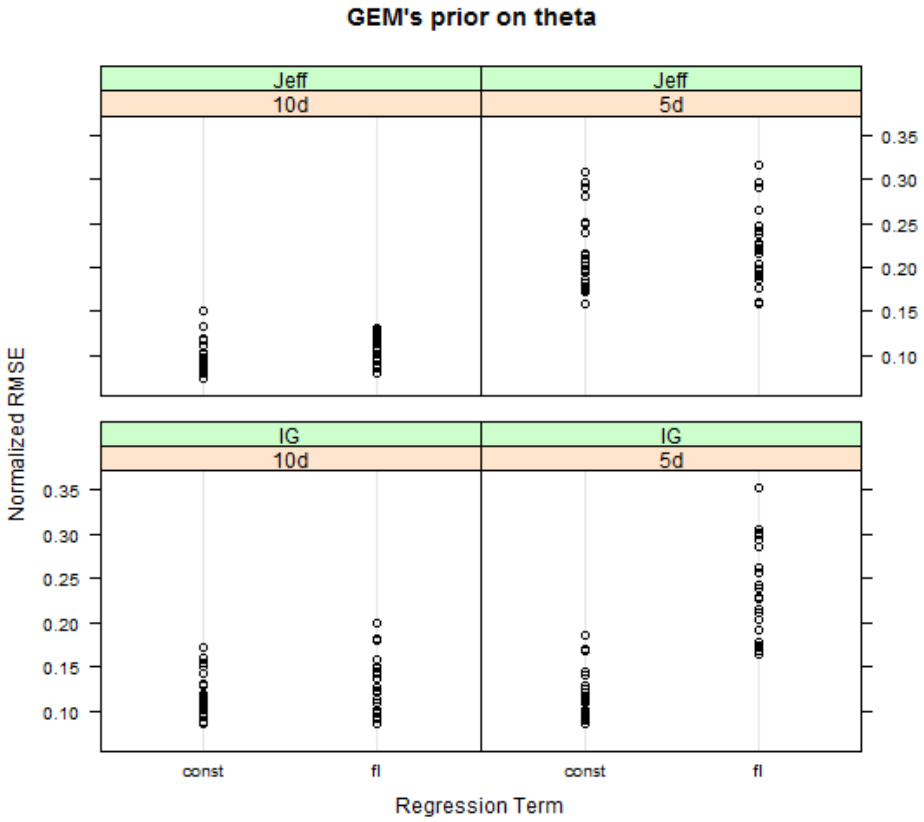

Figure 4.29: Prediction accuracy, given GEM's prior on θ_i, G-protein.

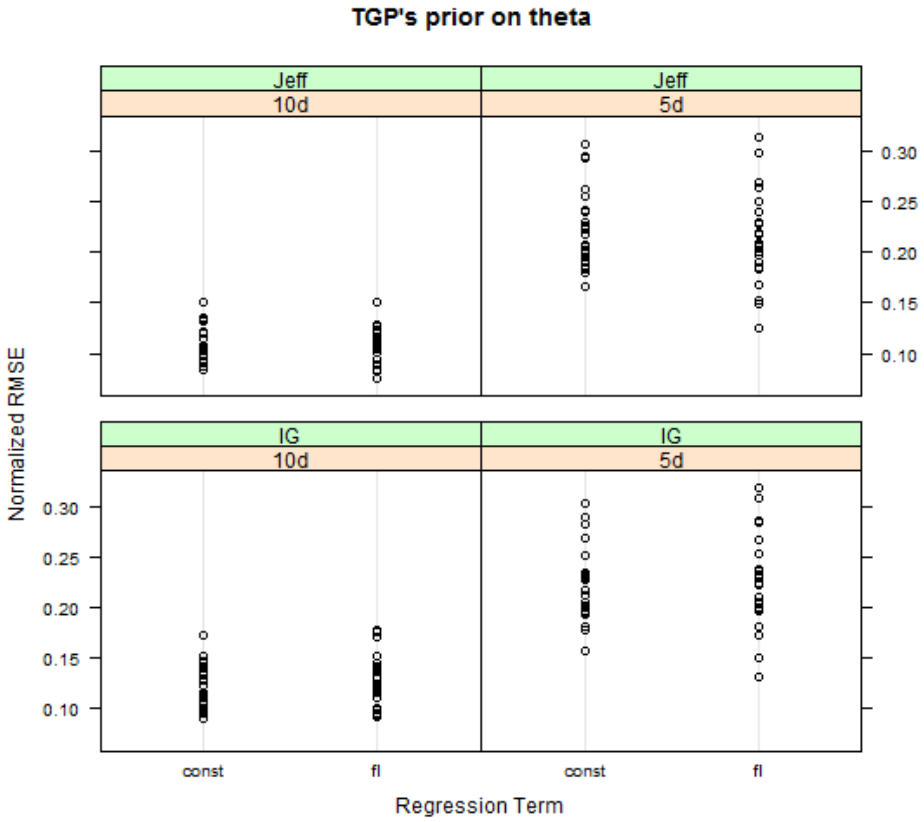

Figure 4.30: Prediction accuracy, given TGP's prior on θ_i, G-protein.

From Figures 4.28, 4.29 and 4.30, we can see that

- A large number of runs ($10d$) has more accurate prediction values than a small number of runs ($5d$).

Considering both the main effects and the interactions, an ANOVA for the Normalized RMSE numbers is given in Table 4.5.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
theta	2	0.03	0.01	10.55	0.00	***
sigma	1	0.00	0.00	1.73	0.19	
reg	1	0.03	0.03	28.01	0.00	***
ss	1	1.49	1.49	1192.70	0.00	***
theta:sigma	2	0.03	0.01	10.16	0.00	***
theta:reg	2	0.03	0.02	13.17	0.00	***
sigma:reg	1	0.02	0.02	17.37	0.00	***
theta:ss	2	0.02	0.00	6.87	0.00	**
sigma:ss	1	0.03	0.03	26.99	0.00	***
reg:ss	1	0.01	0.01	4.89	0.03	*
theta:sigma:reg	2	0.02	0.01	8.09	0.00	***
theta:sigma:ss	2	0.02	0.01	8.96	0.00	***
theta:reg:ss	2	0.02	0.01	8.24	0.00	***
sigma:reg:ss	1	0.01	0.01	11.86	0.00	***
theta:sigma:reg:ss	2	0.03	0.01	10.76	0.00	***
Residuals	576	0.72	0.00			

Table 4.5: ANOVA results for the G-protein computer model.

The ANOVA of the Gprotein data indicates that except the main effect of the prior on σ^2, all of the remaining factors including all of the 2–factor, 3–factor and even 4–factor interactions are significant at significance level 0.05. However, we also observe that the p-values for the prior on θ_i and the regression terms are larger than for the Borehole and PTW datasets. From visual inspections of the plots in Figures 4.31, 4.32 and 4.33, we can not see obvious performance difference for the two "significant" factors— prior on θ and the regression terms. The number of runs ($10d$ versus $5d$) is important, however.

From all of the figures and Table 4.5 , our conclusion is that only the number of runs is obviously important for the prediction accuracy and a-

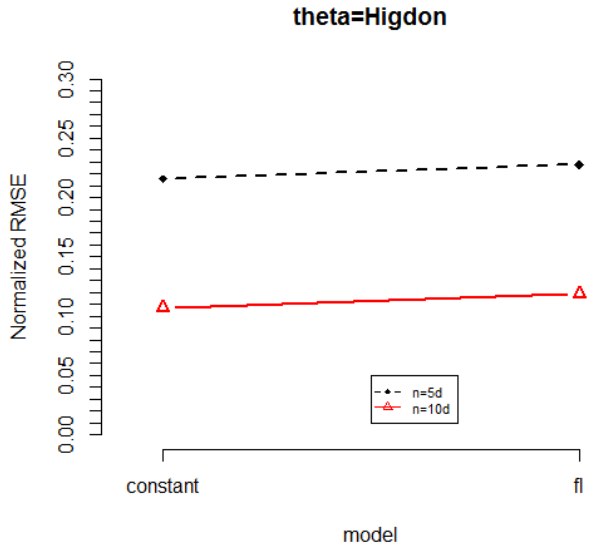

Figure 4.31: Average prediction accuracy, given Higdon's prior on θ_i, G-protein.

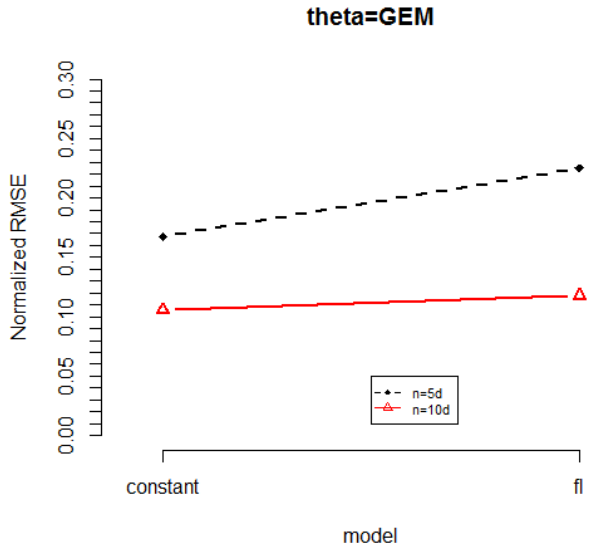

Figure 4.32: Average prediction accuracy, given GEM's prior on θ_i, G-protein.

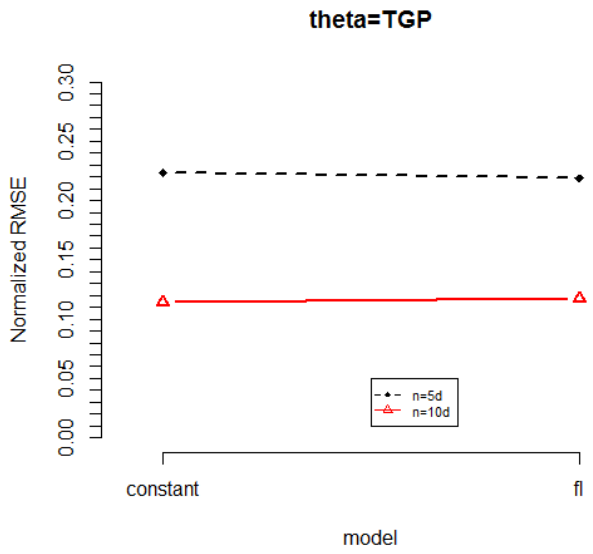

Figure 4.33: Average prediction accuracy, given TGP's prior on θ_i, G-protein.

gain, large number ($10d$) is preferred.

4.4.2 Actual Coverage Probability

For each combination, we report the average value for its ACP in Figures 4.34, 4.35, 4.36 and Figures 4.37, 4.38, 4.39.

The red lines in Figures 4.34, 4.35, 4.36 and Figures 4.37, 4.38, 4.39 represent the correct 90% and 95% coverage probability, respectively. We observe that the ACP for Higdon's prior and GEM's prior show sight under coverage. The worst case for Higdon's prior is 74.105% for 90% Nominal CP and 80.455% for 95% Nominal CP. The worst case for GEM's prior is 77.875% for 90% Nominal CP and 84.135% for 95% Nominal CP. Although the ACP values for the Higdon's prior and GEM's prior are not as good as they were in previous experiments, they are still much better than the MLE. Moreover, we also observe that Higdon's prior and GEM's prior have better ACP with the Jeffreys' prior on σ^2.

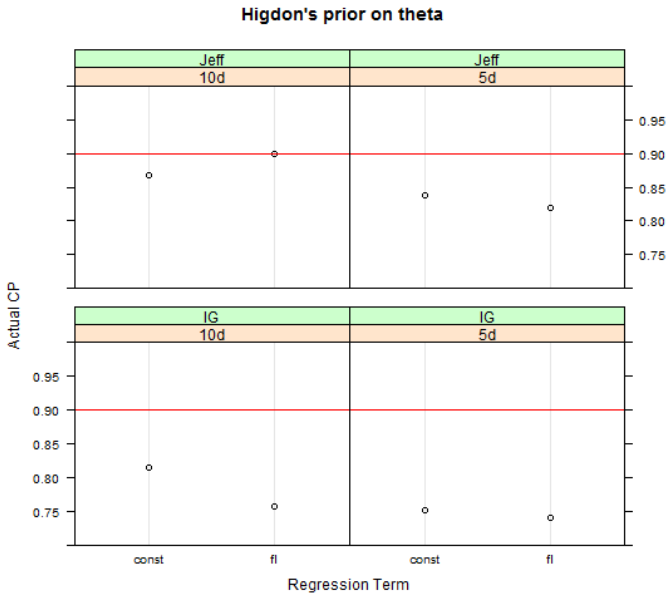

Figure 4.34: Prediction actual coverage probability, given Higdon's prior on θ_i, 90% true CP, G-protein.

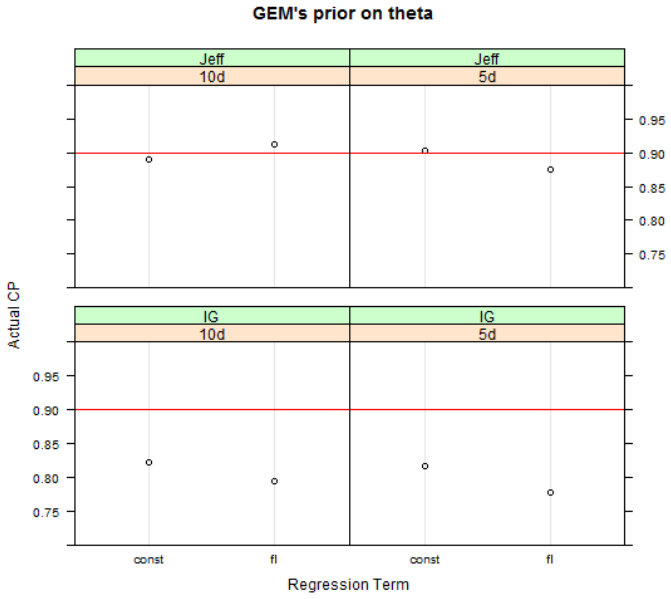

Figure 4.35: Prediction actual coverage probability, given GEM's prior on θ_i, 90% true CP, G-protein.

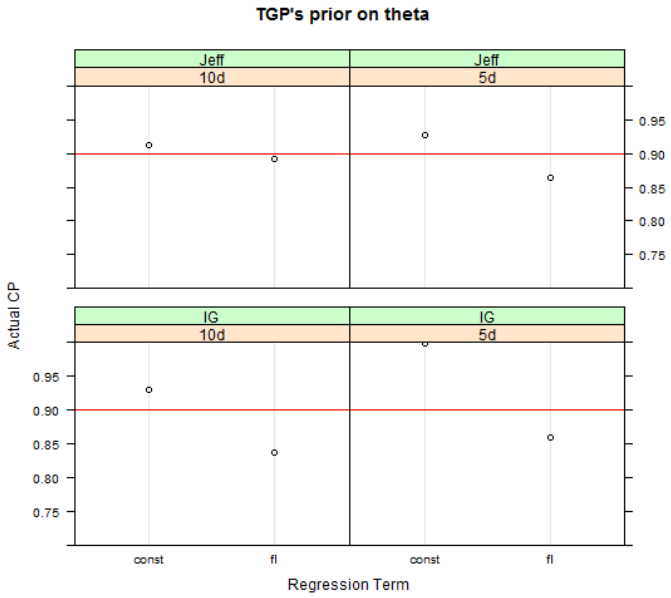

Figure 4.36: Prediction actual coverage probability, given TGP's prior on θ_i, 90% true CP, G-protein.

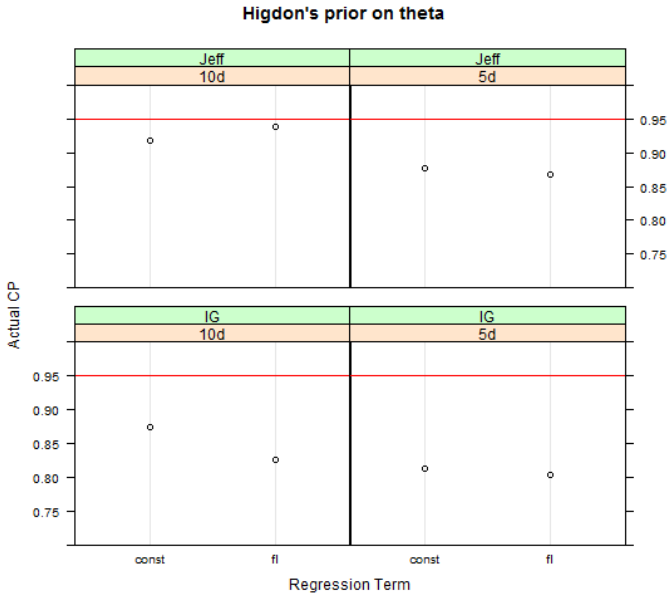

Figure 4.37: Prediction actual coverage probability, given Higdon's prior on θ_i, 95% true CP, G-protein.

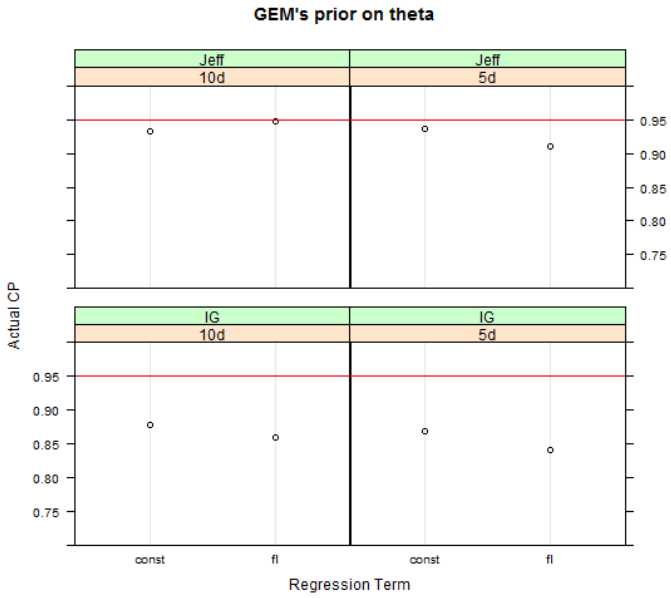

Figure 4.38: Prediction actual coverage probability, given GEM's prior on θ_i, 95% true CP, G-protein.

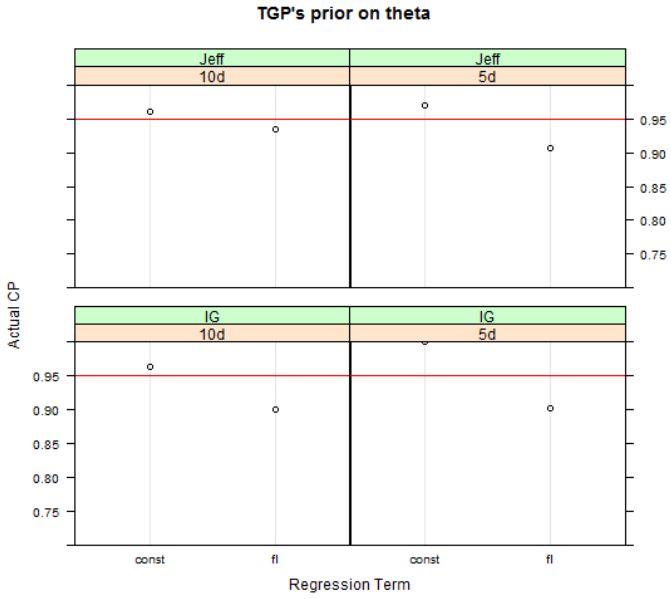

Figure 4.39: Prediction actual coverage probability, given TGP's prior on θ_i, 95% true CP, G-protein.

4.5 Summary of the Real Computer Models

Till now, we have conducted the full factorial design on three real outputs from computer models—Borehole, PTW and Gprotein. We can draw some common conclusions through the above analyses. First of all, in terms of the prediction accuracy, which is measure by Normalized RMSE, we have

- The prior on θ_i usually has significant effect on the prediction accuracy and Higdon's prior on θ_i is preferred.
- The number of runs is significant to the prediction accuracy and a large sample size ($10d$) is favoured.
- There is a lack of evidence to conclude that regression terms for the underlying GP have significant effects if Higdon's prior or GEM's prior on θ_i is assumed.
- There is not enough evidence to prove that the prior on σ^2 significantly affect the prediction accuracy.

Secondly, in term of the prediction Actual Coverage Probability, Higdon's prior and GEM's prior on θ_i combined with Jeffreys' prior on σ^2 show good coverage probability performance for all three applications.

Chapter 5

In-Depth Comparison: Simulation Study

In this chapter, instead of working on outputs from computer models, we will simulate data from a Gaussian Process to conduct experiments.

5.1 Details for the Simulation Study

From previous analyses, we know that the prior on σ^2 does not significantly affect the prediction accuracy and the regression model does not matter if Higdon's prior or GEM's prior is assumed on the correlation parameters. Therefore, starting from this section, we will focus on the two significant factors and drop the remaining factors, i.e., we fix prior on σ^2 as Jeffreys' prior and fix the regression term for the underlying GP as constant. The two varying factors and their levels are summarized below.

- Prior on θ_i: 3 levels, Hidgon's prior, TGP's prior and GEM's prior.
- Number of Runs: 2 levels, $5d$ and $10d$.

Hence, in total, we get $3 \times 2 = 6$ combinations.

5.1.1 Data Generating Procedures

We will simulate data from a Gaussian Process and carry out the full factorial experiments on the simulated data. We fix dimension as 5 and the number of replications as 50. For fixed true θ_i values, the procedures to generate training (input) data are described below:

- Step 1. Generate β_i from $N(0, 300^2)$.
- Step 2. Generate σ^2 from $IG(0.1, 5)$.
- Step 4. Given true θ_i values, compute the correlation matrix R.

- Step 5. Generate $\boldsymbol{y}$ from multivariate normal—$N(\boldsymbol{\beta}, \sigma^2 R)$.
- Step 6. Repeat Step 1 to Step 5 for 50 times.

Then, we generate 1000 testing points based on a random Latin Hypercube Design(LHD) using the R package *lhs* as the test set.

5.1.2 Choice of True θ_i

In general, there are two way to determine the true θ_i. The first way is to fix the true θ_i beforehand and the second way is to sample the true θ_i from a specific distribution at each replication. We have tried both way, but in the thesis, we only report the results obtained from the first way, because some large θ_i values are sampled due to randomness even if the distribution we sample from assigns heavy weights on small θ_i values. Very large θ_i values lead to unpredictable functions.

Fixing the True θ_i According to Canonical Configuration

In order to determine θ_i, we adapt a two-parameter class of canonical configurations, which was proposed by Loeppky et al. (2009). Let $\tau = \sum_{i=1}^{d} \theta_i$ and $\Psi = \sum_{i=1}^{d} \theta_i^2$. The two-parameter class of canonical configuration is defined by

$$\theta_i = \tau \left((1 - \frac{i-1}{d})^b - (1 - \frac{i}{d})^b \right), i = 1, \ldots, d,\ \tau \geq 0,\ b \geq 1. \tag{5.1}$$

- τ is a scale parameter, which control the **magnitude** of the θ_i.
- b is a parameter that control the **sparseness**.

In section 5.2, we let $b = 3$ and $\tau = 1$, so the true θ_i are

$$(0.488, 0.296, 0.152, 0.056, 0.008)$$

We name this simulation scenario 1. Then, in section 5.3, we keep $b = 3$, but increase τ to 3 and the true θ_i change to

$$(1.464, 0.888, 0.456, 0.168, 0.024)$$

We name this simulation scenario 2. In both scenarios, the sparseness is the same but the magnitude of the θ_i of scenario 2 is larger than that of scenario

1.

Except the fact that data are simulated from a Gaussian Process and that only two factors are considered, all of the other details in Chapter 5 are identical to those in Chapter 4.

5.2 Simulation Scenario 1

As we have discussed, in this section, the true θ_i are fixed as

$$(0.488, 0.296, 0.152, 0.056, 0.008)$$

.

5.2.1 Prediction Accuracy

The criterion is still Normalized RMSE. The scatter plots for TGP's prior versus Higdon's prior and GEM's prior versus Higdon's prior are provided by Figures 5.1, 5.2.

We can see that Hidgon's prior on θ is much better than TGP's prior and is slightly better than GEM's prior. In addition, $n = 10d$ has better accuracy than $n = 5d$. Those two conclusions agree with the conclusions we drew from the analyses of the real computer models.

5.2.2 Actual Coverage Probability

For each combination, we have 50 numbers for its ACP. We take the mean for each combination and report the means in Tables 5.1 and 5.2. The true coverage probability is 90% and 95%, respectively.

	5d	10d
Higdon's prior	89.9%	90.7%
TGP's prior	96.6%	95.6%
GEM's prior	94.5%	92.4%

Table 5.1: Actual coverage probability, 90% true CP, scenario 1.

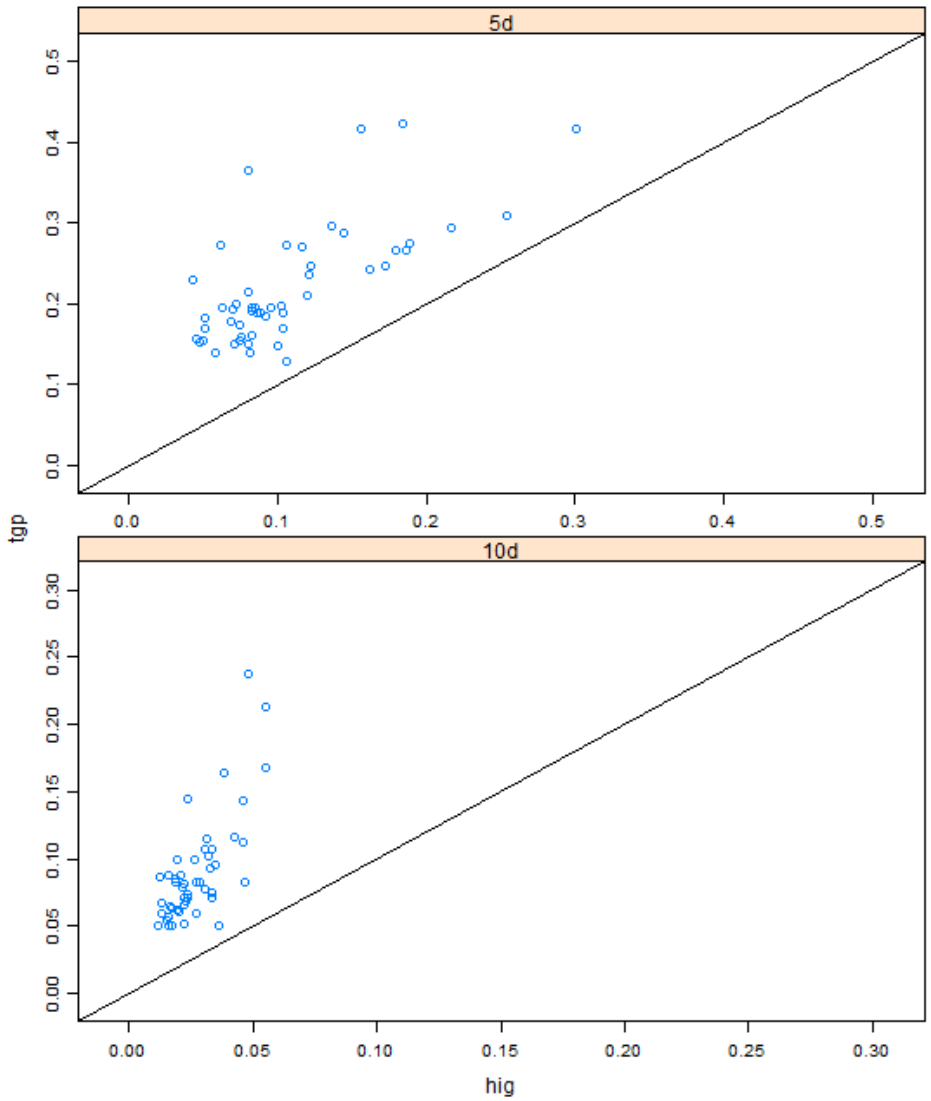

Figure 5.1: Norm-RMSE for TGP's prior versus Higdon's prior, scenario 1.

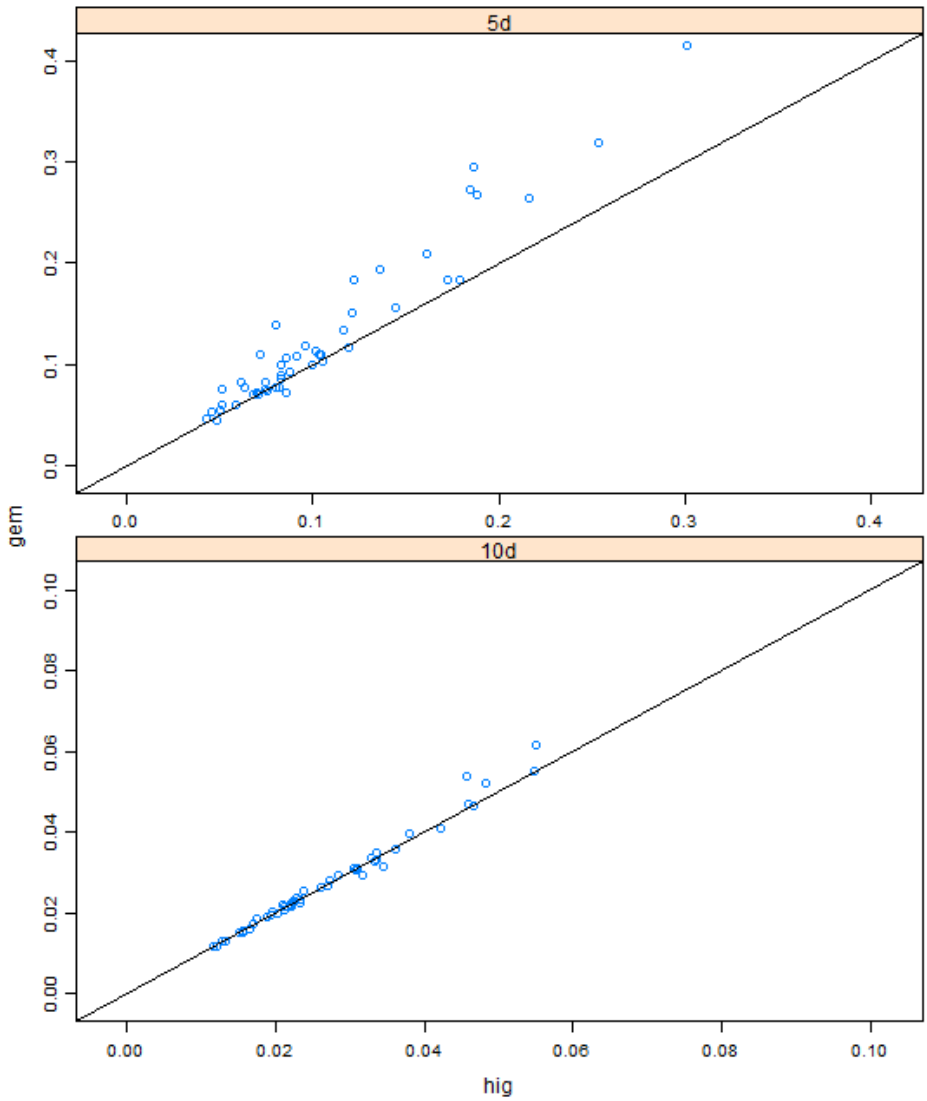

Figure 5.2: Norm-RMSE for GEM's prior versus Higdon's prior, scenario 1.

	5d	10d
Higdon's prior	94.1%	95.3%
TGP's prior	98.7%	98.8%
GEM's prior	96.9 %	96.4%

Table 5.2: Actual coverage probability, 95% true CP, scenario 1.

From Tables 5.1 and 5.2, the ACP of Higdon's prior is the closest to the true coverage probability among the three priors. Besides, we also notice that as the number of runs increases, the ACP tends to be closer to the true coverage probability.

5.3 Simulation Scenario 2

In this section, the true θ_i are fixed as

$$(1.464, 0.888, 0.456, 0.168, 0.024)$$

.

5.3.1 Prediction Accuracy

The scatter plots for TGP's prior versus Higdon's prior and GEM's prior versus Higdon's prior are provided by Figures 5.3 and 5.4.

From Figures 5.3 and 5.4 , we get exactly the same two conclusions as we got from previous analyses. We do not state them again. Comparing Figures 5.3 and 5.4 with Figures 5.1 and 5.2, it is obvious that the prediction accuracy of Scenario 1 is better than the accuracy of Scenario 2. The reason lies in the magnitude of θ_i. In general, large θ_i values reduce the correlation between points in the input space, thus increase the prediction difficulty.

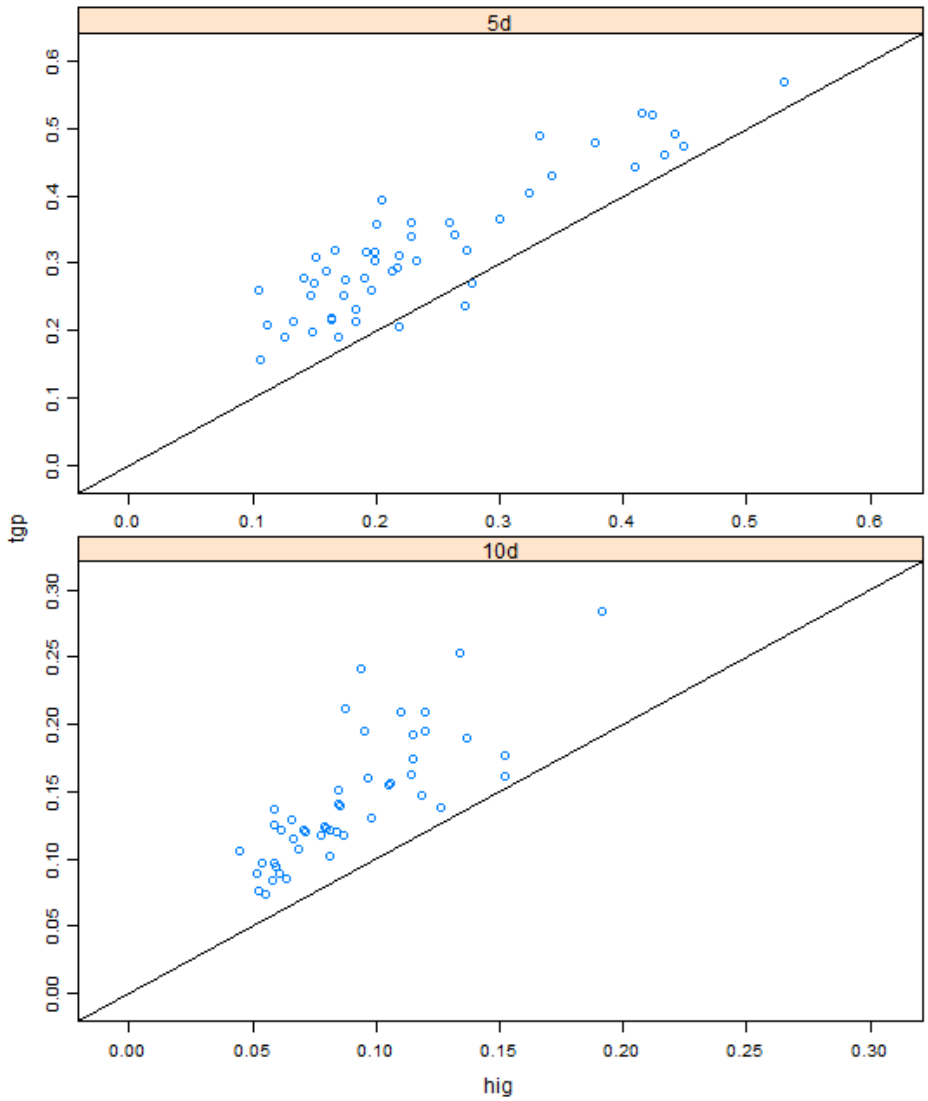

Figure 5.3: Norm-RMSE for TGP's prior versus Higdon's prior, scenario 2.

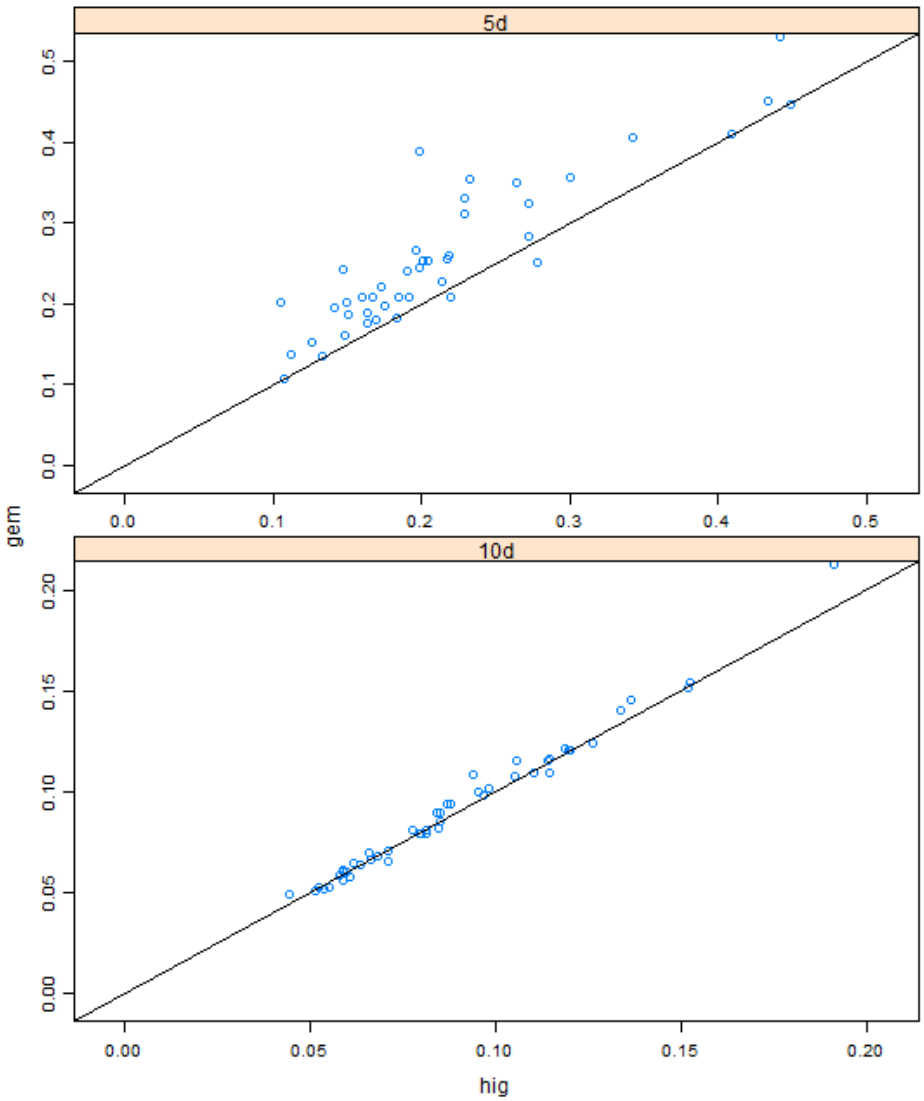

Figure 5.4: Norm-RMSE for GEM's prior versus Higdon's prior, scenario 2.

5.3.2 Actual Coverage Probability

For each combination, we have 50 numbers for its ACP. We take the mean for each combination and report it in Tables 5.3 and 5.4. The true coverage probability is 90% and 95%, respectively.

	5d	10d
Higdon's prior	93.6%	91.1%
TGP's prior	94.2%	93.9%
GEM's prior	95.2%	92.8%

Table 5.3: Actual coverage probability, 90% true CP, scenario 2.

	5d	10d
Higdon's prior	96.4%	95.5%
TGP's prior	97.2%	96.4%
GEM's prior	97.8 %	96.5%

Table 5.4: Actual coverage probability, 95% true CP, scenario 2.

From Tables 5.3 and 5.4, the ACP of Higdon's prior is the closest to the true coverage probability among the three priors. Besides, we also notice that as the number of runs increases, the ACP tends to be closer to the correct coverage probability.

5.4 Summary of the Simulation Study

From the simulation study, we can safely draw the following conclusions

- In terms of both the prediction accuracy and the actual coverage probability, Higdon's prior on θ_i is highly preferred.
- In terms of both the prediction accuracy and the actual coverage probability, a large number of runs ($n = 10d$) is highly preferred.

Chapter 6

Conclusion & Discussion

Currently, Bayesian methods are popular among the statistical community. To be more specific, in the analysis of computer experiments, there exist several statistical methods within the Bayesian paradigm. Before making any concluding remarks, let's first concisely summarize what we have done. We review three well-known Bayesian methods in Chapter 2. The tentative comparison in Chapter 3 suggests that the performances of the three Bayesian approaches are quite different. Motivated by the differences, we specified 4 factors and allocated different levels for the factors according to the three existing methods. With two assessment criteria in mind, full factorial designs were then conducted both on real computer codes in Chapter 4 and on simulated data in Chapter 5, with the purpose of identifying the important factors and their best levels.

Here, we reiterate our findings. Among the 4 factors we specified,

- The prior on the correlation parameters θ_i is significant to the prediction performance, and Higdon's prior, which favours small θ values, is highly preferred.
- The number of runs significantly affects the prediction performance and a large number ($10d$) is favoured.
- So far, there is not enough evidence to conclude that the prior on σ^2 is significant for prediction accuracy.
- So far, there is a lack of sufficient facts to conclude that the regression term for the Gaussian Process is important for prediction accuracy when Higdon's prior or GEM's prior is assumed for the correlation parameters.

Through the previous analyses, we not only identified the significant factors, but also discover the best level for each factor. The best combination that we recommend is

- Higdon's prior for the correlation parameters.

- $n = 10d$ as the number of runs.
- Jeffreys' prior for σ^2.
- Constant model for the underlying Gaussian Process.

Although the prior on σ^2 is not significant for prediction accuracy, we recommend Jeffreys' prior on σ^2 for the benefit of actual coverage probability.

In addition, the constant model is suggested over a full linear model for the Gaussian Process. It is noticed that the full linear model is more computational intensive than the constant model. Although the full linear model has better prediction accuracy when assuming TGP's prior on θ_i, the difference in prediction accuracy does not exist with Higdon's prior on θ_i, which, in fact, is recommended. Also, since our research does not involve the extrapolation of data, we deem a constant model is enough for the Gaussian Process.

In terms of approximation of the output from a computer model, the Gaussian Process model is currently the most popular statistical method that is widely used in practice. It is also of paramount importance to search for a suitable prior that can be set as the default for the correlation parameters in Bayesian analysis. Higdon's prior is one of the useful informative priors that allocates heavy weight to small θ_i values. From all of the analyses we have conducted in this thesis, Higdon's prior is highly preferred and can be trusted as a default prior for future analysis.

A large number of runs is favoured through our analyses. This makes perfect intuitive sense, since large sample size contain more information that can be utilized to make more accurate predictions. To some degree, the prediction accuracy improves as the number of runs increases. However, increasing the number of runs increases the computation of analysis, since the number of rows and columns of the correlation matrix also increase accordingly. Let n be the number of runs. As n becomes very large, the correlation matrix, denoted as $R_{n\times n}$, will be too "cumbersome" to take inverse or do any other matrix manipulations. Currently, we believe that $n = 10d$, where d is the dimension of the data, is enough for well-behaved problems, such as most of the demonstrating examples in the thesis.

In future, we would like to continue our research on the Bayesian approach for the analysis of computer experiments. Based on this thesis, one improvement we can do is to add more choices for the correlation structure. We only focus on the Gaussian correlation structure here. However, it is also recognized (Chen et al. (2013))that the Gaussian correlation structure performs badly under some circumstances. Therefore, adding more correlation structures, such as the Power Exponential is needed in future work.

Bibliography

Abt, M. (1999). Estimating the prediction mean squared error in gaussian stochastic processes with exponential correlation structure. *Scandinavian Journal of Statistics*, 26:563–578.

Ba, S. and Joseph, R. (2012). Composite gaussian process models for emulating expensive functions. *Annals of Appied Statistics*, 6(4):1838–1860.

Bastos, L. and O'Hagan, A. (2009). Diagnostics for gaussian process emulators. *Technometrics*, 51(4):425–438.

Berger, J., Oliveira, V. D., and Sanso, B. (2001). Objective bayesian analysis of spatially correlated data. *Journal of the American Statistical Association*, 96:1361–1374.

Chen, H., Loeppky, J., Sacks, J., and Welch, W. (2013). A laboratory to assess analysis methods for computer experiments.

Gelman, A., Carlin, J., Stern, H., and Rubin, D. (1995). *Bayesian data analysis*. Chapman and Hall.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721C741.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). *Markov Chain Monte Carlo in practice*. Chapman and Hall.

Gramacy, R. (2005). *Bayesian Treed Gaussian Process Models*. PhD thesis, University OF California, Santa Cruz.

Gramacy, R. and Lee, H. (2008). Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130.

Gramacy, R. and Lee, H. (2009). Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2):130–145.

Hartigan, J. (1964). Invariant prior distributions. *Annals of Mathematical Statistics*, 35:836–845.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583.

Higdon, D., Kennedy, M., Cavendish, J., Cafeo, J., and Ryne, R. (2004). Combining field observations and simulations for calibration and prediction. *SIAM Journal of Scientific Computing*, 26:448–466.

Kennedy, M. (2004). Description of the gaussian process model used in gem-sa. Technical report, University of Sheffield.

Kennedy, M. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of Royal Statistical Association, Series B*, 63(3):425–464.

Loeppky, J., Sacks, J., and Welch, W. (2009). Choosing the sample size of a computer experiments: A practical guide. *Technometrics*, 51:366376.

McKay, M., Beckman, R., and Conover, W. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., and Teller, A. (1953). Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, 21:1087C1091.

Morris, M., Mitchell, T., and Ylvisaker, D. (1993). Bayesian design and analysis of computer experiments: Use of derivatives in surface prediction. *Technometrics*, 35:243–255.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, 7:308–313.

Preston, D., Tonks, D., and Wallace, D. (2003). Model of plastic deformation for extreme loading conditions. *Journal of Applied Physics*, 93:211–220.

Robert, C. (2001). *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk metropolis algorithms. *The Annals of Applied Probability*, 7:110–120.

Sacks, J., Welch, W., Mitchell, T., and Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435.

Santner, T., Williams, B., and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer Series in Statistics. Springer Press, first edition.

Shanno, D. F. and Kettler, P. C. (1970). Optimal conditioning of quasi-newton methods. *Mathematics of Computation*, 24:657–664.

Smith, B. (2007). boa: An r package for mcmc output convergence assessment and posterior inference. *Journal of Statistical Software*, 21:1–37.

Welch, W., Buck, R., Sacks, J., Wynn, H., and Toby Mitchell, M. M. (1992). Screening, predicting, and computer experiments. *Technometrics*, 34(1):15–25.

Yi, T.-M., Fazel, M., Liu, X., Otitoju, T., Goncalves, J., Papachristodoulou, A., Prajna, S., and Doyle, J. (2005). *Application of robust model validation using SOSTOOLS to the study of G-protein signaling in yeast*. Proceedings of the First Conference on Foundations of Systems Biology in Engineering, FOSBE.

Appendix A

Posterior Distribution of θ_i for $\pi(\sigma^2) = IG(\alpha_1, \alpha_2)$

We present the detailed derivation for the posterior distribution of θ_i here. The final results were given by equation (4.5) and equation(4.6). The posterior distribution of θ_i is obtained by integrating the σ^2 and $\boldsymbol{\beta}$ out.

(1) If $\pi(\sigma^2) = IG(\alpha_1, \alpha_2)$ and $\pi(\boldsymbol{\beta}) = 1$

$$
\begin{aligned}
& p(\boldsymbol{\theta}|\boldsymbol{y}) \\
& \propto \int\int \pi(\boldsymbol{\theta}, \boldsymbol{\beta}, \sigma^2|\boldsymbol{y}) L(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\beta}, \sigma^2) d\boldsymbol{\beta} d\sigma^2 \\
& \propto \int\int \pi(\boldsymbol{\theta}) IG(\alpha_1, \alpha_2) \frac{1}{(\sigma^2)^{n/2}|R|^{1/2}} \exp\{-\frac{(\boldsymbol{y} - F\boldsymbol{\beta})^T R^{-1} (\boldsymbol{y} - F\boldsymbol{\beta})}{2\sigma^2}\} d\boldsymbol{\beta} d\sigma^2 \\
& \propto \frac{\pi(\boldsymbol{\theta})}{|R|^{\frac{1}{2}}} \int \frac{IG(\alpha_1, \alpha_2)}{(\sigma^2)^{n/2}} \int \exp\{-\frac{(\boldsymbol{y} - F\boldsymbol{\beta})^T R^{-1} (\boldsymbol{y} - F\boldsymbol{\beta})}{2\sigma^2}\} d\boldsymbol{\beta} d\sigma^2 \qquad (1)
\end{aligned}
$$

Let's first integrate out $\boldsymbol{\beta}$. Since $\hat{\boldsymbol{\beta}} = (F^T R^{-1} F)^{-1} F^T R^{-1} \boldsymbol{y}$, we have

$$
\begin{aligned}
& (\boldsymbol{y} - F\boldsymbol{\beta}) R^{-1} (\boldsymbol{y} - F\boldsymbol{\beta}) \\
& = \boldsymbol{\beta}^T F^T R^{-1} F \boldsymbol{\beta} - \boldsymbol{\beta}^T F^T R^{-1} \boldsymbol{y} - \boldsymbol{y}^T R^{-1} F \boldsymbol{\beta} + \boldsymbol{y}^T R^{-1} \boldsymbol{y} \\
& = (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T (F^T R^{-1} F)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) - \hat{\boldsymbol{\beta}}^T F^T R^{-1} F \hat{\boldsymbol{\beta}} + \boldsymbol{y}^T R^{-1} \boldsymbol{y}
\end{aligned}
$$

Therefore, Let E denote $\exp\{-\frac{(\boldsymbol{y} - F\boldsymbol{\beta})^T R^{-1} (\boldsymbol{y} - F\boldsymbol{\beta})}{2\sigma^2}\}$

$$
\begin{aligned}
& \int E d\boldsymbol{\beta} \\
& = \exp\{-\frac{\boldsymbol{y}^T R^{-1} \boldsymbol{y} - \hat{\boldsymbol{\beta}}^T F^T R^{-1} F \hat{\boldsymbol{\beta}}}{2\sigma^2}\} \int \exp\{-\frac{(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T (F^T R^{-1} F)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})}{2\sigma^2}\} d\boldsymbol{\beta} \\
& = \exp\{-\frac{\boldsymbol{y}^T R^{-1} \boldsymbol{y} - \hat{\boldsymbol{\beta}}^T F^T R^{-1} F \hat{\boldsymbol{\beta}}}{2\sigma^2}\} |\frac{F^T R^{-1} F}{\sigma^2}|^{-\frac{1}{2}} \qquad (2)
\end{aligned}
$$

Since

$$\int \underbrace{|\frac{F^T R^{-1} F}{\sigma^2}|^{\frac{1}{2}} \exp\{-\frac{(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^T (F^T R^{-1} F)(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{2\sigma^2}\}}_{\boldsymbol{\beta} \sim N(\hat{\boldsymbol{\beta}}, [\frac{F^T R^{-1} F}{\sigma^2}]^{-1})} d\boldsymbol{\beta} = 1$$

Combining the fact that $\hat{\sigma^2} = \frac{(\boldsymbol{y}-F\hat{\boldsymbol{\beta}})^T R^{-1} (\boldsymbol{y}-F\hat{\boldsymbol{\beta}})}{n-k}$, we continue from (2)

$$\begin{aligned}
&\int E d\boldsymbol{\beta} \\
&= \exp\{-\frac{\boldsymbol{y}^T R^{-1} \boldsymbol{y} - \hat{\boldsymbol{\beta}}^T F^T R^{-1} \hat{\boldsymbol{\beta}}}{2\sigma^2}\} |\frac{F^T R^{-1} F}{\sigma^2}|^{-\frac{1}{2}} \\
&= \exp\{-\frac{(\boldsymbol{y}-F\hat{\boldsymbol{\beta}})^T R^{-1} (\boldsymbol{y}-F\hat{\boldsymbol{\beta}})}{2\sigma^2}\} |\frac{F^T R^{-1} F}{\sigma^2}|^{-\frac{1}{2}} \\
&= \exp\{-\frac{(n-k)\hat{\sigma}^2}{2\sigma^2}\} |\frac{F^T R^{-1} F}{\sigma^2}|^{-\frac{1}{2}}
\end{aligned}$$

Hence, we continue from (1)

$$\begin{aligned}
p(\boldsymbol{\theta}|\boldsymbol{y}) &\propto \int\int \pi(\boldsymbol{\theta}, \boldsymbol{\beta}, \sigma^2 | \boldsymbol{y}) L(\boldsymbol{y} | \boldsymbol{\theta}, \boldsymbol{\beta}, \sigma^2) d\boldsymbol{\beta} d\sigma^2 \\
&\propto \frac{\pi(\boldsymbol{\theta})}{|R|^{\frac{1}{2}}} \int \frac{IG(\alpha_1, \alpha_2)}{(\sigma^2)^{n/2}} \int \exp\{-\frac{(\boldsymbol{y}-F\boldsymbol{\beta})^T R^{-1} (\boldsymbol{y}-F\boldsymbol{\beta})}{2\sigma^2}\} d\boldsymbol{\beta} d\sigma^2 \\
&\propto \frac{\pi(\boldsymbol{\theta})}{|R|^{\frac{1}{2}}} \int \frac{IG(\alpha_1, \alpha_2)}{(\sigma^2)^{n/2}} \exp\{-\frac{(n-k)\hat{\sigma}^2}{2\sigma^2}\} |\frac{F^T R^{-1} F}{\sigma^2}|^{-\frac{1}{2}} d\sigma^2 \\
&\propto \frac{\pi(\boldsymbol{\theta})}{|R|^{\frac{1}{2}} |F^T R^{-1} F|^{\frac{1}{2}}} \int \frac{(\sigma^2)^{-\alpha_1 - 1} \exp\{-\frac{\alpha_2}{\sigma^2}\}}{(\sigma^2)^{(n-k)/2}} \exp\{-\frac{(n-k)\hat{\sigma}^2}{2\sigma^2}\} d\sigma^2 \\
&\propto \frac{\pi(\boldsymbol{\theta})}{|R|^{\frac{1}{2}} |F^T R^{-1} F|^{\frac{1}{2}}} \int (\sigma^2)^{-(\alpha_1 + \frac{n-k}{2} + 1)} \exp\{-\frac{\alpha_2 + \frac{(n-k)}{2}\hat{\sigma}^2}{\sigma^2}\} d\sigma^2 \\
&\propto \frac{\pi(\boldsymbol{\theta})}{|R|^{\frac{1}{2}} |F^T R^{-1} F|^{\frac{1}{2}}} \times \frac{\Gamma(\alpha_1 + \frac{n-k}{2})}{(\alpha_2 + \frac{n-k}{2}\hat{\sigma}^2)^{(\alpha_1 + \frac{n-k}{2})}} \\
&\propto \frac{\pi(\boldsymbol{\theta})}{|R|^{\frac{1}{2}} |F^T R^{-1} F|^{\frac{1}{2}} (\alpha_2 + \frac{n-k}{2}\hat{\sigma}^2)^{(\alpha_1 + \frac{n-k}{2})}} \qquad (3)
\end{aligned}$$

Since

$$\int \underbrace{\frac{(\alpha_2 + \frac{n-k}{2}\hat{\sigma}^2)^{(\alpha_1 + \frac{n-k}{2})}}{\Gamma(\alpha_1 + \frac{n-k}{2})} (\sigma^2)^{-(\alpha_1 + \frac{n-k}{2} + 1)} \exp\{-\frac{\alpha_2 + \frac{n-k}{2}\hat{\sigma}^2}{\sigma^2}\}}_{\mathrm{IG}(\alpha_1 + \frac{n-k}{2}, \alpha_2 + \frac{n-k}{2}\hat{\sigma}^2)} d\sigma^2 = 1$$

Equation (3) is the posterior distribution of $\boldsymbol{\theta}$ given by (4.5), which allows MCMC.

Appendix B

Posterior Distribution of θ_i for $\pi(\sigma^2) \propto 1/\sigma^2$

(2) If $\pi(\sigma^2) \propto 1/\sigma^2$ and $\pi(\boldsymbol{\beta}) = 1$.

The Jeffreys' prior is a special case of Inverse Gamma as $\alpha_1 \to 0, \alpha_2 \to 0$. Let's plug $\alpha_1 = \alpha_2 = 0$ into equation (3), we have

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{y}) &\propto \frac{\pi(\boldsymbol{\theta})}{|R|^{\frac{1}{2}}|F^T R^{-1} F|^{\frac{1}{2}}(0 + \frac{n-k}{2}\hat{\sigma}^2)^{(0+\frac{n-k}{2})}} \\ &\propto \frac{\pi(\boldsymbol{\theta})}{|R|^{\frac{1}{2}}|F^T R^{-1} F|^{\frac{1}{2}}(\hat{\sigma}^2)^{(\frac{n-k}{2})}} \end{aligned} \tag{4}$$

Equation (4) is the posterior distribution of θ_i given by (4.6).