INTEGRATION IN COMPUTER EXPERIMENTS AND BAYESIAN ANALYSIS

By

Stella Wanjugu Karuri

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics
Waterloo, Ontario, Canada, 2005

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Mathematical models are commonly used in science and industry to simulate complex physical processes. These models are implemented by computer codes which are often complex. For this reason, the codes are also expensive in terms of computation time, and this limits the number of simulations in an experiment. The codes are also deterministic, which means that output from a code has no measurement error.

One modelling approach in dealing with deterministic output from computer experiments is to assume that the output is composed of a drift component and systematic errors, which are stationary Gaussian stochastic processes. A Bayesian approach is desirable as it takes into account all sources of model uncertainty. Apart from prior specification, one of the main challenges in a complete Bayesian model is integration. We take a Bayesian approach with a Jeffreys prior on the model parameters. To integrate over the posterior, we use two approximation techniques on the log scaled posterior of the correlation parameters. First we approximate the Jeffreys on the untransformed parameters, this enables us to specify a uniform prior on the transformed parameters. This makes Markov Chain Monte Carlo (MCMC) simulations run faster. For the second approach, we approximate the posterior with a Normal density.

A large part of the thesis is focused on the problem of integration. Integration is often a goal in computer experiments and as previously mentioned, necessary for inference in Bayesian analysis. Sampling strategies

are more challenging in computer experiments particularly when dealing with computationally expensive functions. We focus on the problem of integration by using a sampling approach which we refer to as "GaSP integration". This approach assumes that the integrand over some domain is a Gaussian random variable. It follows that the integral itself is a Gaussian random variable and the Best Linear Unbiased Predictor (BLUP) can be used as an estimator of the integral. We show that the integration estimates from GaSP integration have lower absolute errors. We also develop the Adaptive Sub-region Sampling Integration Algorithm (ASSIA) to improve GaSP integration estimates. The algorithm recursively partitions the integration domain into sub-regions in which GaSP integration can be applied more effectively. As a result of the adaptive partitioning of the integration domain, the adaptive algorithm varies sampling to suit the variation of the integrand. This "strategic sampling" can be used to explore the structure of functions in computer experiments.

Acknowledgements

I thank Prof. Will Welch and Prof. Don McLeish for their supervision. I also thank my thesis committee, Prof. Hugh Chipman and Prof. Ken Sen Tan for their reading and comments on the thesis.

I am grateful to Dr. Shoo Lee of the Centre for Healthcare Innovation and Improvement, Women and Children's Hospital, Vancouver BC, for allowing me to use the centre's resources while completing this thesis.

Finally I would like to acknowledge the support received from my family; this thesis would not have been completed without them.

To Meshack Mahungu Kariuki

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Review

The work in this thesis covers the topic of computer experiments, a statistical application widely used in science and engineering. Computer experiments can be considered as equivalent to physical experiments, but performed on the computer implementations of mathematical functions which represent physical processes. For example, pharmokinetic models are mathematical functions which enable the prediction, distribution, metabolism and excretion of chemicals in the body. These mathematical functions are in turn implemented by computer codes. Inference of the physical process is made by running the codes at specified levels of input, this constitutes a computer experiment. The results from such an experiment would help in optimizing the required drug dosage for treating a patient [20]. Another example is in finite element analysis in which structural properties of a material are broken down into many small blocks then described with sets of mathematical equations. These equations collectively represent the structure of the material and are solved by computer codes [4]. Such functions or simulations have the advantage of being cheaper to implement and control than the physical processes they represent, for instance in finite element analysis, information from running the computer experiment can be used to reduce

the number of prototypes needed in subsequent physical experiments. Ethical issues might also be another motivation for computer experiments, for example pharmokinetic models reduce the risk of exposing patients to potentially harmful dosages in clinical trials.

1.1 Developing a Computer Experiment

There are roughly five stages that go into developing a computer experiment [30]. These are as follows:

1. Formulation of the problem and identification of inputs to a simulation
2. Function implementation with computer codes
3. Derivation of inputs levels or the design and its application to the codes to obtain output
4. Validation of the model with physical data
5. Application of the results from the code to meet engineering goals

The above steps might make up a cyclic process, depending on initial experiment goals. For example, to implement a pharmokinetic model one might go through stages (1) to (4), then narrow down the number of inputs after identifying the important effects to the experiment, obtain a cheaper approximation to the function and repeat the process to optimize for a smaller range of inputs values in step (2).

1.2 Goals in Computer Experiments

Computer experiments often involve finding numerical solutions to large systems of differential equations. For instance operations with pharmokinetic models often involve optimizing over large sets of differential equations with numerous known and unknown time dependent variables which are related to different physiological measurements. Physical processes are complex and as a result the codes that represent them are also complex. The complexity of computer codes makes them costly in terms of run times and this results in small output data sets. Unlike physical experiments where physical processes are random, computer codes are deterministic – the same input to a code results in the same output. This presents a challenge in statistical modelling as randomness is a requirement for probability and inference. In some cases computer codes are black-boxes, the codes can be used by passing set input values to obtain outputs, but their internal structure or means of operation is unknown.

Suppose we have evaluated a function or code of d variables at n input points obtained from the design space. The design space $\mathcal{X}$ is bounded by the upper and lower limits of each of the input variables. Let input point i with $i = 1, \ldots, n$ be denoted as $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})$, the set of points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$, are referred to as the design. The design can be chosen to be optimal such that it satisfies certain criteria, for some commonly used criteria and designs refer to Koehler and Owen [21]. Each input has associated function value $y_i = y(\mathbf{x}^{(i)})$ which may be multivariate. Some common goals in a computer experiments are:

- Prediction – Efficient prediction is central to achieving any goal in computer experiments. This involves estimating $\hat{y}(\mathbf{x}^*)$ at an untried input combination $\mathbf{x}^*$. Often a surrogate for the codes is constructed to enable prediction, for

example a linear, polynomial or kriging model [35]. An example application in the literature is by Sacks, Welch, Mitchell and Wynn [31] who use a kriging model on output from an electric simulator to obtain predictions at a set of 100 points.

- Optimization – Optimization involves searching all allowable input values for a combination that results in a maximum or minimum y; for example, finding the minimum temperature which melts an alloy for a range of electric voltages. An example application is by Jones, Schonlau and Welch [18] who use stochastic processes in response surface modelling.

- Visualization – Computer codes are sometimes black-boxes, visualization helps in understanding the underlying function by helping find discontinuities, singular values or turning points. Visualization of the function is often a preliminary step to achieving other goals in computer experiments.

- Calibration – In some cases inputs to a code might not be known. Calibration involves matching up or tuning the computer code to fit observed data from the physical process. This enables the experimenter to relate the unknown inputs to the physical process.

- Integration – The average of the output for a particular input variable might be of interest, particularly if the input is assumed to be random with some distribution [23]. Schonlau and Welch [34] show that when integrating out effects, the integral itself is a Gaussian process and the Kriging model can be used to estimate the integral. We use this technique to as a tool for integration; more information will be given in the review section and subsequent chapters.

There are two main approaches to statistical analysis of deterministic output from computer experiments. The first approach assumes that the input points $\mathbf{x}$ are random variables with some distribution which is propagated to the output values. This approach was used by McKay, Conover and Beckman [23] in what is regarded as the first application of experimental design in computer simulations.

The second approach assumes that the output points are realizations of a Gaussian stochastic process. Some of the earliest work on the Gaussian stochastic process approach are by Sacks, Schiller and Welch [29] who first applied the kriging approach to output from computer experiments in order to address the issue of computation of efficient designs. They used the best linear predictor to formulate the integrated mean square error of prediction (IMSE) as a criterion to obtaining efficient designs. They applied this criterion to chemical kinetics problems and showed that the stochastic process approach compared to least squares estimation and factorial design reduced the actual square error of prediction. Sacks, Welch, Mitchell and Wynn [31] provided a review of computer experiments and also examined the issue of design by evaluating different criteria for choosing optimal designs. To best describe the kriging approach, we outline some notation and theory based on the paper by Jones et al [18].

1.3 Methodology

Suppose we have evaluated a deterministic function of d variables at n sample points. Let sample point i with $i = 1, \ldots, n$ be denoted as $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)})$ with associated function value $y_i = y(\mathbf{x}^{(i)})$. The observations are assumed to be realizations of

$$Y(\mathbf{x}^{(i)}) = \sum_{h=1}^{d} \beta_h f_h(\mathbf{x}^{(i)}) + \epsilon(\mathbf{x}^{(i)}); \quad (i = 1, \ldots, n). \tag{1.1}$$

The error terms are assumed to have a Normal distribution,

$$\epsilon(\mathbf{x}^{(i)}) \sim N(0, \sigma^2).$$

The model in (1.1) has two components, the first component consists of a response surface which models the drift in the response. The second component models the systematic lack of fit, and is treated as the realization of a stationary Gaussian stochastic process. The assumption of stationarity implies that

$$E(\epsilon(\mathbf{x}^{(i)})) = 0.$$

The covariance between two input points is given as

$$\text{cov}(\epsilon(\mathbf{x}^{(i)}), \epsilon(\mathbf{x}^{(j)})) = \sigma^2 \text{corr}(\epsilon(\mathbf{x}^{(i)}), \epsilon(\mathbf{x}^{(j)})),$$

where the correlation can be specified by various positive definite functions, a discussion of this is presented by Koehler and Owen [21].

A commonly used correlation function is the Gaussian correlation function where

$$\text{corr}(\epsilon(\mathbf{x}^{(i)}), \epsilon(\mathbf{x}^{(j)})) = \exp\left[-d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})\right]. \tag{1.2}$$

The function $d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is the distance function, specified as

$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sum_{h=1}^{k} \theta_h |x_h^{(i)} - x_h^{(j)}|^2 \quad (\theta_h \geq 0).$$

The above representation ensures that the errors are stationary, as the correlation depends on the magnitude of the distance between any pair of sites. The parameter θ_h measures the activity or 'importance' of the variable x_h. A variable h is active if for small values of $|x_h^{(i)} - x_h^{(j)}|$, y_i and y_j are not necessarily similar and they have a low correlation; consequently large values of θ_h will magnify small values of $|x_h^{(i)} - x_h^{(j)}|$

resulting in such low correlations. The properties of the correlation parameters in the Gaussian correlation function are well illustrated by Jones et al. [18]. The use of the Gaussian correlation function assumes that the code has a high degree of smoothness.

We can thus modify (1.1) and assume that the variation in the realizations is taken up entirely by the systematic error,

$$Y(\mathbf{x}^{(i)}) = \mu + \epsilon(\mathbf{x}^{(i)}). \tag{1.3}$$

The assumption of a constant trend is convenient as it reduces the number of parameters in the model that need to be estimated. Furthermore, an example by Sacks, Schiller and Welch [29] compared the constants, linear first order and quadratic trend and found that the three models gave similar results in prediction. Chen [3] also arrived at the same conclusion, he studied different linear specifications against a constant trend using simulations.

The expression in (1.3) implies that

$$\mathbf{Y} = (Y(\mathbf{x}^{(1)}), \ldots, Y(\mathbf{x}^{(n)})) \sim N(\mathbf{1}\mu, \sigma^2\mathbf{R}),$$

where $\mathbf{1} = (1, \ldots, 1)^T$ is a vector of length n, and $\mathbf{R}$ denotes the $n \times n$ design correlation matrix whose $(i, j)th$ entry is $\text{corr}(\epsilon(\mathbf{x}^{(i)}), \epsilon(\mathbf{x}^{(j)}))$. This model has $k + 2$ parameters which are often unknown and need to be estimated: μ, σ^2 and the vector of correlation parameters $\mathbf{\Theta} = (\theta_1, \ldots, \theta_k)^T$.

1.3.1 The Kriging Approach to the Gaussian Stochastic Process Model

Suppose we wish to obtain a prediction $\hat{y}$ at input $\mathbf{x}^*$. We let the correlation between $\epsilon(\mathbf{x}^*)$ and the n design points be denoted as

$$\mathbf{r} = \left(\text{corr}(\epsilon(\mathbf{x}^*), \epsilon(\mathbf{x}^{(1)})), \ldots, \text{corr}(\epsilon(\mathbf{x}^*), \epsilon(\mathbf{x}^{(n)}))\right)^T. \tag{1.4}$$

One approach to prediction is to use linear prediction based on (1.3), which is also known as Kriging in geostatistics. The Best Linear Unbiased Prediction (BLUP) at a new site ($\mathbf{x}^*$) is chosen to be linear in $\mathbf{y}$,

$$\hat{y}(\mathbf{x}^*) = c^T(\mathbf{x}^*)\mathbf{y},$$

with the vector c chosen to minimize the mean square error (MSE) of $\hat{Y}$ where

$$\text{MSE}(Y(\mathbf{x}^*)) = E(c^T\mathbf{Y} - Y(\mathbf{x}^*))^2,$$

subject to the unbiasedness condition

$$E(c^T\mathbf{Y}) = E(Y(\mathbf{x}^*)).$$

It can be shown [29] that the BLUP – assuming the correlation structure is known, is

$$\hat{y} = \hat{\mu} + \mathbf{r}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}), \tag{1.5}$$

with variance equal to

$$\text{var}(\hat{y}) = \hat{\sigma}^2\left[1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}\right], \tag{1.6}$$

where

$$\hat{\mu} = \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}, \tag{1.7}$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{n}. \tag{1.8}$$

From the representation in (1.5), the BLUP is unbiased and linear in the observed output. The BLUP thus gives the prediction at a site $\mathbf{x}^*$ as the generalized least square mean $\hat{\mu}$, adjusted by the correlation of the error of the new site to the errors of the sampled sites. Its variance is the generalized residual sum of squares adjusted by

two terms: the first, the correlation between the new site and the sampled sites, and the second, the fact that μ is estimated by the generalized least squares mean. The estimated response from the BLUP interpolates the observations. This can be seen by obtaining the prediction at points in the design; the predictions are the observed values and their corresponding variance or mean square error equals to zero.

1.3.2 The Bayesian Approach to the Gaussian Stochastic Process Model

The BLUP method assumes the correlation parameters are known. In reality these have to be estimated, often by point estimation methods such as maximum likelihood estimation. Point estimation often results in under estimation of prediction errors and under coverage of the true value in interval estimations. A Bayesian approach on the other hand, can incorporate some model uncertainty in estimation.

The likelihood from the model in (1.3) is

$$L(\mathbf{y}|\mu,\sigma^2,\boldsymbol{\Theta}) \propto \frac{1}{(\sigma^2)^{\frac{n}{2}}\sqrt{|\mathbf{R}|}} \exp(\frac{-1}{2\sigma^2}(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)). \tag{1.9}$$

From Bayesian methodology inference is based on the posterior density of the model parameters which is derived from

$$\text{pr}(\mu,\sigma^2,\boldsymbol{\Theta}|\mathbf{y}) \propto L(\mathbf{y}|\mu,\sigma^2,\boldsymbol{\Theta})\text{pr}(\mu,\sigma^2,\boldsymbol{\Theta}), \tag{1.10}$$

where $\text{pr}(\mu,\sigma^2,\boldsymbol{\Theta})$ is the prior density on the model parameters. As a notational convention we use $\text{pr}(\cdot)$ to denote the probability density with respect to variables $(\cdot)$. To demonstrate the difference a Bayesian approach makes, we examine the 'best' predictor which minimizes the quadratic loss function

$$E(y(\mathbf{x}^*)|\mathbf{y}) = \int_{\boldsymbol{\Theta}} E(y(\mathbf{x}^*)|\mu,\sigma^2,\boldsymbol{\Theta},\mathbf{y})\text{pr}(\boldsymbol{\Theta}\ |\mu,\sigma^2,\mathbf{y})\ d\mu d\sigma^2 d\boldsymbol{\Theta}. \tag{1.11}$$

The variance of this estimate is given as

$$\text{var}(y(\mathbf{x}^*)|\mathbf{y}) = E_{\mu,\sigma^2,\boldsymbol{\Theta}|\mathbf{y}}\text{var}(y(\mathbf{x}^*)|\mu,\sigma^2,\boldsymbol{\Theta},\mathbf{y}) + \text{var}_{\mu,\sigma^2,\boldsymbol{\Theta}|\mathbf{y}}E(y(\mathbf{x}^*)|\mu,\sigma^2,\boldsymbol{\Theta},\mathbf{y}) \tag{1.12}$$

The notation $E_{\mu,\sigma^2,\boldsymbol{\Theta}|\mathbf{y}}(.)$ and $\text{var}_{\mu,\sigma^2,\boldsymbol{\Theta}|\mathbf{y}}(.)$ stand for the expectation and variance with respect to the posterior density. From (1.12) the variance of the Bayesian predictor is the sum of the posterior mean of the BLUP variance and the posterior variance of the BLUP. This illustrates the advantage of a Bayesian approach, since the model parameters have some assumed distribution, then depending on prior chosen, parameter uncertainty is accounted for and error estimates can be more conservative.

1.4 A Review of Bayesian Analysis in Computer Experiments

Currin et al [5] used the following Bayesian formulation in (1.13) in prediction and design selection. Assuming the model parameters are known, and that $Y(\mathbf{x}^*) \sim N(\mu,\sigma^2)$ then it follows that

$$\begin{aligned} Y(\mathbf{x}^*|\mathbf{y},\mu,\sigma^2) &\sim N(\mu_{\mathbf{y}},\sigma^2_{\mathbf{y}}), && (1.13)\\ \text{where } \mu_{\mathbf{y}} &= \mu + \mathbf{r}^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu),\\ \sigma^2_{\mathbf{y}} &= \sigma^2(1-\mathbf{r}^T\mathbf{R}^{-1}\mathbf{r}). && (1.14)\end{aligned}$$

They assumed that the mean, variance and correlation parameters are known, but these were estimated by maximum likelihood estimation. The maximum likelihood estimates of the mean and variance parameter are equal to the generalized least square mean and generalized residual sum of squares given by (1.7) and (1.8) respectively. Their approach basically breaks down to analyzing the BLUP, the only difference is

that their BLUP variance in (1.14) does not take into account the estimation of μ.

Handcock and Stein [15] presented a framework for full Bayesian analysis of the Gaussian stochastic process model. They recommended that inference be based on the Bayesian predictive distribution as this incorporates model uncertainty, especially in cases where there is little information about the model in the available data. They formulated the true predictive density by assuming a non-informative prior of the form

$$\mathrm{pr}(\mu, \sigma^2, \boldsymbol{\Theta}) \propto 1/\sigma^2.$$

On the other hand, assuming the correlation and variance parameters are known and using a diffuse prior on μ, the resulting predictive density is Normal with mean parameter equal to the BLUP and variance parameter equal to the BLUP variance. Handcock and Stein [15] refer to this density as the plug-in predictive distribution. They show that the plug-in predictive distribution is significantly different from the true predictive density and that the difference is dependent on the specified correlation structure. However, it is unclear what method they use for numerical integration to obtain the predictive density.

In most Bayesian approaches in literature, the correlation parameters in the stochastic process model are often estimated by maximum likelihood estimation. Kennedy and O'Hagan's [20] approach to the Gaussian stochastic process model assumed that the process realization was the sum of the scaled computer output; which is dependent on both control and calibrated input, and an independent systematic error. Both of these components were assumed to be Gaussian processes, whose mean and variance functions were modelled hierarchically. Their aim was to base inference

of the model calibration parameters on the posterior density of the calibration parameters, physical data and output from the code. They stopped short of a full Bayesian implementation due to computation limitations and estimated the correlation and scale parameters with the posterior mode of the density based on output from the code. They applied their calibration method to model the deposition of radionuclides with calibration inputs as the source term and deposition velocity.

Reese, Hamada and Ryan [27], outline a Bayesian approach without the assumption of a Gaussian Stochastic process model. Their incorporation of expert opinion, physical outcome and computer code output results in the Recursive Bayesian Hierarchical Model. They assumed a multivariate Normal density on the output from three stages, in the first stage a linear model was assumed on the expert opinion data to formulate priors for coefficients and the variance parameters for the computer experiment data. Assuming a linear model on output from the computer experiment, the priors from the first stage were then used to formulate a posterior density on the computer output. Correlation in the computer output is induced through this hierarchical structure of the prior. In the final stage the posterior densities from the second stage were then used as priors to update the posterior density of the physical data. The method used by Reese et al [27] has the advantages of being computationally tractable and easily interpretable, though it would require large sample sizes due to the number of parameters in the model. The other disadvantage is that the linear form in the computer model is not always practical as one might not know what functional form to specify for the regression terms. Furthermore, their model is sensitive to prior specification and not as flexible as the Gaussian stochastic process model in terms of interpolation in prediction.

In summary, two obstacles to overcome when using a Bayesian approach are

- Specification of Priors – As correlation parameters are difficult to interpret independently, priors are often non-informative. Berger, Oliviera and Sanso [2] give comprehensive guidelines for choosing non-informative priors. They show that the likelihood is bounded away from zero, therefore it is important to have priors that ensure propriety of the posterior. They also show that the uniform prior used by Handcock and Stein [15], with some prior specification of the variance parameter will result in improper posteriors. The Jeffreys prior [17] however, results in a proper posterior. Berger, De Oliveira and Sanso also show that the Jeffreys prior for the correlation parameter is approximately proportional to the inverse function for small values of the parameters.

- Integration – Evaluating the integrals in (1.11) and (1.12) often requires numerical methods. The posterior density $\text{pr}(\boldsymbol{\Theta}|\mathbf{y})$ doesn't have a standard form and cannot be sampled directly, therefore some approaches to evaluation involve techniques such as Markov Chain Monte Carlo. A comprehensive survey of methods available for numerical integration of posterior densities was done by Evans and Swartz [8].

1.5 Thesis Outline

The work in this thesis is focused on implementing a full Bayesian approach to the Gaussian stochastic process model. For consistency in our analysis, we assume the Gaussian correlation function. We also assume that output at an input is univariate though the theory can be extended to multivariate output.

In Chapter 2 we specify prior information on the parameters using the Jeffreys prior. We then use two approximation methods on the posterior. Following from results by Berger, De Oliveira and Sanso [2], we obtain an approximation to the Jeffreys prior for the Gaussian correlation function which we verify for a single variable function, with an equispaced design using Chen's [3] representation for $\mathbf{R}^{-1}$. We then formulate a similar approximation for $d = 2$. We adopt this approximation and use a log transformation on the correlation parameters in the posterior, which in effect specifies a uniform prior on the reparametrized likelihood. MCMC simulations are needed for integration, the advantage of this approximation is that the resulting posterior is less complex and MCMC simulations take a shorter time to obtain. In the second method we approximate the reparametrized posterior with a Normal density. The motivation for this method is the more ellipsoidal form of the log-transformed posterior. This second approach is a far cheaper Bayesian implementation as it is possible to use plain Monte Carlo by drawing samples from the Normal density without resorting to MCMC simulations.

Integration is also a daunting exercise in computer experiments, for example when the input is assumed to be random as previously mentioned in Section 1.2; it is made more complicated when dealing with expensive black-box functions. In Chapter 3 we examine the general integration problem,

$$\bar{g} = \int_{\mathcal{X}} g(\mathbf{x}) d\mathbf{x},$$

where $\mathbf{x} \in \mathcal{X}$. Straightforward Monte Carlo (MC) integration can easily be implemented in low and high dimensions provided that the distribution of $\mathbf{x}$ or its approximation can be sampled. Alternatively one can forgo the idea of random sequences and employ sequences of points that emulate randomness which are specifically tailored for

integration. Quasi Monte Carlo methods have widely been used in number theory and numerical analysis; Fang, Wang and Bentler [9] give a comprehensive review of their use in statistical applications in particular in the areas of generation of sequences for multivariate distributions and the evaluation of expectations of functions. Sequences such as the Halton [14] and Sobol [36] sequences, result in comparatively smaller bounds for the integration error [10] hence have faster convergence rates compared to the MC method. Robinson and Atcitty [28] compared four modifications of Halton sequences and Latin Hypercube Sampling [23] as designs in computer experiments, and found that quasi-random sequences provided estimates with lower integration errors compared to Latin Hypercube sampling in low dimension parameter space.

In Chapter 3 we examine a second approach to integration in which the integrand on a rectangular domain, is assumed to be a Gaussian process of the form (1.3). We call this approach to integration *GaSP integration*. The assumption of randomness follows from the fact that the numerical value of the integrand at a point $\mathbf{x}$, is unknown until $g(\mathbf{x})$ is actually calculated. This method was first applied by O'Hagan [25] who used Bayesian analysis of the quadrature problem. An earlier description and review of Bayesian analysis in numerical analysis is given by Diaconis [7]. Using a non-informative prior on the model parameters, O'Hagan formulated the posterior distribution of $\bar{g}$ based on the evaluation of the integrand at a set of points. The general technique of Bayesian quadrature is to make the fullest possible use of function evaluations, hence it is an ideal method for the numerical integration of costly functions. O'Hagan [25] also showed that the product correlation structure reduced multidimensional integrals into one dimension integrals which are easily approximated. The theory set out by Schonlau and Welch [34] is slightly different; if

the integrand is assumed to be a realization of a Gaussian stochastic process, it then follows that $\bar{g}$ is also a realization of a Gaussian stochastic process and the BLUP can be used to estimate the integral. The BLUP estimate that they formulate is essentially the posterior mean of $\bar{g}$ as derived by O'Hagan. More theory on this technique is illustrated in Chapter 3. Schonlau and Welch [34] used this approach to estimate effects in computer experiments. We present example multidimensional integration using GaSP integration in Chapter 3.

Due to the assumption of stationarity used in (1.2), GaSP integration will not perform well when this assumption is violated for example when dealing with a function with asymptotes. Consider the function

$$f(x_1, x_2) = 1/(1 - x_1 x_2),$$

on the unit square $\mathcal{X} = [0, 1] \times [0, 1]$. A contour plot of the function is given in Figure 1.1. The function has a vertical asymptote at (1,1). Its structure changes with location, consequently its behavior on $[0.5, 1] \times [0, 1]$ is quite different compared to its behavior on $[0.5, 1] \times [0, 1]$. In this case, the assumption of stationarity does not hold. To gain more information on the function, we would like to sample more from regions where the function changes rapidly, however the size of n is limited due to the computation power needed to invert the correlation matrix. In Chapter 4 we introduce the Adaptive Subregion Sampling Integration Algorithm (ASSIA) which partitions the integration region into more homogenous subregions and concentrates sampling where the integration region is most varied. Adaptive integration methods are common in numeric integration [6], they work by dynamically partitioning the integration region so that the integrand is more or less homogenous in respective sub-regions. A local integration rule such as a polynomial integration rule, is applied

to the subregions to obtain an estimate of the integral. Genz [12] employs adaptive integration to deal with functions having dominant peaks after employing split-t transformations [13], which transform the integration region by redistributing the mass about the peak so that it occupies a larger fraction of the integration space. In Chapter 4, we present the workings of ASSIA with both Monte Carlo integration and GaSP integration for two dimensional integration problem.

In Chapter 5 we apply the algorithm to higher dimension problems and make some changes to make computation easier when working in higher dimension, as well as to reduce the number of iterations. We use ASSIA in a five and ten dimension integration problem. ASSIA can also be used as a sampling or design tool, and we present applications in which ASSIA is used to obtain "strategic samples". These samples can be used to further other goals in computer experiments such as visualization and optimization. In the last chapter we present recommendations on improvements to ASSIA and some directions for future work.

1.6 Examples

The main examples deal with both aspects of Bayesian analysis in computer experiments and integration. Computer experiment data was created through simulations using the Gaussian Stochastic Process model in (1.3).

1.6.1 Single Input Simulations

For one sample obtained by Latin Hypercube sampling [23] with $n = 5$ we obtain simulated data sets using $\mu = 0, \sigma^2 = 1$ and a range of values for $\theta = \theta_1$, $(\theta = 0.05, 0.1, 0.5, 1, 5, 10)$. For each θ, we obtain $\mathbf{R}$ and use a $N(\mathbf{0}, \sigma^2\mathbf{R})$ density to

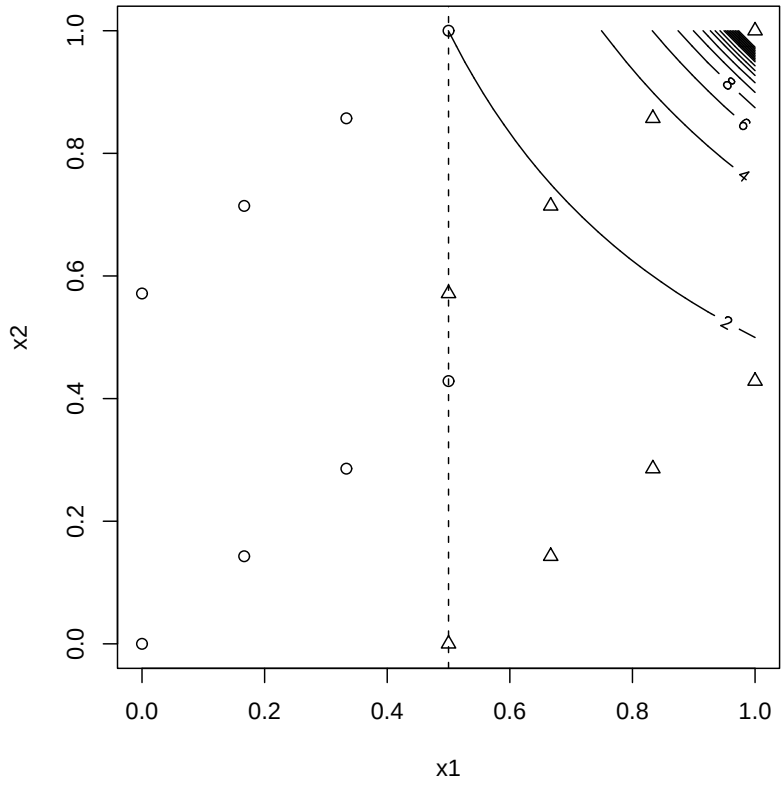


Figure 1.1: Plot of $f(x_1, x_2) = 1/(1 - x_1x_2)$, '$\circ$' for design on $[0, 0.5] \times [0, 1]$, '$\triangle$' for design on $[0.5, 1] \times [0, 1]$.

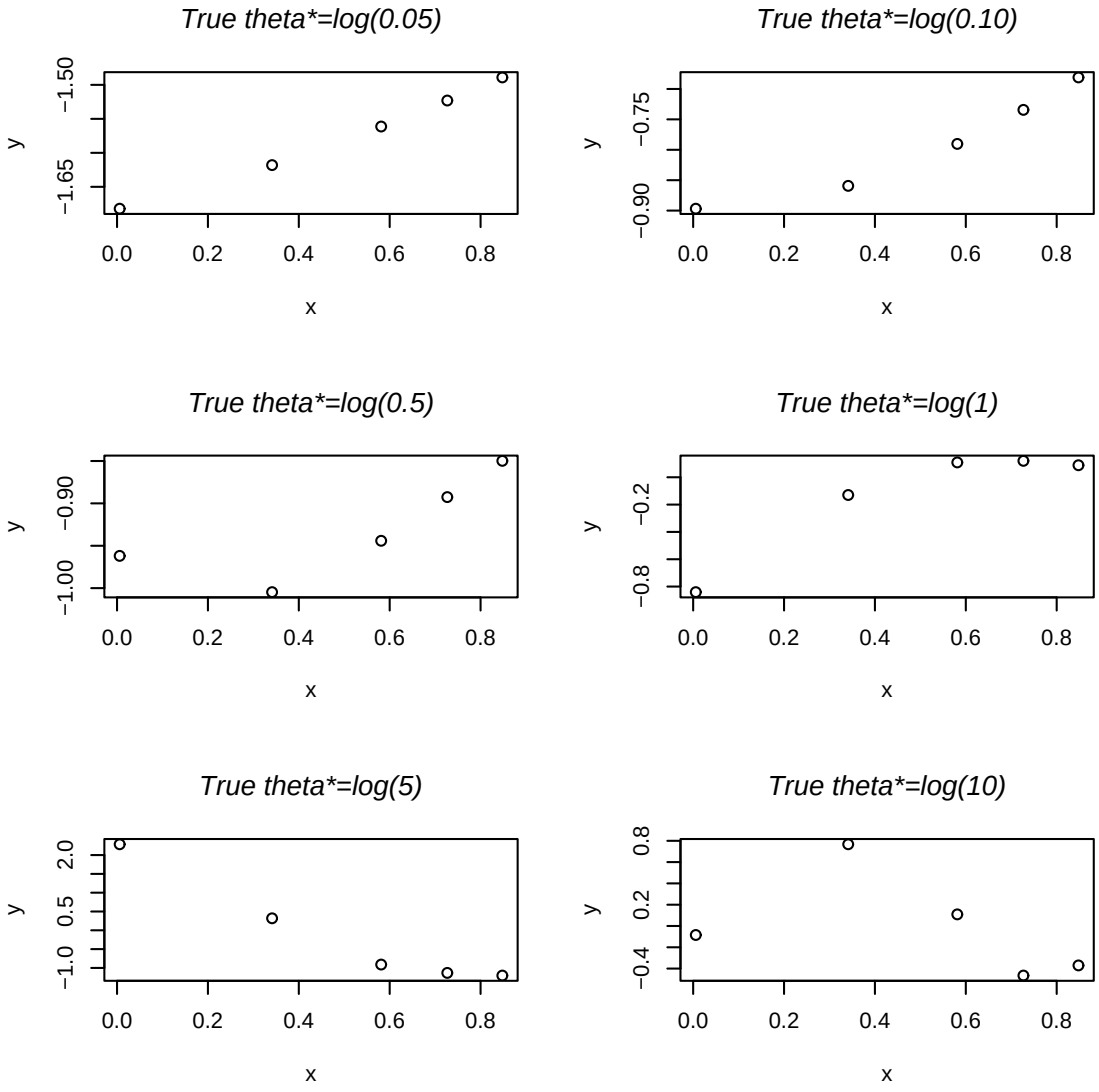


Figure 1.2: Plots of simulated data for $d = 1$

generate realizations. This results in 6 data sets: **1D1**, **1D2**, **1D3**, **1D4**, **1D5** and **1D6**. Plots of the simulated data are given in Figure 1.2. The plots indicate a linear trend for strong correlation in the errors.

x	$y(x)$					
	$\theta = 0.05$	$\theta = 0.1$	$\theta = 0.5$	$\theta = 1$	$\theta = 5$	$\theta = 10$
	1D1	**1D2**	**1D3**	**1D4**	**1D5**	**1D6**
0.00572842	-1.682227	-0.8969145	-0.9619334	-0.84085548	2.2872472	-0.08458607
0.34118175	-1.618032	-0.8593026	-1.0047370	-0.12989661	0.3194758	0.76745245
0.58125444	-1.561303	-0.7903443	-0.9441377	0.10856554	-0.9106261	0.10904981
0.72673939	-1.523015	-0.7343532	-0.8925794	0.12005513	-1.1324536	-0.46569314
0.84790232	-1.489051	-0.6810062	-0.8498170	0.08790122	-1.2034158	-0.37025205

Table 1.1: One dimension Simulation Data

1.6.2 Two Input Simulations

In a similar manner as the one input simulation approach, we simulate data sets for the two input problem with $(\mu = 0, \sigma^2 = 1)$, and $(\theta_1, \theta_2) = (\exp(-1, -1), \exp(0, 0), \exp(3, 2))$. The sites are an unmodified Halton Sequence of 21 points. This resulted in three data sets: **2D1**, **2D2** and **2D3**. Plots of the sites versus the realizations are given by the contour plots in Figures 1.3 – 1.5. The plot indicates that the values of $\theta_1, \theta_2)$ used in the simulations affect the characteristics of the resulting function or output.

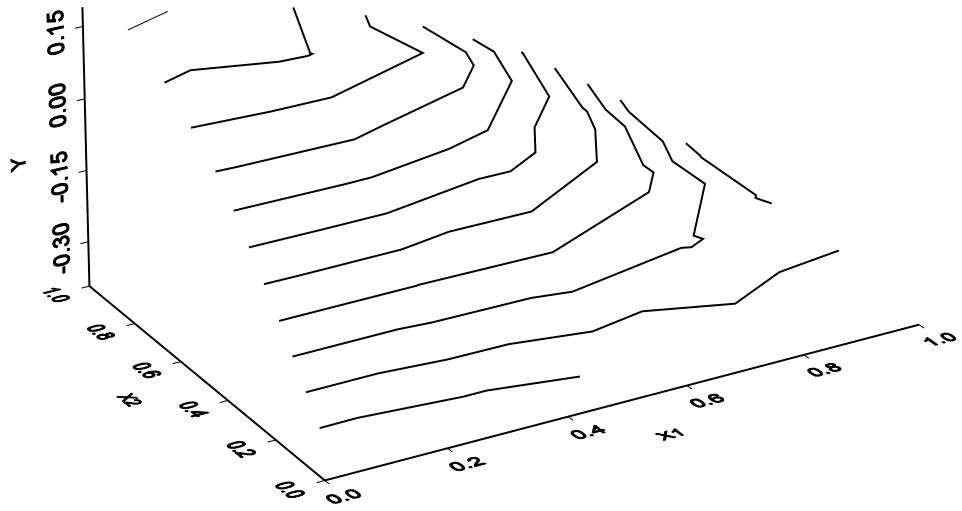


Figure 1.3: Plot of realizations versus input value for **2D1** $(\theta_1^*, \theta_2^*) = (-1, -1)$

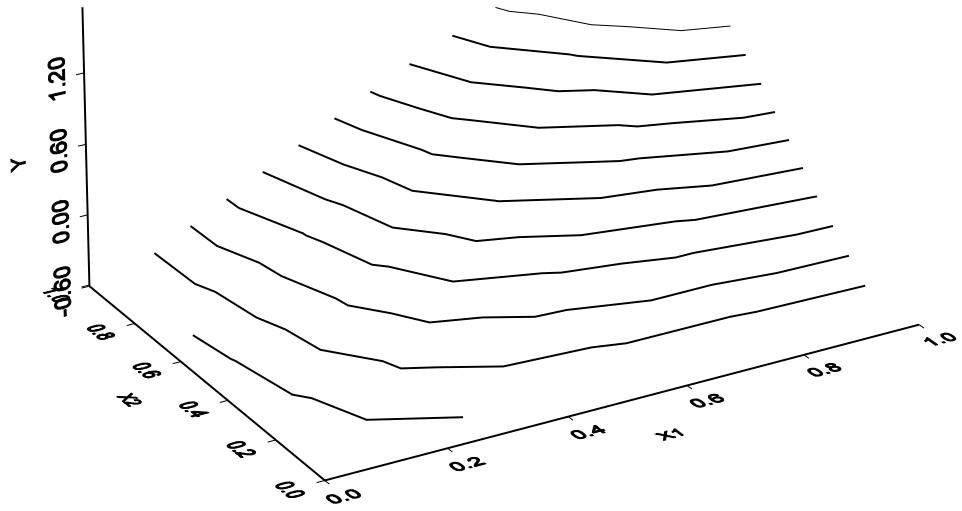


Figure 1.4: Plot of realizations versus input value for **2D2** $(\theta_1^*, \theta_2^*) = (0, 0)$

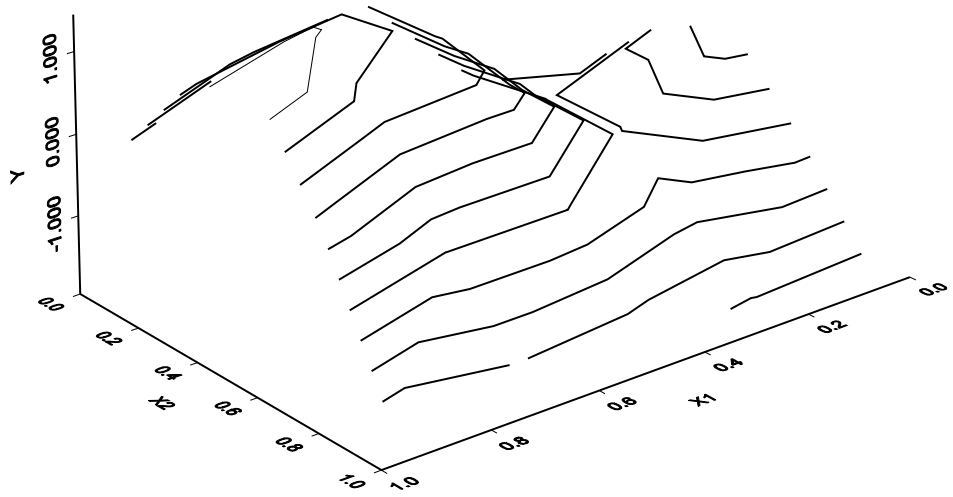


Figure 1.5: Plot of realizations versus input value for **2D3** $(\theta_1^*, \theta_2^*) = (3, 2)$

$\mathbf{x}$		$y(\mathbf{x})$		
		$(\theta_1, \theta_2) = \exp(-1, -1)$ **2D1**	$(\theta_1, \theta_2) = \exp(0, 0)$ **2D2**	$(\theta_1, \theta_2) = \exp(3, 2)$ **2D3**
0.5000	0.3333	-0.1444	0.0717	0.9679
0.2500	0.6667	0.1146	-0.0034	-0.9590
0.7500	0.1111	-0.2696	-0.2592	0.3385
0.1250	0.4444	-0.0252	-0.3087	0.0130
0.6250	0.7778	-0.0358	0.8791	-1.3150
0.3750	0.2222	-0.2075	-0.1820	-0.3119
0.8750	0.5556	-0.2497	0.8835	1.0333
0.0625	0.8889	0.1759	-0.3740	-1.8113
0.5625	0.0370	-0.3199	-0.4027	0.9032
0.3125	0.3704	-0.0865	-0.0779	-0.7602
0.8125	0.7037	-0.1927	1.0896	-0.4006
0.1875	0.1481	-0.2846	-0.3938	-0.6445
0.6875	0.4815	-0.1467	0.5223	0.6326
0.4375	0.8148	0.0892	0.5140	-1.0974
0.9375	0.2593	-0.2761	0.1585	0.9029
0.0313	0.5926	0.0689	-0.4652	-0.2172
0.5313	0.9259	0.0340	0.8038	-1.5988
0.2813	0.0741	-0.3369	-0.4015	-0.7439
0.7813	0.4074	-0.2121	0.4548	0.9720
0.1563	0.7407	0.1541	-0.1924	-1.4574
0.6563	0.1852	-0.2455	-0.1177	1.1014

Table 1.2: Two dimension Simulation Data

Chapter 2

Approximation to the Posterior

2.1 Introduction

In this chapter we present a full Bayesian approach to the Gaussian stochastic Process model. The novelty of our approach is that we incorporate two approximation techniques; the first based on approximating the Jeffreys prior, the second based on approximating the posterior with a Normal density.

Prior information on model parameters in the stochastic process model is often unavailable thus it is specified by non-informative priors. Berger, De Oliveira and Sanso [2] presented a study of non-informative priors that result in proper posterior densities, among these were the Reference and the Jeffreys prior. In this chapter we specify prior information on the parameters using the Jeffreys prior. Apart from producing a proper posterior density, the Jeffreys prior appeal is that it is easy to formulate, and as was shown by Berger, De Oliveira and Sanso, can be approximated by a simpler function. As an exercise, we verify this approximation for a single variable function with an equispaced design using Chen's [3] representation for $\mathbf{R}^{-1}$. This representation for $\mathbf{R}^{-1}$ enables the simplification of the prior using Maple software for

$n = 1, \ldots, 12$. Using the product correlation rule, we formulate a similar approximation for $d = 2$ in the equispaced design. We adopt this approximation and use a log transformation on the correlation parameters in the posterior, which in effect specifies a uniform prior on the reparametrized parameters. The resulting posterior is then the integrated likelihood. This leads to the first approximation method of the posterior. MCMC simulation is needed to obtain samples and for integration. The advantage of this approximation is that the resulting posterior is less complex, hence MCMC simulation take a shorter time. We use simulated data sets to compare moments of the approximation to the true posterior. In the second method we approximate the reparametrized posterior with a Normal density. The motivation of this method is the more ellipsoidal form of the log-transformed posterior. This second approach is a far cheaper Bayesian implementation as it is possible to use plain Monte Carlo by drawing samples from the Normal density without resorting to MCMC simulations. The two approximation techniques are used to obtain predictions of a two dimensional function, and their performance is compared to that of the BLUP.

2.2 Developing the Jeffreys Prior and the Posterior Density

The Jeffreys prior [17] for the stochastic process model in (1.3) is derived as (see Appendix A.1):

$$\begin{aligned} \text{pr}(\mu,\ \sigma^2, \boldsymbol{\Theta}) &\propto \sqrt{\mathbf{I}(\mu,\ \sigma^2, \boldsymbol{\Theta})}, \\ &\propto \text{pr}(\boldsymbol{\Theta})\text{pr}(\mu, \sigma^2), \end{aligned} \tag{2.1}$$

where

$$\begin{aligned} \text{pr}(\mu, \sigma^2) &\propto 1/(\sigma^2)^{3/2}, \\ \text{pr}(\boldsymbol{\Theta}) &\propto \sqrt{(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}|\mathbf{B}_d|)}. \end{aligned}$$

The matrix $\mathbf{B}_d$ is the $(d \times d)$ information matrix of the correlation parameters, formulated in Appendix A.1. The results from (2.1) specify a diffuse prior on μ and sets out the independence of the mean, variance and correlation parameters. Another suggested prior [15] takes the form in (2.1) with

$$\text{pr}(\mu, \sigma^2) = 1/\sigma^{2a}, \tag{2.2}$$

$$\text{pr}(\boldsymbol{\Theta}) \propto 1, \tag{2.3}$$

where a is an arbitrary positive number.

The likelihood from (1.9) is

$$L(\mathbf{y}|\mu, \sigma^2, \boldsymbol{\Theta}) \propto \frac{1}{(\sigma^2)^{\frac{n}{2}}\sqrt{|\mathbf{R}|}} \exp(\frac{-1}{2\sigma^2}(\mathbf{y} - \mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)).$$

In general, the posterior from using a prior of the form $\text{pr}(\mu, \sigma^2, \boldsymbol{\Theta}) \propto \text{pr}(\boldsymbol{\Theta})/\sigma^{2a}$ is

$$\text{pr}(\mu, \sigma^2, \boldsymbol{\Theta}|\mathbf{y}) \propto \frac{\text{pr}(\boldsymbol{\Theta})}{(\sigma^2)^{\frac{n+a}{2}}\sqrt{|\mathbf{R}|}} \exp(\frac{-1}{2\sigma^2}(\mathbf{y} - \mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)). \tag{2.4}$$

Using the fact that a random variable X with an inverse gamma distribution has a density function of the form

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-(\alpha+1)} \exp(-\beta/x)$$

with

$$\alpha, \beta > 0 \text{ and } x > 0,$$

and that

$$\int_0^\infty f(x)dx = 1,$$

we integrate out over σ^2 in (2.4) by multiplying appropriate normalizing constants. Matching up parameters we have

$$\alpha = (n + a - 3),$$

and $\beta = (\mathbf{y} - \mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)/2$, then

$$\text{pr}(\mu, \mathbf{\Theta}|\mathbf{y}) \propto \text{pr}(\mathbf{\Theta})[t(\mu, \mathbf{\Theta})]^{\frac{-(n+a-2)}{2}}, \tag{2.5}$$

where

$$\begin{aligned} t(\mu, \mathbf{\Theta}) &= \hat{\sigma}^2 \left(1 + \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}(\mu - \hat{\mu})^2}{\hat{\sigma}^2}\right), \\ \hat{\mu} &= \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}, \\ \hat{\sigma}^2 &= (\mathbf{y} - \mathbf{1}\hat{\mu})^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}). \end{aligned}$$

If a random variable X has a Student's t distribution, with α degrees of freedom, location parameter λ and scale parameter Σ^2, then its density function has the form:

$$f(x) = \frac{\Gamma[(\alpha + 1)/2]}{\Sigma(\alpha\pi)^{1/2}\Gamma(\alpha/2)} \left(1 + \frac{(x - \lambda)^2}{\alpha\Sigma^2}\right)^{-(\alpha+1)/2}$$

with

$$\alpha, \Sigma^2 > 0 \text{ and } -\infty < \lambda < \infty.$$

If $\mathbf{\Theta}$ were known in (2.5), μ would have a Student's t density with $\alpha = (n + a - 3)$, location parameter

$$\lambda = \hat{\mu},$$

and

$$\alpha\Sigma^2 = \hat{\sigma}^2/(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}).$$

To integrate over μ, we use the same technique as that used to integrate out over σ^2. The results are as follows:

$$\begin{aligned} \text{pr}(\boldsymbol{\Theta}|\mathbf{y}) &\propto \text{pr}(\boldsymbol{\Theta})L^I(\mathbf{y}|\boldsymbol{\Theta}), & (2.6)\\ L^I(\mathbf{y}|\boldsymbol{\Theta}) &= |\mathbf{R}|^{-1/2}(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1/2}(\hat{\sigma}^2)^{-\frac{(n-3)}{2}+a}. & (2.7)\end{aligned}$$

Berger, De Oliveira and Sanso [2] refer to $L^I(\mathbf{y}|\boldsymbol{\Theta})$ as the integrated likelihood. As the integrated likelihood is positive and bounded, they showed that the prior given by (2.2) and (2.3), results in an improper posterior for certain values of a, the Jeffreys prior however results in a proper posterior. They also formulate an approximation for the Jeffreys prior for the case $d = 1$, we verify this approximation in the next section.

2.3 Approximating the Jeffreys Prior

2.3.1 Single Input Function

The Jeffreys prior for a one input design is given by (A.4) in Appendix A.1 and can be written as

$$\text{pr}(\mu, \sigma^2, \theta) \propto \text{pr}_J(\theta)\text{pr}_J(\mu, \sigma^2), \quad (2.8)$$

where

$$\text{pr}_J(\mu, \sigma^2) \propto 1/(\sigma^2)^{3/2}$$

and

$$\text{pr}_J(\theta) \propto \sqrt{(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})|\mathbf{B}_1|}.$$

The notation above is in line with earlier notation, the subscript J denotes the Jeffreys prior on the correlation parameters.

Berger, De Oliveira and Sanso [2] obtained the following approximation with respect to the Jeffreys prior for the correlation parameter,

$$\mathrm{pr}_J(\theta) = O(1/\theta) \text{ as } \theta \to 0.$$

Their approximation is dependent on several assumptions; the two main assumptions are based on the approximation of $\mathbf{R}$ and the derivative matrix $\mathbf{R}_\theta$. These approximations are:

$$\begin{aligned} \mathbf{R} &= \mathbf{1}\mathbf{1}^T + \theta(D + o(1)) \text{ as } \theta \to 0, \\ \mathbf{R}_\theta &= D + o(1) \text{ as } \theta \to 0. \end{aligned}$$

The matrix D is dependent on the distance between pairs of sites and is also assumed to be non-singular. Non-singularity of D is required in the approximation of $\mathbf{R}^{-1}$.

We verify the approximation of Berger, De Oliveira and Sanso with the results of Chen [3]. With one input ($d = 1$) and an equispaced design $(x^{(1)}, x^{(2)}, \ldots, x^{(n)}) = (1/n, 2/n \ldots, 1)$, the correlation between two sites or input points $(x^{(i)}, x^{(j)})$, is given as $\mathbf{R}_{i,j} = \rho^{(i-j)^2}$ with $\rho = \exp(-\theta/n^2)$. The resulting correlation matrix is Toeplitz. This special structure enabled Chen to formulate an expresssion for $\mathbf{R}^{-1}$ in terms of ρ.

Lemma 2.3.1. *The Jeffreys Prior for a one input equispaced design is:*

$$pr_J(\mu, \sigma^2) \propto 1/(\sigma^2)^{3/2}, \tag{2.9}$$

$$pr_J(\theta) \propto \left(\sum_{k=1}^{n} \frac{(\mathbf{1}^T \bar{\mathbf{w}}^k)^2}{1 - Q_{k-1}}\right)^{1/2} \left(\sum_{k=1}^{n}\sum_{j=1}^{n} \frac{nC_{jk}^2 - C_{jj}C_{kk}}{(1 - Q_{k-1})(1 - Q_{j-1})}\right)^{1/2}, \tag{2.10}$$

where

$$C_{jk} = (\bar{\mathbf{w}}^j)^T \mathbf{R}_\theta \bar{\mathbf{w}}^k,$$

and $\bar{\mathbf{w}}^k = (w_1^k, \ldots, w_n^k)$ *with*

$$w_i^k = \begin{cases} u_i^{(k-1)} & i < k, \\ -1 & i = k, \\ 0 & i > k, \end{cases} \tag{2.11}$$

$$u_i^{(k)} = -(-\rho)^i \prod_{j=1}^{i} \left(\frac{\sum_{r=0}^{k-j} \rho^{2r}}{\sum_{r=0}^{j-1} \rho^{2r}} \right), \tag{2.12}$$

and

$$1 - Q_k = (1-\rho^2)^k \prod_{j=1}^{k} \left(\sum_{r=0}^{j-1} \rho^{2r} \right). \tag{2.13}$$

A proof of this lemma is given in Appendix A.2.

Lemma 2.3.1 shows that the Jeffreys prior exists for all values of $\theta \neq 0$. We are then able to verify the approximation by Berger, De Oliveira and Sanso, for $n = 1, \ldots, 12$ using Maple software. The results are presented in Appendix A.3. For an illustration, when $n = 3$, we formulate the integrated likelihood from (2.7) as

$$L^I(\mathbf{y}|\theta) \propto \frac{\sqrt{(1-\rho^4)(\rho^2+2\rho+3)}}{(\rho^3+\rho^2+\rho)(y_2-y_1)(y_3-y_2)+3/2\sum_{i=1}^{3}(y_i-\bar{y})^2}.$$

As $\rho \to 0^+$ or as $\theta \to \infty$, the integrated likelihood is unaffected by θ and is dependent on the variation of the data. As $\rho \to 1$ or as $\theta \to 0$, the integrated likelihood grows dependent on θ. Using Maple software, the Jeffreys prior on θ in (2.1) is

$$\mathrm{pr}_J(\theta) \propto \frac{\rho(1-\rho)}{(1-\rho^4)(1-\rho^2)} \sqrt{(8\rho^6+11\rho^4+14\rho^2+3)(\rho^2+2\rho+3)}.$$

As $\theta \to 0^+$, $\rho = 1 - \kappa\theta + o(1)$,

$$\sqrt{(8\rho^6+11\rho^4+14\rho^2+3)(\rho^2+2\rho+3)} = O(1),$$

and

$$\begin{aligned} \frac{\rho(1-\rho)}{(1-\rho^4)(1-\rho^2)} &= \frac{(1-\kappa\theta+o(\theta))(\kappa\theta+o(\theta))}{(4\kappa\theta+o(\theta))(2\kappa\theta+o(\theta))} \\ &= \frac{\theta(1-\kappa\theta+o(\theta))(\kappa+o(1))}{\theta^2(4\kappa+o(1))(2\kappa+o(1))} \\ &= O\left(\frac{1}{\theta}\right). \end{aligned}$$

Hence

$$\mathrm{pr}_J(\theta) = O\left(\frac{1}{\theta}\right).$$

The plots in Figure 2.1 verify this limiting behavior. Assuming $\mathrm{pr}_J(\theta) \approx 1/\theta$, a transformation θ^* such that this prior is locally uniform, is

$$\theta^*(\theta) \propto \int^{\theta} \frac{1}{t} dt = \log(\theta). \tag{2.14}$$

2.3.2 Analysis of Posterior approximation for Simulated Data, $d = 1$

For the simulated data sets in **1D1** – **1D6** we use plots to study the approximation of the Jeffreys prior. As mentioned in Chapter 1, Latin Hypercube Sampling was used to obtain sample sites with the aim of introducing variation in the equispaced design, thereby assessing the approximation in a non-equispaced design. We compare the **P**osterior for the log **T**ransformed parameters using **J**effreys **P**rior (PTJP) to the **P**osterior for the log **T**ransformed parameters using a **U**niform **P**rior (PTUP) based on (2.14). The constants of proportionality are estimated using importance sampling with the importance function equal to $N(\theta^*, \hat{I}^{-1})$, the normal density with mean equal to the log of the true value and variance equal to the inverse observed Fisher information. Figure 2.2 shows that PTUP approximates PTJP well for smaller values of θ, but performs poorly when the distribution of θ is centered at large values.

2.3.3 Two Input Function

In a two input problem, the correlation between the errors at a pair of sites $(\epsilon(\mathbf{x}^{(i)}), \epsilon(\mathbf{x}^{(j)}))$ from (1.2) is

$$\mathbf{R}_{(i,j)} = \exp\left[-\theta_1(x_1^{(i)} - x_1^{(j)})^2 - \theta_2(x_2^{(i)} - x_2^{(j)})^2\right].$$

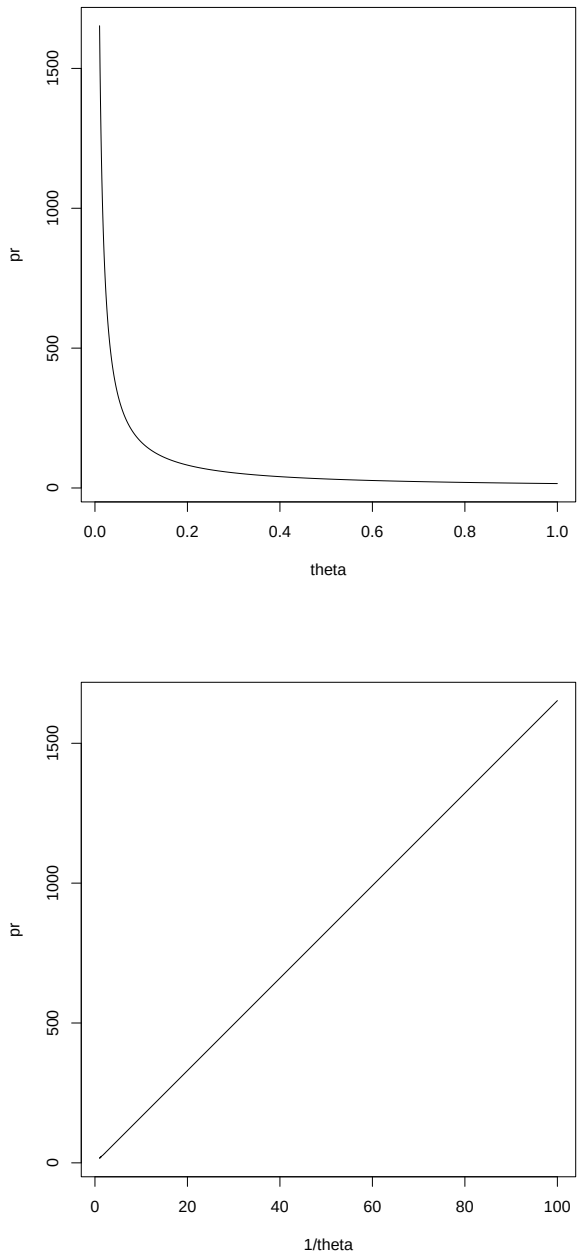


Figure 2.1: Plot of $\mathrm{pr}_J(\theta)$ versus exact θ and $1/\theta$ for $n = 3$.

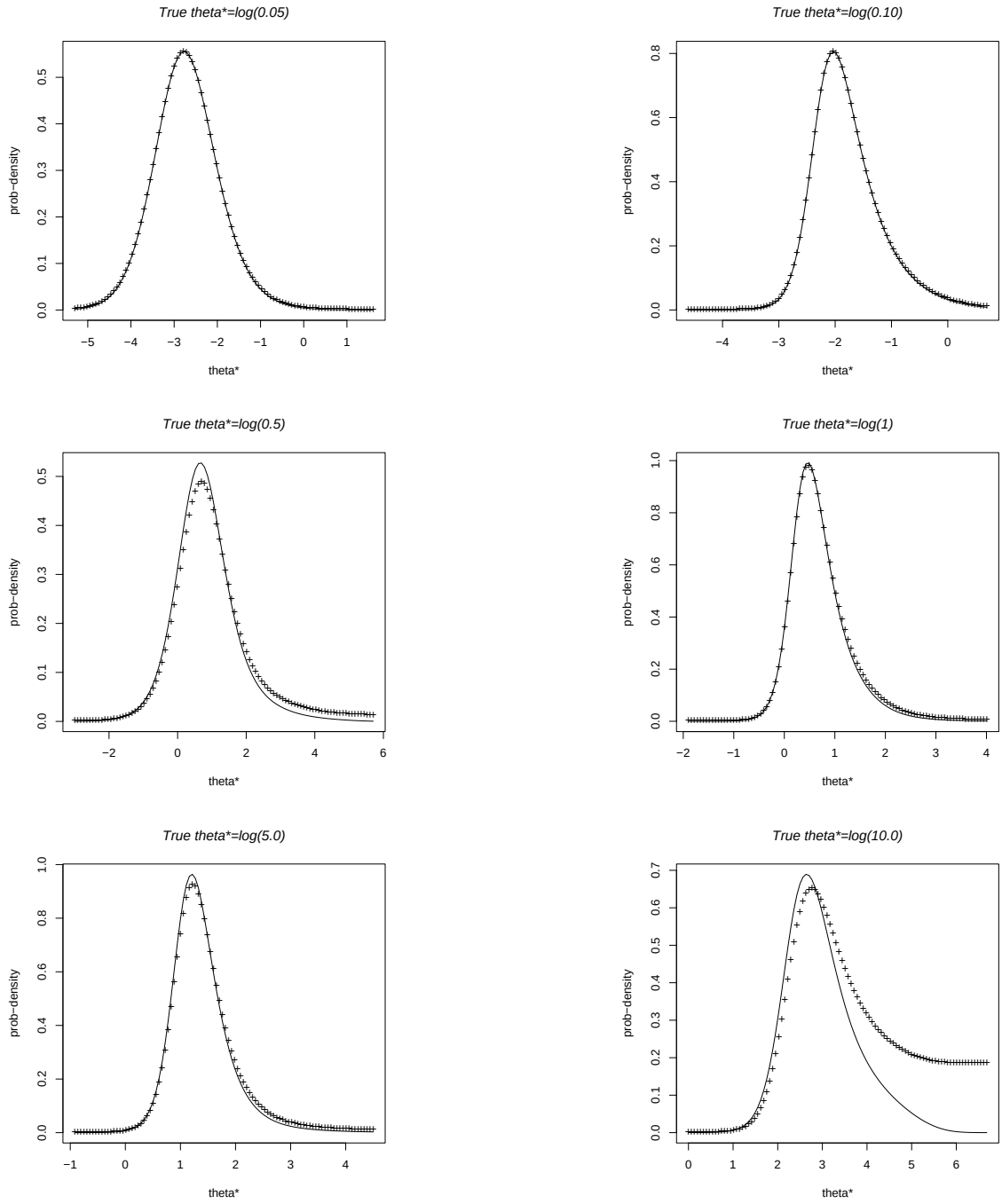


Figure 2.2: Plots of PTJP and PTUP versus $\theta^* = \log(\theta)$, for the same LHS design, solid lines for PTJP, '+' for PTUP.

The Jeffreys Prior derived in Appendix A.1 is

$$\begin{aligned} \mathrm{pr}_J(\mu, \sigma^2) &\propto 1/(\sigma^2)^{3/2}, \\ \mathrm{pr}_J(\theta_1, \theta_2) &\propto \sqrt{(\mathbf{1}\mathbf{R}^{-1}\mathbf{1})|\mathbf{B}_2|}, \end{aligned} \tag{2.15}$$

where

$$|\mathbf{B}_2| \propto \left| \begin{pmatrix} \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_{\theta_1})^2) & \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_{\theta_2})(\mathbf{R}^{-1}\mathbf{R}_{\theta_1})) & \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta_1}) \\ \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_{\theta_2})(\mathbf{R}^{-1}\mathbf{R}_{\theta_1})) & \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_{\theta_2})^2) & \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta_2}) \\ \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta_1}) & \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta_2}) & n \end{pmatrix} \right|.$$

Here $\mathbf{R}_{\theta_1}$ and $\mathbf{R}_{\theta_2}$ denote the matrices derived from differentiating elements of $\mathbf{R}$ with respect to θ_1 and θ_2 respectively.

Lemma 2.3.2. *For an equispaced design on an $n \times n$ grid, $pr_J(\theta_1, \theta_2) = pr_J(\theta_1) pr_J(\theta_2)$ where $pr_J(\theta_1)$ and $pr_J(\theta_2)$ are specified in Lemma 2.3.1.*

Proof of Lemma 2.3.2 is given in Appendix A.4. Lemma 2.3.2 can be extended to d dimension equispaced grid designs, that is $\mathrm{pr}_J(\mathbf{\Theta}) \propto \prod_{i=1}^{d} \mathrm{pr}_J(\theta_i)$. Proof of this can be obtained by induction for n^d points.

Corollary 2.3.3. *For a two-dimension equispaced grid design,*

$$pr_J(\theta_1, \theta_2) = O\left(\frac{1}{\theta_1 \theta_2}\right) \quad \text{as } \theta_1, \theta_2 \to 0^+.$$

Corollary 2.3.3 follows from the approximation results of the one input case.

Proposition 2.3.4. *Corollary 2.3.3 holds for non-equispaced, non-grid designs in d dimensions.*

The above corollary was studied graphically using Halton designs and equispaced designs in two and one dimension, and held for the examples studied.

2.3.4 Metropolis Hasting Algorithm

For inference in Bayesian analysis, the following integration expression often needs to evaluated,

$$E(h(\boldsymbol{\Theta})|y) = \int_{\boldsymbol{\Theta}} h(\boldsymbol{\Theta})\text{pr}(\boldsymbol{\Theta}\ |\mathbf{y})\ d\boldsymbol{\Theta}. \tag{2.16}$$

For example to obtain the posterior expectation of θ in the one dimension case (2.16) needs to be evaluated with $h(\boldsymbol{\Theta}) = \theta$.

Monte Carlo Integration can be used to obtain an estimate of (2.16) with

$$\hat{H} = \frac{1}{m}\sum_{i=1}^{m} h(\boldsymbol{\Theta}_i),$$

and $\boldsymbol{\Theta}_i$ are random samples drawn from the posterior density. If the posterior is not a standard density, Markov Chain Monte Carlo (MCMC) methods may be used to obtain samples.

The Metropolis Hasting Algorithm (MHA), with a normal proposal density was used to obtain samples of $\boldsymbol{\Theta}$ from PTJP and PTUP. A description of the MHA is as follows:

1. At iteration $m-1$ let the transition distribution be $N(\mathbf{0}, \boldsymbol{\Sigma})$. We choose $\boldsymbol{\Sigma} = \hat{\mathbf{I}}^{-1}$, the inverse observed information matrix evaluated at the mode of the integrated likelihood,

2. Propose $\boldsymbol{\Theta}^m = \boldsymbol{\Theta}^{m-1} + \delta$, where δ is drawn from $N(\mathbf{0}, \boldsymbol{\Sigma})$,

3. Let $p(\boldsymbol{\Theta}^m, \boldsymbol{\Theta}^{m-1}) = \left\{\min \frac{\text{pr}(\boldsymbol{\Theta}^m\ |\mathbf{y})}{\text{pr}(\boldsymbol{\Theta}^{m-1}\ |\mathbf{y})}, 1\right\}$,

4. Accept the new value $\boldsymbol{\Theta}^m$ with probability $p(\boldsymbol{\Theta}^m, \boldsymbol{\Theta}^{m-1})$, otherwise set $\boldsymbol{\Theta}^m = \boldsymbol{\Theta}^{m-1}$.

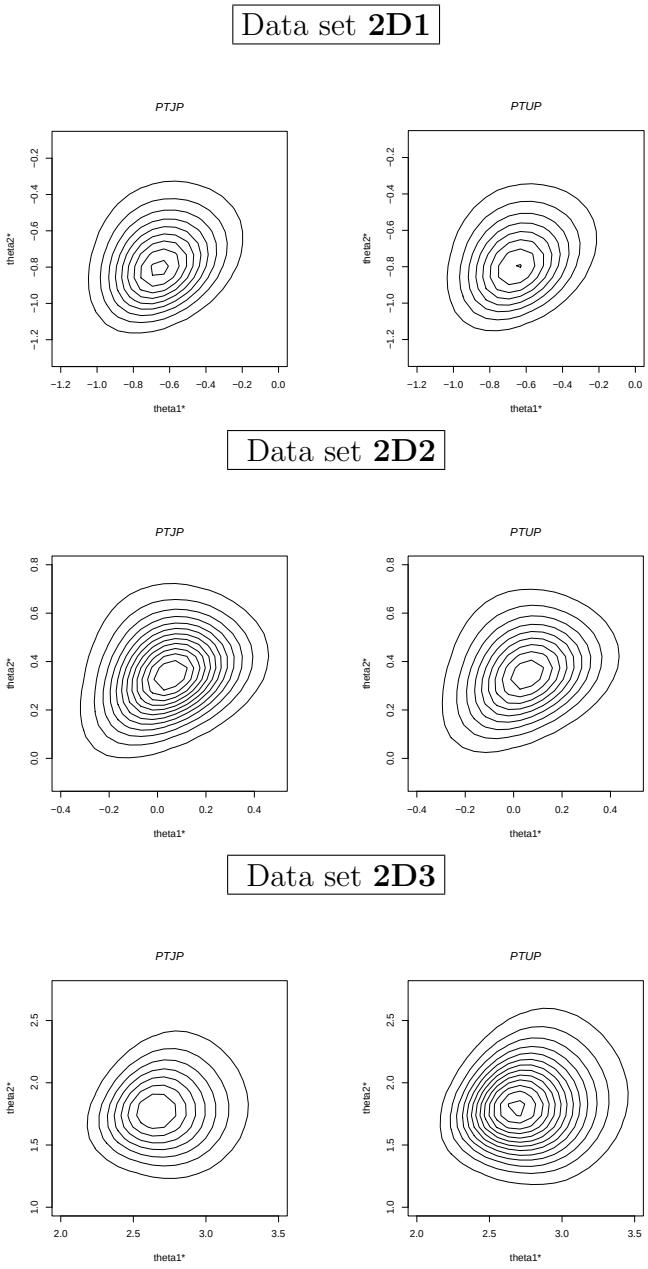


Figure 2.3: Posterior density plots for simulated data sets.

2.3.5 Analysis of Posterior Approximation for Simulated Data, $d = 2$

We use the simulated data sets **2D1**, **2D2** and **2D3** to compare the densities from samples of PTJP and PTUP. Figures 2.3 is composed of the density plots for the data sets. We observe ellipsoidal symmetry with a slight skewness in all the densities. There is not much difference in the shape of PTUP contours as compared to PTJP contours across the three simulations.

The Metropolis Hasting algorithm is used to obtain a sample of size 4000 from 50000 iterations from both PTJP and PTUP. We allow for 10000 iteration burn-ins then pick every 10th iteration to obtain samples with low correlation [26]. Based on 1000 iterations, MHA on PTUP takes an average of 32 seconds of CPU time (on a 900MHz AMD Athlon 4 Processor), whereas PTJP takes 241 seconds. The rejection rate for both densities are shown in Table 2.1. The rejection rates vary across the data sets but are approximately equal in the simulation of the two densities. The difference in MHA simulation time in the two densities is brought about by the extra operations required to compute the Jeffreys prior in PTJP. Not taking into account the operations needed to invert $\mathbf{R}$; the information matrix $\mathbf{B}_2$ has 5 elements which require operations of magnitude 21^3. The determinant of $\mathbf{B}_2$ is obtained by LU decomposition in R which requires operations of magnitude 3^3.

The marginal densities $\text{pr}(\theta_i^*|\mathbf{y})$ are presented in the Figure 2.4. Table 2.2 presents the posterior expectations of the parameters. The plots in Figure 2.4 indicate that the marginal densities in the two dimension case are similar to the one dimension densities shown in Figure 2.2. From Figure 2.4 and Table 2.2, PTUP approximates PTJP well for smaller values of the correlation parameters and have lower rejection rates. By

Data Set	True Value	PTUP %rejected	PTJP %rejected
2D1	(-1,-1)	41.38	41.59
2D2	(0, 0)	49.97	49.71
2D3	(3,2)	52.04	53.19

Table 2.1: MHA Rejection Rates for **2D1**, **2D2**, **2D3**

2D1 - True value =(-1,-1)		
	PTJP	PTUP
$\theta 1^*$	-0.63025 (0.00331)	-0.63723 (0.00330)
$\theta 2^*$	-0.74424 (0.00330)	-0.74806 (0.00326)
2D2 - True value =(0,0)		
	PTJP	PTUP
$\theta 1^*$	0.04382 (0.00283)	0.04744 (0.00287)
$\theta 2^*$	0.36081 (0.00262)	0.36948 (0.00261)
2D3 - True value =(3,2)		
	PTJP	PTUP
$\theta 1^*$	2.75133 (0.00460)	2.78920 (0.00476)
$\theta 2^*$	1.81022 (0.00480)	1.86219 (0.00533)

Table 2.2: Posterior expectations of simulated data, the standard errors are given in brackets

running MHA with different realizations, we found that for realizations simulated with values of $(\theta_1^*, \theta_2^*) > (3, 3)$, PTUP was improper hence MCMC simulations did not converge, however MCMC simulations converged for PTJP.

2.4 Normal Approximation of the Posterior

An alternative to MCMC is to approximate the posterior density with a standard density that is easy to sample. A simplified approach to the Normal approximation is as follows. Let $\hat{\mathbf{\Theta}}$ be the mode of $\text{pr}(\mathbf{\Theta}\ |\mathbf{y})$. A Taylor series expansion of the t log posterior around $\hat{\mathbf{\Theta}}$ can be written as:

$$\log(\text{pr}(\mathbf{\Theta}\ |\mathbf{y})) = \log(\text{pr}(\hat{\mathbf{\Theta}}\ |\mathbf{y})) - \frac{1}{2}(\mathbf{\Theta} - \hat{\mathbf{\Theta}})^T \hat{\mathbf{I}} (\mathbf{\Theta} - \hat{\mathbf{\Theta}}) + R(\mathbf{\Theta}) \tag{2.17}$$

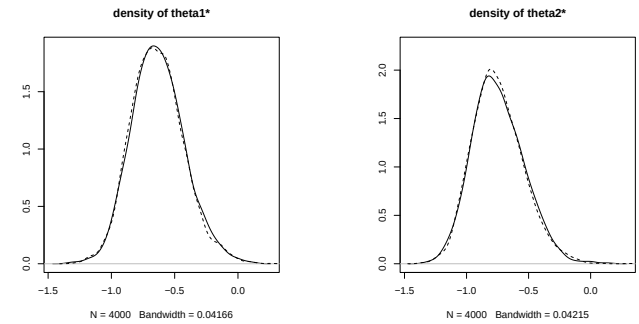
The matrix $\hat{\mathbf{I}}$ has $(i,j)^{th}$ element $\frac{-\partial^2 \log(\text{pr}(\mathbf{\Theta}\ |\mathbf{y}))}{\partial\theta_i \partial\theta_j}$ evaluated at the posterior mode. The last term in Equation (2.17) is a remainder term which is assumed to be small. By taking the exponent of Equation (2.17), $\mathbf{\Theta} \sim N(\hat{\mathbf{\Theta}}, \hat{\mathbf{I}}^{-1})$, the posterior density is approximately d-variate Normal with mean equal to the posterior mode and variance equal to the posterior modal dispersion matrix. Approximations can be improved by reparametrization. The log transformation reduces skewness by mapping the parameter space from $(0, \infty)$ to $(-\infty, \infty)$. This approximation technique is applied in the next section.

2.5 Posterior Inference for the Output Variable

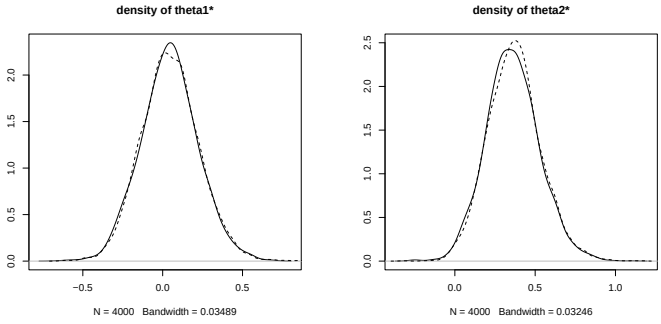
We evaluate the function used by Currin et al. [5]:

$$f(x_1, x_2) = [1 - \exp(-1/(2x_2))] \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20}, \tag{2.18}$$

$(\theta_1^*, \theta_2^*) = (-1, -1)$



$(\theta_1^*, \theta_2^*) = (0, 0)$



$(\theta_1^*, \theta_2^*) = (3, 2)$

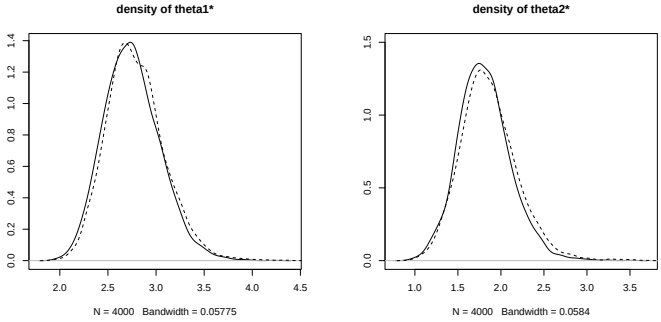


Figure 2.4: Marginal plots for PTJP (Solid line) and PTUP (dashed lined).

at 21 sites, sites chosen using a Halton sequence with the aim of obtaining predictions for a grid of 100 points.

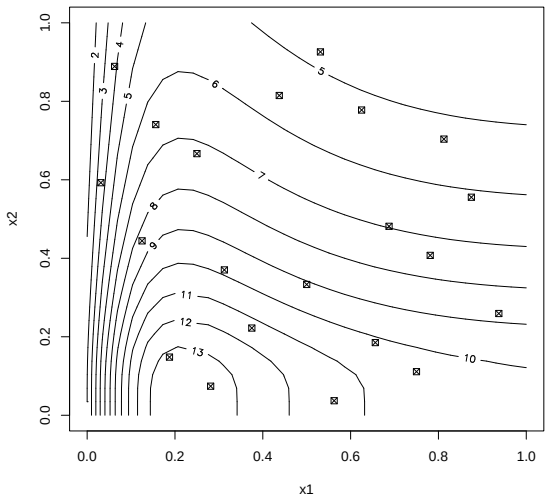


Figure 2.5: Contour plot of the test function, ⊠ shows the sampled sites.

The plot of the function and sampled sites are shown by the contour plot in Figure 2.5. The test function presented an interesting case study as the function changes relatively fast close to the point $x_2 = 0$. The BLUP being an interpolator, might not capture these changes well unless a large number of points were sampled close to this point. The design is not a very efficient one, the main purpose of the exercise is to compare prediction results. Owing to the degree of smoothness of the function, the use of the Gaussian correlation function is justified. To validate the GaSP model [18], we obtained the BLUP using MLE estimates for (θ_1, θ_2) for $y_1, \ldots, y_{21}$ and plotted the cross validated errors (CVE):

$$\epsilon_i = \frac{\hat{y}_i - y_i}{\sqrt{\text{var}(\hat{y}_i)}}, \qquad i = 1, \ldots, 21.$$

If the model were correct and ignoring uncertainty in estimating (θ_1, θ_2), the CVE

would have standard normal density. Figure 2.6 gives a plots of the predicted versus true values and CVE quantiles versus standard normal quantiles. The quantile plot does not give any indication of a departure from normality of the CVE. It is evident from the plots that the BLUP performed well, the BLUP resulted in accurate predictions except for small values of the response.

The contour plots for the Integrated Likelihood using the untransformed and log-transformed parameters are shown in Figure 2.7 and Figure 2.8. The positive skewness of the likelihood is apparent in the plots, this is reduced by reparametrization. The integrated likelihood takes on a maximum value when the estimates of the log transformed parameters are $\hat{\boldsymbol{\Theta}}^* = (1.8969, 0.6993)$. The inverse observed information matrix

$$\hat{\mathbf{I}}^{-1} = \begin{pmatrix} 0.14192362 & 0.09043233 \\ 0.09043233 & 0.09300352 \end{pmatrix}.$$

We aim to compare the performance of the Bayesian posterior expectation for $y(\mathbf{x}^*)$ to the BLUP. To do this, we obtain predictions over an equally spaced grid of 10×10 points using the following techniques,

1. BLUP predictions using $\hat{\boldsymbol{\Theta}}^*$ as true values.

2. Predictions using MHA on PTJP. We obtain a sample of size 2000 from 40000 iterations by selecting every 20th iteration after allowing 10000 burn-ins. We then use Monte Carlo Integration on Equation (1.11) and Equation (1.12) to obtain estimates of predictions and variances.

3. Predictions using MHA on PTUP. We use a similar technique as outlined above for PTJP to obtain predictions.

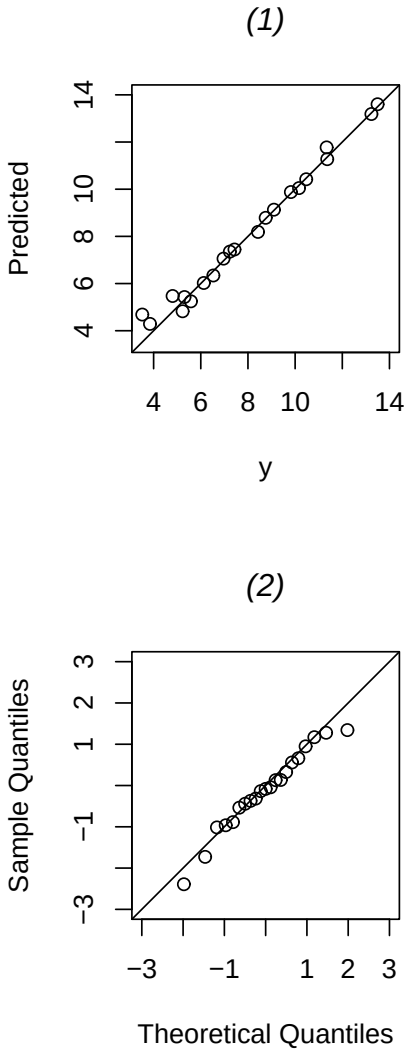


Figure 2.6: (1) Prediction versus true value, (2) CVE quantiles versus standard Normal quantiles, the lines in the plots have slope equal to one.

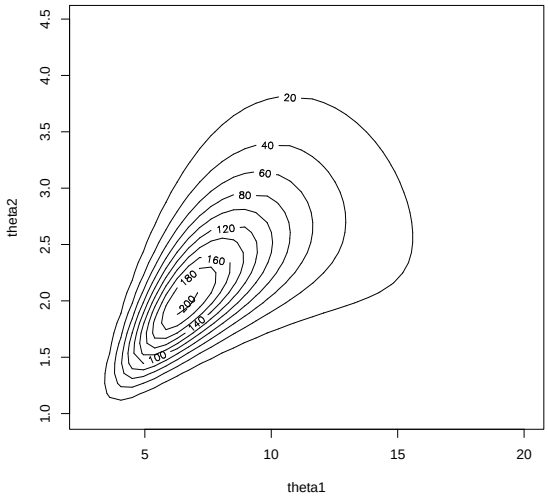


Figure 2.7: Integrated likelihood contour plot for test function data – untransformed parameters.

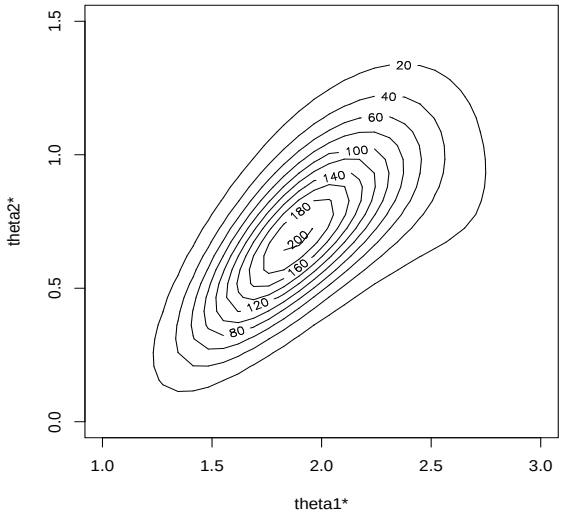


Figure 2.8: Integrated likelihood contour plot for test function data – log-transformed parameters.

4. Predictions by approximating PTJP with $N(\hat{\boldsymbol{\Theta}}^*, \hat{\mathbf{I}}^{-1})$. Monte Carlo integration is employed on Equation (1.11) and Equation (1.12) on samples from this density.

Figures 2.9 and 2.10 show the marginal density plots for θ_1^* and θ_2^* using the last three strategies above. There is close agreement between the approximating densities and PTJP.

The contours of the true and predicted functions are plotted in Figure 2.11. The plot shows that the prediction surfaces are quite similar across the four methods. Figure 2.12 is a quantile-quantile (QQ) plot of the standardized prediction errors from the other three prediction techniques versus the standardized errors using PTJP. The QQ-plots helps in comparing the distributions of the predictive errors. The extreme low and high values of the BLUP quantiles indicate left and right skewness (or heavy tails) in the sample distribution of the standardized errors. This phenomena is attributed to smaller BLUP prediction errors, compared to PTJP prediction errors. The QQ-plot also shows that the Normal approximation results in a predictive distribution which is closer to the predictive distribution from PTJP compared to PTUP.

2.6 Discussion

The Jeffreys Prior appeal is that it results in a proper posterior, it can also be approximated by a simple function which enables the use of a uniform prior on the transformed scale. This approximation works well when for small values of the correlation parameters. We found by working on various examples that the approximation works well if the maximum likelihood estimate of $\theta \leq 5$. We recommend that the

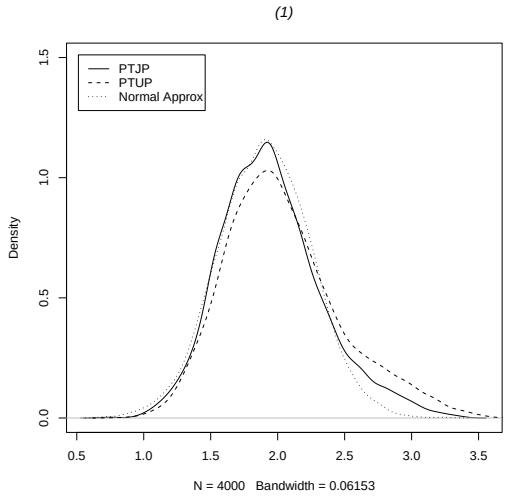


Figure 2.9: Comparative Density Plots for θ_1^* – test function data

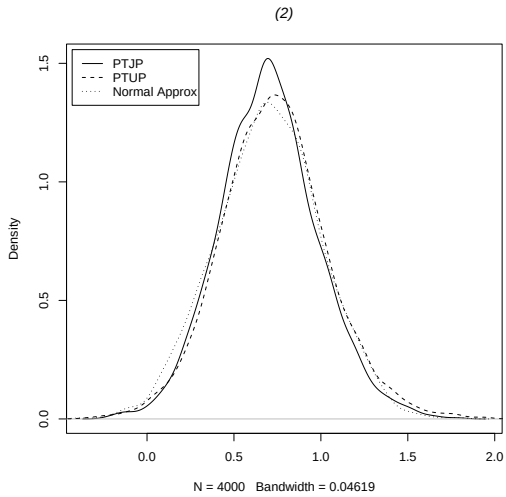


Figure 2.10: Comparative Density Plots for θ_2^* – test function data.

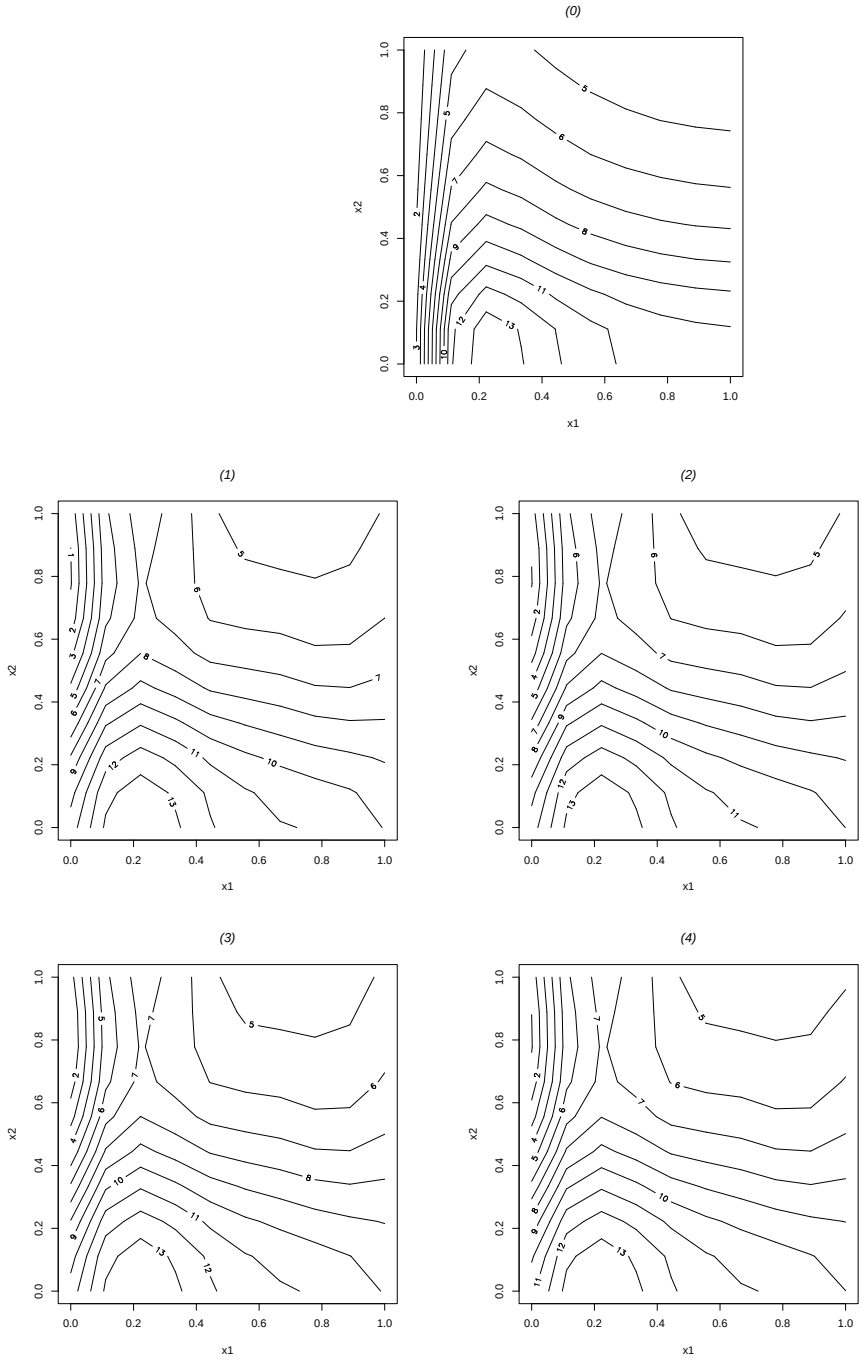


Figure 2.11: True functions and predictions: (0) – True value, (1) – BLUP, (2) – MCMC on PTJP, (3) – MCMC on PTUP, (4) – Monte Carlo on Normal approximation to PTJP.

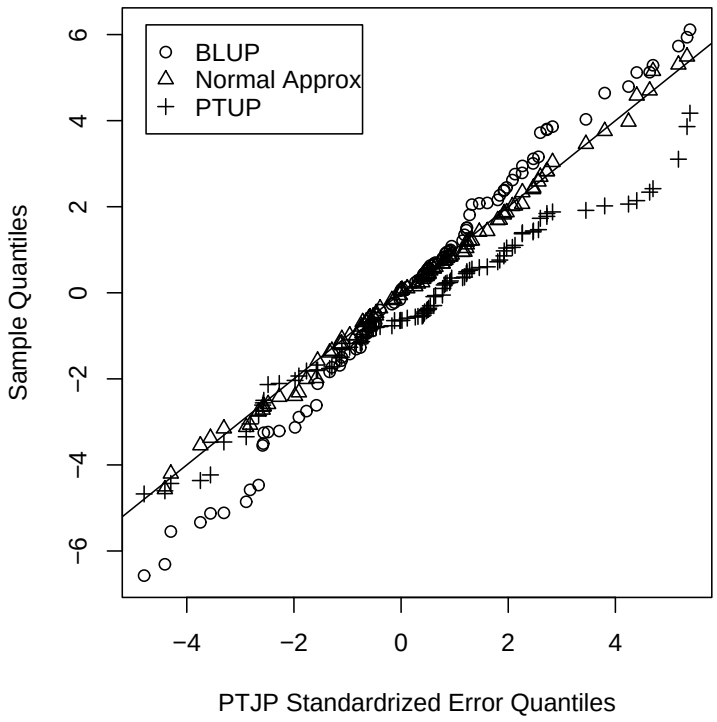


Figure 2.12: Quantile-quantile Plot of Standardized Errors from Prediction Estimates

mode and the observed information matrix be used to make inference about the dispersion of θ^*, this would give an indication on the effectiveness of this approximation technique. Though we only looked at the one and two input cases, we can extend the theory to approximate the Jeffreys prior in $d > 2$ dimensions. An educated guess would be $\text{pr}_J(\boldsymbol{\Theta}) \propto \prod_{i=1}^{d} \theta_i^{-1}$.

In the example in Section 2.5, a Normal approximation of PTJP works just as well as PTUP. This is a promising result as it makes MCMC simulations unnecessary. In other applications it might be necessary to obtain other reparametrizations in order to obtain a more ellipsoidal form for the posterior, and enable the Normal approximation.

Chapter 3

GaSP Integration

3.1 Introduction

A common problem in Bayesian analysis and computer experiments is that of integration. We often need to evaluate

$$\bar{g} = \int_{\mathcal{X}} g(\mathbf{x}) d\mathbf{x}. \tag{3.1}$$

For example in the previous chapter, we obtained predictions by evaluating

$$\bar{g} = \int_0^{\infty} E(y(\mathbf{x}^*)|\mathbf{\Theta}, \mathbf{y}) \text{pr}(\mathbf{\Theta}|\mathbf{y}) d\mathbf{\Theta}.,$$

here

$$\begin{aligned} g(\mathbf{x}) &= E(y(\mathbf{x}^*)|\mathbf{\Theta}, \mathbf{y}), \\ d\mathbf{x} &= \text{pr}(\mathbf{\Theta}|\mathbf{y}) d\mathbf{\Theta}. \end{aligned}$$

In a computer experiment setting, suppose that $\mathbf{x}^*$ is the target value of the input vector, but the input vector is assumed to be random with some distribution $d\mathbf{x}$; then we need to find the average value of y over this distribution [21],

$$\bar{g} = \int_{\mathbf{x}} y(\mathbf{x}) d\mathbf{x}. \tag{3.2}$$

One standard approach to numerical integration is MCMC as used in the previous chapter. In this chapter we examine another approach which we call *GaSP integration*. This method was first applied by O'Hagan [25] and more recently by Schonlau and Welch [34] in screening input variables in computer experiments. In the next section we outline GaSP integration based on Schonlau and Welch's technique and also point out the differences and similarity to O'Hagan's Bayesian quadrature. We give detailed one dimension illustrations in Section 3.3. The novelty of the work in this chapter is based on application. Unlike Schonlau and Welch's screening approach where they integrate out over partial sets of input variables, we integrate out over the whole set of variables. O'Hagan's application was geared to finding optimal designs in low dimensions, we use random and Halton sequences with fairly good results in low dimensions in section 3.4. GaSP integration is enabled by William J. Welch's GaSP program. This program iteratively obtains the maximum likelihood estimates of the correlation parameters and works out GaSP integration estimates. This program is also widely used in computer experiments applications and optimization [32].

We use the assumption that the domain of integration is rectangular and finite and of the form

$$\mathcal{X} = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_D, b_D]$$

where $a_1, \ldots, a_D, b_1, \ldots, b_D$ are constants. For cases where the domain is infinite, a truncation or transformation can be used to map to a finite domain. A change of notation for dimension is used in this chapter and subsequent chapters, we use D to avoid confusion with the differential d.

3.2 GaSP Integration Outline

Suppose the integrand in Equation (3.1) has D variables, starting with a sampling design of n points on the integration domain we denote this as $\mathbf{x}_1, \ldots, \mathbf{x}_n$ where

$$\mathbf{x}_i = (x_i^{(1)}, \ldots, x_i^{(D)}),$$

we evaluate

$$\mathbf{y}^T = (y_1 = g(\mathbf{x}_1), \ldots, y_n = g(\mathbf{x}_n))^T.$$

We use subscripts for the sampling design as opposed to the superscript notation from Chapter 1 and Chapter 2 so as to make the integration problem distinct from the computer experiment problem. If we assume that $g(\mathbf{x})$ is a realization of the process

$$G(\mathbf{x}) = \mu_G + \epsilon(\mathbf{x}) \tag{3.3}$$

where

$$\epsilon(\mathbf{x}) \sim N(0, \sigma_G^2),$$

and the errors are correlated, their correlation can be specified by the Gaussian correlation function,

$$\text{corr}(\epsilon(\mathbf{x}_i), \epsilon(\mathbf{x}_j)) = R(\mathbf{x}_i, \mathbf{x}_j) = \prod_{k=1}^{D} \exp(-\theta_k(\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)})^2).$$

From previous notation, the correlation matrix $\mathbf{R}$ has $(i,j)^{th}$ element equal to $R(\mathbf{x}_i, \mathbf{x}_j)$. It follows that $\bar{g}$ is also a realization of the Gaussian stochastic variable [25],

$$\bar{G}(\mathbf{x}) = \mu_{\bar{G}} + \bar{\epsilon}(\mathbf{x}), \tag{3.4}$$

where

$$\begin{aligned}\hat{\mu}_{\bar{G}} &= \bar{u}\mu_G, \\ \bar{u} &= \prod_{i=1}^{D}(b_i - a_i),\end{aligned}$$

and

$$\bar{\epsilon}(\mathbf{x}) = \int_{\mathcal{X}} \epsilon(\mathbf{x})d\mathbf{x} \sim N(0, \sigma^2_{\bar{G}}),$$

with

$$\begin{aligned}\sigma^2_{\bar{G}} &= \sigma^2_G \int_{\mathcal{X}}\int_{\mathcal{X}} R(\mathbf{x}, \mathbf{x}')d\mathbf{x}d\mathbf{x}', \\ &= \sigma^2_G \prod_{i=1}^{D}\int_{\mathcal{X}}\int_{\mathcal{X}} \exp(-\theta_k(x - x')^2)dxdx'. \end{aligned} \tag{3.5}$$

An estimate of $\bar{g}$ can be obtained using the BLUP from (3.4). The steps to deriving the BLUP of $\bar{g}$ are similar to those outlined in the introduction and are shown by Schonlau and Welch [34]. The BLUP of $\bar{g}$ is

$$\hat{\bar{g}} = \bar{u}\hat{\mu}_G + \bar{\mathbf{r}}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}_G) \tag{3.6}$$

and its estimated variance is given as

$$\text{var}(\hat{\bar{g}}) = \hat{\sigma}^2_{\bar{G}} - \hat{\sigma}^2_G\bar{\mathbf{r}}^T\mathbf{R}^{-1}\bar{\mathbf{r}} + \hat{\sigma}^2_G\frac{(\bar{u} - \mathbf{1}^T\mathbf{R}^{-1}\bar{\mathbf{r}})^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}. \tag{3.7}$$

From previous notation, $\mathbf{1}$ is a vector of ones of length n and

$$\begin{aligned}\hat{\mu}_G &= \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}, \\ \hat{\sigma}^2_G &= (\mathbf{y} - \mathbf{1}\hat{\mu}_G)^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}_G)/n.\end{aligned}$$

The vector $\bar{\mathbf{r}}$ has i^{th} element

$$\begin{aligned} \bar{r}_i &= \int R(\mathbf{x}, \mathbf{x}_i) d\mathbf{x}, \\ &= \prod_{k=1}^{D} \int_{\mathcal{X}_k} \exp(-\theta_k (x - x_i^{(k)})^2) dx. \end{aligned}$$

O'Hagan's [25] Bayesian approach uses the fact that

$$(G(\mathbf{x}_1), \ldots, G(\mathbf{x}_n))^T | \mu_G, \sigma_G^2 \sim N(\mu_G, \sigma_G^2 \mathbf{R}),$$

to specify a prior of the form

$$\text{pr}(\mu_G, \sigma_G^2) \propto 1/\sigma^2. \tag{3.8}$$

It then follows that the posterior distribution $\bar{g}$ given $\mathbf{y}$ is a shifted t-density with mean equal to $\hat{\bar{g}}$ and whose variance equals $\text{var}(\hat{\bar{g}})/(n-3)$. These results are consistent with the general Bayesian approach to the Gaussian stochastic process model, if the correlation parameters are assumed to be known, then the prior in (3.8) results in a student-t density with mean equal to the BLUP and variance equal to the scaled BLUP variance [15].

The correlation parameters $\theta_1, \ldots, \theta_D$ are estimated by maximum likelihood estimation. We refer to this approach as "GaSP integration" due to the fact that it is based on the assumption that the integrand is a realization of the Gaussian stochastic process. The BLUP of $\bar{g}$ has the same form as (1.5) but with $\mathbf{r}$ replaced by $\bar{\mathbf{r}}$, and its variance involves the extra terms $\hat{\sigma}_{\bar{G}}^2$ and $\bar{u}$. The estimate of $\bar{g}$ is a linear combination of elements contained in $\bar{\mathbf{r}}$, which depend on the sampling design on the domain of integration.

Apart from the advantage of variance reduction, the immediate appeal of this method is that it breaks down multidimensional integration problems into one and

two dimensional problems, the resulting integrals from (3.6) and (3.7) are easy to approximate. The choice of the Gaussian correlation function, as will be shown in the next section, greatly simplifies computations.

3.3 One Dimension Illustration

Consider the simple integration problem

$$\begin{aligned} \bar{g} &= \int_0^1 \sin(\pi x)dx \\ &= \frac{2}{\pi}, \\ &= 0.6366198. \end{aligned} \tag{3.9}$$

(3.10)

We choose the points

$$(x_1, x_2, x_3, x_4) = (1/4, 2/4, 3/4, 1),$$

then

$$\mathbf{y} = (1/\sqrt{2}, 1, 1/\sqrt{2}, 0).$$

First we estimate the correlation parameter $\theta_1 = \theta$, using maximum likelihood estimation. For this we use the **ms** function in Splus on the likelihood in (1.9) to get

$$\hat{\theta} = 3.30778.$$

The next step to deriving the BLUP is to evaluate elements of $\bar{\mathbf{r}}$ using the estimated

value of θ. Note that

$$\begin{aligned}
\bar{r}_i &= \int_0^1 \exp(-\theta(x-x_i)^2)dx \\
&= \sqrt{\frac{\pi}{\theta}} \int_0^1 \left(\sqrt{\frac{\pi}{\theta}}\right)^{-1} \exp(-\theta(x-x_i)^2)dx \\
&= \sqrt{\frac{\pi}{\theta}} \left(\Phi(\sqrt{2\theta}(1-x_i)) - \Phi(-\sqrt{2\theta}x_i)\right), \qquad (3.11)
\end{aligned}$$

where $\Phi(\cdot)$ is the cumulative standard Normal function. From (3.11)

$$\bar{r}_i = \sqrt{\frac{\pi}{\theta}} P(0 \leq X \leq 1) \qquad (3.12)$$

where

$$X \sim N(x_i, \frac{1}{2\theta}).$$

This shows that the $\bar{r}_i$ are the weighted normal probabilities of the sampled points being within the domain of integration. Points close to the middle will have a higher value of $\bar{r}_i$ than points close to the end points.

Using the MLE estimate of θ we estimate

$$\begin{aligned}
\hat{\mu}_G &= 0.2519199, \\
\hat{\sigma}_G^2 &= 0.3444129,
\end{aligned}$$

and using (3.6) we estimate

$$\hat{\hat{g}} = 0.6651143.$$

For the BLUP variance,

$$\begin{aligned}
\hat{\sigma}_{\bar{G}}^2 &= \hat{\sigma}_G^2 \int_0^1 \int_0^1 \exp(-\theta(x-x')^2)dxdx', \\
&= \hat{\sigma}_G^2 \sqrt{\frac{\pi}{\theta}} \int_0^1 \Phi(\sqrt{2\theta}(1-x')) - \Phi(-\sqrt{2\theta}x')dx'. \qquad (3.13)
\end{aligned}$$

The expression in (3.13) can be evaluated using numerical methods such as Simpson's rule or the rectangular rule. Using rectangular rule on (3.13) we get

$$\hat{\sigma}^2_{\bar{G}} = 0.2319448.$$

Calculations using (3.7) yield

$$\text{var}(\hat{\bar{g}}) = 0.0003679454.$$

The standard error of 0.0191 is less than the absolute error of 0.0285. Figure 3.1 is a graphical summary of this example, $\bar{r}_i$ are parabolic and symmetric about $x = 0.5$.

From the expression of $\bar{r}_i$ and $\hat{\sigma}^2_{\bar{G}}$, it is evident that the GaSP integration estimates are dependent on the value of θ. Note that when $\theta \to \infty$ then $\bar{r}_i \to 0$ and the correlation matrix is the identity matrix. It follows that

$$\begin{aligned} \hat{\bar{g}} &\approx \frac{\mathbf{1}^T\mathbf{y}}{\mathbf{1}^T\mathbf{1}}, \\ &= \sum_{i=1}^{4} y_i/4, \\ &= \bar{y}. \end{aligned}$$

Similarly, when $\theta \to \infty$, the first two components in Equation (3.7) go to zero and $\text{var}(\hat{\bar{g}}) \to \sum(y_i - \bar{y})^2/n^2$. The above observation is reinforced by Figure 3.2 which is a plot of $\hat{\bar{g}}$ as a function of θ; when there is no correlation between the points, GaSP integration is Monte Carlo integration. Figure 3.2 shows that $\hat{\bar{g}}$ attains a maximum value of 0.68 when $\theta = 23$ and as $\theta \to 0^+$ the rate of change of $\hat{\bar{g}}$ increases. Figure 3.3 plots the standard error of $\hat{\bar{g}}$ for different values of θ. The plot suggests that estimates with low absolute errors don't imply lower GaSP errors. The plot also indicates that smaller values of θ yield smaller errors. This is consistent with the findings of Yong et al. [38]. They showed that for values of $\theta \to 0$ the BLUP from (3.3) is a polynomial

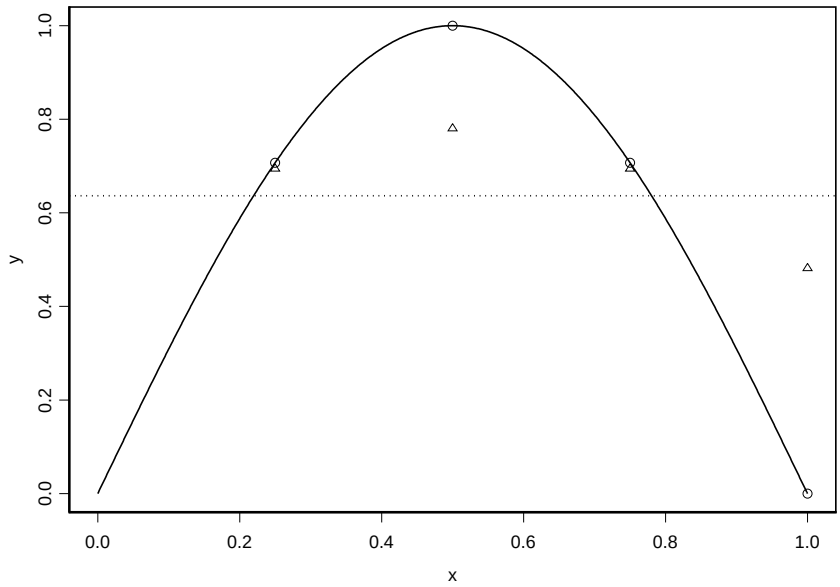


Figure 3.1: Plot of $\sin(\pi x)$, the dotted line represents $\hat{\bar{g}}$, '$\triangle$' represent $\bar{\mathbf{r}}$, '$\circ$' represent $\mathbf{y}$.

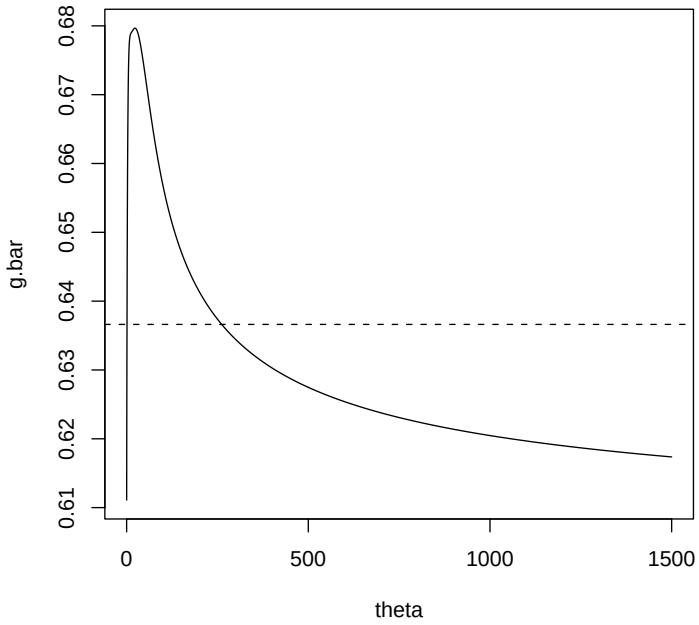


Figure 3.2: Plot of different estimates of $\hat{\bar{g}}$ versus θ, the dashed line represents the true value $\bar{g}$.

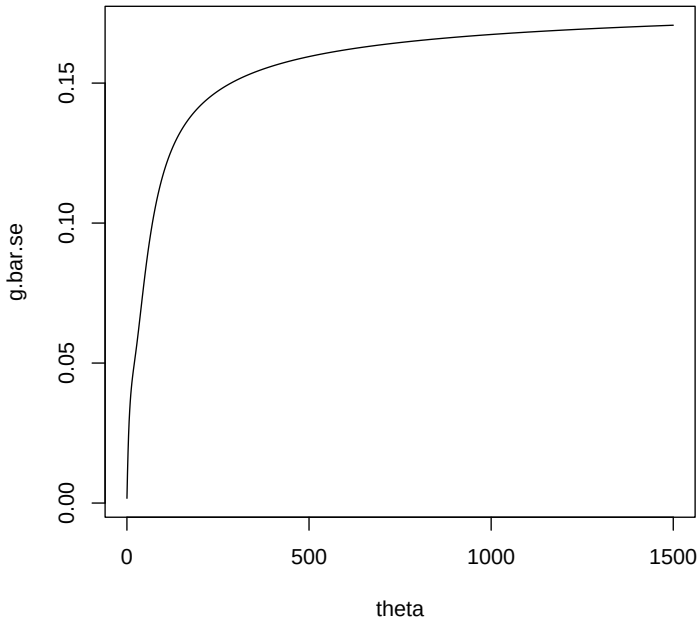


Figure 3.3: Plot of $\sqrt{\text{var}(\hat{\bar{g}})}$ versus θ.

and if $g(\mathbf{x})$ is a polynomial, it can be approximated without error – in which case the BLUP for the integral of $g(\mathbf{x})$ approximates it with no error.

3.3.1 GaSP Integration and Designs

To further analyze the effect of the design on GaSP estimates, consider the following more complex integration problem

$$\begin{aligned} \bar{g} &= \int_0^1 \sin\left(\frac{1}{0.1+x}\right) dx, \\ &= 0.5945062784. \end{aligned} \tag{3.14}$$

We use the following methods to obtain four designs:

1. Maxima and minimum points and 4 points in between which are selected randomly between the turning points and the endpoints,
2. Six points selected randomly,
3. Six equi-spaced points,
4. Six points selected by Latin Hypercube Sampling (LHS).

Integration results are presented in Table 3.1. The function and resulting design points are given in Figure 3.4. Random sampling in this case, (design (2)) concentrates sampling in the middle. Design (1) on the other hand concentrates sampling about the turning points. Smaller estimates for θ in Table 3.1 suggests smaller absolute errors. GaSP errors are smaller than absolute errors, with the exception of design (1).

We repeat the exercise 90 times for random designs and LHS designs, each time obtaining a different design of size six. For random designs we also obtain MC estimates

and standard errors. The average estimate for the integral using GaSP integration on LHS is 0.5557 with a standard error of 0.0151. The average estimate for the integral using GaSP integration on random samples is 0.5473 with a standard error of 0.0208. Normal quantile-quantile plots of the standardized errors of prediction are given in Figure 3.5. GaSP error quantiles have much heavier tails than MC estimates, indicating smaller errors. It is also clear that GaSP integration with LHS provides estimates with smaller errors compared to GaSP integration with random samples.

Design	$\hat{\theta}$	$\hat{\bar{g}}$	$\sqrt{\text{var}(\hat{\bar{g}})}$	$\lvert\hat{\bar{g}} - \bar{g}\rvert$
(1)	635.2	0.3542	0.2681	0.2403
(2)	24.6	0.3650	0.1982	0.2295
(3)	22.3	0.5257	0.0392	0.0688
(4)	16.2	0.5461	0.0250	0.0484

Table 3.1: GaSP estimates from the integration of $\sin(1/(0.1+x))$.

3.3.2 GaSP Integration and Design Size

The purpose of this exercise is to compare GaSP and MC estimates and their relationship to sample size. We obtain different random samples of size $n = 2, .., 90$ and to each sample, estimate the integral in (3.9) using both GaSP and MC integration.

The results are summarized by the plots in Figure 3.6. The plot of estimates of $\bar{g}$ versus n indicate more variation in MC estimates than GaSP estimates. The GaSP estimate with the least absolute error had $n = 34$ and $\hat{\bar{g}} = 0.5855$ with an estimated standard error of 3.3049×10^{-3}. For $n > 40$, sample size does not seem to have as much an effect on GaSP estimates, the estimates seem to take on a value of approximately

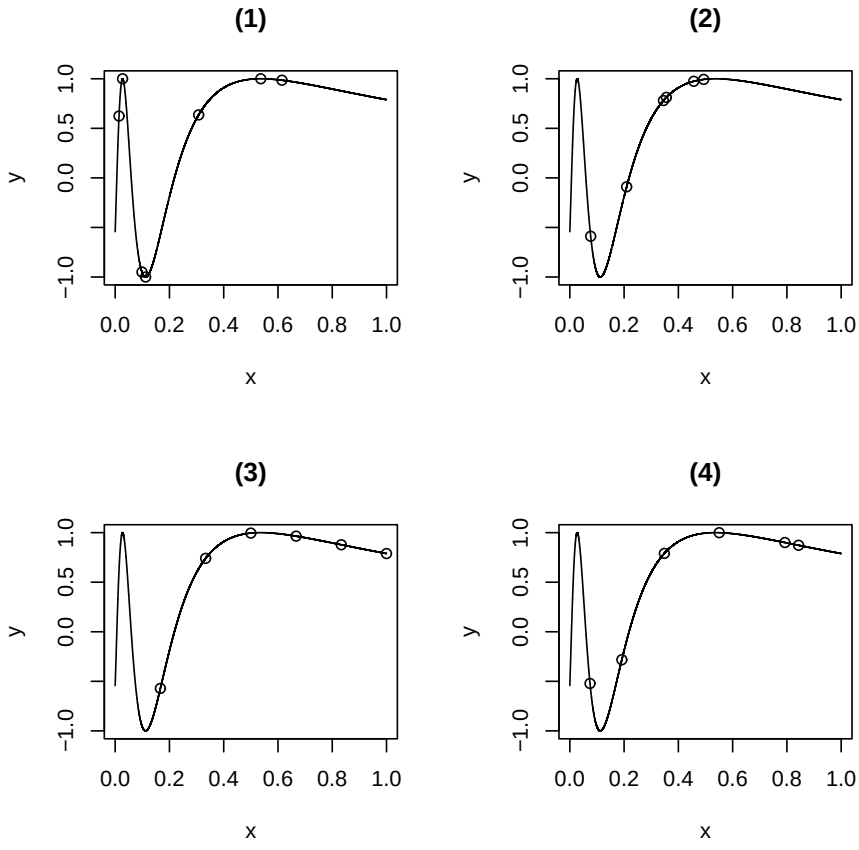


Figure 3.4: Designs used for integrating $\sin(1/(0.1+x))$

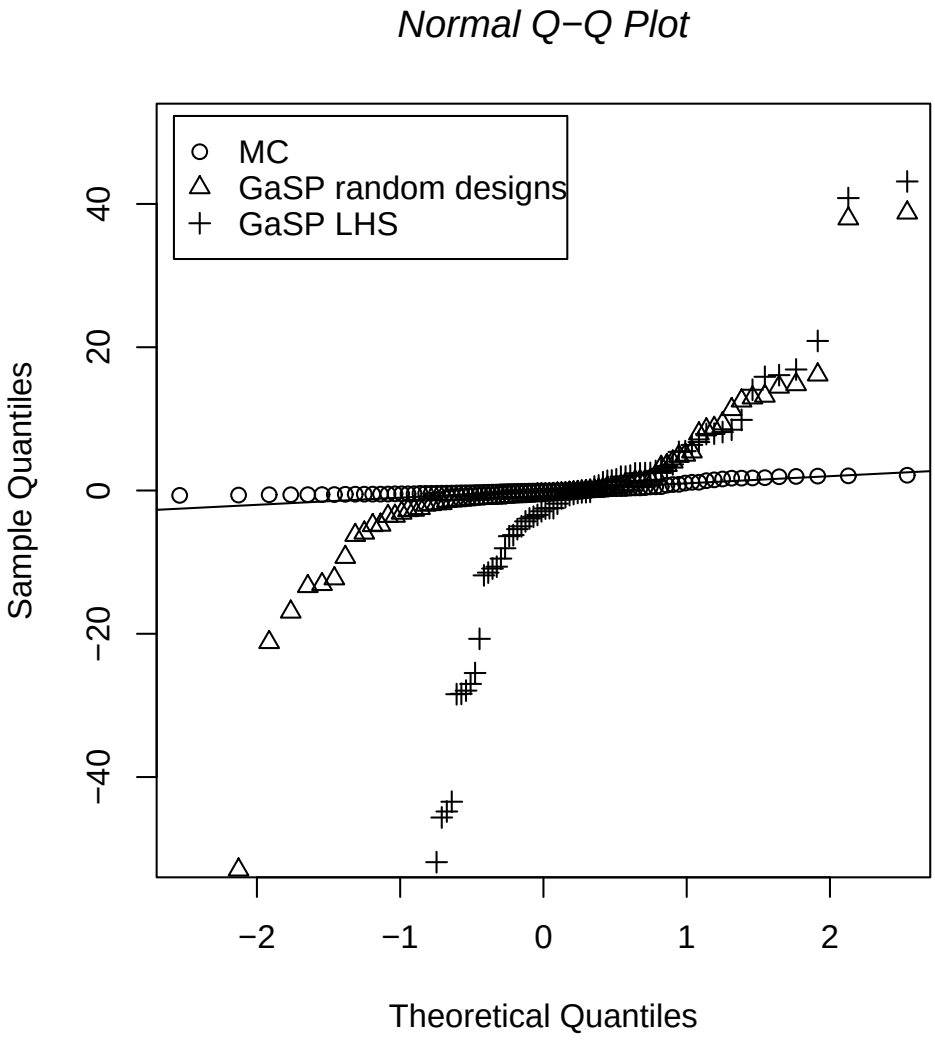


Figure 3.5: Normal QQ-Plot for standardized GaSP estimates using random sequences and LHS, and MC estimates, the solid line has intercept equal to zero and slope equal to one

0.5. This phenomena can be explained by the structure of the integrand. The change in value of $g(x)$ between any pair of sample points varies greatly by location, this possibly violates the assumption of stationarity in the model specified in (3.4).

The plot of standard errors in Figure 3.6 indicates that the estimated GaSP errors are smaller than MC errors. The variation in GaSP errors is higher than MC errors. GaSP integration errors seem to have smaller error bounds than MC errors. A least square fit of the log errors versus n supports this, the fit yields a slope of -1.5 with standard error of 0.18 for GaSP, and -0.5 with standard error of 0.03 for MC.

3.4 Multidimension Examples

3.4.1 Integration Strategies

The purpose of this exercise is to compare integration strategies, these are

1. Monte Carlo integration,

2. GaSP integration using random samples for the design.

The same random samples are used for both integration strategies. We obtain estimates of

$$\begin{aligned} \bar{f}_D &= \int_0^1 \cdots \int_0^1 \prod_{i=1}^{d} \left(\frac{\pi}{2} \sin(\pi x_i) \right) dx_1, \ldots, dx_D \\ &= 1, \end{aligned} \tag{3.15}$$

for $D = 3, 5, 10$. The integrand is a smooth continuous symmetric function, and takes on a maximum value of 1 at $x_i = 0.5, \ \forall \ i$. We obtain three estimates of the integrand in each dimension for particular n using different random samples.

The results are presented in Tables 3.2 – 3.4. From Table 3.2, GaSP integration provides estimates with lower absolute errors, the absolute errors and GaSP errors

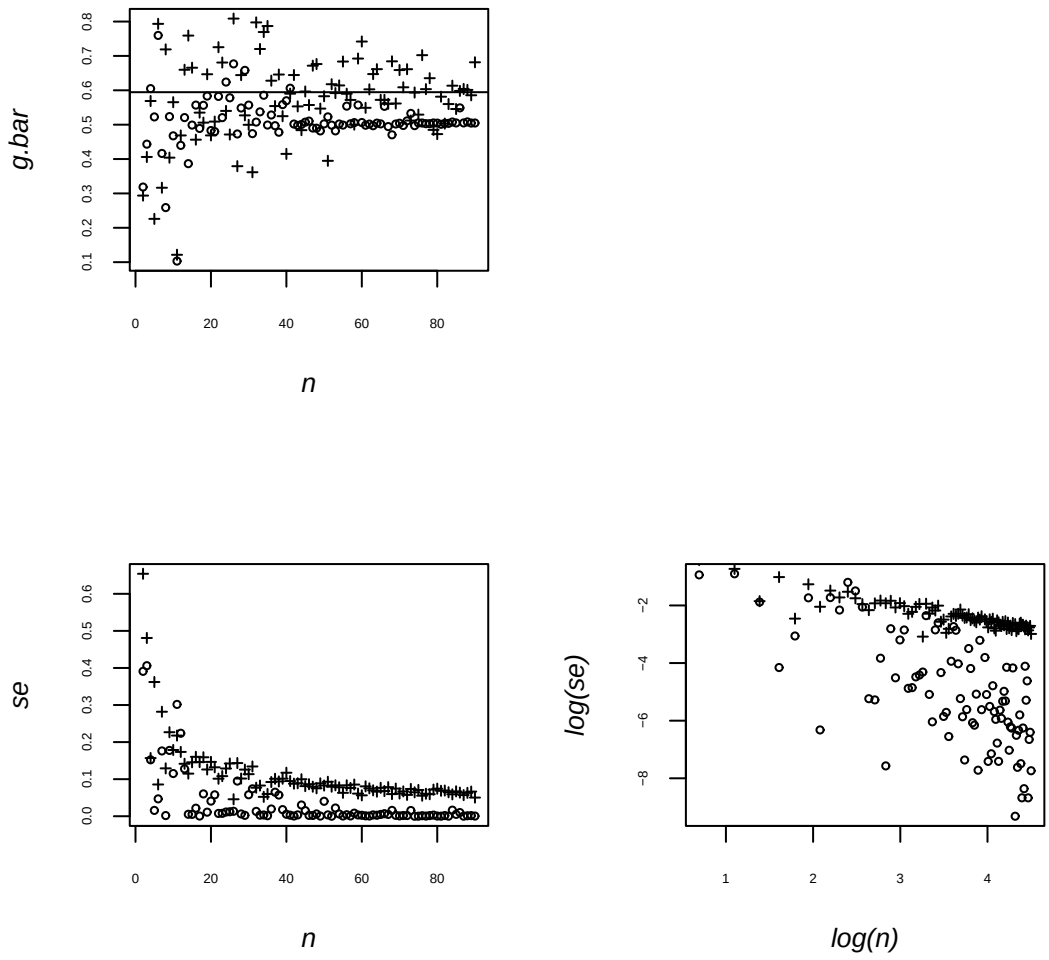


Figure 3.6: Plot of GaSP (∘) and MC (+) estimates and errors by n, the line in the top plot represents the true value of the integral

decreasing with increasing n. GaSP standard errors are substantially smaller than MC errors. It is worth noting that GaSP standard errors are almost proportional to the squared MC errors. This relationship in standard errors is not evident in estimates for $\bar{f}_5$ and $\bar{f}_{10}$. In Table 3.3, GaSP estimates have lower absolute errors. Due to computation limitations when estimating the correlation parameters for estimates of $\bar{f}_{10}$, we were only able to sample up to $n = 300$ points. In Table 3.4 the difference in the two methods is less discernable, increasing n seems to have no effect on the GaSP estimates and errors. The explanation for the poor performance of GaSP integration for $\bar{f}_{10}$ can be obtained from Figure 3.7. Calculations show that the area of the integral between $0.4 \leq p \leq 6$ which is where the integrand changes the most, is approximately equal to 0.3. In ten dimensions this translates to a volume of $(0.3)^{10}$ which is a very small proportion of the integration domain. It gets harder to find a design that explores the integration domain without using an unrealistically large number of points in higher dimensions. In the ten dimensional case, random designs are a "hit or miss" affair, the standard errors estimates for different n show this – despite the fact that we use three times more points ($n = 300$), the estimates and the standard errors do not improve.

3.4.2 Sampling and Integration Strategies

In addition to the previous integration strategies, we compare sampling designs:

1. Random sampling

2. Halton Sequences.

Halton sequences were chosen because of their equidistributed property as well as their ease in construction, we do not need to recompute the series when the number

n	MC Estimate (SE) *AE*	GaSP Estimate (SE) *AE*
30	1.0695 (0.1593) *0.0695*	1.0445 (0.0523) *0.0445*
	1.2901 (0.1766) *0.2901*	1.0411 (0.0743) *0.0411*
	0.8940 (0.1564) *0.1060*	0.9940 (0.0336) *0.0060*
60	0.9099 (0.1076) *0.0901*	0.9992 (0.0129) *0.0008*
	1.1460 (0.1298) *0.1460*	0.9945 (0.0082) *0.0055*
	1.0460 (0.1316) *0.0460*	1.0062 (0.0113) *0.0062*
90	1.0921 (0.1101) *0.0921*	0.9981 (0.0032) *0.0019*
	1.0374 (0.1045) *0.0374*	0.9997 (0.0021) *0.0003*
	1.0260 (0.1040) *0.0260*	0.9997 (0.0025) *0.0003*
120	1.1334 (0.0902) *0.1334*	1.0007 (0.0009) *0.0007*
	0.9673 (0.0866) *0.0327*	1.0020 (0.0025) *0.0020*
	1.0742 (0.0899) *0.0742*	1.0016 (0.0017) *0.0016*

Table 3.2: Estimated values, Standard Errors (SE) and Absolute Errors (AE) of $\bar{f}_3$.

n	MC Estimate (SE) *AE*	GaSP Estimate (SE) *AE*
50	0.8833 (0.1739) *0.1167*	0.9201 (0.1518) *0.0799*
	0.8568 (0.1633) *0.1432*	1.1076 (0.1210) *0.1076*
	0.8191 (0.1959) *0.1809*	0.6580 (0.1516) *0.3420*
100	1.0668 (0.1318) *0.0668*	0.9683 (0.0619) *0.0317*
	1.2452 (0.1570) *0.2452*	0.9734 (0.0768) *0.0266*
	1.0312 (0.1756) *0.0312*	1.0713 (0.0624) *0.0713*
150	0.8769 (0.1024) *0.1231*	1.0332 (0.0405) *0.0332*
	0.7628 (0.0945) *0.2372*	1.0517 (0.0420) *0.0517*
	0.8105 (0.0924) *0.1895*	0.9491 (0.0425) *0.0309*
200	0.8830 (0.0847) *0.1170*	1.0242 (0.0243) *0.0242*
	0.9501 (0.0911) *0.0499*	1.0036 (0.0278) *0.0036*
	0.9254 (0.1011) *0.0746*	0.9857 (0.0265) *0.0143*

Table 3.3: Estimated values, Standard Errors (SE) and Absolute Errors (AE) of $\bar{f}_5$.

n	MC Estimate (SE) *AE*	GaSP Integration (SE) *AE*
100	0.8417 (0.1757) *0.1583*	0.9849 (0.1671) *0.0151*
	1.0167 (0.2797) *0.0167*	0.9990 (0.2262) *0.0010*
	0.9165 (0.2335) *0.0835*	0.9424 (0.4928) *0.0576*
200	1.1683 (0.1995) *0.1683*	1.4885 (0.2597) *0.4885*
	1.0084 (0.1617) *0.0084*	0.9845 (0.1255) *0.0155*
	0.8448 (0.1346) *0.1552*	1.1675 (0.0942) *0.1675*
300	0.8376 (0.1150) *0.1624*	0.9626 (0.0879) *0.0374*
	1.3644 (0.2178) *0.3644*	1.3930 (0.1543) *0.3930*
	1.0162 (0.1351) *0.0162*	1.0602 (0.1105) *0.0602*

Table 3.4: Estimated values, Standard Errors (SE) and Absolute Errors (AE) of $\bar{f}_{10}$.

of points or dimensions increases. The integral $\bar{f}_d^*$ is

$$\int_0^1 \dots \int_0^1 \prod_{i=1}^{d} \frac{\Gamma(3.5)}{\sqrt{6\pi}\Gamma(3)} (p_i(1-p_i))^{-1} \left(1 + \frac{1}{6}\left[\log\left(\frac{p_i}{1-p_i}\right)\right]\right)^{-3.5} dp_1 \dots dp_d \quad (3.16)$$

This is a d-variate density for independent Student-t variables with 6 degrees of freedom, mapped on to the unit cube $[0,1]^d$ using the logistic transformation $p_i = 1/(1+e^{-t_i})$. The resulting transformed function which is shown in Figure 3.7 for $d = 1$, is smooth and symmetric about $p_i = 0.5$, and evaluates to one for all d.

Table 3.5 gives the estimation results for $d = 2, 5, 10$ as well as the respective absolute errors. We note that overall, GaSP integration produces estimates with smaller absolute error. For $d = 2, 5$, the Halton sequences tend to provide more accurate estimates. In the case where $d = 10$, however, any advantage of the Halton sequences over random sampling is less clear. This phenomena can be explained by the correlation of the radix inverse function, which causes clustering in higher dimensions [10]. To illustrate this, we obtained plots of points in of the projections $\bar{\mathbf{r}}$ on each axis in ten dimensions using a Halton design of 100 points. For this exercise we arbitrarily let $\theta = 0.5$. The plot in Figure 3.8 shows some clustering effects in the higher dimensions; consequently Halton sequences in higher dimensions do not explore the domain of integration as well as random sampling.

3.5 Discussion

In this chapter, we outlined GaSP integration. From the representation given in Equation (3.6), GaSP integration can be viewed as the averaged BLUP interpolators. This is one reason why GaSP performs well. We also showed that the performance of GaSP estimates depends on:

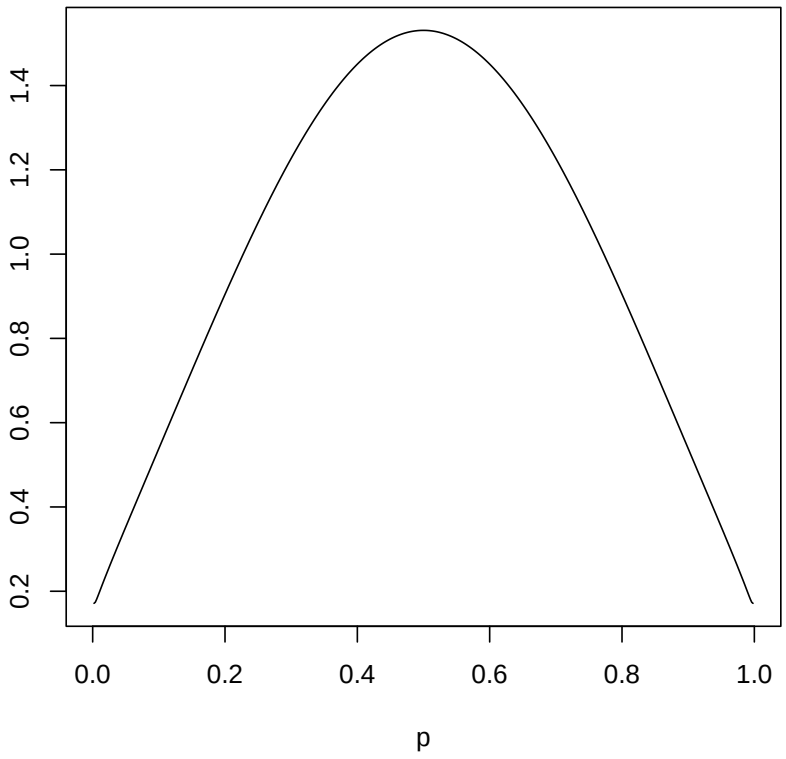


Figure 3.7: Density plot of logit transformed student-t variable with 6 degrees of freedom.

$d = 2$	**HALTON SEQUENCES**				**RANDOM SAMPLING**			
	Estimate		Absolute Error		Estimate		Absolute Error	
N	GaSP	MC	GaSP	MC	GaSP	MC	GaSP	MC
20	0.9989	1.0160	0.0011	0.0160	1.0022	0.7645	0.0022	0.2355
30	1.0000	1.0127	0.0005	0.0127	0.9989	1.0701	0.0011	0.0701
60	0.9999	1.0068	0.0002	0.0068	1.0001	1.0476	0.0002	0.0476
80	0.9999	1.0121	0.0001	0.0121	1.0003	0.9664	0.0003	0.0336
100	0.9999	1.0058	0.0003	0.0058	0.9993	0.9324	0.0006	0.0676
$d = 5$ N	GaSP	MC	GaSP	MC	GaSP	MC	GaSP	MC
50	1.0013	0.9309	0.0013	0.0691	0.9790	1.1088	0.0210	0.1088
100	0.9978	0.9681	0.0022	0.0319	1.0136	1.0042	0.0136	0.0042
150	0.9864	0.9674	0.0136	0.0326	0.9849	0.9771	0.0151	0.0229
200	0.9904	0.9674	0.0096	0.0329	1.0040	0.9771	0.0040	0.0804
250	0.9910	0.9768	0.0091	0.0232	1.0212	1.0804	0.0212	0.0478
$d = 10$ N	GaSP	MC	GaSP	MC	GaSP	MC	GaSP	MC
100	0.6886	0.6817	0.3114	0.3183	1.1077	1.3455	0.1077	0.3455
200	0.8102	0.7875	0.1898	0.2125	1.2449	1.1046	0.2449	0.1046
300	0.8817	0.8500	0.1183	0.1500	1.0589	1.0280	0.0589	0.0280
400	0.9444	0.9141	0.0556	0.0859	1.0083	1.0196	0.0083	0.0196
500	0.9433	0.9231	0.0566	0.0768	1.0175	0.9612	0.0175	0.0388

Table 3.5: Integration Results for $\bar{f}_d^*$.

Figure 3.8: Plot of projections of $\bar{\mathbf{r}}$ on the Halton design.

- The nature of the integrand – GaSP estimates have lower absolute errors than MC estimates in lower dimensions, provided that the changes in the integrand are gradual over the integration domain.

- Design – GaSP estimates with low absolute errors were obtained when the design was well spaced out over the integration domain. This is why LHS showed an improvement over random sampling in Section 3.3.1.

- Size of the design – When the above two items are taken into consideration, the GaSP estimates improve with n. However the size of the design imposes limitations in higher dimension when computing GaSP estimates. This is due to the computation power needed to invert the correlation matrix $\mathbf{R}$.

One point of concern is the small values of the estimated error in GaSP integration in some applications. This is possibly due to two reasons:

- The violation of the model assumptions by the function. This would affect the estimation of the correlation parameters and therefore the errors.

- The use of point estimates for θ which excludes the uncertainty in the estimation of θ. The results in Section 3.3 show that different designs will result in different estimates of θ.

Due to time constraints, we do not investigate this phenomena further. One way of overcoming this is by running GaSP integration several times with different designs so as get an idea of the uncertainty of the estimates.

We solve GaSP integration's shortcomings in the next chapter by defining an integration algorithm which allows for adaptive sampling. The algorithm also subdivides the integration domain into regions where the integrand is more or less uniform

which enable better performance of GaSP integration within the sub-regions.

Chapter 4

Adaptive Sub-region Sampling Integration Algorithm

4.1 Introduction

In this chapter, we introduce the Adaptive Sub-region Sampling Integration algorithm (ASSIA). The algorithm is outlined in Section 4.3. The motivation of ASSIA was to develop a numeric method which requires few evaluations of expensive integrands and enables GaSP integration on sections of the domain where the changes to the integrand are more uniform. The algorithm works by dynamically dividing the region of integration into more homogenous sub-regions until a maximum number of iterations or work level is achieved. Instead of GaSP integration, Monte Carlo integration can be used for estimation within the sub-region. We compare GaSP and MC integration in two dimension integration examples in Section 4.4 and illustrate the use of GaSP integration in Bayesian integration in computer experiments in Section 4.4.3. Recommendations and concluding remarks are presented in Section 4.5.

4.2 Methodology

Revisiting the integration problem,

$$\bar{g} = \int_{\mathcal{X}} g(\mathbf{x}) d(\mathbf{x}), \tag{4.1}$$

we aim to obtain finer subdivisions of the original integration region $\mathcal{X}$, with smaller sub-regions where the integrand varies most, once this is done, it can be assumed that the integration domain has been divided into sub-regions where the integrand satisfies the model assumptions in each sub-region. GaSP integration is then employed within the sub-regions. The estimate of the integral is a weighted sum of all the GaSP estimates of the sub-regions.

Suppose the integration domain has been sub-divided into m independent rectangular sub-regions. We use the notation $\mathcal{X}_i$ to denote a sub-region in $\mathcal{X}$, the random sample $\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}$ is used to obtain the GaSP estimates for the integral in this sub-region, which is denoted as

$$\bar{g}_i = \int_{\mathcal{X}_i} g(\mathbf{x}) d(\mathbf{x}). \tag{4.2}$$

Recall from (3.6) and (3.7) the estimate of (4.2) is

$$\hat{\bar{g}}_i = \bar{u}_i \hat{\mu}_{G_i} + \bar{\mathbf{r}}_i^T \mathbf{R}_i^{-1} (\mathbf{y}_i - \mathbf{1}\hat{\mu}_{G_i}) \tag{4.3}$$

with variance

$$\begin{aligned} V_i &= \text{var}(\hat{\bar{g}}_i) \\ &= \hat{\sigma}^2_{\bar{G}_i} - \hat{\sigma}^2_{G_i} \bar{\mathbf{r}}_i^T \mathbf{R}_i^{-1} \bar{\mathbf{r}}_i + \hat{\sigma}^2_{G_i} \frac{(\bar{u}_i - \mathbf{1}^T \mathbf{R}_i^{-1} \bar{\mathbf{r}}_i)^2}{\mathbf{1}^T \mathbf{R}_i^{-1} \mathbf{1}}, \end{aligned} \tag{4.4}$$

The subscript in (4.3) and (4.4) indicate that computations are done within the sub-region using the sampled sites. Assuming the $\hat{\bar{g}}_i$ are independent, the estimate of the

integral in (4.1) is given as,

$$\bar{\hat{\bar{g}}} = \sum_{k=1}^{m} w_k \hat{\bar{g}}_k \tag{4.5}$$

with variance

$$V = \sum_{k=1}^{m} w_k^2 V_k, \tag{4.6}$$

where w_k are weights corresponding to the volume of $\mathcal{X}_k$, that is if

$$\begin{aligned} \mathcal{X}_k &= [a_{k1}, b_{k1}] \times [a_{k2}, b_{k2}] \times [a_{kD}, b_{kD}] \\ w_k &= \prod_{l=1}^{D} (b_{kl} - a_{kl}) \end{aligned}$$

To obtain sub-regions and sample points, we do this adaptively using the **A**daptive **S**ub-region and **S**ampling **I**ntegration **A**lgorithm (ASSIA). More details are given in the next section.

4.3 The Adaptive Sub region Sampling Integration Algorithm

Suppose that at some stage in the algorithm the region of integration $\mathcal{X}$ has been subdivided into m subregions, the relevant pieces of information are kept in a list

$$\mathcal{S} = \{(\mathcal{X}_1, \hat{\bar{g}}_1, w_1^2 V_1, n_1), (\mathcal{X}_2, \hat{\bar{g}}_2, w_2^2 V_2, n_2), \ldots, (\mathcal{X}_m, \hat{\bar{g}}_m, w_m^2 V_m, n_m)\}.$$

The sampled points and computed values of the function are stored in matrices Xdata and Ydata, these matrices have a key which associates points with elements in $\mathcal{S}$. A description of the algorithm is as follows:

For a given integrand $g(\mathbf{x})$, region of integration $\mathcal{X}$, and maximum work, $\mathcal{W}$:

1. Compute a global estimate for the integrand $\hat{\bar{g}}$, and its variance estimate using an initial sample size n_0. Typically we use the rule of thumb, $n_0 = 10 \times D$, where D is the dimension of the integration space. Initialize

$$\mathcal{S} = \{(\mathcal{X}_1 = \mathcal{X}, \hat{\bar{g}}_1 = \hat{\bar{g}}, w_1^2 V_1 = \prod_{i=1}^{D}(b_i - a_i)^2 \text{var}(\hat{\bar{g}}), n_1 = n_0)\},$$

2. **while** (work $< \mathcal{W}$)

   (a) Pick a sub-region to sub-divide and its associated points and remove its information from the list $\mathcal{S}$. This is the region with the largest variance, $\mathcal{X}^* = \mathcal{X}_l$ where $\max(w_1^2 V_1, \ldots, w_m^2 V_m) = w_l^2 V_l$ with $1 \leq l \leq m$. If $m = 1$, then $\mathcal{X}^* = \mathcal{X}$.

   (b) Verify that the number of points in $\mathcal{X}^*$ is at least n_{top}. If not increase the points to n_{top} by random sampling.

   (c) Determine the axis to sub-divide. To divide $\mathcal{X}^*$ we sequentially sub-divide each co-ordinate axis into half. For each split across a co-ordinate axis j, $j = 1, \ldots, D$, the estimated variance using the sampled points in both halves, V_1^{*j} and $V_2^{*j}$are used to estimate the sub-region variance,

$$V^{*j} = V_1^{*j} + V_2^{*j}.$$

   The co-ordinate axis to be subdivided $\bar{j}$, is such that $\min(V^{*1}, \ldots, V^{*D}) = V^{*\bar{j}}$. Divide across this axis to get $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$

   (d) Obtain estimates $\hat{\bar{g}}_1^*$ and $\hat{\bar{g}}_2^*$ by applying GaSP integration to points already in $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$.

   (e) Update $\mathcal{S}$, Xdata, Ydata by inserting the integration information of the new sub-regions $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$.

3. **end(while)**

The sequence of steps in (2) consists of one iteration and is continuously repeated with $\mathcal{S}$ being updated until a maximum work level $\mathcal{W}$ is achieved. The work level $\mathcal{W}$ can be quantified in various ways, for example it could be the number of iterations, or the maximum number of computations of the integrand. After m sub-regions have been obtained, the total number of sampled points is

$$n_{total} = \sum_{i=1}^{m} n_i.$$

We denote $\bar{\hat{g}}_{\{m\}}$ and $V_{\{m\}}$ as the integral estimate and its variance after m iterations.

A demonstration of how splitting is carried out is given by Figure 4.1 for the first 15 iterations of the integration problem given in the previous chapter,

$$\int_0^1 \sin(1/(0.1+x))dx,$$

with

$$\begin{aligned} n_0 &= 10, \\ n_{top} &= 10, \\ \mathcal{W} &= \text{max 150 points.} \end{aligned}$$

The figure shows that as the algorithm progresses, smaller sub-regions are allocated to areas where the function is rapidly changing.

A demonstration of how points are allocated by the algorithm is better shown in Figure 4.2 with

$$\bar{g} = \int_0^1 \int_0^1 1/(1-x_1x_2)dx_1dx_2,$$

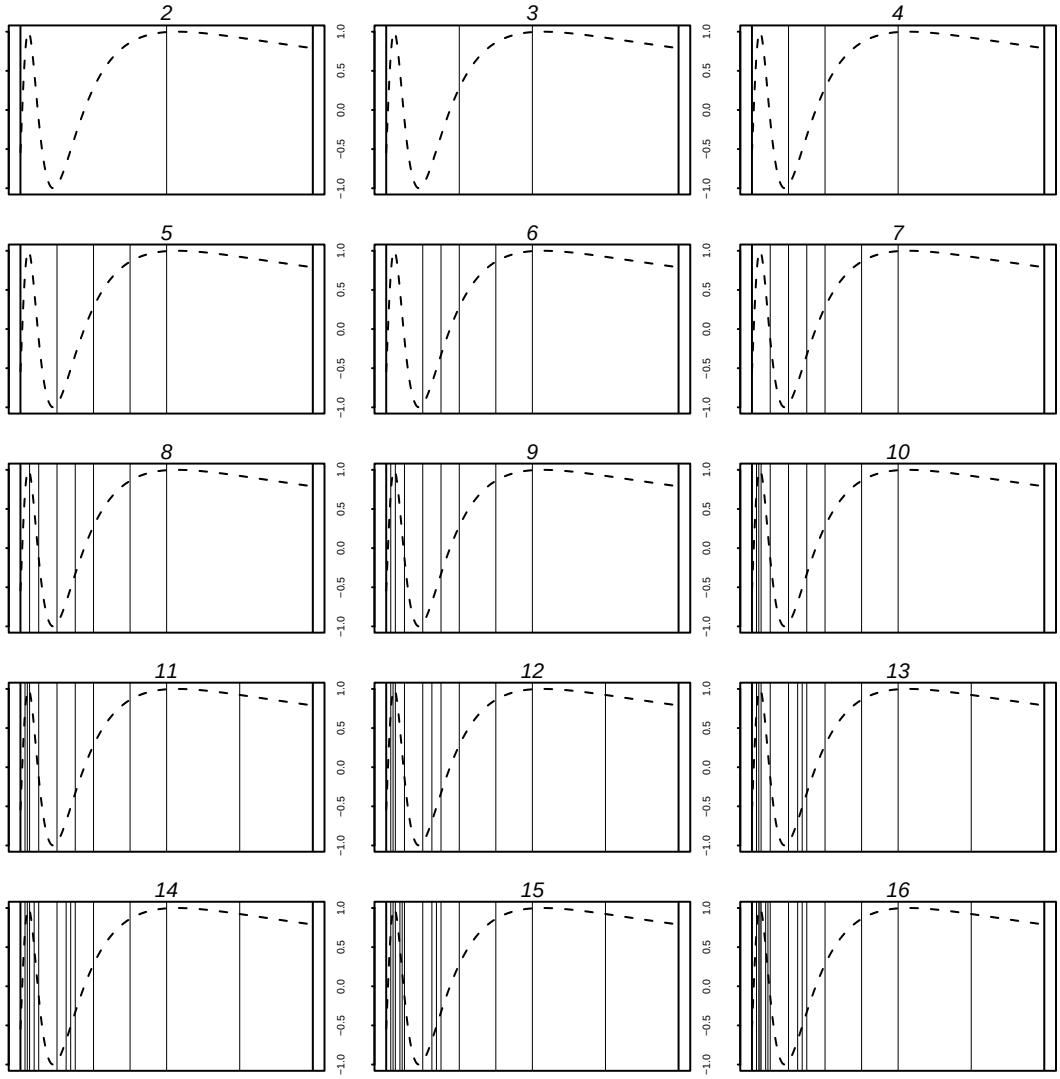


Figure 4.1: ASSIA-GaSP splitting of $\sin(1/(0.1+x))$, main title gives number of sub-regions, vertical lines indicate splits.

and

$$\begin{aligned} n_0 &= 20, \\ n_{top} &= 20, \\ \mathcal{W} &= 12 \text{ iterations.} \end{aligned}$$

The plot shows that the algorithm locates the rapid changes close to (1,1) and samples close to this point. Though the algorithm is primarily meant to improve GaSP integration, MC integration can be applied within sub-regions. Instead of GaSP estimates in sub-region i, with MC integration we have

$$\hat{\bar{g}}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} g(\mathbf{x}_{ik}) \tag{4.7}$$

$$V_i = \frac{1}{n_i} \frac{1}{n_i - 1} \sum_{k=1}^{n_i} \left(g(\mathbf{x}_{ik}) - \hat{\bar{g}}_i \right)^2 \tag{4.8}$$

We denote GaSP integration within ASSIA as ASSIA-GaSP and MC integration within ASSIA as ASSIA-MC.

Since the algorithm serves to minimize the estimated standard error, the combined error estimate at the end of a run is small and does not reflect the error of the estimate of the integral, to overcome this we repeat the algorithm a number of times to obtain some measure of uncertainty.

4.4 Applications

4.4.1 One Variable Function

Five runs of ASSIA-GaSP on the problem

$$\int_0^1 \sin(1/(0.1 + x))dx$$

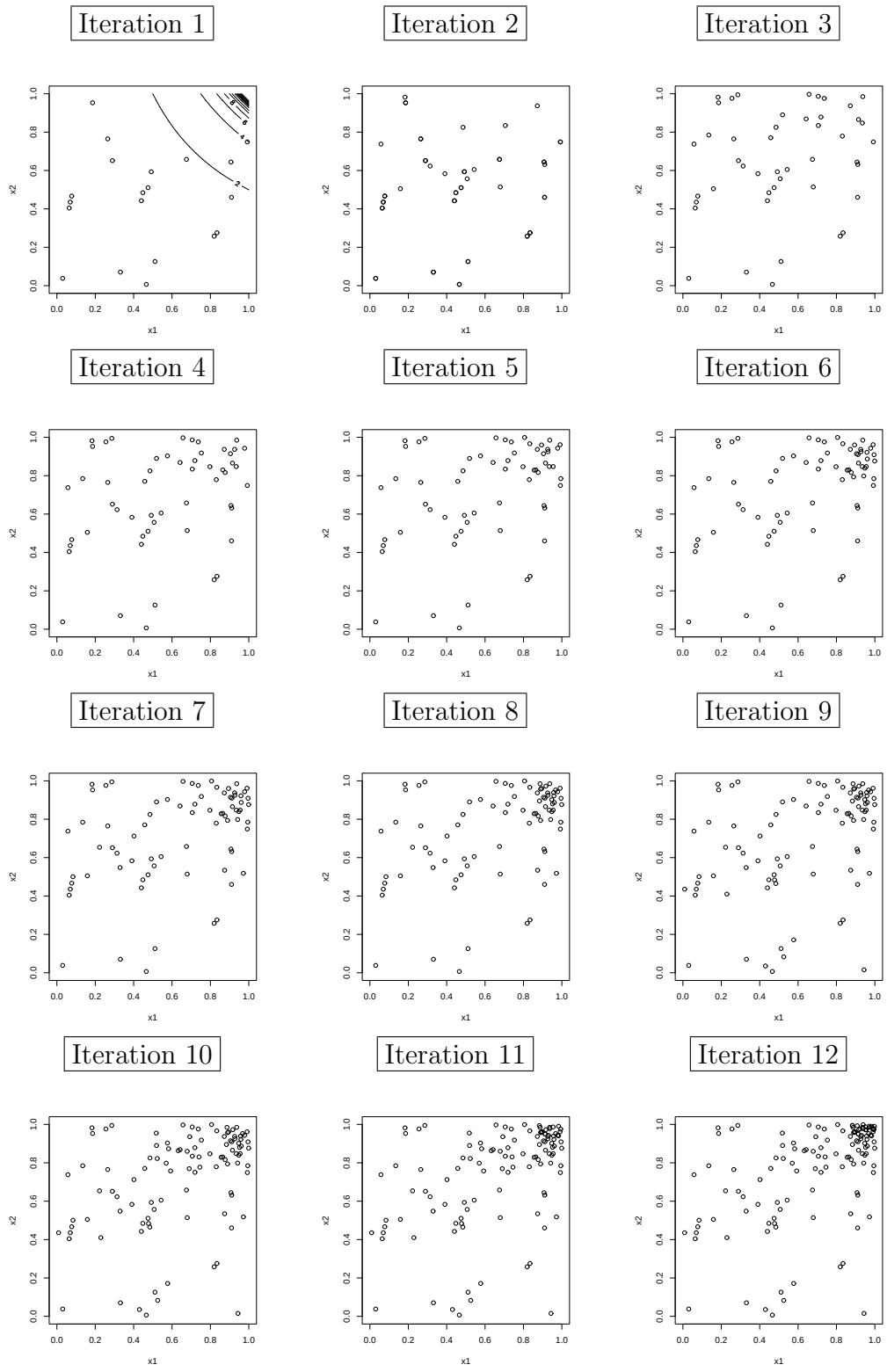


Figure 4.2: Demonstration of ASSIA-GaSP point allocation with integrand $1/(1 - x_1x_2)$.

were obtained with

$$\begin{aligned} n_0 &= 10, \\ n_{top} &= 10, \\ \mathcal{W} &= 4 \text{ iterations} \end{aligned}$$

The choice of four iterations was based on results from GaSP integration in the previous chapter. The results from four runs are presented in Table 4.1. The true value of the integral, correct to four decimal places is 0.5945. The ASSIA-GaSP results are promising, with roughly 50 points we are able to obtain 3 decimal place accuracy for the integral.

$\bar{\hat{\bar{g}}}$	$\lvert\bar{g} - \bar{\hat{\bar{g}}}\rvert$	n_{total}
0.5943	0.0002	54
0.5936	0.0009	53
0.5942	0.0003	51
0.5944	0.0001	50
0.5942	0.0003	50

Table 4.1: ASSIA-GaSP results for the integration of $\sin(1/(0.1+x))$

4.4.2 Two Variable Functions

We applied ASSIA to estimate

$$\int_0^1 \int_0^1 g(x_1, x_2) dx_1 dx_2$$

where

$$g(x_1, x_2) = f_1 = \sin(2\pi x_1) + \sin(2\pi x_1 + \pi x_2) \quad (4.9)$$

$$g(x_1, x_2) = f_2 = \exp(-(x_1^2 + x_2^2)) \quad (4.10)$$

$$g(x_1, x_2) = f_3 = 1/(1 - x_1 x_2) \quad (4.11)$$

$$g(x_1, x_2) = f_4 = \sqrt{|x_1 - x_2|}. \quad (4.12)$$

The true values are

$$\begin{aligned} \bar{g} &= 0 \text{ for } f_1, \\ \bar{g} &= 0.557746 \text{ for } f_2, \\ \bar{g} &= 1.644931 \text{ for } f_3, \\ \bar{g} &= 0.533333 \text{ for } f_4. \end{aligned}$$

We chose the above functions because of their varied characteristics, f_1 is an oscillatory function, f_2 is a smooth well behaved function whose higher derivatives are also well behaved, f_3 blows up at $(1, 1)$ and the derivative for f_4 does not exist along the line $x_1 = x_2 = 1$.

We applied GaSP and MC integration within ASSIA. The parameters to ASSIA using both these techniques were:

$$\begin{aligned} n_0 &= 10, \\ n_{top} &= 20, \\ \mathcal{W} &= 30 \text{ iterations}. \end{aligned}$$

GaSP integration provides very good results for the better behaved functions with $n_0 = 20$, which is why we start with a smaller value of $n_0 = 10$; this makes it easier

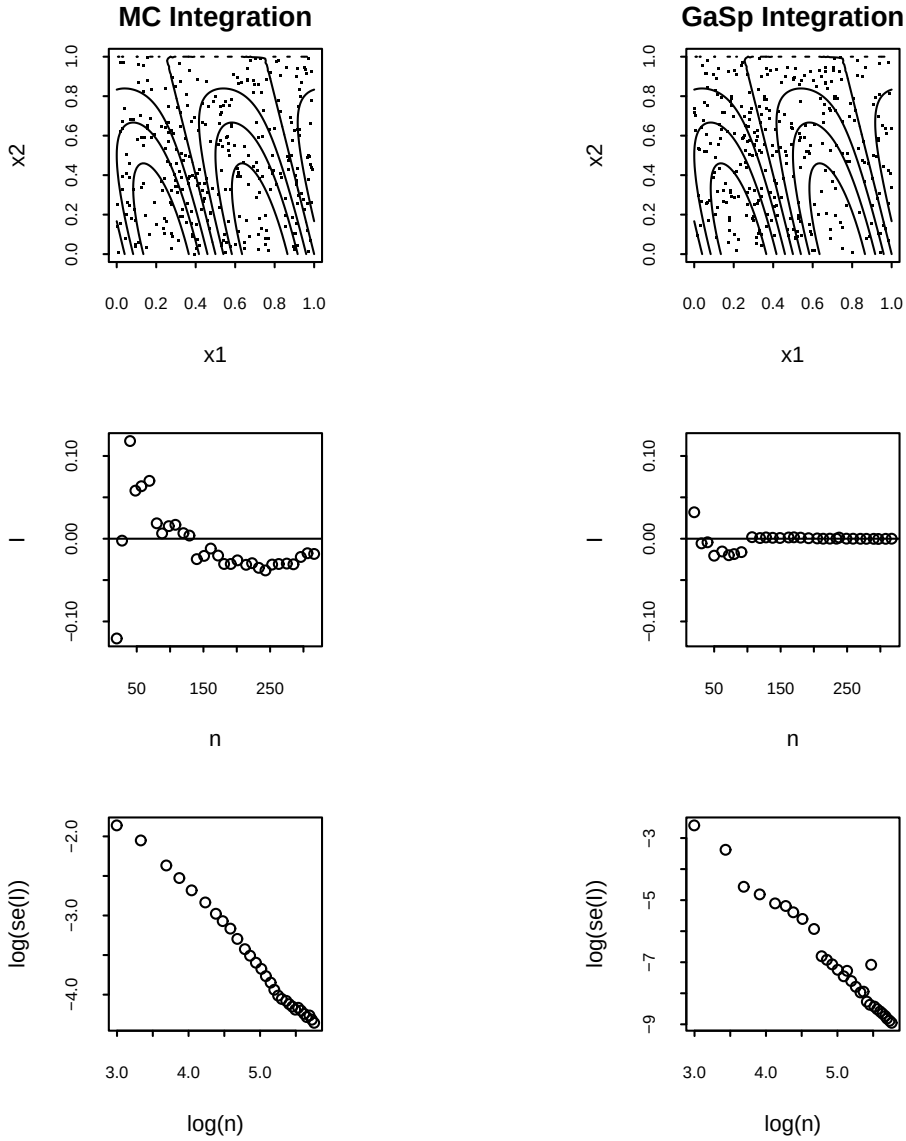


Figure 4.3: ASSIA integration results for f_1, '$\circ$' represents estimates at an iteration in the run.

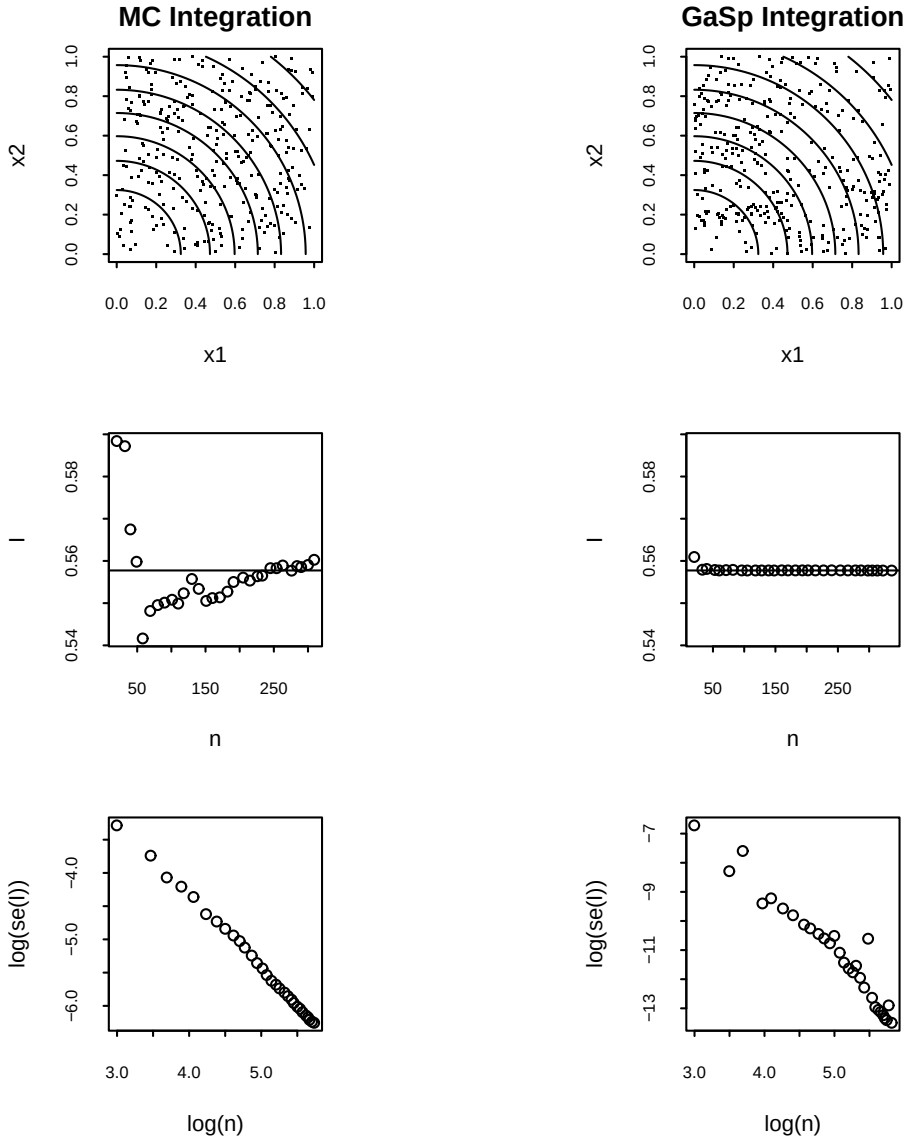


Figure 4.4: ASSIA integration results for f_2, '$\circ$' represents estimates at an iteration in the run.

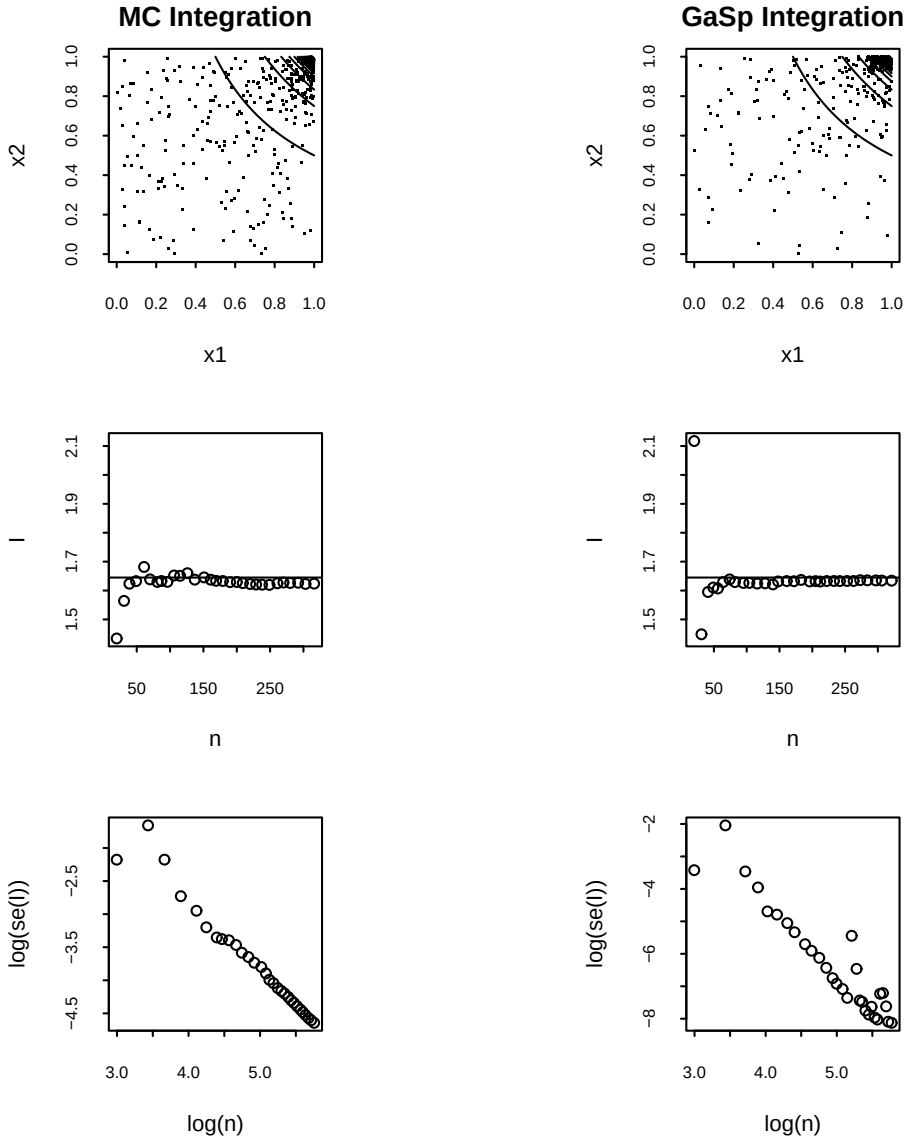


Figure 4.5: ASSIA integration results for f_3, '$\circ$' represents estimates at an iteration in the run.

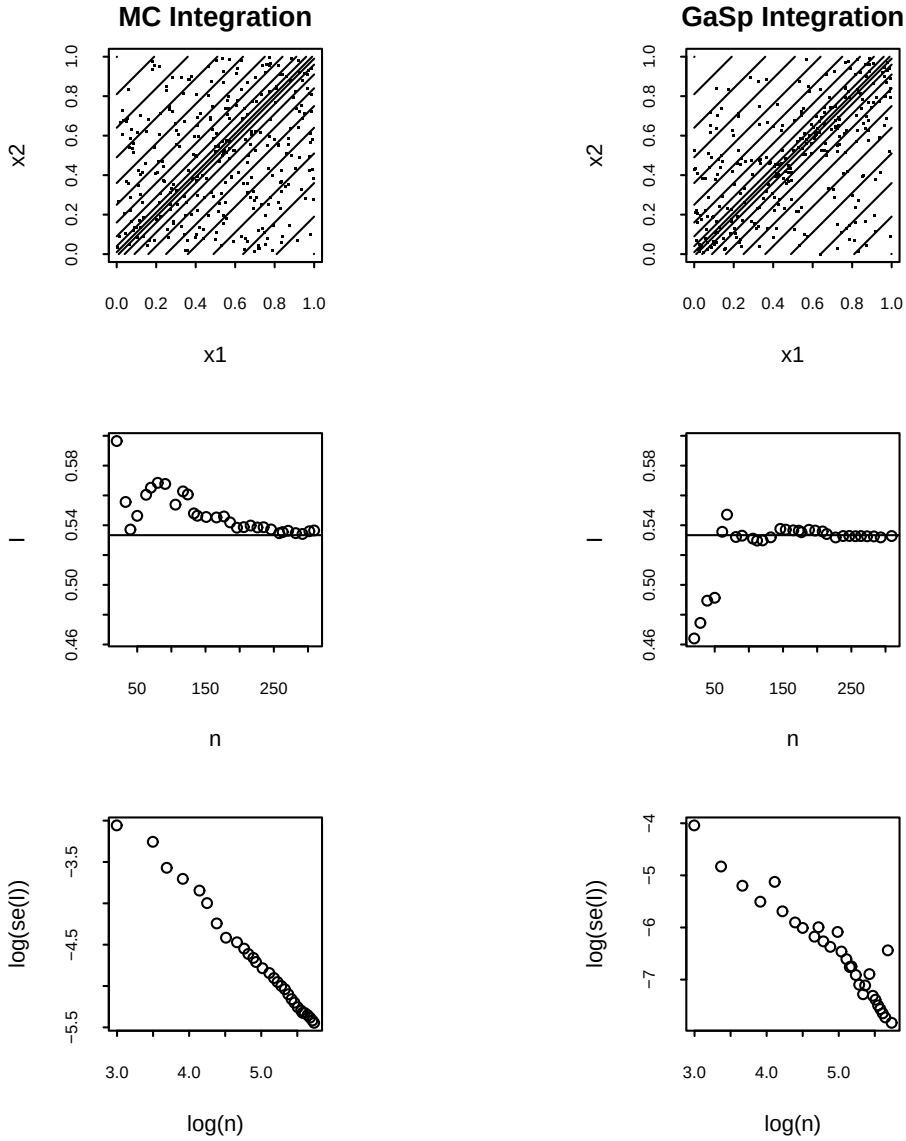


Figure 4.6: ASSIA integration results for f_4, '$\circ$' represents estimates at an iteration in the run.

to trace and compare ASSIA-GaSP with ASSIA-MC. Results for a single run of the algorithm are given in Table 4.2 and presented graphically in Figures 4.3 – 4.6. For each figure, the first column gives ASSIA-MC results, the second column gives ASSIA-GaSP results, the first row shows contours of function and points allocated by the algorithm, the second row consists of plots of $\bar{\hat{g}}_{\{m\}}$ against n_{total} per iteration with the solid line at the true value, the third row consists of plots of $\log(\sqrt{V_{\{m\}}})$ versus $\log(n_{total})$ per iteration.

From plots in Figures 4.3 – 4.6, ASSIA works as it is supposed to, it locates and concentrates sampling in 'trouble regions' when they exist. ASSIA-GaSP estimates have lower absolute errors compared to ASSIA-MC estimates. It seems that the performance of ASSIA-GaSP depends on the behavior of the integrand, the two well behaved functions f_1 and f_2 have very accurate ASSIA-GaSP estimates, this level of accuracy is not evident in f_3 and f_4. From the third rows of Figures 4.3 – 4.6, the error reduction is more consistent with ASSIA-MC, there are observable 'jumps' in the errors with ASSIA-GaSP. The reason for this is that GaSP variances are more sensitive to sample sizes, regions with small n_i tend to have large GaSP variances and consequently tend to be chosen for sub-division. This is why ASSIA-GaSP ends up with larger n_{total}. This variance sensitivity in GaSP is a good feature in a way, as sample sizes are more uniform in the sub-regions and there is a potential for the algorithm to revisit sub-regions which might have previously been overlooked.

Summary results for four runs of the algorithm are given in Table 4.3, which helps determine how ASSIA results vary with different runs. Table 4.3 indicates that ASSIA-GaSP estimates for the integral of f_2 have much lower variation with different runs than the ASSIA-MC estimates. Results in the table also reinforce the sensitivity

Function	ASSIA-MC Estimates			ASSIA-GaSP Estimates		
	n_{total}	$\bar{\hat{\bar{g}}}$	$\lvert\bar{g} - \bar{\hat{\bar{g}}}\rvert$	n_{total}	$\bar{\hat{\bar{g}}}$	$\lvert\bar{g} - \bar{\hat{\bar{g}}}\rvert$
f_1	316	-0.018413	0.018413	317	-0.000013	0.000013
f_2	309	0.560270	0.002524	336	0.557726	0.000020
f_3	316	1.623920	0.021011	321	1.634126	0.010805
f_4	309	0.536499	0.003166	309	0.532772	0.000561

Table 4.2: Estimates using a single run of ASSIA.

Function	ASSIA-MC			ASSIA-GaSP		
	n_{total}	$\bar{\hat{\bar{g}}}$	$\lvert\bar{g} - \bar{\hat{\bar{g}}}\rvert$	n_{total}	$\bar{\hat{\bar{g}}}$	$\lvert\bar{g} - \bar{\hat{\bar{g}}}\rvert$
f_1	314	0.006848	0.006848	332	0.000167	0.000167
	306	-0.012812	0.012812	333	-0.000006	0.000006
	320	-0.008385	0.008385	328	-0.000137	0.000137
	328	-0.002443	0.002443	346	-0.000258	0.000258
f_2	309	0.559477	0.001731	334	0.557717	0.000029
	310	0.5590853	0.001339	348	0.557747	0.000001
	314	0.5592934	0.001547	338	0.557746	0.000000
	307	0.555751	0.001995	339	0.557746	0.000000
f_3	306	1.600367	0.044564	333	1.641825	0.003106
	325	1.632575	0.012356	330	1.638313	0.006618
	313	1.626849	0.018082	326	1.6427300	0.002201
	315	1.621515	0.023416	311	1.641690	0.003241
f_4	317	0.536168	0.002835	315	0.534863	0.001530
	312	0.533312	0.000021	314	0.533837	0.000504
	315	0.545870	0.012537	315	0.531419	0.001914
	302	0.540607	0.007274	318	0.533854	0.000251

Table 4.3: ASSIA results based on four runs.

of GaSP variances, ASSIA-GaSP apportions more points that ASSIA-MC for the same number of iterations.

4.4.3 Posterior Inference in Computer Experiments

To illustrate the use of ASSIA-GaSP integration in Bayesian analysis in computer experiments, we applied plain GaSP integration and ASSIA-GaSP to obtain the first posterior moments of the log transformed correlation parameters (PTJP), using the simulated data sets analyzed in Chapter 2 and presented in Chapter 1.

Recall that in PTJP, we adopt the parametrization $\theta_k^* = \log(\theta_k)$, and the integral

to be evaluated is

$$\bar{g} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} cM(\theta_1^*, \theta_2^*)\sqrt{|\mathbf{B}_2^*|}|\mathbf{R}|^{-1/2}(\hat{\sigma}^2)^{-\frac{21}{2}}\partial\theta_1^*\partial\theta_2^*. \tag{4.13}$$

The functions

$$M(\theta_1^*, \theta_2^*) = \begin{cases} \theta_1^* & \text{to find } E(\theta_1^*), \\ \theta_2^* & \text{to find } E(\theta_2^*). \end{cases}$$

The matrix $\mathbf{B}_2^*$ is the information matrix with respect to the log-transformed parameters and c is the constant of proportionality. Plots of the posterior density were given in Figure 2.3.

Based on preliminary graphical analysis, we truncated the integration space to the values given in Table 4.4.

	Truncation interval	
	θ_1^*	θ_2^*
2D1	(-1.5, 0.25)	(-1.5, 0.25)
2D2	(-0.6, 0.75)	(-0.25, 1.25)
2D3	(1.75, 4.5)	(0.75, 3.25)

Table 4.4: Parameter space truncation.

We set the parameters in ASSIA to

$$\begin{aligned} n_0 &= 20, \\ n_{top} &= 20, \\ \mathcal{W} &= 80 \text{ iterations and } \sqrt{V_{\{m\}}} > 0.0001 \text{ for 1st moments.} \end{aligned}$$

For plain GaSP integration, we obtained the design using uniform random samples with $n = 60$. The size was chosen based on the reasonably good estimates on a similar problem in Section 3.4.2.

To obtain the constant of proportionality c in ASSIA-GaSP integration, we let $M(\theta_1^*, \theta_2^*) = 1$, and ran the algorithm for 100 iterations. To avoid dealing with ratio estimates, we assumed the resulting value to be the true value for c. To obtain the constant of proportionality in GaSP integration, we used plain Monte Carlo integration with 10000 evaluations and also assumed that this estimate had no error associated with it.

Data Set	Moment	MCMC (MHA) Estimate	GaSP Estimate	ASSIA-GaSP Estimate n_{total}
2D1	$E(\theta_1^*)$	-0.6302	-0.6009	-0.6349 812
	$E(\theta_2^*)$	-0.7442	-0.7474	-0.7454 835
2D2	$E(\theta_1^*)$	0.0438	0.0450	0.0476 584
	$E(\theta_2^*)$	0.3608	0.3557	0.3622 701
2D3	$E(\theta_1^*)$	2.7513	2.9499	2.7742 847
	$E(\theta_2^*)$	1.8102	1.8807	1.8092 843

Table 4.5: Estimated moments of simulated data sets

The results are presented in Table 4.5 for a single run of GaSP and ASSIA-GaSP integration. There is no way of finding out the true posterior moments of the correlation parameters, we can gain a fair assessment of ASSIA-GaSP estimates by comparing them to the MCMC estimates obtained in Chapter 2. Table 4.5 shows that ASSIA-GaSP estimates are almost equivalent to MCMC estimates. Considering the number of rejections in the Metropolis Hasting algorithm as well as the number of burn ins allowed, ASSIA-GaSP estimates involve fewer evaluations of the posterior

Data Set	Parameter	Variance	CI
2D1	θ_1^*	0.04359	(-1.0441, -0.2257)
	θ_2^*	0.04382	(-1.1557, -0.3351)
2D2	θ_1^*	0.03186	(-0.3022 , 0.3975)
	θ_2^*	0.02767	(0.0362 , 0.6882)
2D3	θ_1^*	0.08089	(2.2167 , 3.3316)
	θ_2^*	0.09217	(1.2141 , 2.4042)

Table 4.6: Posterior variances and marginal confidence intervals calculated using ASSIA-GaSP integration

density. Consequently, ASSIA-GaSP runs much faster than the Metropolis Hasting Algorithm; for instance, the ASSIA-GaSP estimate of $E(\theta_1^*)$ took approximately 230 seconds of CPU time (on a 900MHz AMD Athlon 4 Processor) while the MCMC estimate took 9.64×10^3 seconds. We also used ASSIA-GaSP to compute the posterior variances for the correlation parameters. Table 4.6 shows the estimated variances and frequentist 95% confidence intervals for (θ_1^*, θ_2^*).

4.5 Discussion

In this Chapter we introduced ASSIA to enable GaSP integration in functions with varied characteristics. There are a few areas that can be improved for higher dimensional problems. A lot of overhead goes into estimating the correlation parameters when using GaSP integration within sub-regions, particularly when deciding which direction to split. For example for one iteration with $D = 2$ we need to estimate the correlation parameters 4 times, for n^* iterations with $D = D^*$ we need to this

$2 \times D^* \times n^*$ times. Estimating correlation parameters will be a computational burden if we run ASSIA for many iterations. The algorithm also splits sub-regions in half, we can make the splits more flexible to increase sub-regions for a set work level. The next chapter has more details on modifications made to ASSIA as well as example problems in higher dimensions.

Chapter 5

Further Applications with ASSIA-GaSP Integration

In this chapter, we demonstrate the use of ASSIA-GaSP for higher dimension integration and "strategic sampling". Samples are strategic in that they conform to the function as ASSIA-GaSP sampling is more intense where the changes in a function are rapid. Sampling or design is important in computer experiments in building informative prediction models. In Section 5.4, we will show by example how ASSIA-GaSP sampling can be used as a preliminary tool in exploring a function's structure. We first introduce modifications to ASSIA-GaSP with a view to decreasing computation costs. These changes are outlined in Section 5.1.

5.1 Modification to ASSIA-GaSP Integration

1. We use sample variances as given in (4.8) in Step (2) part (c), instead of GaSP variances. This eliminates the computation time needed in estimating maximum likelihood estimates of the correlation parameters in the decision stage, which means GaSP variances are only computed at the end of an iteration when $\mathcal{S}$ is updated. By looking at various trace plots of the parameter estimates we found

that this modification had no effect on the ASSIA-GaSP estimates.

2. Using the above modification, we improvised on sub-divisions of the integration domain. Instead of splitting the axis into halves, we obtain 'variance stabilizing' (**VS**) splits . Initially we divide a particular axis in half, then move the mid-point in steps of Δ until either the variance in the two subdivided regions is roughly equivalent or the number of points in a sub-region is at least two. Typically we choose $\Delta = 0.01 \times |\mathcal{X}^*|_i$, where $|\mathcal{X}^*|_i$ is the length in direction i of sub-region $\mathcal{X}^*$. An illustration in one dimension is shown in Figure 5.1 for the integration $\sin(1/(0.1 + x))$, with the same parameters that were used in Chapter 4 to generate Figure 4.1. The effect of VS splits is visible by comparing the 10 sub-region plot to that in Figure 4.1; the sizes of the sub-regions close to the minimum are less uniform using VS splits. As will be shown by examples later on, this improvisation does not necessarily improve the estimates, however it has the effect of isolating areas where the function is most varied using fewer points, which increases the overall number of splits for a set maximum number of points in the work level. The example in Section 5.3 is a good illustration of this feature.

   The modified algorithm is as follows:

   Modified ASSIA-GaSP Algorithm

   For a given integrand $g(\mathbf{x})$, region of integration $\mathcal{X}$, and maximum work, $\mathcal{W}$:

   (a) Compute a global estimate for the integrand $\hat{\bar{g}}$, and its variance estimate using an initial sample size n_0. Typically we use the rule of thumb, $n_0 =$

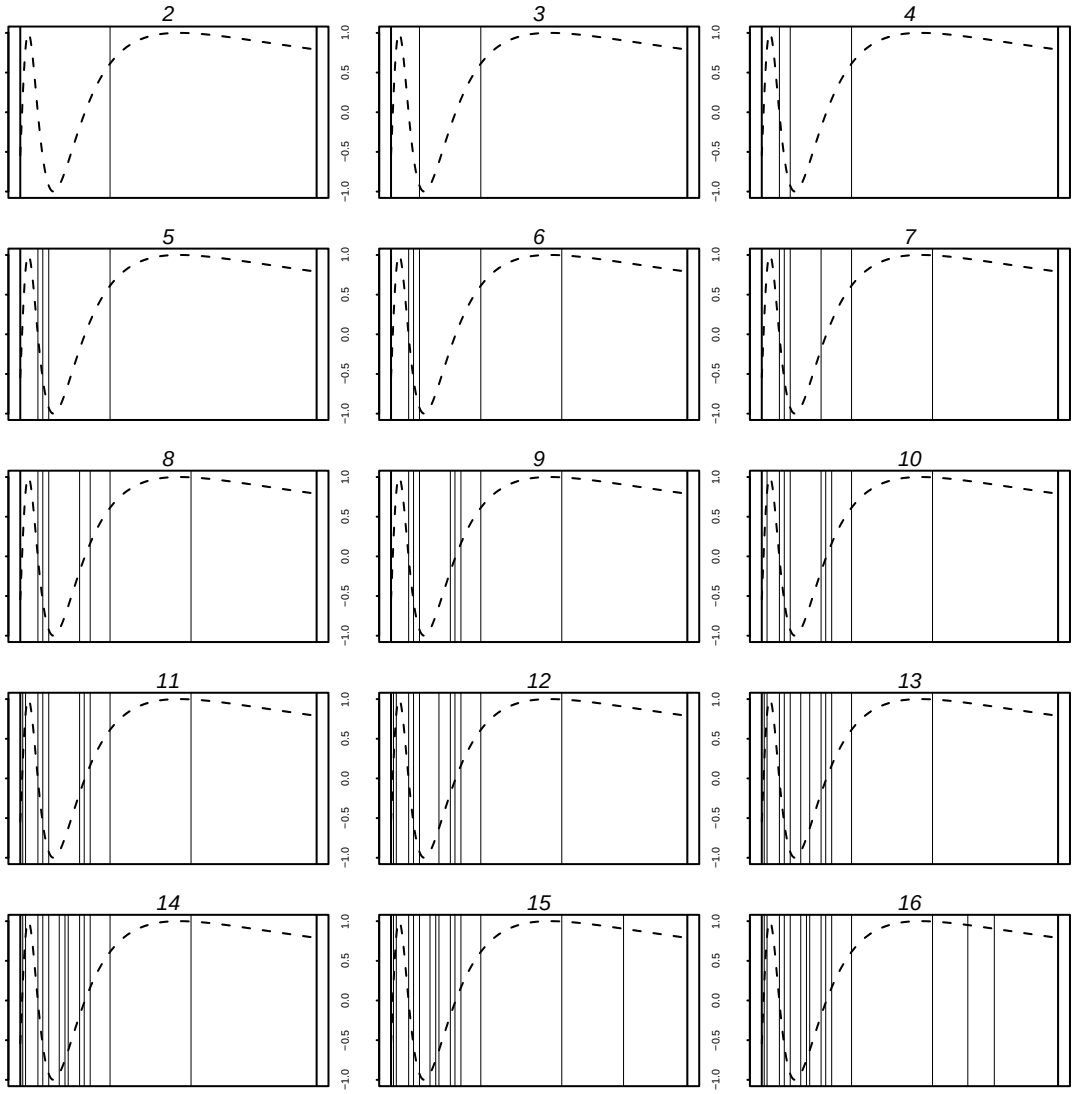


Figure 5.1: ASSIA-GaSP VS splits for $\sin(1/(0.1+x))$. The headings give the number of sub-regions.

$10 \times D$, where D is the dimension of the integration space. Initialize

$$\mathcal{S} = \{(\mathcal{X}_1 = \mathcal{X}, \hat{\bar{g}}_1 = \hat{\bar{g}}, w_1^2 V_1 = \prod_{i=1}^{D}(b_i - a_i)^2 \mathrm{var}(\hat{\bar{g}}), n_1 = n_0)\},$$

(b) **while** (work $< \mathcal{W}$)

i. Pick a sub-region to sub-divide and its associated points and remove its information from the list $\mathcal{S}$. This is the region with the largest variance, $\mathcal{X}^* = \mathcal{X}_l$ where $\max(w_1^2 V_1, \ldots, w_m^2 V_m) = w_l^2 V_l$ with $1 \leq l \leq m$. If $m = 1$, then $\mathcal{X}^* = \mathcal{X}$.

ii. Verify that the number of points in $\mathcal{X}^*$ is at least n_{top}. If not increase the points to n_{top} by random sampling.

iii. Determine the axis to sub-divide. To divide $\mathcal{X}^*$ we sequentially sub-divide each co-ordinate axis into half. For each split across a co-ordinate axis j, $j = 1, \ldots, D$, move the mid-point in steps of Δ until either the variance in the two subdivided regions is roughly equivalent or the number of points in a sub-region is at least two. The weighted estimated sample variance obtained, V_1^{*j} and $V_2^{*j}$are used to estimate the sub-region variance,

$$V^{*j} = V_1^{*j} + V_2^{*j}.$$

The co-ordinate axis to be subdivided $\bar{j}$, is such that $\min(V^{*1}, \ldots, V^{*D}) = V^{*\bar{j}}$. Divide across this axis to get $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$

iv. Obtain estimates $\hat{\bar{g}}_1^*$ and $\hat{\bar{g}}_2^*$ by applying GaSP integration to points already in $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$.

v. Update $\mathcal{S}$, Xdata, Ydata by inserting the integration information of the new sub-regions $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$.

(c) **end(while)**

5.2 Example 1: Five Dimension Integration

This example is taken from Evans and Swartz [8] where they compute the orthant probability,

$$\bar{g} = \int_0^\infty \dots \int_0^\infty \frac{1}{\sqrt{2\pi|\Sigma|}} \exp(-0.5\mathbf{x}^T\Sigma^{-1}\mathbf{x}) dx_1, \dots, dx_6, \tag{5.1}$$

where $\Sigma^{-1/2} = \text{diag}(0, 1, 2, 3, 4, 5) + J$, and J is a 6×6 matrix of ones. This is a computation of the multivariate normal probability $\mathbf{X} \geq 0$, where $\mathbf{X} \sim N_6(\mathbf{0}, \Sigma)$. The exact value as given by Evans and Swartz correct to 10 decimal places is 0.166625×10^{-4}.

We make use of a sequence of parametrizations which exploit the features of the integrand to transform the domain of integration onto the hypercube. These were first presented by Genz [11]. We define a new variable $\mathbf{u} = C^{-1}\mathbf{x}$ where C is the lower triangular Cholesky factor of Σ. This alters the integration problem to one of obtaining probabilities of independent Normal random variables $u_1, \dots, u_6$. The second transformation is $v_i = \Phi(u_i)$, where Φ is the $N(0, 1)$ distribution. The interval of integration for v_i is $(a_i, 1)$, where $a_1 = 0.5$, $a_i = -\sum_{j=1}^{i-1}(C_{ij}\Phi^{-1}(v_j))/C_{ii}$ for $i > 1$ and C_{ij} is the $(i, j)^{th}$ element of C. The final transformation is given by $w_i = (v_i - a_i)/(1 - a_i)$ for all i, which maps the integral's domain onto the hypercube $[0, 1]^6$. The final integral has the form,

$$\bar{g} = \int_0^1 \dots \int_0^1 \prod_{i=1}^{6}(1 - b_i) dw_6 dw_5 dw_4 dw_3 dw_2 dw_1, \tag{5.2}$$

where

$$\begin{aligned} b_1 &= 0.5, \\ b_i &= \Phi\left(\sum_{j=1}^{i-1}\left(C_{ij}\Phi^{-1}\left((1-b_j)w_j+b_j\right)\right)/C_{ii}\right). \end{aligned}$$

Though the integral seems more complicated after the transformation, the dimension of the domain is reduced to five due to the constant term in the inner most integral. We than thus rewrite (5.2) as,

$$\bar{g} = 0.5\int_0^1 \ldots \int_0^1 \prod_{i=1}^{5}(1-b_i)dw_5dw_4dw_3dw_2dw_1. \tag{5.3}$$

We applied ASSIA-GaSP integration using both equal and VS splits with the following parameters:

$$\begin{aligned} n_0 &= 50, \\ n_{top} &= 50, \\ \mathcal{W} &= 5000 \text{ points}. \end{aligned}$$

We chose the work level as 5000 points after an initial run of ASSIA-GaSP with work level set to 10000 points; the estimates achieved some stability between 4000 and 5000 points.

The average value for 10 runs using equal splits was $\bar{\hat{g}} = 1.60848 \times 10^{-5}$ with a standard error of 2.31908×10^{-7}. The average for 10 runs using VS splits was $\bar{\hat{g}} = 1.44872 \times 10^{-5}$ with a standard error of 4.53106×10^{-7}. Full results are given in Tables B.1 and B.2 in the appendix. On average, a single run took approximately 6 minutes with VS splits and 5 minutes for equal splits.

Using 10^6 evaluations and $N_6(\mathbf{0}, \Sigma)$ as the importance function, Evans and Swartz estimated $\bar{g}$ as 1.90000×10^{-5} with an absolute coefficient of variation equal to 0.229,

with 10^8 computations they obtained a more precise estimate of 1.63000×10^{-5} which took about 100 minutes of CPU time. Running ASSIA-GaSP with equal splits with fewer evaluations results in better estimates – 9982 evaluations yielded a value of 1.65668×10^{-5}, 50000 evaluations yielded a value of 1.66363×10^{-5}. The computation time needed to obtain these results is considerably longer which highlights an important point in choice of methods, the importance sampling method is faster than ASSIA-GaSP though needs considerably more evaluations. Ultimately, the choice of integration method would depend on other factors such as the computation cost of the integrand, or computing power available.

5.3 Example 2: Ten Dimension Integration

The integral to be evaluated is:

$$\begin{aligned} \bar{g} &= \int_0^1 \dots \int_0^1 \prod_{i=1}^{10} f_i(p_i) dp_1 \dots dp_d, \\ &= 1, \end{aligned} \tag{5.4}$$

where

$$f_i(p_i) = \frac{\Gamma((i+1)/2)}{\Gamma(i/2)\sqrt{\pi i}} p_i(1-p_i) \left(1 + \frac{1}{i}\left[\log\left(\frac{p_i}{1-p_i}\right)\right]\right)^{-(i+1)/2} \tag{5.5}$$

The functions f_i is a transformed Student's-t density with i degrees of freedom which has been mapped to [0,1] using the logit transformation,

$$p = 1/(1 + \exp(-t)).$$

The integrand presents an interesting problem as the characteristics of f_i change with different values of i as shown in Figure 5.2. When $i = 1$ the function has

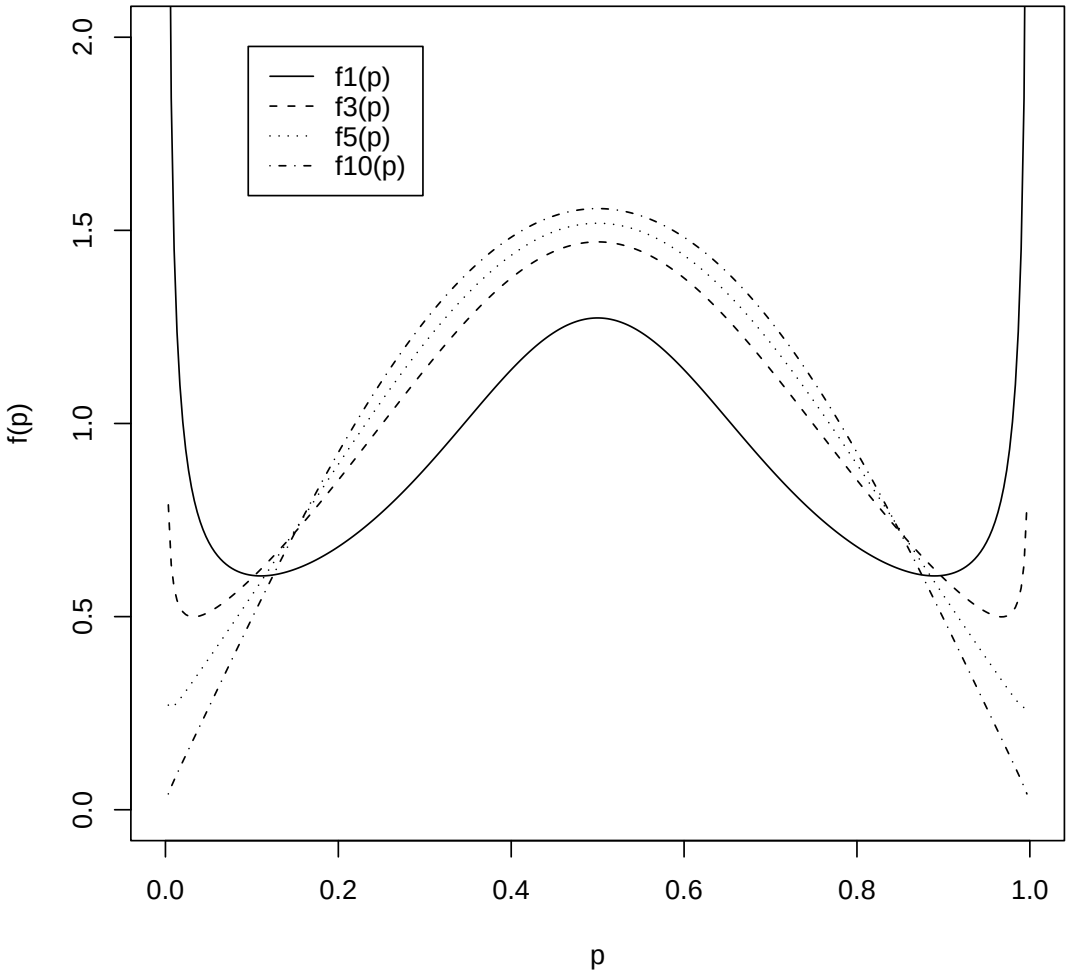


Figure 5.2: Plot of $f_i(p_i)$ $i = 1, 3, 5, 10$.

asymptotes at both end points and a global maximum at 0.5, as i increases the maximum is less pronounced (or less sharper) and f_i has a more parabolic shape. We ran ASSIA-GaSP 20 times using the equal and VS split methods, each time starting with a different random design with the following parameters:

$$\begin{aligned} n_0 &= 100, \\ n_{top} &= 100, \\ \mathcal{W} &= 1000 \text{ points.} \end{aligned}$$

The complete set of results are given in Appendix B. The average $\bar{\hat{\bar{g}}}$ value without 'VS splits' was 0.967452 with a standard error equal to 0.0706515. The average $\bar{\hat{\bar{g}}}$ value was 1.008044 with a standard error equal to 0.1045328. Within the work level specified, on average, equal splits in the algorithm used 18 sub-divisions while VS splits used 19 sub-divisions.

5.4 Strategic Sampling

The approach exploits ASSIA-GaSP's varied sampling to get a more representative sample of a function. We can learn about the function by observing the pattern of sampling. This can be done for example by using a kernel density estimator on the sampled points. The following two sub-sections give applications of this technique.

5.4.1 Two Dimension Strategic Sampling

We chose two functions with different behavior on $[0,1] \times [0,1]$. These were given by equations in (4.9) and (4.12), namely,

$$\begin{aligned} f_1 &= \sin(2\pi x_1) + \sin(2\pi x_1 + \pi x_2), \\ f_4 &= \sqrt{|x_1 - x_2|}. \end{aligned}$$

Adopting the two modifications in Section 5.1, we used the following parameters in ASSIA-GaSP:

$$\begin{aligned} n_0 &= 20, \\ n_{top} &= 20, \\ \mathcal{W} &= 500 \text{ points}. \end{aligned}$$

We then used a two-dimensional kernel density estimator on the resulting points from Xdata. Kernel density estimation was enabled by the function '**kde2d**' from the library **MASS** in the **R** software, which uses an axis-aligned bivariate normal kernel, evaluated on a square grid.

Figures 5.4 and 5.3 are contour plots of the estimated densities. Figure 5.3 shows that sampling is slightly concentrated away from the edges of the domain. Figure 5.4 shows that sampling is concentrated on the line $x_1 = x_2$. The sampling obtained by ASSIA-GaSP indicate that changes in f_1 are everywhere quite gradual compared to f_4.

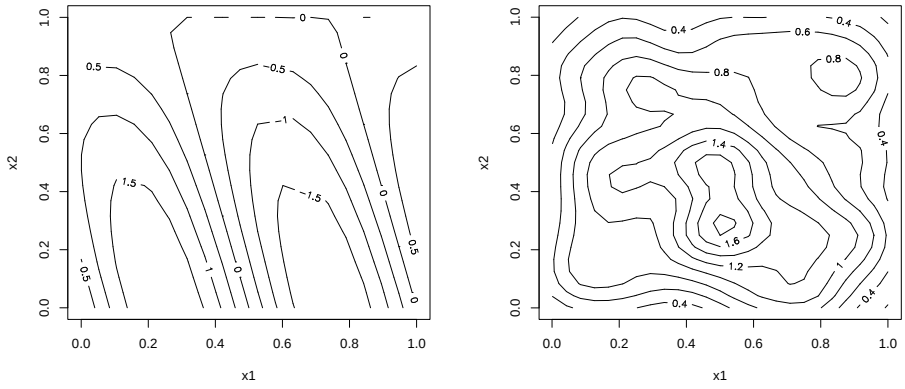


Figure 5.3: ASSIA-GaSP visualization of $\sin(2\pi x_1) + \sin(2\pi x_1 + \pi x_2)$.

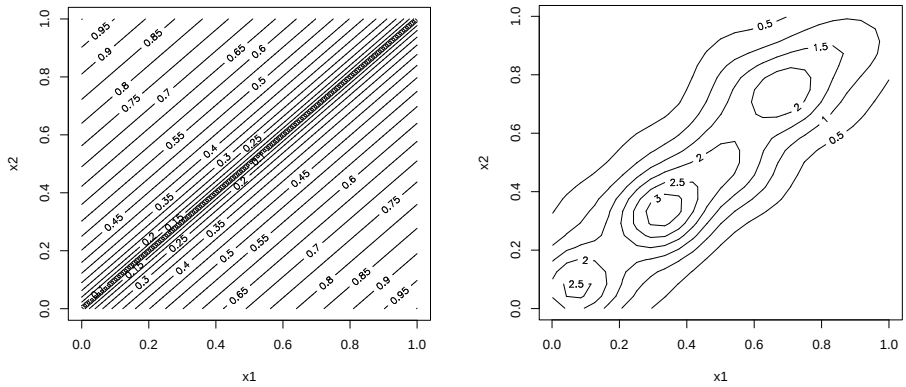


Figure 5.4: ASSIA-GaSP visualization of $\sqrt{|x_1 - x_2|}$.

5.4.2 Three Dimension Strategic Sampling

The function of interest is a three dimension version of (5.4),

$$g(p_1, p_2, p_3) = \prod_{i=1}^{3} f_i(p_i) \quad \text{where } 0 < p_i < 1, \tag{5.6}$$

and f_i are given by (5.5). Cross sectional plots of the function are given by Figures 5.5, 5.6 and 5.7. The cross sectional topography of the function is the same, the difference in the plots is brought about by the range of the plotted values. In Figure 5.5 the minimum and maximum values for p_1 and p_2 are 0.01 and 0.99 respectively; in Figures 5.6 and 5.7 these are 0.0001 and 0.9999. In all plots we used a grid of 100×100 points. The difficulty in visualizing the function is due to the asymptotes at the end points.

Assuming no prior knowledge of the function, we ran ASSIA-GaSP using both equal and VS splits separately. The parameters to both methods were:

$$\begin{aligned} n_0 &= 30, \\ n_{top} &= 30, \\ \mathcal{W} &= 5000 \text{ points.} \end{aligned}$$

We then used the **kde2d** function for two dimension Kernel density smoothing on sample points from pairs of axis. Contour plots on the smoothed data are presented in Figures 5.8 and 5.9. Sampling using both methods is concentrated in the center and the edges of the domain. The most amount of splitting occurs on the p_3 axis, this is evident from the well defined contours in the plot on the top left hand side of Figures 5.8 and 5.9. This is consistent with the function, Figure 5.2 demonstrates that $g(p_1, p_2, p_3)$ has the most pronounced or sharper peak when it is projected on

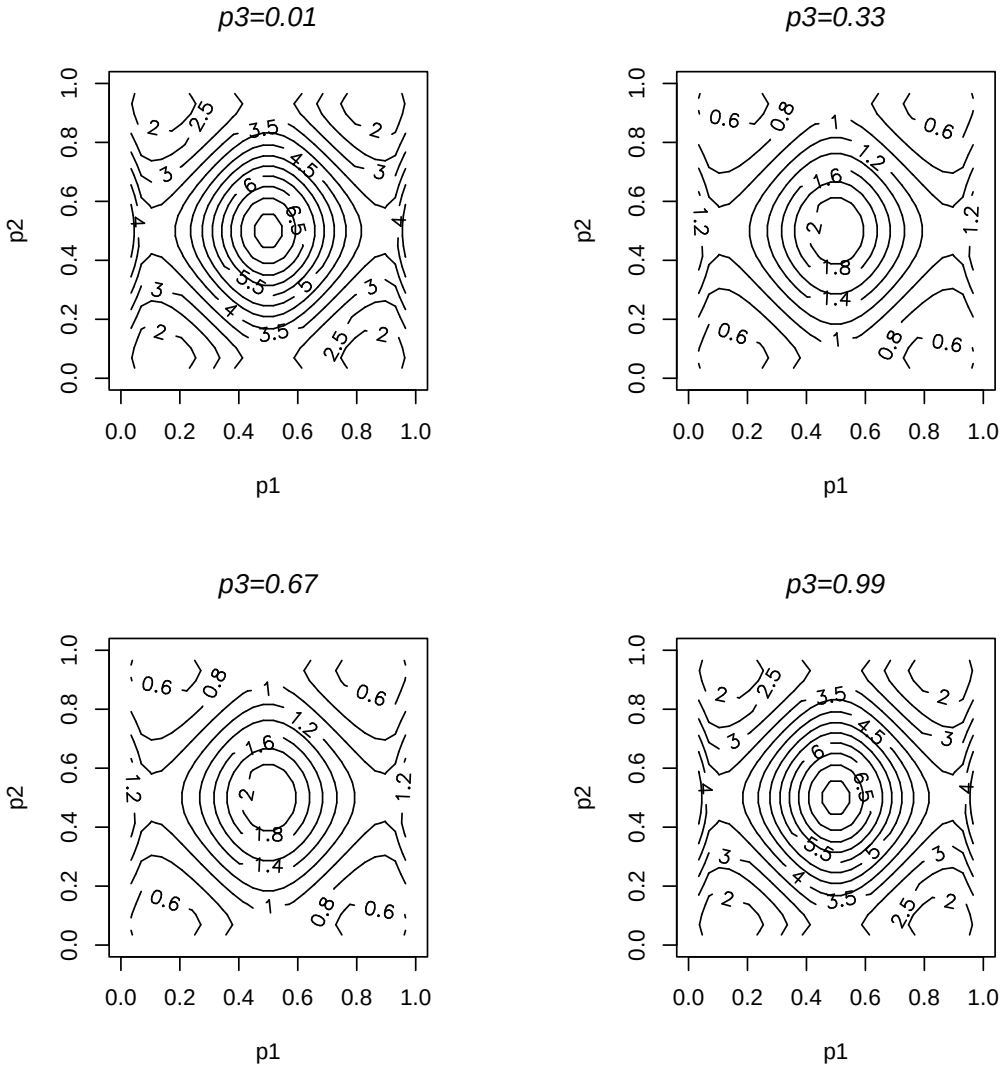


Figure 5.5: Projection of $g(p_1, p_2, p_3)$ on p_3.

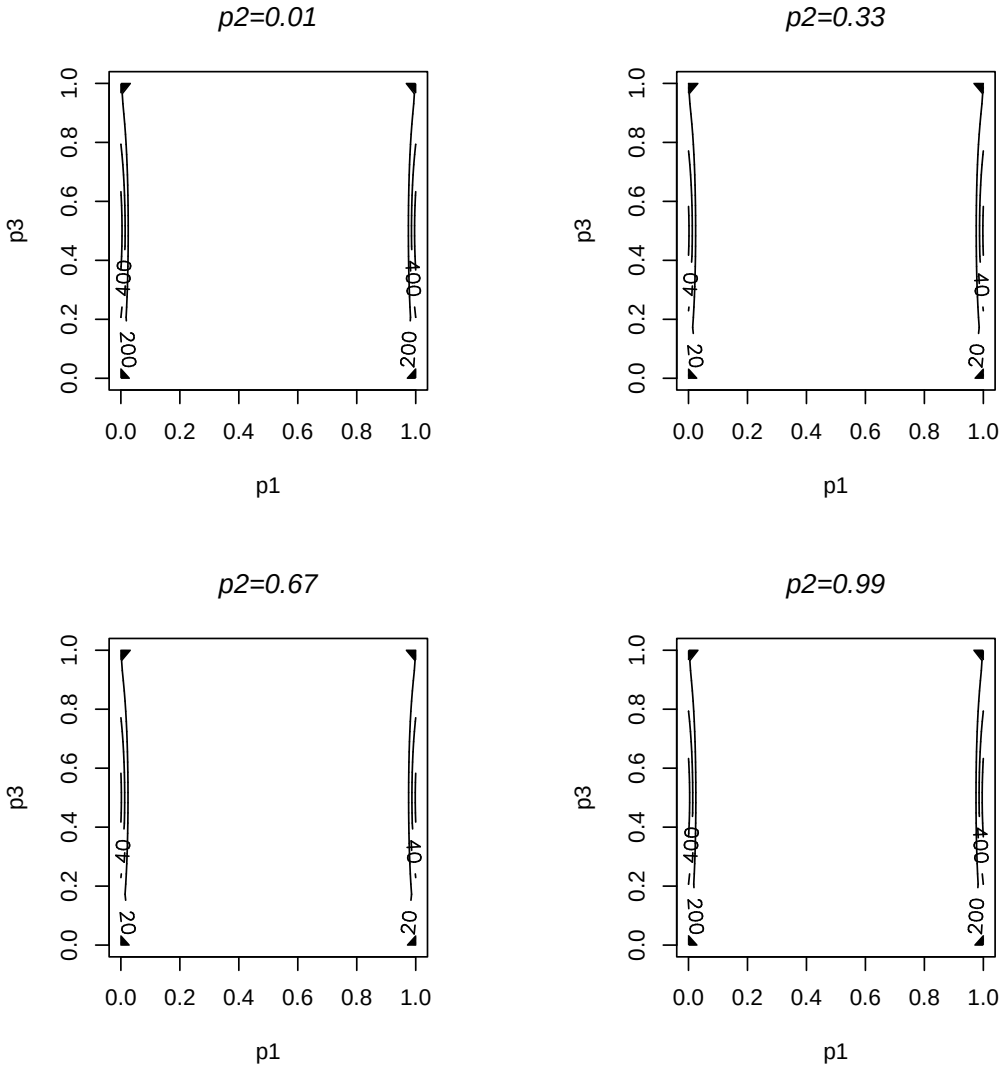


Figure 5.6: Projection of $g(p_1, p_2, p_3)$ on p_2.

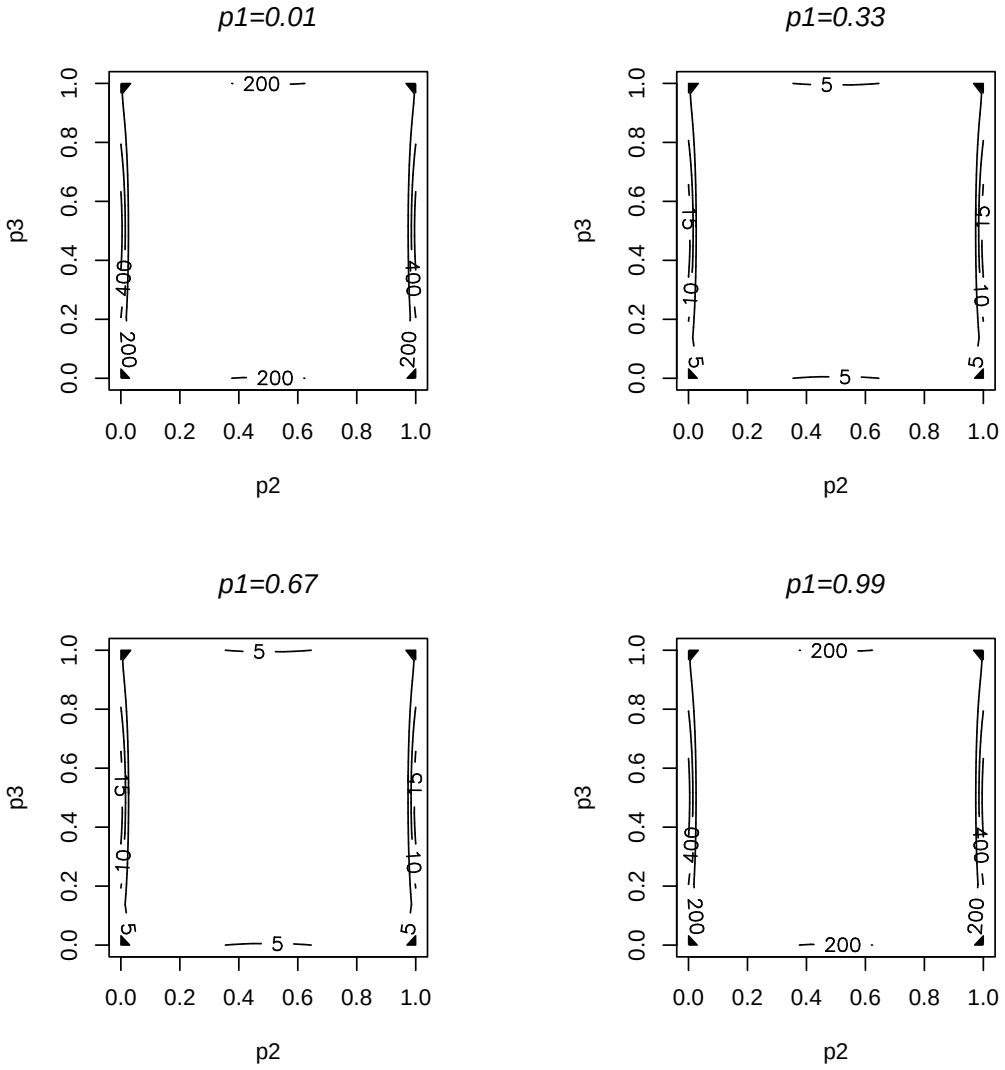


Figure 5.7: Projection of $g(p_1, p_2, p_3)$ on p_1.

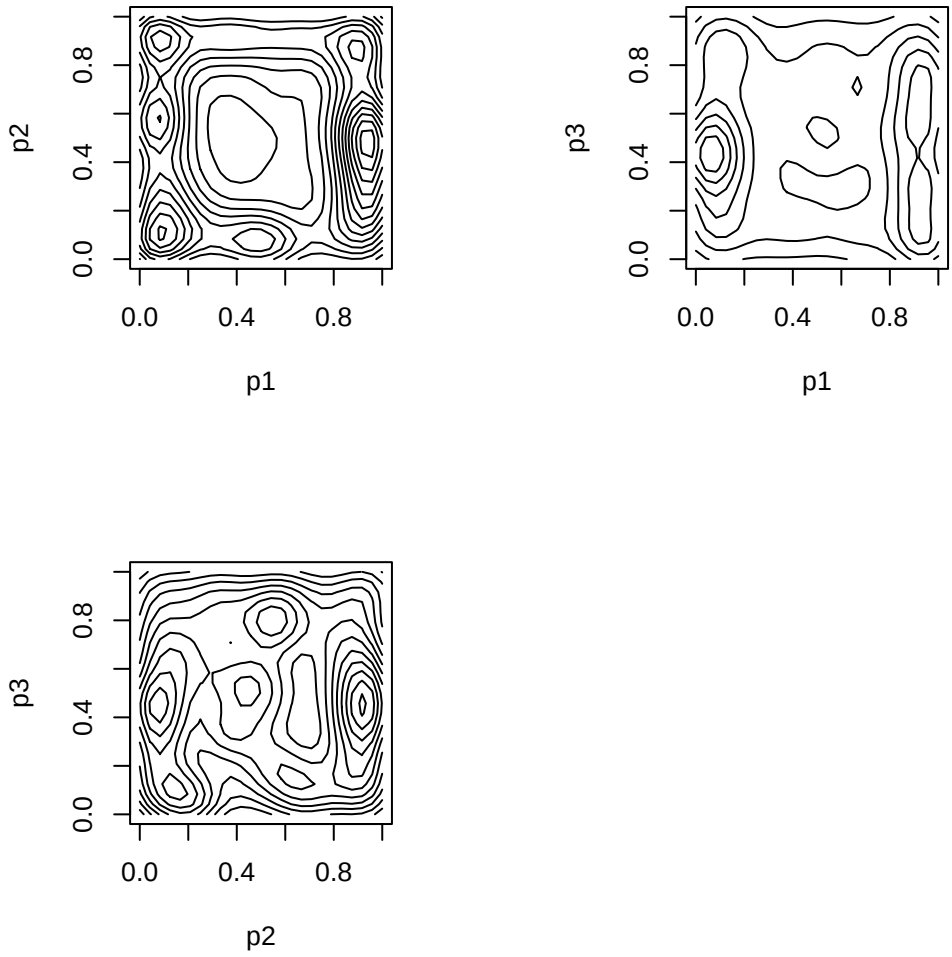


Figure 5.8: Cross-sectional contour plots for smoothed points using equal splits on ASSIA-GaSP.

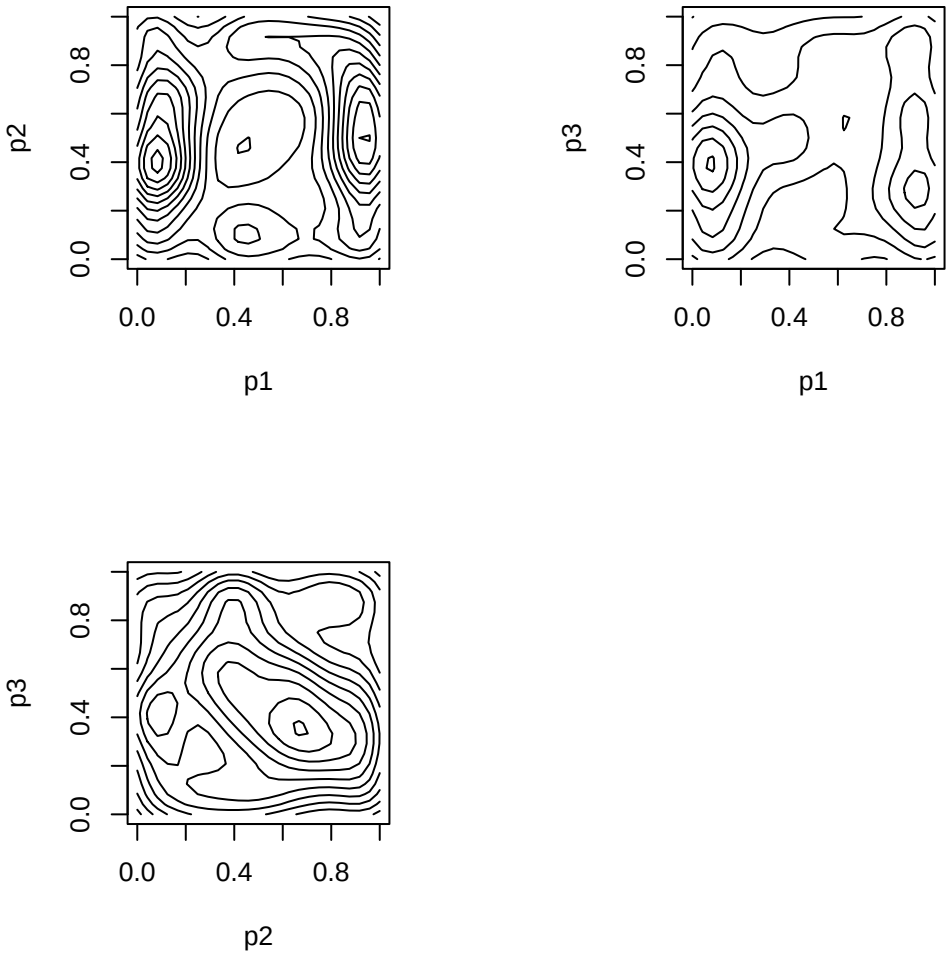


Figure 5.9: Cross-sectional contour plots for smoothed points using VS splits on ASSIA-GaSP.

p_3. Equal splits provide a better sample than VS splits, this is due to the symmetry of the function.

5.5 Discussion

The modifications made to ASSIA-GaSP in this chapter enabled high dimension integration with less computation costs. The windows used in the VS splits might be considered too small – with n points in a sub-region, we need at most $n-1$ comparisons. However, considering that we are comparing sample variances, which are cheap computation wise; extra comparisons are a minor inconvenience.

ASSIA-GaSP samples could be a basis for kriging models within sub-sections. These models would be better at approximating the true function, particular non-smooth functions. The samples can also help achieve other goals in computer experiments. For example, 'crude' optimization can be carried out by comparing the value of the integral in sub-section and matching it up with points in Xdata and Ydata, as well as observing the pattern of clustering in the samples as carried out in Section 5.4.

Chapter 6

Conclusion and Recommendations

This thesis presented a Bayesian implementation in computer experiments. In Chapter 2 we looked at prior formulation and proposed two approximation techniques based on the log scaled posterior of the correlation parameters. The first technique involved approximating the Jeffreys prior with a diffuse prior, which enabled MCMC simulations to run faster. The second technique approximated the posterior with the Normal density, which avoids the use of MCMC simulations altogether. It should be mentioned that the approximation of the Jeffreys prior is limited to the range of correlation parameters. While the approximation techniques we formulated are based on the Gaussian correlation function, work by Berger, De Oliveira and Sanso [2], suggests that these same techniques with some limitations, may be applicable to the Matern, Spherical and Rational Quadratic correlation function. The application of the approximation approach demonstrated in Chapter 2 involved two input variables, however this approach can be extended to more than two input variables. For cases where skewness persists, an alternative suggestion is to use mixed distribution for example Geweke split-t densities [13]. The work in Chapter 2 did not cover a key element of computer experiments - finding an optimal design. An optimal design is

important in building an informative predictive model. Nonetheless, we do address the issue of design by using the ASSIA method.

In Chapters 3, 4, and 5, we dwelt on the general subject of integration. We analyzed GaSP integration in Chapter 3 and found that GaSP integration in low dimensions provided estimates with lower absolute errors compared to MC integration. GaSP errors were considerably lower than MC errors, there was evidence of GaSP integration having faster convergence rates (in terms of error bounds) compared to MC integration. Though this phenomena was not studied rigorously, it presents an interesting area for future work. Limitations to GaSP integration were solved by the adaptive algorithm presented in Chapter 4. While ASSIA was used with the aim of integration, we demonstrated that it can also be used as a tool for sampling, the resulting samples can be used to gain information about a function's structure. Further recommendations and directions for future work are:

1. Investigation into the relationship between GaSP integration errors and MC errors – The results in Chapter 3 indicate a relationship between GaSP integration errors and MC errors, primarily when uniform random samples are used in the design.

2. Investigation into the sensitivity of GaSP integration estimates to the correlation parameters – The effect of correlation functions as well as the effect of the estimates of the correlation parameters on GaSP integration estimates is a possible area for investigation. The simple example in Section 3.3 presented the possibility of fixed designs having GaSP estimates with minimal variation for certain ranges of correlation parameters.

3. Investigation into the uniformity or variability of functions – It is important before application that one gain some idea of the uniformity or variability of the function. Maximum likelihood estimates of the correlation parameters from an initial set of sub-regions can be used. This can be done by running ASSIA-GaSP for a small number of iterations.

4. Simplifications in higher dimension integrations – In higher dimensions integration, some dimension reduction can also be achieved by transformation or finding dependencies between variables.

5. Improvement to sub-region sampling – Sampling may be improved by using more equi-spaced designs in the sub-regions. Derek Bingham, in a conversation, suggested that the initial design be space filling and adaptive Latin Hypercube Sampling be used to top-up sub-regions.

6. Effective increase of points in sub-regions – The number of points increased to a sub-region could be done more adaptively, for example one could use an adaptive Neymann allocation taking into account the complexity of the integrand.

7. Investigation into a Bayesian approach on GaSP integration – This would involve specifying priors on the correlation parameters, the guidelines set by Berger, De Oliveira and Sanso [2] can be used. A Bayesian approach would incorporate parameter uncertainty in the estimate of the integral, thereby addressing the issue of parameter sensitivity.

Appendix A

Chapter 2 Proofs

A.1 Jeffreys Prior

For any correlation function the likelihood and log likelihood equations are:

$$\begin{aligned} L &\propto \frac{1}{(\sigma^2)^{\frac{n}{2}}\sqrt{|\mathbf{R}|}}\exp(\frac{-1}{2\sigma^2}(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)), \qquad (A.1)\\ l &= constant - \frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log|\mathbf{R}| - \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu). \end{aligned}$$

For a one dimension or one input problem, the information matrix is

$$\mathbf{I}(\mu,\sigma^2,\theta) = \begin{bmatrix} \frac{1}{\sigma^2}\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1} & 0 & 0 \\ 0 & \frac{1}{2}\log_{\theta^2}|\mathbf{R}| + \frac{1}{2}\mathrm{tr}(\mathbf{R}_{\theta^2}^{-1}\mathbf{R}) & -\frac{1}{2\sigma^2}\mathrm{tr}(\mathbf{R}_{\theta}^{-1}\mathbf{R}) \\ 0 & -\frac{1}{2\sigma^2}\mathrm{tr}(\mathbf{R}_{\theta}^{-1}\mathbf{R}) & \frac{n}{2(\sigma^2)^2} \end{bmatrix}. \qquad (A.2)$$

The notation for the partial derivatives above is as follows:

$$\begin{aligned} \mathbf{R}_{\theta^s} &= \frac{\partial^s}{\partial\theta^s}\mathbf{R},\\ \mathbf{R}_{\theta^s}^{-1} &= \frac{\partial^s}{\partial\theta^s}\mathbf{R}^{-1},\\ \log_{\theta^s}|\mathbf{R}| &= \frac{\partial^s}{\partial\theta^s}\log|\mathbf{R}| \text{ for } s=1,2. \end{aligned}$$

Expanding the (2,2) entry in the above matrix and using the fact that

$$\log_\theta |\mathbf{R}| = \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_\theta),$$

and

$$\mathbf{R}_\theta^{-1} = -\mathbf{R}^{-1}\mathbf{R}_\theta\mathbf{R}^{-1},$$

then

$$\log_{\theta^2} |\mathbf{R}| = -\mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_\theta\mathbf{R}^{-1}\mathbf{R}_\theta) + \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta^2}).$$

Similarly

$$\mathrm{tr}(\mathbf{R}_{\theta^2}^{-1}\mathbf{R}) = 2\mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_\theta\mathbf{R}^{-1}\mathbf{R}_\theta) - \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta^2}).$$

Adding the above two equations gives a simplification for the (2,2) entry in the information matrix. The rest of the entries in the matrix can be simplified in a similar manner. The result is

$$\mathbf{I}(\mu, \sigma^2, \theta) = \begin{bmatrix} \frac{1}{\sigma^2}\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1} & 0 & 0 \\ 0 & \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_\theta)^2) & -\frac{1}{2\sigma^2}\mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_\theta) \\ 0 & -\frac{1}{2\sigma^2}\mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_\theta) & \frac{n}{2(\sigma^2)^2} \end{bmatrix}. \tag{A.3}$$

The matrix in (A.3) reflects the independence between the mean and correlation and variance parameters. The Jeffreys prior is proportional to the square root of the determinant of the information matrix,

$$\mathrm{pr}(\mu, \sigma^2, \theta) \propto \frac{\sqrt{(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})|\mathbf{B}_1|}}{(\sigma^2)^{3/2}}, \tag{A.4}$$

with

$$|\mathbf{B}_1| \propto \left| \begin{pmatrix} \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_\theta)^2) & \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_\theta) \\ \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_\theta) & n \end{pmatrix} \right|. \tag{A.5}$$

In the notation given in (A.4), we can view the Jeffreys prior as as specifying a uniform prior on the mean parameter μ. Using the same working as the the one input case,

the Jeffreys prior for the two dimension case is given as:

$$\mathrm{pr}(\mu, \sigma^2, \theta_1, \theta_2) \propto \frac{\sqrt{(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})|\mathbf{B}_2|}}{(\sigma^2)^{3/2}},$$

where

$$|\mathbf{B}_2| \propto \left| \begin{pmatrix} \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_{\theta_1})^2) & \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_{\theta_2})(\mathbf{R}^{-1}\mathbf{R}_{\theta_1})) & \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta_1}) \\ \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_{\theta_2})(\mathbf{R}^{-1}\mathbf{R}_{\theta_1})) & \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_{\theta_2})^2) & \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta_2}) \\ \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta_1}) & \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta_2}) & n \end{pmatrix} \right|. \quad \text{(A.6)}$$

The notation for the partial derivatives follows from that in (A.3) with

$$\mathbf{R}_{\theta_i} = \frac{\partial}{\partial\theta_i}\mathbf{R} \qquad \text{for } i = 1, 2.$$

A.2 Proof of Lemma 2.3.1

In the one input, and using Chen's results [3] for the equispaced design:

$$\mathbf{R}^{-1} = \sum_{k=1}^{n} \frac{\bar{\mathbf{w}}^k(\bar{\mathbf{w}}^k)^T}{1 - Q_{k-1}} \quad \text{(A.7)}$$

where $\bar{\mathbf{w}}^k$ is given by (2.11). The first term in (A.4)

$$\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1} = \sum_{k=1}^{n} \frac{(\mathbf{1}^T\bar{\mathbf{w}}^k)^2}{1 - Q_{k-1}}.$$

By using the commutative property of the trace of a matrix, we formulate an expression for $|\mathbf{B}_1|$ as follows;

$$|\mathbf{B}_1| = n\mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_\theta)^2) - (\mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_\theta))^2. \quad \text{(A.8)}$$

Considering terms in (A.8)

$$\begin{aligned} \mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_\theta) &= \sum_{k=1}^{n} \frac{(\bar{\mathbf{w}}^k)^T\mathbf{R}_\theta\bar{\mathbf{w}}^k}{1 - Q_{k-1}} \\ &= \sum_{k=1}^{n} \frac{C_{kk}}{1 - Q_{k-1}}, \end{aligned} \quad \text{(A.9)}$$

where

$$C_{jk} = (\bar{\mathbf{w}}^j)^T \mathbf{R}_\theta \bar{\mathbf{w}}^k.$$

$$\begin{aligned}
\mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_\theta)^2) &= \mathrm{tr}\left[\left(\sum_{k=1}^{n} \frac{\bar{\mathbf{w}}^k(\bar{\mathbf{w}}^k)^T\mathbf{R}_\theta}{1-Q_{k-1}}\right)\left(\sum_{j=1}^{n} \frac{\bar{\mathbf{w}}^j(\bar{\mathbf{w}}^j)^T\mathbf{R}_\theta}{1-Q_{j-1}}\right)\right] \\
&= \sum_{k=1}^{n}\sum_{j=1}^{n} \frac{\mathrm{tr}\left(\bar{\mathbf{w}}^k(\bar{\mathbf{w}}^k)^T\mathbf{R}_\theta\bar{\mathbf{w}}^j(\bar{\mathbf{w}}^j)^T\mathbf{R}_\theta\right)}{(1-Q_{j-1})(1-Q_{k-1})} \\
&= \sum_{k=1}^{n}\sum_{j=1}^{n} \frac{\mathrm{tr}((\bar{\mathbf{w}}^j)^T\mathbf{R}_\theta\bar{\mathbf{w}}^k(\bar{\mathbf{w}}^k)^T\mathbf{R}_\theta\bar{\mathbf{w}}^j)}{(1-Q_{j-1})(1-Q_{k-1})} \\
&= \sum_{k=1}^{n}\sum_{j=1}^{n} \frac{((\bar{\mathbf{w}}^j)^T\mathbf{R}_\theta\bar{\mathbf{w}}^k)((\bar{\mathbf{w}}^k)^T\mathbf{R}_\theta\bar{\mathbf{w}}^j)}{(1-Q_{j-1})(1-Q_{k-1})} \\
&= \frac{C_{jk}^2}{(1-Q_{j-1})(1-Q_{k-1})}. \qquad \text{(A.10)}
\end{aligned}$$

Substituting (A.10) and the square of (A.9) into (A.8) yields the results of Lemma 2.3.1.

A.3 Verification of the Jeffreys Prior Approximation for $n = 1, \ldots, 12$

The Jeffreys prior for $d = 1$ is

$$\mathrm{pr}(\theta) \propto \sqrt{(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})|\mathbf{B}_1|}$$

Using Chen's results

$$\begin{aligned}
\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1} &= \sum_{k=1}^{n} \frac{(\mathbf{1}^T\bar{\mathbf{w}}^k)^2}{(1-Q_{k-1})} \\
&= \sum_{k=1}^{n}\prod_{i=1}^{k-1}\left(\frac{1-\rho^{2i}}{(1-(-\rho)^i)^2}\right),
\end{aligned}$$

where

$$\prod_{i=1}^{k-1}\frac{(1-\rho^{2i})}{(1-(-\rho)^{i})^2}=\begin{cases}\prod_{i=1}^{m}\frac{(1+\rho^{2i})(1-\rho^{2i-1})}{(1-\rho^{2i})(1+\rho^{2i-1})} & k=2m\ (k\text{ even}),\\ \frac{1-\rho^{k}}{1+\rho^{k}}\prod_{i=1}^{m}\frac{(1+\rho^{2i})(1-\rho^{2i-1})}{(1-\rho^{2i})(1+\rho^{2i-1})} & k=2m+1\ (k\text{ odd}).\end{cases} \tag{A.11}$$

Assuming $\rho_{\theta\to 0^+}=1-\kappa\theta+o(1)$, where κ is some constant, the term

$$\begin{aligned}\prod_{i=1}^{m}\frac{(1+\rho^{2i})(1-\rho^{2i-1})}{(1-\rho^{2i})(1+\rho^{2i-1})} &= \prod_{i=1}^{m}\frac{(1-\rho^{4i})(1-\rho^{2i-1})^2}{(1-\rho^{2i})^2(1-\rho^{4i-2})}\\ &= \prod_{i=1}^{m}\frac{(4i\kappa\theta+o(\theta))((2i-1)\kappa\theta+o(\theta))^2}{(2i\kappa\theta+o(\theta))^2((4i-2)\kappa\theta+o(\theta))}\\ &= \prod_{i=1}^{m}\frac{\theta^3(4i\kappa+o(1))((2i-1)\kappa+o(1))^2}{\theta^3(2i\kappa+o(1))^2((4i-2)\kappa+o(1))}\\ &= O(1). \end{aligned} \tag{A.12}$$

As $\theta\to 0^+$, in the odd terms in (A.11),

$$\begin{aligned}\frac{1-\rho^{k}}{1+\rho^{k}} &= \frac{(1-\rho^{k})^2}{1-\rho^{2k}}\\ &= \frac{\theta^2(k\kappa+o(1))}{\theta(2k\kappa+o(1))}\\ &= o(1). \end{aligned} \tag{A.13}$$

Using results from (A.12) and (A.13), terms with k even in (A.11) tend to constant values and terms with k odd vanish hence

$$\left(\sum_{k=1}^{n}\frac{(\mathbf{1}^T\bar{\mathbf{w}}^k)^2}{1-Q_{k-1}}\right)=O(1)\ \text{ as }\theta\to 0^+. \tag{A.14}$$

For the second term in the square root sign in Lemma 2.3.1, the numerator has a quadratic form and is bounded, the denominator shows that $J(\mu,\sigma^2,\theta)\to\infty$ as $\theta\to 0^+$. For cases $n=2,\ldots,12$ we work out the expression in Lemma 2.3.1 using Maple software.

When n = 2:

$$\left(\sum_{k=1}^{2}\sum_{j=1}^{2}\frac{nC_{jk}^2-C_{jj}C_{kk}}{(1-Q_{k-1})(1-Q_{j-1})}\right)=\frac{\rho^2}{(1-Q_1)^2}. \tag{A.15}$$

Assuming $\rho_{\theta\to 0^+} = 1 - \kappa\theta + o(1)$ and using (2.13), then

$$\begin{aligned}\frac{\rho^2}{(1-Q_1)^2} &= \frac{1-2\kappa\theta+o(\theta)}{(2\kappa\theta^2+o(\theta))^2(2-2\kappa\theta+o(\theta))^2} \\ &= \frac{1-2\kappa\theta+o(\theta)}{\theta^2(2\kappa+o(1))^2(2-2\kappa\theta+o(\theta))^2} \\ &= O\left(\frac{1}{\theta^2}\right). \end{aligned} \tag{A.16}$$

When $\mathbf{n = 3, .., 12}$:

For $n = 3$,

$$\left(\sum_{k=1}^{3}\sum_{j=1}^{3}\frac{nC_{jk}^2 - C_{jj}C_{kk}}{(1-Q_{k-1})(1-Q_{j-1})}\right) = \rho^2\Big(\sum_{l=0}^{S_3}\alpha_{3,l}\rho^{2l}\Big)\left[\frac{(1-Q_1)}{(1-Q_2)}\right]^2. \tag{A.17}$$

For $n = 2m, m = 2, \ldots, 5$,

$$\left(\sum_{k=1}^{n}\sum_{j=1}^{n}\frac{nC_{jk}^2 - C_{jj}C_{kk}}{(1-Q_{k-1})(1-Q_{j-1})}\right) = \rho^2\Big(\sum_{l=0}^{S_n}\alpha_{n,l}\rho^{2l}\Big)\left[(1-Q_1)(1-Q_{m-2})\frac{(1-Q_{m-1})}{(1-Q_{2m-1})}\right]^2. \tag{A.18}$$

For $n = 2m-1, m = 3, \ldots, 6$; the left hand side of (A.18) is

$$= \begin{cases} \rho^2\big(\sum_{l=0}^{S_n}\alpha_{n,l}\rho^{2l}\big)\left[(1-Q_1)(1-Q_{m-3})\frac{(1-Q_{m-1})}{(1-Q_{2(m-1)})}\right]^2 & \text{if } m-1 \text{ even}, \\ \rho^2\big(\sum_{l=0}^{S_n}\alpha_{n,l}\rho^{2l}\big)\left[\frac{(1-Q_2)}{(1-Q_1)}(1-Q_{m-3})\frac{(1-Q_{m-1})}{(1-Q_{2(m-1)})}\right]^2 & \text{if } m-1 \text{ odd}. \end{cases} \tag{A.19}$$

The $\alpha_{n,l}$ are positive integers and the terms

$$(S_3, \ldots, S_{11}) = (6, 14, 22, 38, 46, 70, 86, 110, 126).$$

Using (2.13)

$$\prod_{i=1}^{t}(1-Q_{\beta_i}) = (1-\rho^2)^{\beta_1+\ldots+\beta_t}\prod_{i=1}^{t}f_{\beta_i}(\rho) \tag{A.20}$$

where

$$f_r(\rho) = \prod_{p=0}^{r-1}\sum_{r=0}^{p-1}\rho^{2r},$$

and as $\theta \to 0^+,\ f_r(\rho) = O(1)$. Using these results in (A.17), (A.18) and (A.19), and the fact that the term before the square brackets tend to a constant as $\theta \to 0^+$; as $\theta \to 0^+$

$$\begin{aligned}\left(\sum_{k=1}^{n}\sum_{j=1}^{n}\frac{nC_{jk}^2 - C_{jj}C_{kk}}{(1-Q_{k-1})(1-Q_{j-1})}\right) &= \frac{O(1)}{(2\kappa\theta + o(\theta))^2} \\ &= \frac{O(1)}{\theta^2(2\kappa + o(1))^2} \\ &= O\left(\frac{1}{\theta^2}\right) \qquad \text{(A.21)}\end{aligned}$$

Multiplying the square root of (A.14) to the square roots of (A.16) and (A.21) yields

$$\mathrm{pr}(\theta) = O\left(\frac{1}{\theta}\right). \qquad \text{(A.22)}$$

A.4 Proof of Lemma 2.3.2

Assuming that the sampled points are obtained from the grid formed by $x_1 = (1/n, 2/n, \ldots, 1)$ and $x_2 = (1/n, 2/n, \ldots, 1)$. The sample is therefore of size n^2. Due to the product correlation rule, we can express the correlation matrix

$$\mathbf{R} = \mathbf{R}_1 \otimes \mathbf{R}_2,$$

and

$$\mathbf{R}^{-1} = \mathbf{R}_1^{-1} \otimes \mathbf{R}_2^{-1},$$

whose elements are,

$$\begin{aligned}\mathbf{R}_{1_{i,j}} &= \exp(-\theta_1(i-j)^2/n), \\ \mathbf{R}_{2_{i,j}} &= \exp(-\theta_2(i-j)^2/n).\end{aligned}$$

The term (the subscript on the $\mathbf{1}$ vectors give the size),

$$\mathbf{1}_{n^2}^T\mathbf{R}^{-1}\mathbf{1}_{n^2} = (\mathbf{1}_n^T\mathbf{R}_1^{-1}\mathbf{1}_n)(\mathbf{1}_n^T\mathbf{R}_2^{-1}\mathbf{1}_n).$$

The determinant of the matrix given in (A.6) can be simplified by the fact that

$$\begin{aligned}\mathbf{R}_{\theta_1} &= \mathbf{R}_1' \otimes \mathbf{R}_2, \\ \mathbf{R}_{\theta_2} &= \mathbf{R}_1 \otimes \mathbf{R}_2'.\end{aligned}$$

For example

$$\begin{aligned}\mathrm{tr}(\mathbf{R}^{-1}\mathbf{R}_{\theta_1}) &= \mathrm{tr}((\mathbf{R}_1^{-1} \otimes \mathbf{R}_2^{-1})(\mathbf{R}_1' \otimes \mathbf{R}_2)), \\ &= \mathrm{tr}(\mathbf{R}_1^{-1}\mathbf{R}_1' \otimes I) \text{ I is the identity matrix of size } n \times n, \\ &= n\mathrm{tr}(\mathbf{R}_1^{-1}\mathbf{R}_1'); \\ \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_{\theta_1})^2) &= \mathrm{tr}((\mathbf{R}_1^{-1}\mathbf{R}_1' \otimes I)(\mathbf{R}_1^{-1}\mathbf{R}_1' \otimes I)), \\ &= \mathrm{tr}((\mathbf{R}_1^{-1}\mathbf{R}_1')^2 \otimes I), \\ &= n\mathrm{tr}((\mathbf{R}_1^{-1}\mathbf{R}_1')^2); \\ \mathrm{tr}((\mathbf{R}^{-1}\mathbf{R}_{\theta_2})(\mathbf{R}^{-1}\mathbf{R}_{\theta_1})) &= \mathrm{tr}((\mathbf{R}_1^{-1}\mathbf{R}_1' \otimes I)(I \otimes \mathbf{R}_2^{-1}\mathbf{R}_2')), \\ &= \mathrm{tr}(\mathbf{R}_1^{-1}\mathbf{R}_1' \otimes \mathbf{R}_2^{-1}\mathbf{R}_2'), \\ &= \mathrm{tr}(\mathbf{R}_1^{-1}\mathbf{R}_1')\mathrm{tr}(\mathbf{R}_2^{-1}\mathbf{R}_2').\end{aligned}$$

If we let $\mathbf{K} = \mathbf{R}_1^{-1}\mathbf{R}_1'$ and $\mathbf{L} = \mathbf{R}_2^{-1}\mathbf{R}_2'$, then using the above results in (A.6),

$$\mathbf{B}_2 = \begin{bmatrix} n\mathrm{tr}(\mathbf{K}^2) & \mathrm{tr}(\mathbf{K})\mathrm{tr}(\mathbf{L}) & n\mathrm{tr}(\mathbf{K}) \\ \mathrm{tr}(\mathbf{K})\mathrm{tr}(\mathbf{L}) & n\mathrm{tr}(\mathbf{L}^2) & n\mathrm{tr}(\mathbf{L}) \\ n\mathrm{tr}(\mathbf{K}) & n\mathrm{tr}(\mathbf{L}) & n^2 \end{bmatrix}, \tag{A.23}$$

and

$$\det(\mathbf{B}_2) = n^2[n\mathrm{tr}(\mathbf{L}^2) - (\mathrm{tr}(\mathbf{L}))^2][n\mathrm{tr}(\mathbf{K}^2) - (\mathrm{tr}(\mathbf{K}))^2].$$

Matching this up, with (A.8) results in the Lemma.

Appendix B

Chapter 5 Results

B.1 Example 1 Results

Table B.1 and B.2 give the integration results for the Evans and Swartz integration problem given in 5.1 in Chapter 5. ASSIA-GaSP was run ten times with the following parameters:

$$\begin{aligned} n_0 &= 60, \\ n_{top} &= 60, \\ \mathcal{W} &= 5000 \text{ points.} \end{aligned}$$

B.2 Example 2 Results

Table B.3 and B.4 give the integration results for the ten dimension function given in 5.4 in Chapter 5. ASSIA-GaSP was run twenty times with the following parameters:

$$\begin{aligned} n_0 &= 100, \\ n_{top} &= 100, \\ \mathcal{W} &= 1000 \text{ points.} \end{aligned}$$

$\hat{\bar{g}}$	n_{total}
1.59881E-05	4998
1.61114E-05	4990
1.65398E-05	4979
1.40115E-05	4992
1.65299E-05	4991
1.58063E-05	4985
1.63954E-05	4983
1.64788E-05	4981
1.64676E-05	4979
1.65189E-05	4989

Table B.1: Example 1 – ASSIA-GaSP equal splits results

$\hat{\bar{g}}$	n_{total}
1.54858E-05	4956
1.57354E-05	4960
1.47144E-05	4993
1.34215E-05	4985
1.63039E-05	4996
1.53213E-05	4995
1.45729E-05	4988
1.12807E-05	4965
1.39465E-05	4958
1.40901E-05	4983

Table B.2: Example 1 – ASSIA-GaSP VS splits results

$\bar{\hat{\bar{g}}}$	n_{total}	Splits
0.987592	957	18
0.962551	965	18
0.931489	955	18
1.104676	951	18
0.955446	964	18
0.998076	996	19
0.874613	963	18
1.054870	967	18
0.925419	968	18
0.836372	978	18
0.922752	961	18
0.896432	963	18
0.951598	975	18
1.006268	954	18
1.014820	970	18
0.891344	947	17
0.922136	979	18
0.983736	981	18
1.060298	986	18
1.068549	964	18

Table B.3: Example 2 ASSIA-GaSP equal splits results

$\hat{\bar{g}}$	n_{total}	Splits
0.928862	974	19
0.972346	954	18
0.906371	952	17
0.967908	953	18
0.814890	937	17
0.875118	983	18
1.107279	968	19
0.990829	967	19
1.086943	996	19
0.964148	951	18
0.941743	969	18
1.033072	966	19
1.258541	986	20
1.034896	981	19
1.137172	955	19
1.107009	979	19
0.993488	960	18
0.982118	988	19
0.933884	963	18
1.124254	966	19

Table B.4: Example 2 ASSIA-GaSP VS splits results

Appendix C

R Programs

C.1 Metropolis Hasting Algorithm

```
met.has<-function(start=theta.mode,iter,I.inv=I.inv,index,transf=c(2,2)){

p<-c(2,2); n<-nrow(X1);
if(index==1)
func<-function(x) lhood(theta=x,p=p,n=n,transf=transf) else {func<-function(x)
post(th=x,p=p,transf=transf,n2=n)}

theta.mat<-matrix(nrow=iter,ncol=2)
theta.mat[1,]<-start

for (i in 2:iter){
## Propose new state
theta.prop<-theta.mat[i-1,]+rmultnorm(1,c(0,0),I.inv)
u.star<-min(func(theta.prop)/func(theta.mat[i-1,]),1)
u<-runif(1,0,1)
if(u <= u.star) {theta.mat[i,]<-theta.prop} else {theta.mat[i,]<-theta.mat[i-1,]}
}
theta.mat
}
```

C.2 ASSIA-GaSP

```
ASSIA.GS<-function(d,lc,uc,n0,n.top,max.pts,err.lim=0.001){
# n0, initial no of points
# d, dimension, func defined globally
# lc, lower co-ords; uc, upper co-ords;
```

```
# n.top points to top box with
# max.pts limit of points
# err.lim minimum error



xval<-genU.matrix(lc,uc,n0,d)
yval<-apply(xval,1,func)
counter<-1

res.int<-0
res.sigma<-0
number<-0
results<-decide1.func(lc,uc,xval,yval)
no.points<-n0

## Iterative step --
while( (no.points<max.pts)){
if (counter==1){
xout<-matrix(c(results$out1$mat0,counter),nrow=1)
xdata<-cr.xdata(counter,results$out1$xmat)
ydata<-cr.xdata(counter,as.matrix(results$out1$ymat))
counter<-2
xout<-rbind(xout,c(results$out2$mat0,counter))
xdata<-rbind(xdata,cr.xdata(counter, results$out2$xmat))
ydata<-rbind(ydata,cr.xdata(counter,as.matrix(results$out2$ymat)))
}
max.se<-max(xout[,(2*d+2)])

err.prev<-sum(xout[,2*d+1])/sqrt(sum((xout[,2*d+2])^2))

locator<-xout[xout[,(2*d+2)]==max.se,(2*d+3)]

#pick out box's points
xmat<-xdata[xdata[,1]==locator, 2:(d+1)]
if(is.vector(xmat)) xmat<-t(as.matrix(xmat))
ymat<-ydata[ydata[,1]==locator,2]

#Specify position of box
lc<-xout[locator==xout[,(2*d+3)],1:d]
uc<-xout[locator==xout[,(2*d+3)],(d+1):(2*d)]

#Check if to topup xmat
```

```
if (is.vector(xmat)) xmat<-t(as.matrix(xmat))
inc.pts<-0
if (nrow(xmat) < n.top) {
        inc.pts<-n.top-nrow(xmat)
        xmat.top<-topupU.func(xmat,lc,uc,n.top)
                if(is.vector(xmat.top)) xmat.top<-t(as.matrix(xmat.top))
        ymat.top<-apply(xmat.top,1,func)
        xmat<-rbind(xmat,xmat.top)
        ymat<-c(ymat,ymat.top)
        }

#remove points from list
xdata<-xdata[!(xdata[,1]==locator),]
ydata<-ydata[!(ydata[,1]==locator),]

#remove location from list
xout<-xout[!(xout[,(2*d+3)]==locator),]

#integrate
results<-decide1.func(lc,uc,xmat,ymat)

#store integration results
xout<-rbind(xout,c(results$out1$mat0,locator),c(results$out2$mat0,counter+1))
xdata<-rbind(xdata,cr.xdata(locator, results$out1$xmat),
cr.xdata(counter+1, results$out2$xmat) )
ydata<-rbind(ydata,cr.xdata(locator,as.matrix(results$out1$ymat)),
cr.xdata(counter+1,as.matrix(results$out2$ymat)))

#increase counter
counter<-nrow(xout)

res.int<-c(res.int,sum(xout[,2*d+1]))
err.cur<-sum(xout[,2*d+1])/sqrt(sum((xout[,2*d+2])^2))
err.diff<-abs(err.cur-err.prev)
res.sigma<-c(res.sigma,sqrt(sum((xout[,2*d+2])^2)))
number<-c(number,nrow(xdata))
no.points<-nrow(xdata)
}

val1<-cbind(res.int,res.sigma,number)[-1,]
list(val1=val1,xdata=xdata,xout=xout,ydata=ydata)
}
```

```
decide1.func<-function(lc,uc,xval,yval){
d<-length(lc)
splits<-opt.split(lc,uc,xval,yval)
val1<-(xval[,splits[6]]>=splits[1]) & (xval[,splits[6]]<=splits[2])

xval1<-xval[val1,]
if(is.vector(xval1)) xval1<-t(as.matrix(xval1))
yval1<-yval[val1]
xval2<-xval[!val1,]
if(is.vector(xval2)) xval2<-t(as.matrix(xval2))
yval2<-yval[!val1]
if (d==1) {xval1<-matrix(xval1,ncol=1); xval2<-matrix(xval2,ncol=1)}

lc1<-lc
tmp<-uc
tmp[splits[6]]<-splits[2]
uc1<-tmp
tmp<-lc
tmp[splits[6]]<-splits[2]
lc2<-tmp
uc2<-uc
###GASP integration
write.mx(xval1,'x.mat')
write.mx(yval1,'y.mat')
writedesc.mx(lc1,uc1)
system('C:/Gasp/gasp.exe C:/Gasp/fit.gsp')
av1<-prod(uc1-lc1)*read.mx('summary.mat')$Average
se1<-prod(uc1-lc1)*read.mx('summary.mat')$SE.Average

write.mx(xval2,'x.mat')
write.mx(yval2,'y.mat')
writedesc.mx(lc2,uc2)
system('C:/Gasp/gasp.exe C:/Gasp/fit.gsp')
av2<-prod(uc2-lc2)*read.mx('summary.mat')$Average
se2<-prod(uc2-lc2)*read.mx('summary.mat')$SE.Average

out1<-list(mat0=c(lc1,uc1,av1,se1),xmat=xval1,ymat=yval1)
out2<-list(mat0=c(lc2,uc2,av2,se2),xmat=xval2,ymat=yval2)
list(out1=out1,out2=out2)
}
```

Bibliography

[1] Abt, M., Welch, W. J., (1998), Fisher Information and Maximum-likelihood estimation of Covariance Parameters in Gaussian Stochastic Processes, *Canadian Journal of Statistics*, 26:127-137

[2] Berger, J. O., De Oliveira, V., and Sanso, B., (2001), Objective Bayesian Analysis of Spatially Correlated Data, *Journal of the American Statistical Association*, 96, 1361-1374.

[3] Chen, X., (1996) Properties of Models for Computer Experiments, Ph.D. Thesis, University of Waterloo, Waterloo, Ontario,Canada.

[4] Bayarri, M. J., Berger, J. O., Higdon, A., Kennedy, M. C., Kottas A., Paulo, R., Sacks, J., Cafeo J. A., Cavendish, J. C., Lin, C. H., Tui, J., (2002), A Framework for Validation of Computer Models, NISS Technical Report Number 128

[5] Currin, C., Mitchell, T., Morris, M. and Ylvisaker, D., (1991), Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments, *Journal of the American Statistical Association*, 86: 953-963.

[6] Davis, P. J., Rabinowitz, P., (1975), Methods of Numerical Integration, *Academic Press*

[7] Diaconis, P., (1988), Bayesian Numerical Analysis, *Statistical Decision Theory and Related Topics IV* (S.S. Gupta and J. Berger, Eds.), Springer-Verlag, New York, 1:163-175

[8] Evans, M., Swartz, T., (1996) Methods for Approximating Integrals in Statistics with Special Emphasis on Bayesian Integration Problems, *Statistical Science* 10:254-272

[9] Fang, K., Wang Y., Bentler P. M., (1994) Some Applications of Number-Theoretic Methods in Statistics, *Statistical Science*

[10] Fox, B. L., (1986), Algorithm 647: Implementation and Relative Efficiency of Quasirandom Sequence Generators, *ACM Transactions on Mathematical Software* 12:362-376

[11] Genz, A., (1992), Numerical Computation of Multivariate Normal Probabilities, *Journal of Computational and Graphical Statistics*, 1:141-150.

[12] Genz, A. and Kass, R. E., (1997), Subregion-Adaptive Integration of Functions Having a Dominant Peak, *Journal of Computational and Graphical Statistics*, 6, 92-111.

[13] Geweke, J., (1989) Bayesian Inference in Econometric Models Using Monte Carlo Integration, *Econometrica*, 57: 1317-1339

[14] Halton, J. H., (1960) On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numerical Mathematics* 2:84-90

[15] Hancock, M. S. and Stein M. L., (1993), A Bayesian Analysis of Kriging, *Technometrics*, 35: 403-410.

[16] Hoel, P. G., Port, S. C., Stone, J. C., (1972), Introduction to Stochastic Processes, *Wavelnad Press, Inc., Illinois*

[17] Jeffreys, H., (1961) Theory of Probability, London: Oxford University Press.

[18] Jones, D. R., Schonlau, M. and Welch, W. J., (1998) Efficient Global Optimization of Expensive Black-Box Functions, *Journal of Global Optimization*, 13: 455-492.

[19] Kennedy, M. C. and O'Hagan, A., (1998), Predicting the Output from a Complex Computer Code when Fast Approximations are Available, Department of Mathematics, University of Nottingham, Technical Report 98-09

[20] Kennedy, M. C., O'Hagan A., (2001), Bayesian calibration of computer models (with discussion), *Journal of the Royal Statistical Society*, Series B. 63, 425-464

[21] Koehler, J. R., Owen, A., B., (1996), Computer Experiments, *Handbook of Statistics*, 13: 261-308.

[22] Le, N. D. and Zidek, J. V., (1992), Interpolation with Uncertain Spatial Covariance: A Bayesian Alternative to Kriging, *Journal of Multivariate Analysis*, 43: 351-374.

[23] McKay, M. D., Conover, W. J., Beckman, R. J., (1979), A comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, *Technometrics* 21:239-145

[24] Neal, R. M., (2003) Slice sampling, *Annals of Statistics*, 31:705-767

[25] O'Hagan, A., Bayes-Hermite Quadrature, (1991), *Journal of Statistical Planning and Inference*, 29:245-260

[26] O'Hagan, A., Bayesian Inference, (1994), Kendall's Advanced Theory of Statistics, Volume 2B.

[27] Reese, C. S., Wilson, A. G., Hamada, M., Martz, H. F., Integrated Analysis of Computer and Physical Experiments, (2004) *Technometrics*, 46:153-154

[28] Robinson, D., Atcitty, C., Comparison of Quasi and Pseudo-monte Carlo Sampling for Reliability and Uncertainity Analysis, American Institure of Aeronautics and Astronautics, Technical Report 99-1589

[29] Sacks, J., Schiller, S., B., Welch, W., J., (1989) Designs for Computer Experiments, *Technometrics*, 31:41-47

[30] Sacks, J., Welch, W., J., Workshop on Design and Analysis of Computer Experiments for Engineering, (2002) *SSC Annual Metting, Business and Industrial Statistics Section, Hamilton*

[31] Sacks, J., Welch, W., J., Mitchell, T., J., Wynn, P., (1989) Design and Analysis of Computer Experiments, *Statistical Science*, 4:409-423

[32] Schonlau, M., (1997), Computer Experiments and Global Optimization, Ph.D. Thesis, University of Waterloo, Waterloo, Ontario,Canada.

[33] Schonlau, M., Welch, W., Jones, D., (1998), Global Versus Local Search in Constrained Optimization of Computer Models, *New Developments and Applications in Experimental Design* 34:11-25

[34] Schonlau, M., Welch, W., J., Screening the Input Variables to a Computer Model Via Analysis of Variance and Visualization, Unpublished Manuscript version April 9, 2004

[35] Simpson, T. W., Lin, D. K. J., Chen W., (2001) Sampling Strategies for Comuter Experiments: Design and Analysis, *International Journal of Reliability and Application*

[36] Sobol, L. M., (1976) Uniformly Distributed Sequences with an Additional Uniform Property, *Comput. Math. Math. Phys* 16:236-242

[37] Sobol, L. M., (1979) On the Systematic Search in a Hypercube, *SIAM Journal of Numerical Analysis* 16:790-793

[38] Yong, B. L., Sacks, J., Studden, W. J., Welch W. J., (2001) Design and Analysis of Computer Experiments when the Output is Highly Correlated Over the Input Space, *The Canadian Journal of Statistics* 29