Finite Element Approximation of Minimum Generalised Cross Validation Bivariate Thin Plate Smoothing Splines

PENELOPE HANCOCK BAPPSC(HONS)
The Centre for Resource and Environmental Studies

A thesis submitted for the degree of Doctor of Philosophy of
the Australian National University
October 2002

Declaration

The work in this thesis is my own unless otherwise stated.

Hancock

Penny Hancock

Acknowledgements

Throughout the course of this project I have received help from many generous people. Thanks firstly to Michael Hutchinson, for consistently and extensively supporting me throughout this project. Through our lengthy weekly meetings, I have been able to greatly advance my understanding of and interest in computational mathematics. Without this help, the project would not have been nearly so satisfying or so successful. Thanks also to Frank deHoog, for monitoring the project and providing constructive advice throughout the last 3 years. The quality of this thesis has greatly improved as a result of this interaction.

I am also extremely grateful to Karl Nissen, whose assistance was vital in many a LaTeX saga, and Barry Croke, who has been helpful for just about anything. Thanks to John Stein for help with GIS matters. And thanks to Ole Neilson, for answering rather vague questions about mathematics in general, and to David Young, for generously providing LaTeX wisdom.

My experience at CRES has been most enjoyable thanks to the many friendly people here, particularly the bunch that hang out in the tea room. My PhD has been all the more fun thanks to my friends Ryan and Megan and people from the ANU Mountaineering Club. Thanks also to my family for your support.

Finally I would like to thank the ANU and CSIRO Division of Mathematics and Information Sciences for PhD scholarships. Thanks also to Rob Vertessy, for giving me a work experience opportunity with the CRC for Catchment Hydrology.

Abstract

The application of thin plate smoothing splines to the spatial interpolation of large data sets has in the past been limited by the associated computational cost. To obtain the analytic solution to the thin plate smoothing spline equations, $O(n^3)$ operations are required, making routine computation on the average workstation infeasible for data sets with more than around 2000 data points. Methods for fast computation of analytic thin plate smoothing splines have been developed by past studies, and finite element approaches have also been widely employed. These studies have investigated the thin plate spline system mainly from a numerical perspective, without an automatic mechanism for optimising 'smoothness'. The smoothness of a thin plate smoothing spline function is controlled by a smoothing parameter λ. This parameter determines how closely the data points are fitted. Depending on the value of λ, the data points can be exactly interpolated, or globally smoothed.

For practical smooth spatial interpolation problems, surface smoothness is a central issue. The motivation for this study was the spatial interpolation of large environmental data sets, particularly climate data sets. Observations of environmental processes often contain a high noise component, so the extent to which the data are smoothed has a considerable influence on the predictive accuracy of the fitted function. Optimising the smoothing parameter by minimising the generalised cross validation (GCV) is well known to be a suitable method for minimising the predictive error of the thin plate smoothing spline surface, and has been used in many applications of thin plate smoothing splines. It was therefore the goal of this study to develop a computationally efficient numerical strategy for fitting approximate minimum GCV thin plate smoothing splines to large data sets.

Hutchinson [67] has presented an iterative procedure for calculating finite element approximations to thin plate smoothing splines, optimising smoothness

to deliver a user specified residual sum of squares from the data. This smoothness criterion is appropriate in the context of interpolating topography, where an estimate of the data error is known. This study modifies the methods in Hutchinson [67], to iteratively obtain finite element approximations to minimum GCV thin plate smoothing splines.

The procedure involved discretising the thin plate smoothing spline equations, and using a nested grid multigrid iterative strategy to solve the discretised system. The nested grid framework facilitates iteration on grids of varying resolution, starting at a coarse resolution and sequentially refining the grids. To optimise smoothness, the solution process incorporated a double iteration to simultaneously update both the estimate of the discretised solution, and the estimate of the minimum GCV smoothing parameter. A Taylor series expansion was used to estimate the value of λ corresponding to the minimum GCV. A stochastic approximation to the GCV was used to estimate the GCV for given λ values.

The investigations in this study led to an understanding of the process of double iteration for the case of the thin plate smoothing spline problem. It was found that the iteration converged efficiently, except when the thin plate smoothing spline system was poorly conditioned. Conditioning generally deteriorated as the grid resolution was refined, particularly when the smoothing parameter was large. Poor conditioning resulted in degradation of the efficiency of the iterative processes. This caused the double iteration to become poorly synchronised, in that the solution estimate could not be efficiently adjusted in response to changes in the smoothing parameter estimate. In these circumstances the double iteration did not always converge.

Throughout extensive testing of the procedure, a number of strategies for overcoming the above problems were identified. Firstly, the type of discretisation was varied. Discretisation of the spline system using basis elements composed of quadratic B-splines was found to stabilise the double iteration considerably in comparison to a simpler, piecewise constant discretisation. The improvement was attributed to the first order continuity of the quadratic B-spline approximation, which allowed continuous, broadscale functions to be accurately approximated at coarse grid resolutions. Accurate methods of transferring quadratic B-splines from coarse to fine grid resolutions also improved the efficiency of the algorithm.

Convergence was further improved by appropriately setting the initial value of the λ estimate, limiting the amount by which λ could be updated, and emplacing suitable lower and upper bounds on the λ estimate. A first order correction, applied to the solution estimate after each smoothing parameter update, was also found to improve the performance of the algorithm, by allowing the solution estimate to respond more quickly to the smoothing parameter update. This depended on obtaining an efficient estimate of the derivative of the solution with respect to the smoothing parameter. The Taylor series expansion of the GCV in terms of the smoothing parameter also used this derivative estimate.

The algorithm produced by this study was tested on several simulated data sets, with varying spatial complexity, noise level, and distribution. Accurate approximations to the analytic minimum GCV thin plate smoothing spline were produced for all test data sets. Given that the intended application was the spatial interpolation of environmental data, the algorithm was also tested on environmental data sets, including temperature data for the African and Australian continents. Accurate estimates of the analytic solution were generated by the algorithm for both the African and the Australian temperature data sets. This demonstrates that the algorithm constructed by this study is suitable for fitting approximate minimum GCV thin plate smoothing splines to large environmental data sets.

Contents

List of Figures

List of Tables

Part I

Methods

Chapter 1

Introduction

1.1 Overview and motivation

The techniques involved in this study are designed to estimate spatially dependent processes throughout a given region by spatially interpolating large numbers of noisy point observations of the process. The principal intended application is the prediction of spatial processes that occur in natural ecological systems. Many natural processes, such as climate, topography, soil and vegetation, have underlying spatial coherence, in that two observations that are close together are more likely to have similar values than two observations that are far apart. The interpolation procedure is designed to describe this coherence, by finding spatially dependent trends in data observations taken at particular locations in the study area. Interpolation of these trends allows prediction at locations where measurements have not been taken. Interpolated values are used to create regular two dimensional grids of predictions, known as surfaces, which can be incorporated into geographic information systems to visualise spatial patterns and detect spatial relationships. The appropriate resolution of the interpolated grid depends on the complexity of the process and the density of the data.

Research on the prediction of surface climate provided the underlying motivation for further investigation of the thin plate smoothing spline methodology in this study. Studies by Mackey [86] and Mackey et al. [87, 88] emphasise the need for information about the spatial variation of ecosystem characteristics as input to landuse decision making. Given that vegetation communities are generally inadequately sampled and mapped, climate variables that are known to be correlated with plant and animal distributions are often used to predict vegetation characteristics. Bioclimatic indices such as rainfall, temperature and solar radiation are widely known to discriminate between different vegetation types, as demonstrated by Nix [93] and Mackey et al. [88]. Spatial interpolation of climate data has been shown to effectively predict spatial patterns in vegetation and agriculture [71]. The use of interpolated surfaces to detect spatially varying trends also provides much needed spatially predictive, as well as descriptive capacity. In this way they are more informative than other techniques of spatially mapping vegetation, such as aerial photography and remote sensing. Furthermore, ecosystem attributes do not always form distinctive and recognizable photographic patterns or spectral signatures [86].

Spatial interpolation of surface climate variables is also an integral part of temporal climate prediction via stochastic generation of climate data [64]. Elevation dependent spatially interpolated surfaces are used to predict long term spatial climate variability . These techniques are linked to the development of space-time stochastic weather models through the process of spatially extending the parameters of point simulation models. Methodologies for constructing such models are discussed in Hutchinson [63, 64] and Guenni and Hutchinson [43].

The thin plate smoothing spline method of spatial interpolation used in this study can be motivated by the following data model. Consider data observations $(z_i, x_{1i}, x_{2i}, \ldots, x_{di})$ measuring a dependent variable z and a set of d predictor variables $x_1, \ldots, x_d$. For example, surface climate is often well predicted using latitude, longtitude and elevation. If it is assumed that z has both continuous long range variation as well as short range variation that is discontinuous and random, then we can propose the following model

$$z_i = g(x_{1i}, \ldots, x_{di}) + \epsilon_i \qquad i = 1, \ldots, n \tag{1.1}$$

where n are the number of data observations, g is a slowly varying continuous function and ϵ_i are realisations of a random variable ϵ. The function g represents the spatially continuous long range variation in the process measured by z_i. The errors ϵ_i are assumed to be independent with mean zero and variance σ^2. They are assumed to be due to measurement error, and short range microscale variation that occurs over a range smaller than the resolution of the data set. The microscale variation may be spatially continuous, but the data is not spatially dense enough to represent it, so it is usually assumed to be discontinuous noise.

We aim to estimate the process g by a suitably continuous function f. The function f must be able to separate the continuous signal g from the discontinuous noise ϵ_i. This function can be estimated by minimising

$$\frac{1}{n}\sum_{i=1}^{n}(z_i - f_i)^2 + \lambda J_m^d(f) \tag{1.2}$$

over functions $f \in \mathcal{X}$, where $\mathcal{X}$ is a space of functions whose partial derivatives of total order m are in $\mathcal{L}^2(E^d)$ [113]. The f_i are values of the fitted function at the i^{th} data point, λ is a fixed smoothing parameter, and $J_m^d(f)$ is a measure of the roughness of the function f in terms of m^{th} order partial derivatives. The form of $J_m^d(f)$ depends on m and the number of independent variables d. For example, if $m = 2$, which is a typical value, and $d = 2$, then

$$J_2^2(f) = \int_{-\infty}^{\infty} f_{x_1x_1}^2 + 2f_{x_1x_2}^2 + f_{x_2x_2}^2 dx_1 dx_2 \tag{1.3}$$

[113]. Expression (1.2) represents a trade off between fitting the data as closely as possible whilst maintaining a degree of smoothness. The smoothing parameter λ controls the separation of signal and noise. If $\lambda = 0$ the function f exactly interpolates the data, implying zero noise, where as if λ is very large the function approaches a hyperplane. It is shown in Craven and Wahba [24] that the λ corresponding to the spline function f that best represents the underlying process g can be accurately estimated by minimising the generalised cross-validation, or GCV. The GCV is a measure of predictive error, and will be discussed in Chapter 2.

The solution to this minimisation problem is well known to be a thin plate smoothing spline function [99, 31, 92, 113]. Multivariate thin plate splines are not piecewise polynomial functions like the traditional univariate splines. They are termed 'splines' because the solution to (1.2) for the univariate case, with $m = 2$, is a natural cubic spline.

The application of thin plate smoothing splines to the representation of broadscale trends in noisy data is well documented in past studies over the last 20 years, in a wide variety of fields. The thin plate smoothing spline methodology is often an integral part of spatial modelling of environmental processes, including surface climate processes [49, 63, 118], topography [58], remote sensing [10], pollutant dispersion [73] and plankton distributions [116]. They are also commonly used in other fields such as image analysis [11], medical research [83] and data mining [54, 53].

Smoothing splines have several attractive characteristics that explain their popularity across such a wide range of disciplines. They are robust in that accurate predictions can be achieved in the presence of significant data error. They are global, in that they use information from all the data observations to calculate a prediction at any given location in the study area. Furthermore, the algorithms associated with smoothing spline computation are now efficient and operationally straightforward. Smoothing splines are also directly associated with a statistical framework that allows calculation of pointwise standard errors [113], as well as summary statistics such as the generalised cross-validation (GCV) and the degrees of freedom of the fitted model. This facilitates quantitative assessment of the ability of the function to represent the underlying data generation process [63].

To put thin plate smoothing splines in the context of other methods of spatial interpolation of noisy data, we can think of two approaches to smoothing noisy data. A penalty term, such as $\lambda J_m^d(f)$, can be added to the residual sum of squares term $\frac{1}{n}\sum_{i=1}^{n}(z_i - f_i)^2$ to impose smooth interpolation, or the space of functions that make up f can be restricted so that exact interpolation is not possible. For example, we could solve

$$\text{Minimise } \frac{1}{n}\sum_{i=1}^{n}(z_i - f_i)^2 \tag{1.4}$$

where

$$f(x_1, x_2) = \sum_{j=1}^{N}\sum_{k=1}^{M} c_{jk} \sin jx_1 \sin kx_2 \tag{1.5}$$

where c_{jk} are the fitted coefficients and $N \times M << n$, to achieve a low dimensional bivariate Fourier sine series fit that smooths the data.

There are also a number of local techniques for smooth spatial interpolation, which constrain the form of $f(x)$ by considering only certain portions of the data to generate predictions in a certain area. Techniques such as Thiessen polygons [109] and deLauny triangulations [1] paritition the study area into small elements and fit simple functions on each element. The data are exactly interpolated on each element. These methods are not easily extended to higher dimensions [60]. There are also inverse distance weighting [101] and moving average methods [26, 91, 82]. These methods involve a subjective choice of weighting function, which is usually defined in terms of a radius of influence beyond which data points are ignored [60].

The geostatistical method known as kriging is regarded as the main competitor to thin plate splines as a method for spatial interpolation [25, 60]. Kriging originated in the mining industry, to help improve spatial estimation of ore reserves [111]. Like splines, kriging is based on the model in (1.1), but assumes g is a realisation of a spatially correlated random function [74]. Both splines and kriging have been shown by Kimeldorf and Wahba [79], Matheron [90] and Duchon [29, 30] to be formally equivalent, although they are operationally different. Both generally provide higher predictive capacity than the simpler interpolation methods mentioned above, and require fewer guiding covariates [63, 75]. They are also easily extended to higher dimensions, although there are natural restrictions on the dimensions of the fitted surfaces if they are to be robustly determined from observed data [22]. Splines tend to be operationally simpler than kriging, because the kriging method requires separate fitting and calibration of a spatial covariance structure [65].

As an example of thin plate smoothing spline interpolation of climate processes, consider the cases of temperature and precipitation. Thin plate smoothing spline functions of latitude, longtitude and elevation have been shown to accurately describe long term annual and monthly mean surface temperature and precipitation [57, 60, 118, 119, 63, 77]. Temperature is the simpler of the two spatial processes, and has a roughly linear dependence on elevation that is independent of location [60]. Hutchinson [60] demonstrated that the following partial spline model is a sensible model for temperature

$$z_i = g(x_i, y_i) + \beta h_i + \epsilon_i \tag{1.6}$$

where x_i is the longtitude, y_i is the latitude and h_i is the elevation at data measurement location i. The resulting solution incorporates a bivariate thin plate smoothing spline $f(x, y)$ and a spatially constant linear trend on elevation with slope, or 'elevation lapse rate' β.

Rainfall is a more complex and localised process than temperature, and it is not reasonable to assume a constant dependence on elevation throughout the study area. Rainfall can be accurately modelled by a thin plate spline with three independent variables, according to the following model

$$z_i = g(x_i, y_i, h_i) + \epsilon_i \tag{1.7}$$

[57]. This allows for a spatially varying dependence on elevation. Other independent variables, such as slope and aspect, have been found to provide minimal additional explanatory power [63].

As rainfall is more spatially complex than temperature, it requires more data points for accurate prediction over a given area. Approximately 14000 points have been used to construct a reliable rainfall surface for the entire Australian continent, where as temperature surfaces have been constructed from 1000 data points [60]. This raises the issue of computational efficiency, which has in the past been a major practical problem associated with thin plate smoothing splines [119, 54, 53]. The analytic calculation of thin plate smoothing splines described in [113] requires $O(n^3)$ operations, where n is the number of data points. This method quickly becomes unworkable when the number of data points increases. To fit a thin plate smoothing spline using 14000 data points would require a number of operations within an order of magnitude of 10^{12}. As a result, the production of a mean rainfall surface covering Australia by Hutchinson and Kesteven [70] required joining together a number of smaller surfaces. There are clearly many other spatial data sets that consist of thousands, or even millions, of data points. Also, one may wish to generate several surfaces for one process, corresponding to many different seasons or years. In this case, a fast, straightforward procedure is required.

This study aims to widen the applicability of minimum GCV thin plate smoothing splines to the modelling of climate and other environmental processes by increasing the computational efficiency of the spline fitting process. An optimal numerical strategy whose computational speed depended linearly on the number of unknowns would greatly increase the efficiency of thin plate spline computation. This corresponds to an $O(N)$ algorithm, where N is the number of unknowns. There are methods described in recent literature for fast computation of analytic thin plate smoothing splines for large data sets. Techniques utilizing conjugate gradient preconditioning techniques [102] and ideas from multipole expansions and Lagrangian junctions [8, 9, 7] calculate the analytic thin plate smoothing spline solution to (1.2). These methods achieve $O(n)$ workload, but involve complex data structures and algorithms.

Finite element approaches have also been used in a number of studies, including those by Terzopoulos [108], O'Sullivan [94] and Szeliski [106]. Earlier methods discuss discretisation of the roughness penalty, denoted above as $J^d_m(f)$, from the perspective of interpolation with minimum curvature [18, 105] . Hegland et al. [53] also present a methodology for calculating discrete thin plate smoothing splines based on first order techniques similar to mixed finite element techniques for the biharmonic equation [23, 50]. Similar techniques were adopted by Ramsay [97], who presents an approach for bivariate spline smoothing over complex domains.

All of these methods tend to focus on the numeric-analytic properties of thin plate smoothing splines rather than their statistical properties, and therefore do not incorporate an automatic mechanism for optimising smoothness. For practical spatial interpolation problems, surface smoothness is a central issue given that the data observations contain a significant noise component. The amount of smoothing will affect the predictive accuracy of the fitted surface, so clearly it should be optimised. The smoothness of the fitted spline directly corresponds to the ratio of the signal, or the effective number of parameters of the fitted model, and the noise, or the degrees of freedom of the error [61]. Estimates of the noise due to measurement error and microscale variation therefore depend on the smoothness of the fitted surface. Optimising smoothness also provides insight into the variability of the data, and the scale and coherence of the underlying data generation process.

The minimum GCV criterion previously mentioned is generally appropriate for optimising the smoothness of thin plate smoothing spline fits to noisy data [24, 113, 62]. The GCV has been used to optimise smoothness for most thin plate smoothing spline applications to climate, eg. [57, 63, 118, 96, 77]. Minimising the GCV provides an objective criterion for comparing the predictive capacity of different spline models [65, 66]. The signal corresponding to the minimum

GCV surface can also be used to assess the reliability of the surface and its associated statistics [62]. For example, if the signal is equal to n, the spline has exactly interpolated the data and therefore has not separated the signal from the noise.

The primary objective of this study was therefore to construct a simple, fast, grid-based algorithm for calculating numerical approximations of thin plate smoothing splines, incorporating procedures to minimise GCV. The intention was to make the process of optimising smoothness as efficient as possible. A double iteration was therefore used to simultaneously produce increasingly accurate estimates of the minimum GCV smoothing parameter and the smoothing spline solution. Such a process would be considerably faster than fully solving the spline equations for a number of different levels of smoothness in order to find the optimal solution.

Hutchinson [58, 67] developed a simple multigrid based strategy which calculates finite element approximations to thin plate smoothing splines for elevation data in $O(N)$ operations, where N is the number of grid points. This method emphasises the statistical framework of thin plate smoothing splines [61], and optimises smoothness to yield a user specified residual sum of squares. This criterion is appropriate in the context of interpolating topography, where an estimate of the amount of noise is available [58]. A study by Altas et al. [2] showed that multigrid methods are very efficient for solving a finite difference discretisation of the biharmonic equation, whilst standard iterative methods such as Jacobi and Gauss-Seidel exhibit very slow convergence. The discretised thin plate smoothing spline equations have a similar structure to the discretised biharmonic equation, as discussed in Chapter 6. The results of Altas et al. [2] therefore indicate that the success of the Hutchinson [58, 67] method is largely due to the use of multigrid techniques.

The methodology on which this study is based is a variation on the Hutchinson [67] method, in that it seeks to minimise the GCV rather than achieve a prescribed residual sum of squares. Quadratic B-splines were used as the basis elements for the discretised solution, incorporating the favourable properties of first order continuity into the solution estimate. No more than two dimensions were considered, given that is the minimum number required for spatial interpolation. Other dimensions can then be incorporated using additive models [113, 72]. A conceptual overview of the method developed during the course of this study is given below. Hancock and Hutchinson [48], a publication resulting from this study, presents results of implementing the methodology to approximate minimum GCV univariate smoothing splines.

1.2 Summary of the research process

The process undertaken to obtain the discretised minimum GCV thin plate smoothing splines is shown in Figure 1.1. The thin plate smoothing spline equations were first discretised to allow the solution to be approximated by a series of discrete elements. Various multigrid iterative schemes were then tested with the aim of selecting an optimal method for solving for the coefficients of the discretised system. Additional procedures were then incorporated to allow simultaneous solution of the system and optimisation of smoothness by minimising GCV.

Although this study is interested in bivariate thin plate smoothing splines, much of the early algorithm analysis and development was done for the univariate case. This was because the process of testing and optimising the methodology was much simpler and more transparent in one dimension. The results of the univariate testing provided guidelines for the development of the bivariate algorithm.

The methods involved in the three stages in Figure 1.1 were continually refined throughout the preliminary univariate analysis. A number of different approaches were investigated and optimal strategies were selected. The most important progression was the type of discretisation. The system was originally discretised using piecewise constants. Further analysis indicated that a quadratic B-spline discretisation was better suited to the thin plate smoothing spline equations, due to its ability to represent smooth processes at coarse grid resolutions. This progression explains the chapter organisation for this thesis.

1.3 Summary of each chapter

The following summary of each chapter is intended to be a non-technical overview of the processes used in this study. The aim is to give the reader a conceptual grasp of the more important features of the analysis before the full procedures are explained in detail.

PART 1: METHODS

The first part of the thesis presents the technical details of the methods used

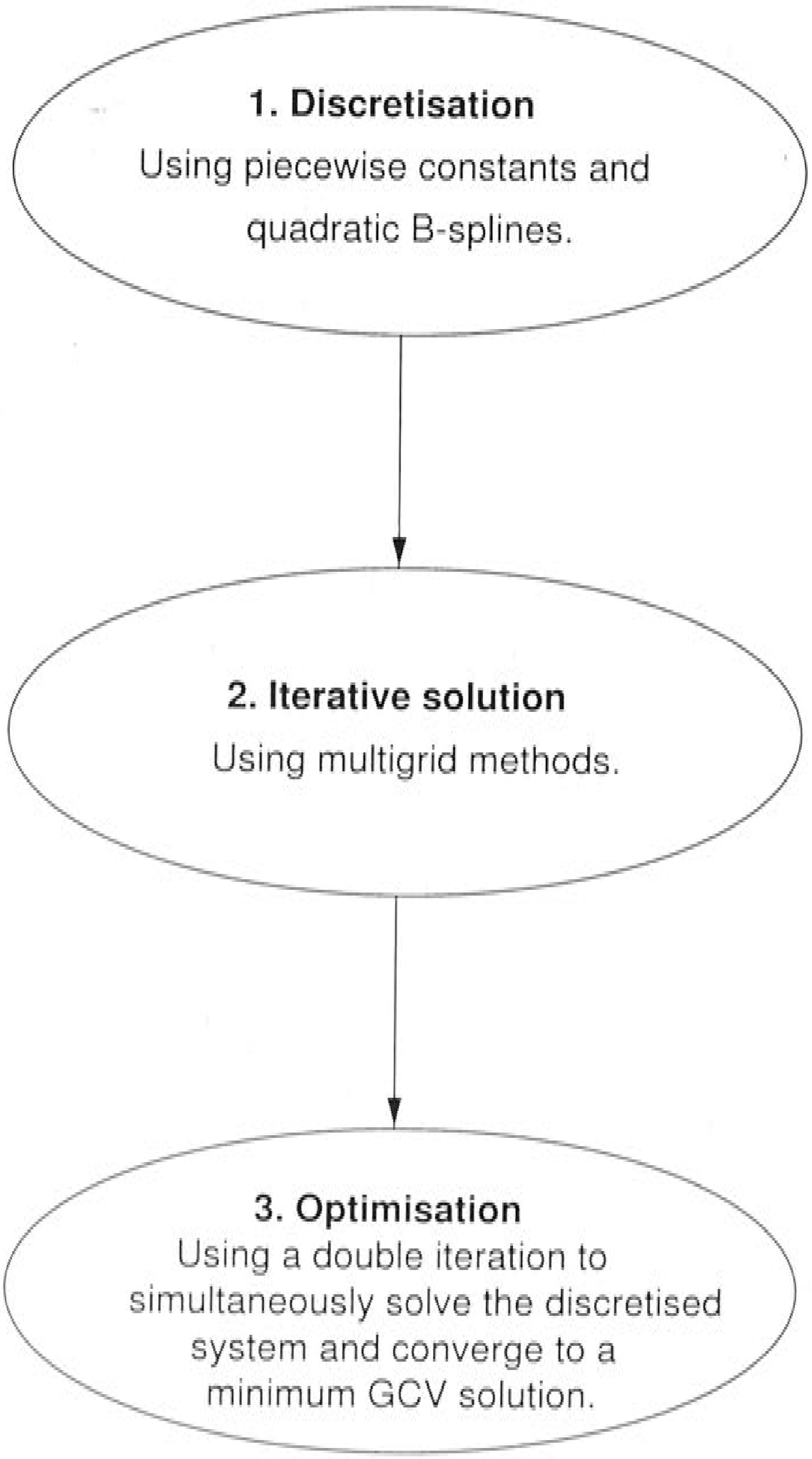

1. Discretisation
Using piecewise constants and quadratic B-splines.
2. Iterative solution
Using multigrid methods.
3. Optimisation
Using a double iteration to simultaneously solve the discretised system and converge to a minimum GCV solution.

Figure 1.1: Stages in developing the algorithm to iteratively solve for discretised minimum generalised cross-validation thin plate smoothing splines.

in this study, from the underlying thin plate smoothing spline model to methods of discretising the spline system, to the iterative algorithm for solving the equations and optimising smoothness.

Chapter 2 - Smoothing splines.

The necessity of optimising the smoothness of spline functions fitted to noisy data is easily depicted in the univariate case. Assume we want to estimate a process X that has both smooth, broadscale variation and short range variation, as shown in Figure 1.2. We collect data measuring the phenomenon X, shown in Figure 1.3. The data set contains significant noise due to measurement error. Smoothing spline functions are ideal for representing the process X from such data, as they can be used to detect broadscale trends that can be reliably interpolated into data sparse regions.

The smoothing parameter λ in equation (1.2) provides a lot of flexibility in the way smoothing splines fit noisy data. A very small value of λ produces a spline function that exactly interpolates the data, as shown in Figure 1.4. This is clearly a poor representation of the process X because it incorporates the errors in the data. Conversely, if the λ value is too large, we can oversmooth the data, as shown in Figure 1.5.

One method of optimising the smoothness of the fitted spline to estimate the process X is to minimise the GCV. The GCV measures the predictive capacity of the fitted spline by essentially determining how well the function predicts withheld data. The GCV calculation implicitly involves removing each data point in turn and summing, with appropriate weighting, the square of the discrepancy of each omitted data point from a surface fitted to all other data points. A plot of the GCV as a function of the logarithm of the smoothing parameter for the data set in figure 1.3 is shown in Figure 1.6. Figure 1.7 shows the result of minimising the GCV to optimise smoothness for this data set. The numerical methods in this study aim to approximate this optimal smoothing spline function in the bivariate case.

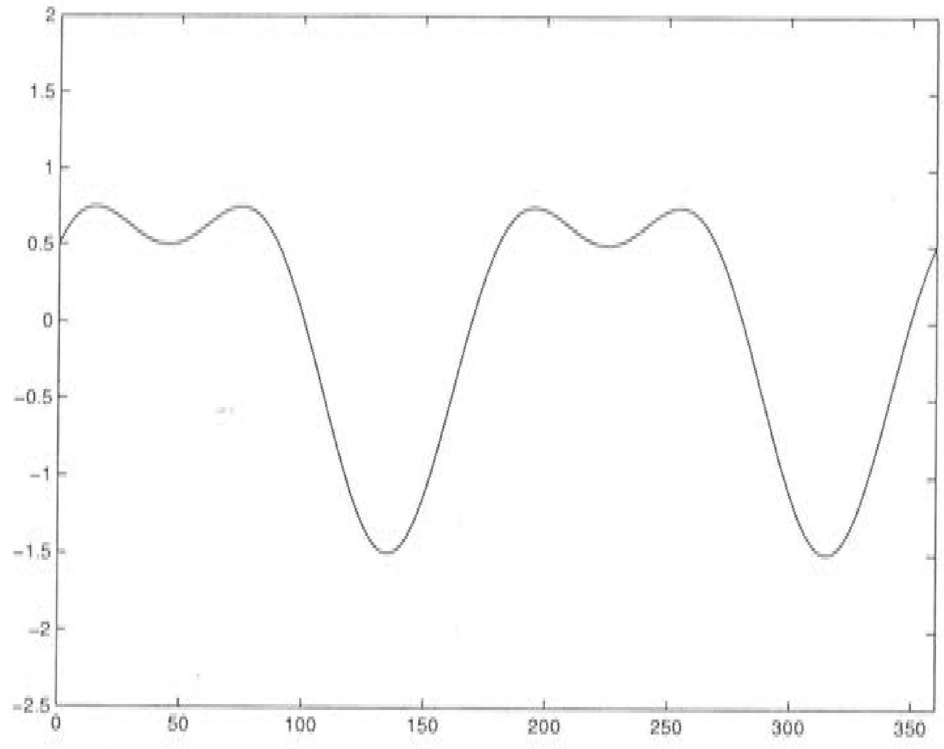


Figure 1.2: The process X.

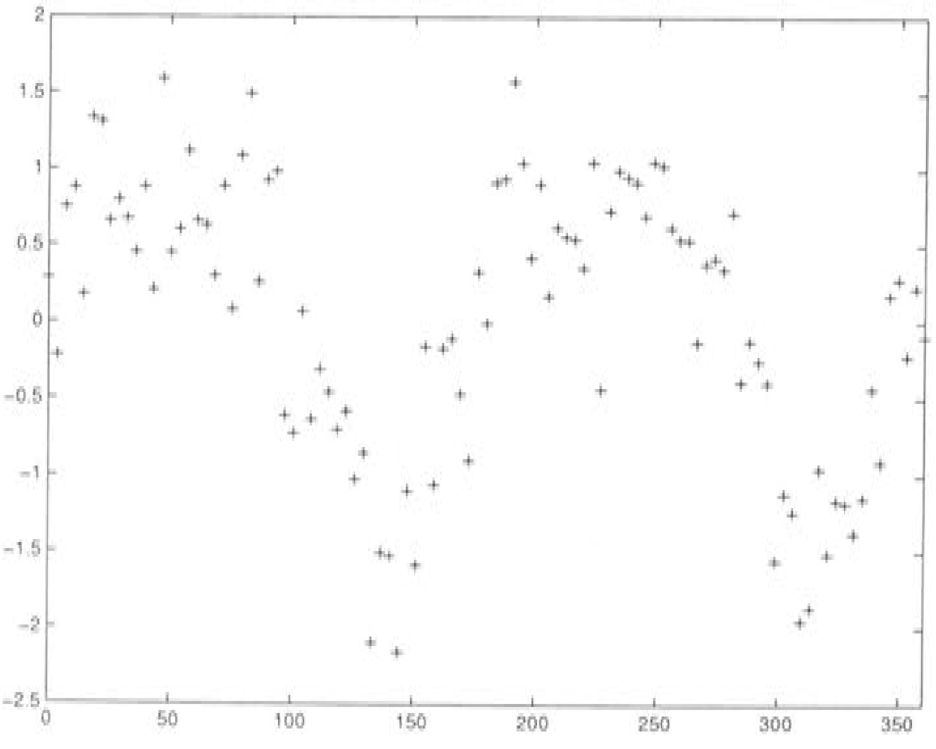


Figure 1.3: Noisy data observations of the process X.

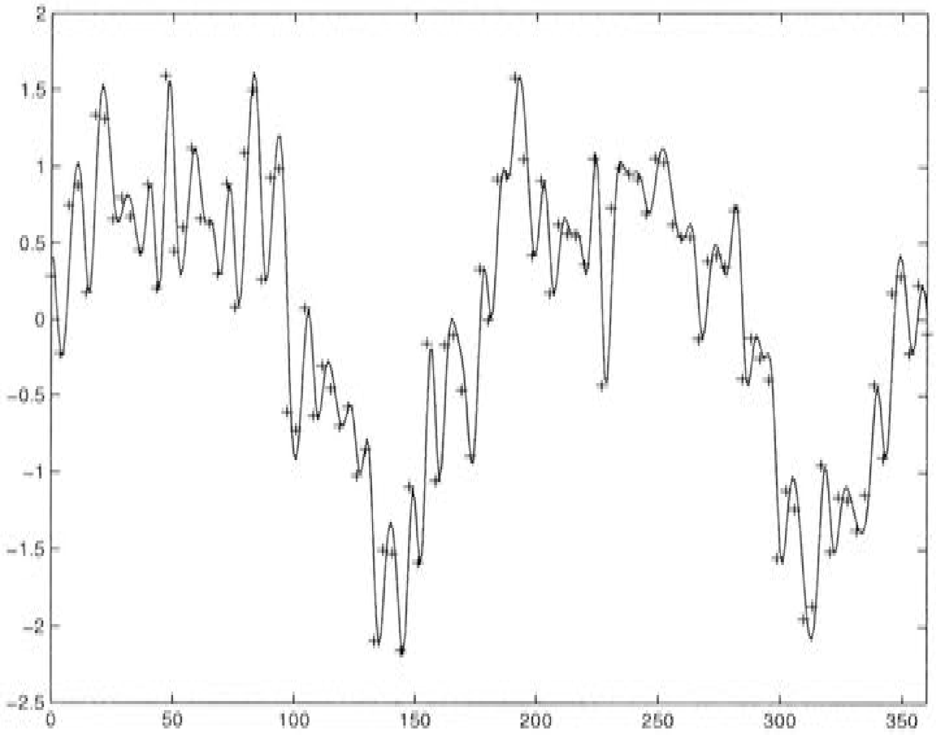


Figure 1.4: Smoothing spline fit to data observations using a small smoothing parameter.

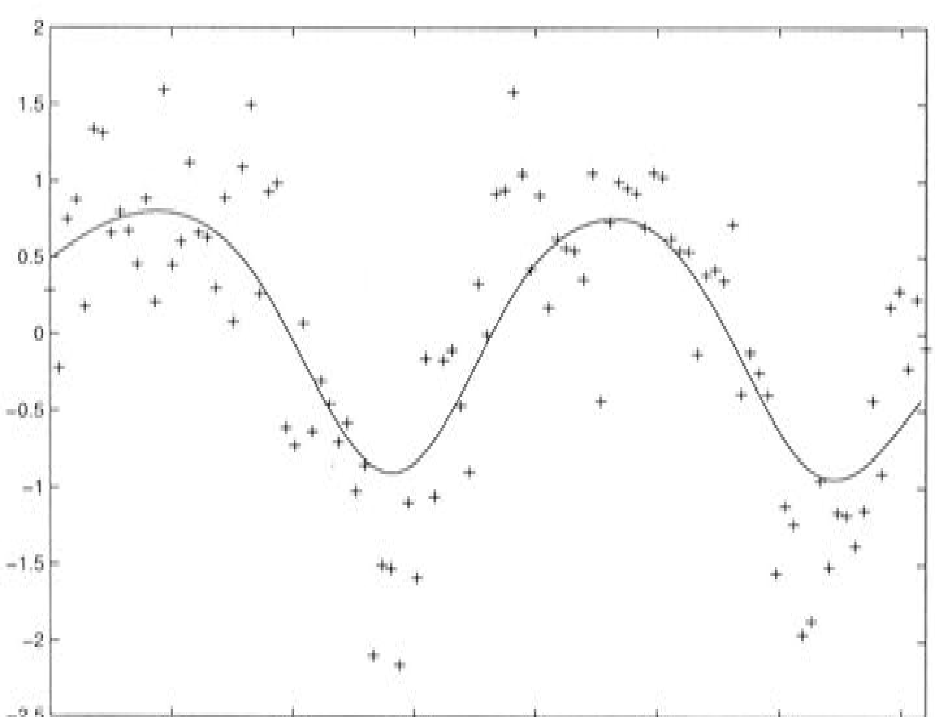


Figure 1.5: Smoothing spline fit to data observations using a large smoothing parameter.

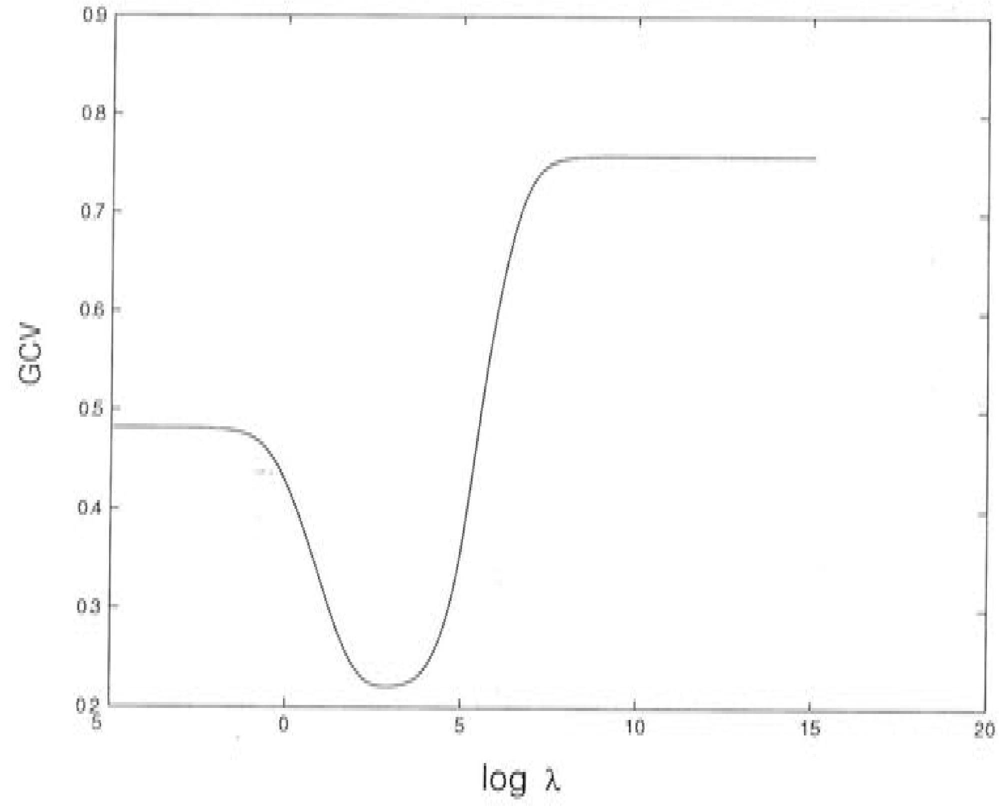


Figure 1.6: The GCV as a function of the smoothing parameter, for the data set in Figure 1.3.

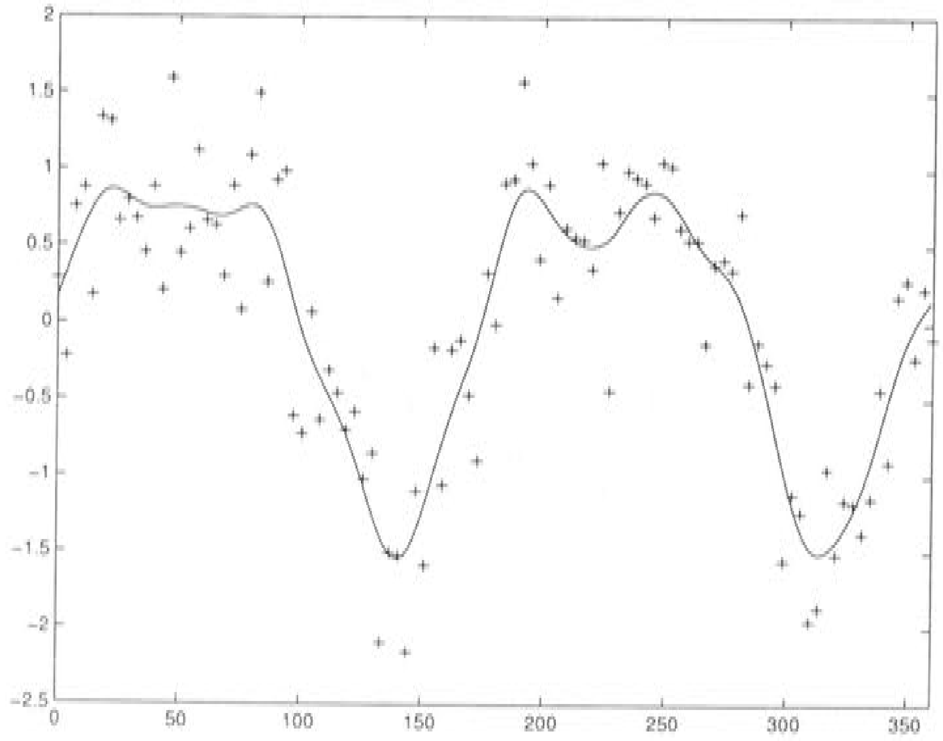


Figure 1.7: Smoothing spline fit to data observations using the minimum GCV smoothing parameter.

Chapter 3 - Discretisation of the univariate smoothing spline equations.

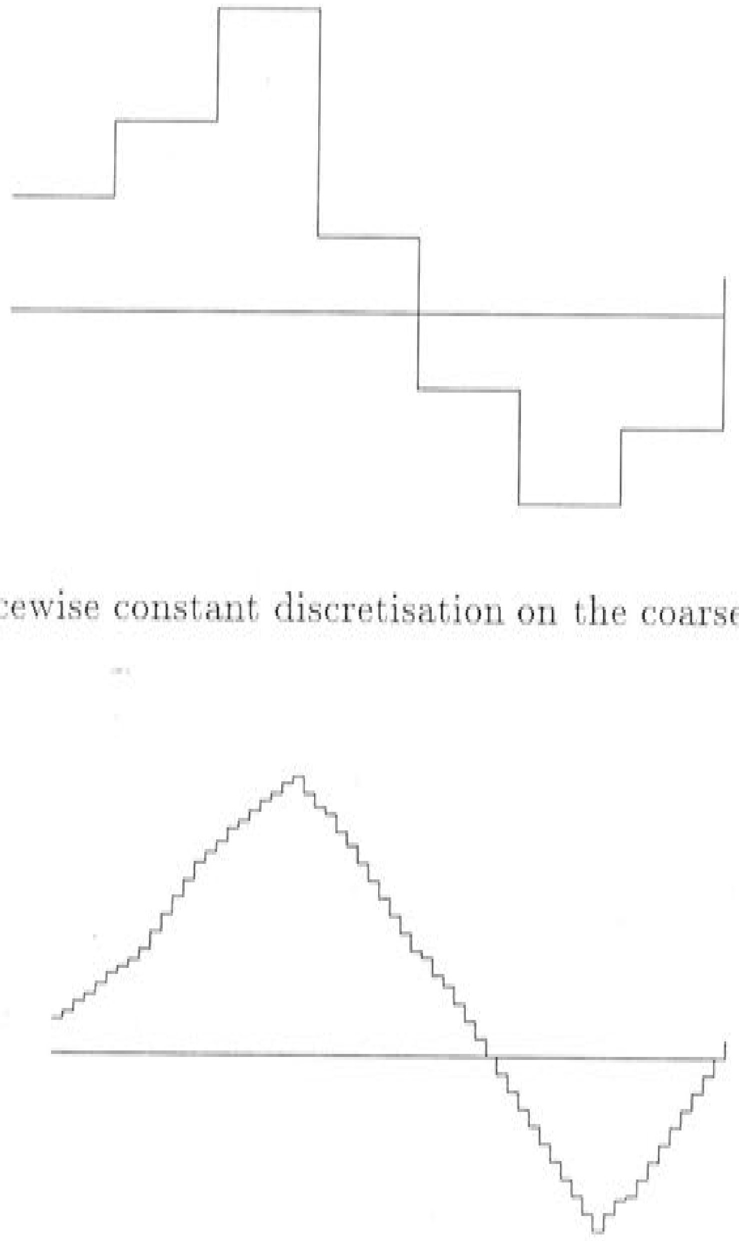
This chapter first presents a piecewise constant discretisation of the univariate smoothing spline problem. However, preliminary analysis demonstrated that the rectangular elements composing a piecewise constant representation were not an ideal approximation for the smoothing spline problem. This was primarily due to their inability to represent smooth functions at coarse discretisations. The piecewise constant discretisation approaches smoothness only at fine discretisations, as is demonstrated by Figure 1.8.

A discretisation that was smooth at coarse resolutions was therefore developed using quadratic B-spline basis elements. Quadratic B-spline elements are localised piecewise quadratic functions. A visual depiction is given in Figure 1.9. The process of constructing a B-spline approximation from the basis elements is shown in Figure 1.10. Unlike the piecewise constant discretisation, this approximation is always smooth, with a continuous first derivative, regardless of the resolution. This discretisation is therefore ideal for smoothing spline functions, that are inherently smooth.

Chapter 4 - Multigrid methods.

Past studies indicate that multigrid methods are efficient for iteratively solving the discretised thin plate smoothing spline system [58, 2, 67]. The multigrid principle provides the initial concept of decomposing a solution into components of different scale, and representing these components on grids of different resolutions. Iterations are performed on grids of varying coarseness during the solution process, as shown in Figure 1.11. This improves the convergence, because fine scale components of the solution are solved more efficiently on fine resolutions and coarse components can be obtained more efficiently on coarse resolutions. By visiting a variety of resolutions, all components of the solution can be quickly estimated.

Although based around a common sense idea, the multigrid method is associated with a somewhat subtle and abstract body of theory. The review in this chapter extracts some practical guidelines from multigrid theory. This study demonstrates the advantages of an empirical, experimental approach to multigrid implementation.



Piecewise constant discretisation on the coarse grid

Piecewise constant discretisation on the fine grid

Figure 1.8: Piecewise constant discretisation on coarse and fine grids.

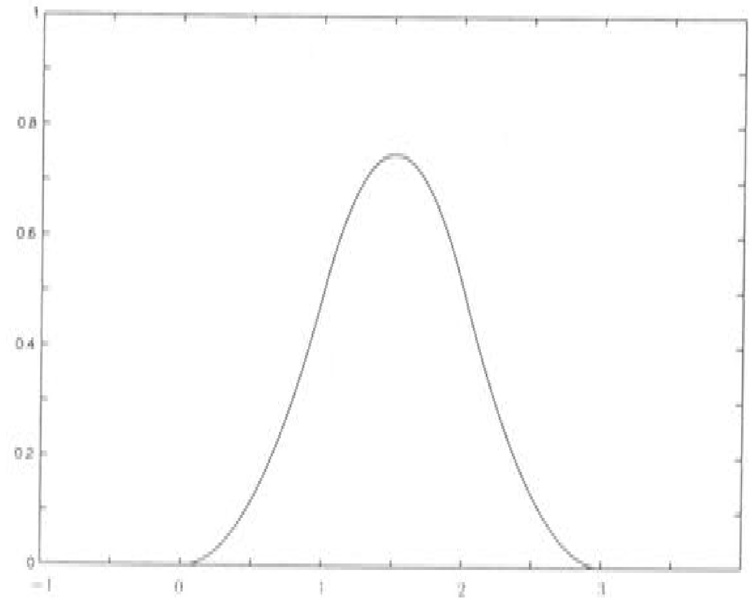


Figure 1.9: A quadratic B-spline basis element.

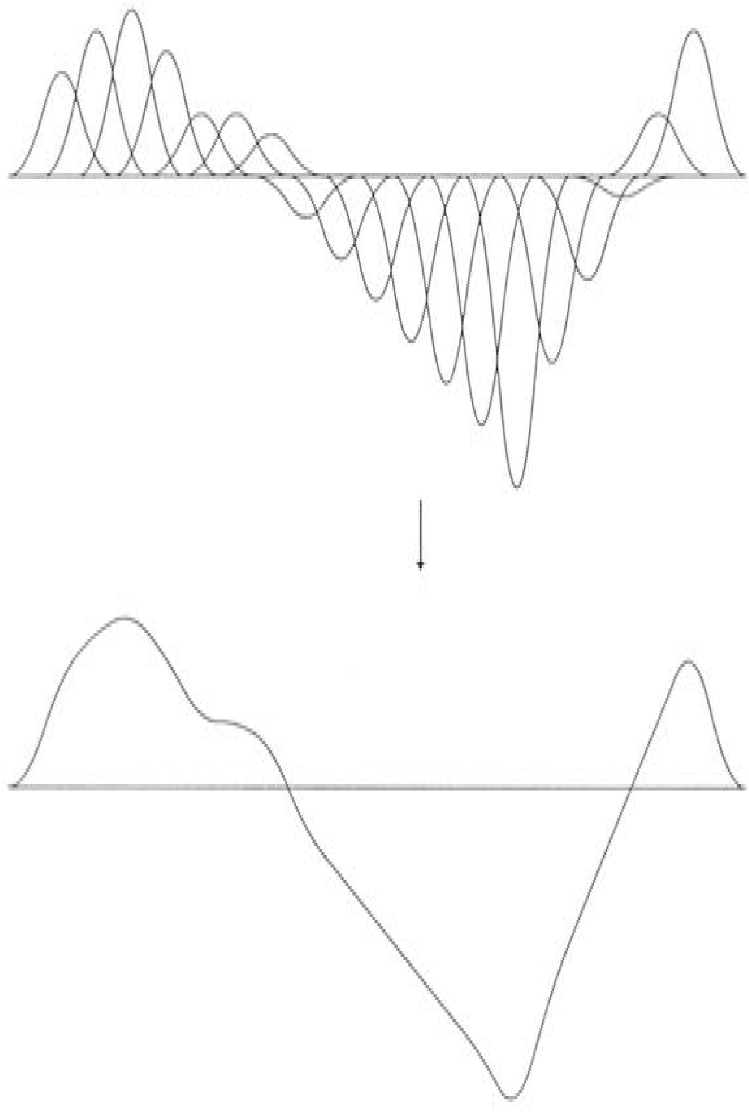


Figure 1.10: The process of constructing a quadratic spline by summing a linear combination of the basis elements.

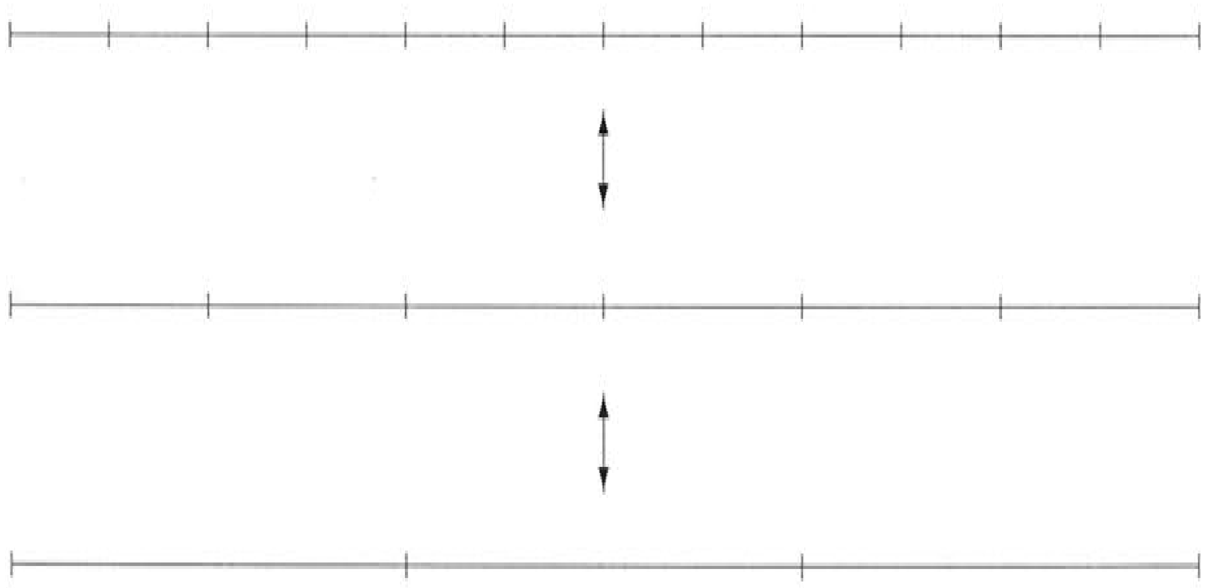


Figure 1.11: The multigrid process of transferring the solution estimate to and from grids of varying coarseness.

Chapter 5 - Prolongation and restriction of univariate quadratic B-splines.

Prolongation and restriction are multigrid processes for transferring the solution estimate to and from grids of different resolution. Prolongation, the process of transferring a solution estimate from a coarse grid to a fine grid, is performed in standard multigrid algorithms by interpolating the coarse grid solution. However, it was realised during the course of the univariate analysis that these transfer techniques are not needed for the discretised smoothing spline problem, because a coarse grid solution can be exactly represented on the fine grid by refining the B-spline basis elements. This refinement process does not change the solution estimate. Thus issues associated with errors introduced by interpolating the solution estimate to a finer grid are avoided. This chapter presents a B-spline refinement process, using a hierarchial B-spline framework.

The opposite process of transferring from a fine grid to a coarse grid, known as restriction, is commonly done by taking local weighted averages of grid values, or simply by taking every second grid point. This study obtains a coarse grid representation that is a least squares estimate of the fine grid solution. The advantage of restricting in this way is that a quantitative measure of the additional variation explained by the finer grid can be obtained. This technique makes use of the quadratic B-spline framework.

Chapter 6 - Discretisation of the bivariate thin plate smoothing spline equations.

The bivariate thin plate smoothing spline solution was discretised using tensor product quadratic B-splines. These basis elements are depicted in Figure 1.12. The discretisation process is more technically complex for the bivariate system than for the univariate case, and involves calculating integrals of products of unidimensional B-splines on a 2 dimensional grid.

Chapter 7 - Prolongation and restriction of bivariate quadratic B-splines.

Prolongation and restriction operations for bivariate quadratic B-splines are

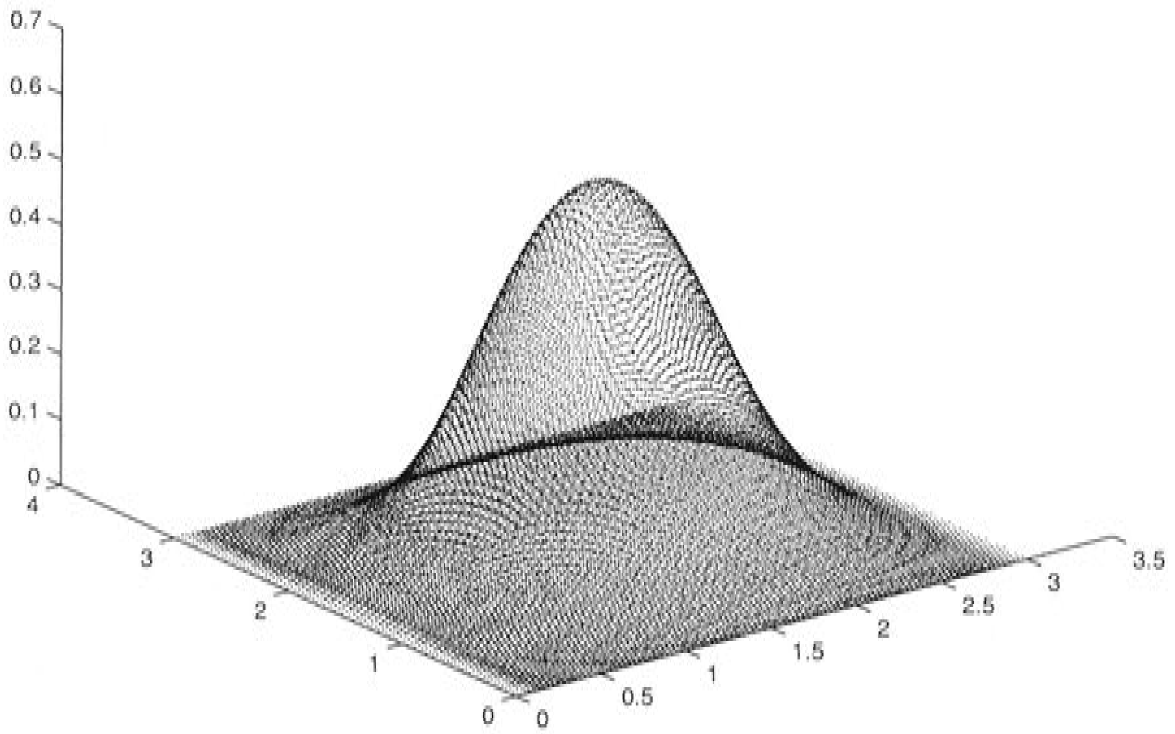


Figure 1.12: A bivariate quadratic B-spline basis element.

combinations of the univariate operations discussed in Chapter 5. The bivariate operations can be expressed as Kronecker, or 'tensor', products of the matrices for the univariate transfer operations.

Chapter 8 - Optimisating the smoothing parameter.

After establishing the iterative solution framework for the discretised thin plate smoothing spline system, the methods of Hutchinson [67] were extended to construct a procedure for optimising the smoothing parameter λ. This involved a double iteration to produce increasingly accurate estimates of the minimum GCV smoothing parameter λ and update the solution estimate accordingly. The algorithm therefore produced two sequences of updates for the solution $\mathbf{u}$ and the smoothing parameter λ, converging to the discretised minimum GCV smoothing spline solution. A conceptual demonstration is given in Figure 1.13. The vector $\mathbf{u}_0$ is an estimate of the discretised smoothing spline solution corresponding to the initial smoothing parameter λ_0. As the estimates of $\mathbf{u}$ get more accurate, the approximations of the smoothing parameter approach the minimum GCV value.

$$\begin{tikzcd}
& \mathrm{u}_0 \arrow[dr] & & \mathrm{u}_1 & & & \mathrm{u}_k \\
\lambda_0 \arrow[ur] & & \lambda_1 \arrow[ur] & & \cdots & \lambda_k \arrow[ur] &
\end{tikzcd}$$

Figure 1.13: The process of double iteration, obtaining increasingly accurate estimates of the solution $\mathbf{u}$ and the minimum GCV smoothing parameter λ

PART II: RESULTS

The second part of this thesis documents the behaviour of the techniques described above when they were applied to the thin plate smoothing spline problem, for both the univariate and bivariate case.

Chapter 9 - Multigrid testing.

As mentioned above, the design of an optimal multigrid scheme for a particular system is an empirical process that involves testing to understand the behaviour of the procedures for the specific problem considered. Testing was performed for the univariate smoothing spline system using simulated data. The results emphasised the fact that, for fine discretisations, the equations are poorly conditioned, especially if λ is large. This meant that the multigrid schemes that relied on fine grids performed poorly. It became clear that a nested grid multigrid algorithm was an efficient iterative solver for the smoothing spline equations. The nested grid algorithm starts with a very coarse grid and periodically refines the grids. By developing an initial solution estimate on coarse grids the algorithm avoids visiting unnecessarily fine grids, where the smoothing spline system is slow to converge.

Chapter 10 - Minimising GCV for the univariate piecewise constant smoothing spline system.

After deciding on a suitable multigrid solver, procedures to optimise the smoothing parameter λ were incorporated. Before optimising the GCV, a preliminary algorithm, which optimises λ to give a prescribed residual sum of squares from the data, was implemented. This algorithm is similar in structure to the procedure for minimising the GCV, but it is simpler and has been well tested by past studies [58, 67]. It was used in this preliminary analysis to test the optimisation framework, and understand its behaviour in the context of the smoothing spline problem.

While the algorithm converged on coarse grids, the initial results revealed the damaging effects of poor conditioning on fine grids. Due to the slowness of the iterative solution process on the fine grids, the double iteration procedure became poorly synchronised. This resulted in the development of an oscillatory pattern in the λ updates, as shown in Figure 1.14. The λ updates kept overshooting the optimal value, because the solution estimate represented old λ updates more strongly than current updates. For particularly poorly conditioned systems, divergence patterns such as those in Figure 1.15 were observed.

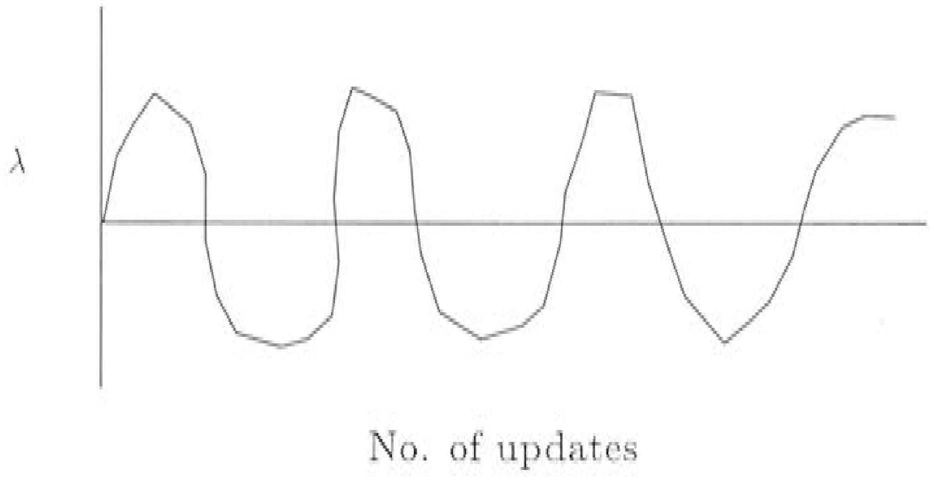


Figure 1.14: The oscillatory behaviour of λ updates.

When the optimisation framework was modified to minimise GCV, similar behaviour to the above was observed on fine grids. It was also found that the piecewise constant discretisation disturbed the solution structure such that a unique local minimum GCV value did not exist on some grids. These factors made the algorithm unstable. This problem was largely solved by the quadratic B-spline discretisation.

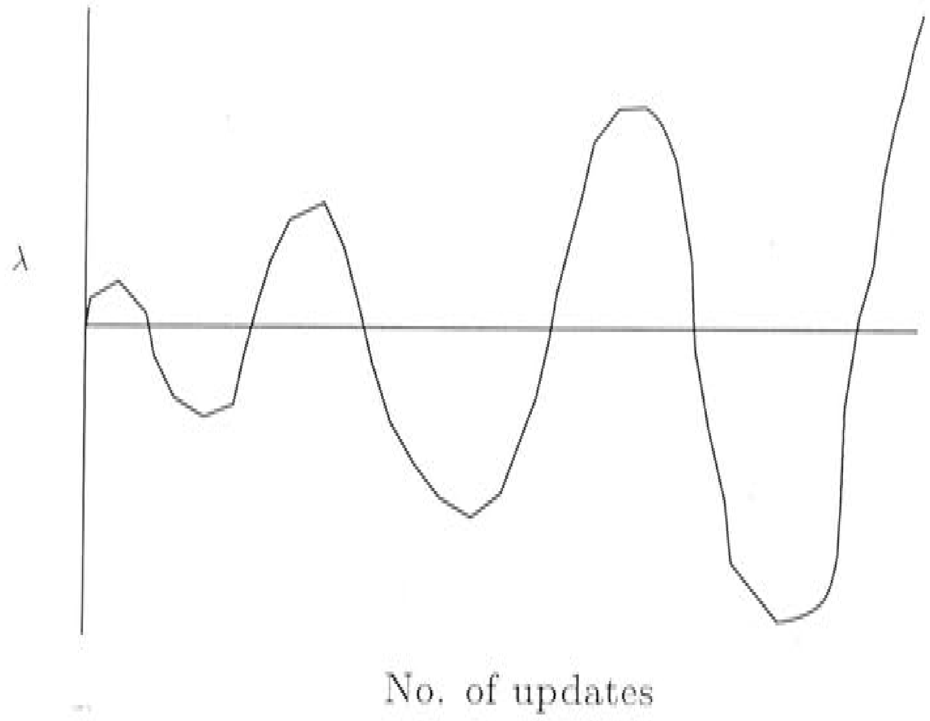


Figure 1.15: Divergent oscillatory behaviour of λ updates.

Chapter 11 - Minimising GCV for the univariate quadratic B-spline smoothing spline system.

Replacing the piecewise constant discretisation with the quadratic B-spline discretisation allowed a more accurate approximation to the solution to be obtained on coarse grids. This also stabilised the algorithm considerably, allowing faster convergence on coarse grids, and convergence on fine grids where the algorithm had diverged using piecewise constants. Interestingly, a unique minimum GCV could always be found using the quadratic B-spline discretisation, demonstrating that this discretisation gives a better representation of the true structure of the smoothing spline solution. The iterative algorithm, termed the MINGCV algorithm, was found to be stable, efficient and accurate at grid resolutions that matched the scale of the data generation process. Thus the univariate testing provided a promising framework on which to base development of the bivariate MINGCV algorithm.

The efficiency of the MINGCV algorithm was further improved by making a simple first order correction to the solution estimate after each smoothing parameter update. This allowed the solution estimate to respond more quickly to the smoothing parameter updates, avoiding the oscillatory patterns discussed above.

Chapter 12 - Minimising GCV for the bivariate quadratic B-spline thin plate smoothing spline system.

The univariate analysis provided a good practical understanding of the techniques used in the algorithm and developed a number of guidelines which defined the structure of the bivariate algorithm. The bivariate procedure was tested on a more diverse range of simulated data sets than the univariate algorithm. The level of noise was varied widely, and clumped data distributions with large data sparse areas were constructed for testing. This led to a series of modifications to the algorithm to control the magnitude of the λ update, the initial value of λ, the lower and upper bounds on λ, updating of the GCV derivatives and the final resolution of the grid. The final MINGCV algorithm converged for all test data sets.

Chapter 13 - Performance of the MINGCV algorithm for large temperature data sets.

The development MINGCV algorithm was motivated by the need for a computationally efficient procedure for spatially interpolating data observations for climate and other environmental processes. Environmental processes can feature complex fine scale interactions with the landscape that are difficult to emulate by a mathematical function. It was therefore important to test the ability of the bivariate MINGCV algorithm to approximate the thin plate spline surfaces that represent 'real' environmental processes. Sea-level temperature data for Africa and Australia were used for this purpose.

Accurate representations of the analytic smoothing spline solution were obtained for these data sets using the MINGCV algorithm. However, the MINGCV algorithm was unstable for the African temperature data set, due to the fine grid resolution required and the large areas devoid of data in the grid. The shape of the African continent is non-rectangular, leaving large areas in the corners of the grid which have no data. The analysis in earlier chapters showed that the smoothing spline equations are poorly conditioned when a highly smooth solution must be obtained on a fine resolution. This can cause poor synchronisation of the MINGCV iterations, leading to instability. In this situation, one alternative presented in this chapter is to subdivide the grid to avoid large holes at grid boundaries. An accurate solution for the African temperature data set was obtained in this way using the MINGCV algorithm. For the Australian continent, which is more rectangular in shape, the algorithm converged efficiently despite the fine resolution of the final grid, and the clumped, sparse distribution of the data. The results presented in this chapter indicate that the MINGCV algorithm is suitable for application to environmental data sets.

Chapter 2

Smoothing splines

When faced with the task of choosing a function to fit a series of data observations (z_i, x_i), $i = 1, \ldots, n$, it is generally desirable to choose one that does not feature significant local variation, given that the data observations usually contain error. A function capable of fast oscillation is likely to be highly sensitive to local fluctuations, and may 'overfit' the data, representing false trends and disguising broader scale trends in the data network. This diminishes the predictive capacity of the fitted model. Green and Silverman [40] note that even when the given observations are known to be extremely accurate, it is often of interest to regard the local variation as random noise in order to study the broader scale trend in the data.

Spline functions have many favourable characteristics that are useful for this purpose, and are thus covered by a vast body of literature. Theory on splines and their advantageous mathematical properties generally relates to the fact that they can vary slowly whilst still being flexible. To introduce spline functions, we begin with an overview of univariate splines

2.1 Univariate splines

To define a univariate spline function, consider a strictly increasing sequence of real numbers, known as 'knots', on some interval $[a, b]$

$$\gamma := \gamma_r, \qquad r = 0, \ldots, N \tag{2.1}$$

where N is the number of knot intervals, or 'spans'. Typically, $\gamma_0 = a$ and $\gamma_N = b$. An example grid is shown in Figure 2.1. A function is a spline if two conditions are satisfied. Firstly, a spline function of degree k is specified within individual spans $[\gamma_r, \gamma_{r+1}]$ by polynomials of degree at most k. Secondly, the function is in $C^{k-1}[a,b]$, the space of all functions that are continuous, and have continuous derivatives of order up to $k-1$, on the interval $[a,b]$. Note that in the general case spline functions can have coincident knot points. This breaks this continuity property, but such cases will not be considered in this study. Note also that in Figure 2.1 the knots are regularly spaced, but this need not be the case.

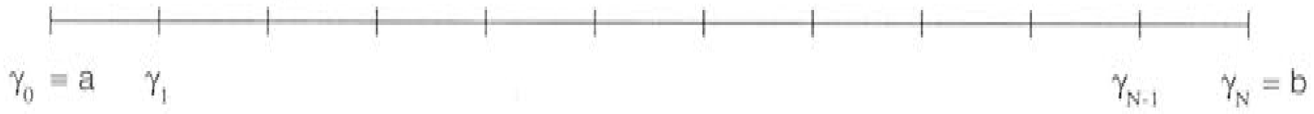


Figure 2.1: Knot positions for a univariate smoothing spline.

The polynomial piece on each span can be expressed as

$$s_{k,r}(x) = \sum_{d=0}^{k} A_{d,r}(x - \gamma_r)^d \tag{2.2}$$

where

$$\gamma_r \leq x \leq \gamma_{r+1}, \qquad r \in \{0, \ldots, N-1\} \tag{2.3}$$

The coefficients $A_{d,r}$ are constrained to meet the continuity conditions at the knot points, so that

$$s_{k,r-1}^{(m)}(\gamma_r) = s_{k,r}^{(m)}(\gamma_r) \tag{2.4}$$

where $r \in \{1, \ldots, N-1\}$, $m \in \{0, \ldots, k-1\}$ and the superscript m denotes differentiation of order m.

The collection of spline functions of degree k with a given knot sequence γ form a vector space, $\mathcal{S}_{k,\gamma}$. The dimension of $\mathcal{S}_{k,\gamma}$ is easy to calculate. A single polynomial curve of degree k belongs to a space that has dimension $k+1$. A spline function is made up of N polynomial pieces. The polynomial in the first span $[\gamma_0, \gamma_1]$ has dimension $k+1$. Polynomials in the remaining spans are constrained by k continuity conditions at the knot points, and therefore contribute only 1 dimension each. The dimension of the spline space $\mathcal{S}_{k,\gamma}$ is therefore $k+N$.

2.1.1 Smoothing splines

Roughness penalties

The use of splines as a smooth interpolant can be rationalised by the roughness penalty approach. The 'roughness' of a twice differentiable curve f defined on $[a, b]$ is often measured by its integrated squared second derivative

$$\int_a^b [f''(x)]^2 dx \tag{2.5}$$

This quantity is known as the 'roughness penalty'. There are various motivations for this measure of roughness, some of which are discussed in [40]. For example, if a thin piece of flexible wood, termed a 'spline', is constrained to pass through the points (z_i, x_i), then the leading term in the strain energy is proportional to $\int g''^2 dx$, where g is the graph of the spline [40]. This was a common method of drawing smooth curves before the age of computer graphics. The quantity in (2.5) is also a natural measure of smoothness as it is an approximation to the curvature of g, and the addition of a constant or a linear function does not change the curvature [40]. It was shown by Schoenberg [99] that, among all curves f in $C^2[a, b]$ interpolating the points in (z_i, x_i), the one minimising the roughness (2.5) is a natural cubic spline.

A cubic spline is given by $s_{k,r}(x)$, $r \in 0, \ldots, N-1$, with $k = 3$. A natural cubic spline is a cubic spline $f(x)$ with the following 'natural' boundary conditions

$$\begin{aligned} f''(a) &= f''(b) = 0 \\ f'''(a) &= f'''(b) = 0 \end{aligned} \tag{2.6}$$

These conditions result in $f(x)$ being linear on the two extreme intervals $[a, \gamma_1]$ and $[\gamma_{N-1}, b]$.

Smoothing

When modelling environmental phenomena it is usual to encounter noisy data sets. In this case it is logical to choose not to interpolate the data exactly. The following statistical framework is generally proposed. The n data observations

z_i, measured at positions x_i, are decomposed into

$$z_i = g(x_i) + \epsilon_i, \qquad i = 1, \dots, n \tag{2.7}$$

where $g(x)$ is a smooth continuous process [113, 63]. The observations incorporate the error term ϵ_i, which is modelled by

$$\begin{aligned} E(\mathbf{e}) &= 0 \\ E(\mathbf{e}\mathbf{e}^{\mathbf{T}}) &= V\sigma^2 \end{aligned} \tag{2.8}$$

where $\mathbf{e}^T = (\epsilon_1, \dots, \epsilon_n)$, V is a known, positive definite $n \times n$ matrix, and σ^2 is unknown.

The errors are usually assumed to be independent from one location to the next and thus V is assumed to be diagonal. Provided this is the case, the model effectively decomposes the observed data into two components - a coherent signal and spatially discontinuous noise [61]. This operates under the assumption that the phenomenon being interpolated has an underlying continuous component that can be sensibly represented by a smooth function [61]. The discontinuous noise is assumed to be due to measurement error or microscale effects below the resolution of the data network. The aim is to estimate $g(x)$, the signal or the smooth broadscale component of the continuous phenomenon, and remove the discontinuous noise.

The roughness penalty approach can be applied to obtain a suitable estimate for $g(x)$, using the penalised sum of squares [40]. The penalised sum of squares $P(f)$ is defined as

$$P(f) = (\mathbf{z} - \mathbf{f})^T V^{-1} (\mathbf{z} - \mathbf{f}) + \lambda \int_a^b [f''(x)]^2 dx \tag{2.9}$$

where $\mathbf{z}$ is a vector containing the data values z_i and $\mathbf{f}$ is a vector of the values of f at the data point locations x_i. The curve $g(x)$ is estimated by the function f that minimises $P(f)$. The first term is the sum of the square of the residuals of the data values from the values of the function f at the data point locations. The minimiser represents a tradeoff between fidelity to the data, as represented by the weighted residual sum of squares, and smoothness of the solution, as represented by the roughness penalty [113]. The relative importance of each of these components is controlled by the smoothing parameter λ, which is determined by the method discussed in section 2.2.1. It was again shown by Schoenberg [99] that the minimiser of $P(f)$ over $C^2[a,b]$ is a natural cubic spline $f_\lambda(x)$. A demonstration of this is also given in [40].

Unless the smoothing parameter λ is zero, the function $f_\lambda(x)$ no longer interpolates the data exactly. It is designed to represent smooth, broadscale trends that can be reliably interpolated into data sparse regions, rather than representing localised trends that are heavily reliant on individual data points. An example of how the function $f_\lambda(x)$ smooths noisy data is shown in Figure 2.2 .

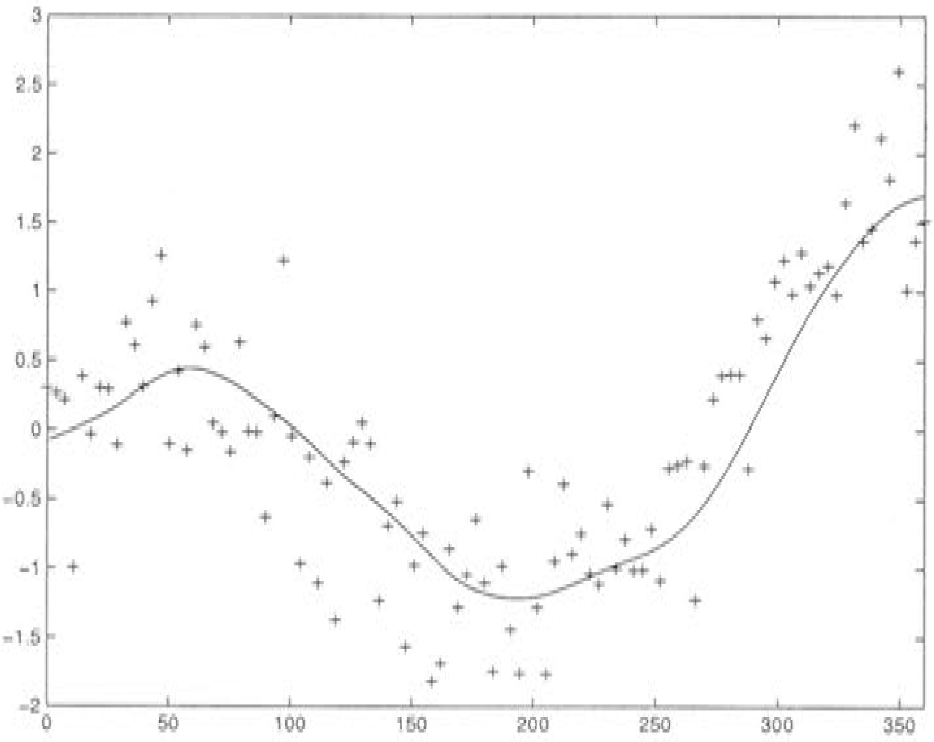


Figure 2.2: Smoothing spline fit to noisy data.

2.2 Thin plate smoothing splines

The generalisation of the smoothing spline problem to higher dimensions begins by stating the underlying model (2.7) in terms of the set of coordinates in multidimensional space, $\mathbf{t}_i$, as follows:

$$z_i = g(\mathbf{t}_i) + \epsilon_i, \qquad i = 1, \ldots, n \tag{2.10}$$

The penalised sum of squares (2.9) in higher dimensions is then expressed as

$$P(f) = (\mathbf{z} - \mathbf{f})^T V^{-1}(\mathbf{z} - \mathbf{f}) + \lambda J_m^d(f) \tag{2.11}$$

where $J_m^d(f)$ is the thin plate 'roughness' penalty functional, d is the number of independent variables and m is the order of the partial derivatives of f in the expression for $J_m^d(f)$ [113, 61]. For example, if $m = 2$, which is a commonly used value, and the $\mathbf{t}_i$ represent bivariate coordinates x and y, then

$$J_2^2(f) = \int_{-\infty}^{\infty} f_{xx}^2 + 2f_{xy} + f_{yy}^2 dxdy \tag{2.12}$$

[113]. The thin plate penalty function measures the roughness of the multidimensional function f.

Duchon [31] and Meinguet [92] obtained the function f that minimises expression (2.11) over $\mathcal{X}$, where $\mathcal{X}$ is a space of functions whose partial derivatives of total order m are in $\mathcal{L}^2(E^d)$ [113]. The solution is a multivariate function known as a thin plate spline, denoted $f_\lambda(\mathbf{t})$. This group of functions includes univariate splines as a special case. A demonstration of how a bivariate thin plate smoothing spline smooths noisy data is shown in Figure 2.3.

The thin plate smoothing spline solution, first obtained by Duchon [31], can be expressed as

$$f_\lambda(\mathbf{t}) = \sum_{j=1}^{M} a_j \phi_j(\mathbf{t}) + \sum_{i=1}^{n} b_i \psi(d_i) \tag{2.13}$$

where a_j and b_i are the coefficients, ϕ_j are a set of M low order monomials forming a basis for the null space of the roughness penalty and $\psi(d_i)$ are the natural scalar radial basis functions, where d_i is the euclidean distance between $\mathbf{t}$ and $\mathbf{t}_i$ [61]. Both M and the function ψ depend on the dimension of $\mathbf{t}$ and the order of the derivative m [113, 61]. For the bivariate case with $m = 2$, ϕ are $1, x$ and y, and $\psi(d_i) = c\, d_i^2 \ln d_i$ where c is a constant.

A necessary condition for obtaining the thin plate smoothing spline solution is that $2m - d > 0$ [113]. This ensures that the space $\mathcal{X}$ endowed with the seminorm $J_m^d(f)$ is a reproducing kernal Hilbert space. It can then be shown that the vector $\mathbf{f}$ is

$$\mathbf{f} = T\mathbf{a} - K\mathbf{b} \tag{2.14}$$

where $\mathbf{a}^T = (a_1, \ldots, a_M)$ and $\mathbf{b}^T = (b_1, \ldots, b_n)$ are vectors of coefficients. The matrix K is a symmetric matrix containing values of the natural scalar radial basis functions at the data point locations. Thus K is given by $K_{ij} = \psi(||\mathbf{t}_i - \mathbf{t}_j||)$. The matrix T is the $n \times M$ matrix defined by $T_{ij} = \phi_j(\mathbf{t}_i)$ [61]. The roughness penalty term in (2.11) also contains the matrix K, and can be expressed as

$$J_m^d(f) = \mathbf{b}^T K \mathbf{b} \tag{2.15}$$

The minimisation problem (2.11) is thus

$$\frac{1}{n}||W^{-T}(\mathbf{z} - T\mathbf{a} - K\mathbf{b})||^2 + \lambda \mathbf{b}^T K \mathbf{b} \tag{2.16}$$

where W results from the Cholesky decomposition of V ie.

$$V = W^T W \tag{2.17}$$

The coefficients $\mathbf{b}$ are restricted to satisfy the boundary conditions $T^T\mathbf{b} = 0$, which ensures the function f is a plane at infinity. Minimising (2.16) with this boundary condition gives the following system of $n + M$ equations

$$\begin{aligned}(K + \lambda V)\mathbf{b} + T\mathbf{a} &= \mathbf{z} \\ T^T\mathbf{b} &= 0\end{aligned} \tag{2.18}$$

[113]. An efficient procedure for solving equations (2.18) is given in [61]. Solution requires $O(n^3)$ operations.

2.2.1 Optimisation using generalised cross validation (GCV)

It can be seen from equation (2.11) that the thin plate spline solution to the minimisation problem will depend on the smoothing parameter λ. The next step is therefore to determine the value of λ that produces the best approximation by the thin plate spline to the actual continuous surface g that the spline is attempting to represent [113]. Craven and Wahba [24], in their analysis of the case where the $\mathbf{t}_i$ are unidimensional, argue that the ideal solution would

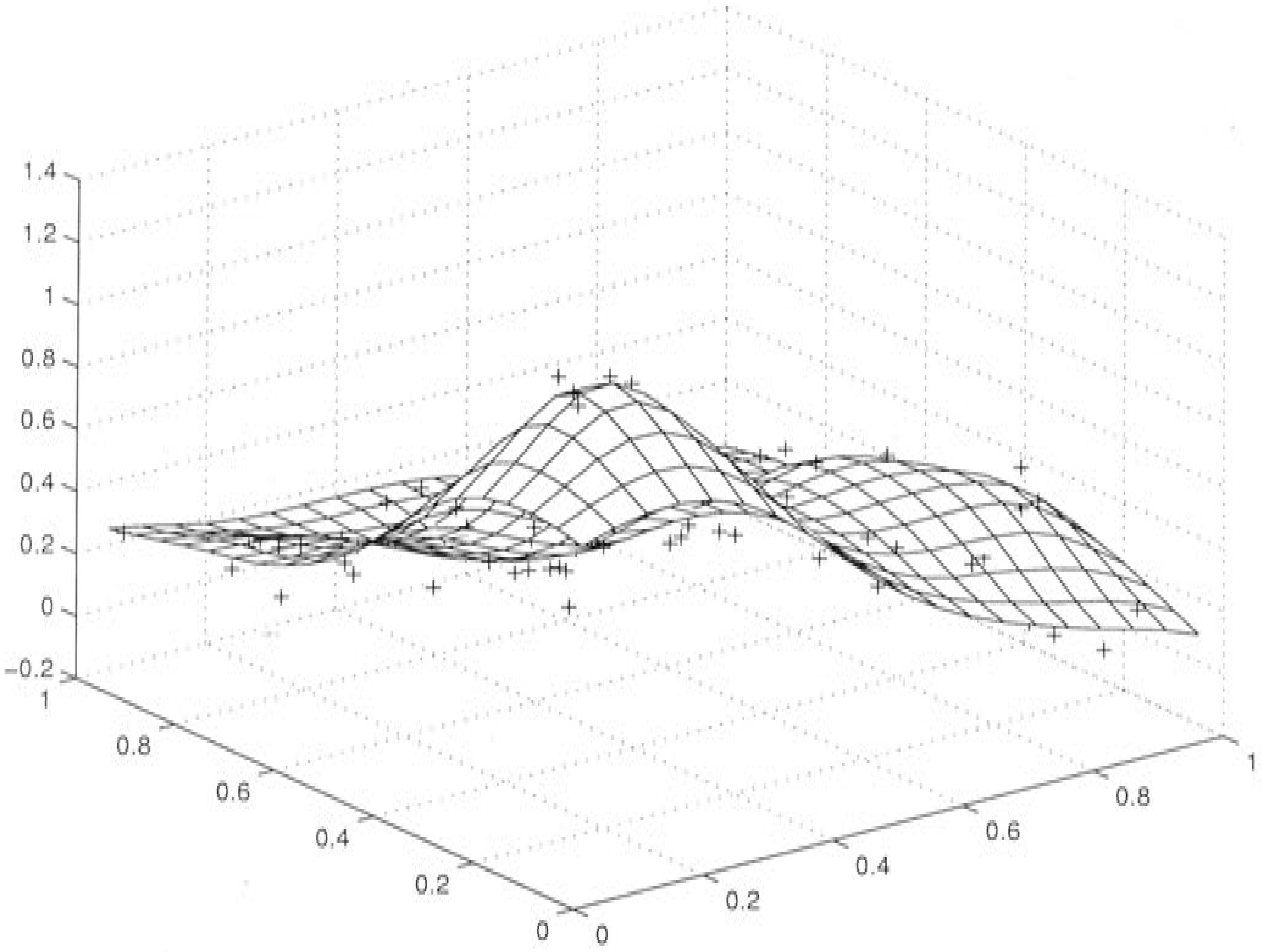


Figure 2.3: Thin plate smoothing spline fit to bivariate noisy data.

be to minimise the true error, given in [24] as

$$R(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(f_\lambda(\mathbf{t}_i) - g(\mathbf{t}_i))^2 \tag{2.19}$$

It is shown in [24] that, using mathematical expectation and assuming that the errors are independent, an unbiased estimate of $R(\lambda)$ is given by

$$\hat{R}(\lambda) = \frac{1}{n}||I - A(\lambda)\mathbf{z}|| - \frac{2\sigma^2}{n}tr(I - A(\lambda)) + \frac{\sigma^2}{n}trA^2(\lambda) \tag{2.20}$$

where

$$E(\epsilon_i\epsilon_j) = \sigma^2\delta_{ij}, \text{ and } E(\epsilon_i) = 0 \tag{2.21}$$

Here σ^2 is the variance of the noise, δ_{ij} is the Kronecker delta and $A(\lambda)$ is an $n \times n$ matrix, known as the 'influence' matrix. The influence matrix takes the vector of data values to the vector of fitted values. It is thus defined by

$$\mathbf{f} = A(\lambda)\mathbf{z} \tag{2.22}$$

where $\mathbf{f}$ is the vector containing the values of the fitted spline at the data point locations. The value of λ that minimises expression (2.20) is difficult to ascertain because σ^2 is generally unknown [24]. The optimal λ value is therefore chosen to be that which minimises a function known as the generalised cross validation (GCV) [24, 113, 39].

The GCV is a measure of the predictive error of the fitted surface and is effectively calculated by removing each data point in turn and summing, with appropriate weighting, the square of the discrepancy of each omitted data point from a surface fitted to all other data points [63]. This is a relatively common concept in statistical analysis [74]. It is shown in [24] that, using the 'leaving out one' lemma [113] the generalised cross validation for the multidimensional case can be calculated implicitly and hence efficiently by

$$GCV(\lambda) = \frac{(\mathbf{z} - A(\lambda)\mathbf{z})^T V^{-1} (\mathbf{z} - A(\lambda)\mathbf{z})/n}{[tr(I - A(\lambda))/n]^2} \tag{2.23}$$

It is demonstrated in Hutchinson [61], following a once only $O(n^3)$ tridiagonalisation of an $(n - M) \times (n - M)$ positive definite matrix, the value of expression (2.23) for a given λ can be calculated in $O(n)$ operations. A theoretical justification for using the GCV to determine the optimum thin plate spline function is given in [24], where it is shown that, if $\hat{\lambda}$ is the minimiser of the true error and λ^* is the minimiser of the GCV, then

$$\lim_{n \to \infty} \frac{R(\hat{\lambda})}{R(\lambda^*)} = 1 \tag{2.24}$$

Thus, in theory, as the number of data points increases, the minimiser of the GCV approaches the minimiser of the true error.

2.2.2 Estimating the variance of the noise

According to Wahba [113], determining the optimum surface by minimising the GCV also yields an estimate of σ^2, the variance of the noise. The estimate is given by

$$\hat{\sigma}^2 = \frac{(\mathbf{z} - A(\lambda)\mathbf{z})^T V^{-1} (\mathbf{z} - A(\lambda)\mathbf{z})}{tr(I - A(\lambda))} \tag{2.25}$$

Hutchinson [61] explains that $tr(I - A(\lambda))$ may be interpreted as the degrees of freedom of the residual sum of squares, and thus equation (2.25) is analogous to the estimate of σ^2 obtained in linear regression [113, 98]. It follows that the effective number of parameters of the fitted model, known as the *signal*, is given by $tr(A(\lambda))$ [62].

2.2.3 Interpretation of the signal

Hutchinson and Gessler [62] present evidence to show that the value of the signal is a useful diagnostic in its own right. They state that, in most applications, if the signal exceeds $n/2$, it is likely that the data are too sparse to adequately support spline interpolation. In the extreme case, exact interpolation corresponds to a signal equal to the number of data points. This implies that there is no measurement error and no microscale variation, which is generally an unrealistic assumption. It may indicate that the optimisation procedures have failed due to insufficient data [62], short range correlation in the data values [65], or autocorrelation in the error structure that has been unaccounted for by the model [28]. On the other hand, when the signal reaches its minimum value, a number which depends on the number of independent variables and the order of the derivative [68], the fitted spline is equivalent to a least squares regression of the data on the M monomials ϕ_j [62]. This results in complete global smoothing of the data. Extreme signal values can indicate a lack of spatial structure in the data.

2.2.4 Standard error estimates

According to Wahba [112] and Hutchinson [61], it can be shown using the multivariate prior distribution which gives rise to splines that the posterior covariance of the vector of the fitted values is given by the symmetric matrix $A(\lambda)V\sigma^2$. This result allows the estimation of the pointwise standard errors of the fitted spline estimate of g, as shown by Hutchinson [61]. The same result can be obtained for the kriging method. This involves using the multivariate prior distribution which underlies the kriging equations [63].

2.2.5 Geostatistical models

A useful interpretation of the process of fitting a thin plate smoothing spline to spatial, environmental data is provided by geostatistical models [25, 111, 74].

Geostatistical models are statistical models designed to incorporate, and take advantage of, the spatial dependencies inherent in environmental data [25]. Thin plate smoothing splines achieve essentially the same goal as geostatistical methods, and have a close association with the geostatistical framework. Certain aspects of geostatistical models of spatial data are therefore discussed below.

The variogram model

Starting with the model in equation (2.10), geostatistical methods model the process g as a spatially correlated random function. According to Wackernagel [111], the concept of a random function can be understood as follows. Take the data locations $\mathbf{t}_i$ and construct at each of the n locations a random variable $Z(\mathbf{t}_i)$. Now assume that these random variables are a subset of an infinite collection of random variables called a random function $Z(\mathbf{t})$ defined at any location $\mathbf{t}$ throughout the domain of interest. The data are assumed to be realisations of the random function at each data point location.

The random function is assumed to have 'intrinsic stationarity', which requires the stationarity of the first two moments of the difference of a pair of values at two points ie.

$$E[Z(\mathbf{t}+\mathbf{h}) - Z(\mathbf{t})] = 0, \tag{2.26}$$

$$\mathrm{var}[Z(\mathbf{t}+\mathbf{h}) - Z(\mathbf{t})] = 2v(\mathbf{h}) \tag{2.27}$$

where $\mathbf{h}$ is the displacement away from $\mathbf{t}$ and $v(\mathbf{h})$ is known as the variogram function. This assumes that the relationship between all pairs of random variables $Z(\mathbf{t}_i)$, $Z(\mathbf{t}_j)$ is the same joint probability distribution. The underlying implication is that the relationship between $Z(\mathbf{t})$ and $Z(\mathbf{t}+\mathbf{h})$ depends only on the length and orientation of $\mathbf{h}$, but not on the position of $\mathbf{h}$. In other words, the spatial trends depend only on the covariance structure of the random function $Z(\mathbf{t})$. In practice, the orientation of $\mathbf{h}$ is often not incorporated into the models of $v(\mathbf{h})$, and then the measure of spatial dependence varies only with the length of $\mathbf{h}$.

Using this geostatistical model, the method of kriging is used to predict the value of $Z(\mathbf{t})$ at a certain location $\mathbf{t}_0$. This is done by constructing a weighted linear combination of the realisations at the sample locations as follows

$$\hat{Z}(\mathbf{t}_0) = \sum_{i=1}^{n} w_i Z(\mathbf{t}_i) \tag{2.28}$$

where $\hat{Z}(\mathbf{t}_0)$ is the prediction of $Z(\mathbf{t}_0)$. The weights w_i are chosen such that $\hat{Z}(\mathbf{t}_0)$ is an unbiased, minimum error variance estimate. The importance of the covariance structure of $Z(\mathbf{t}_i)$ is in determining the weights. The estimation of the w_i requires calibration of a covariance function $C(\mathbf{h})$, which is the covariance between random variables separated by a displacement $\mathbf{h}$:

$$C(\mathbf{h}) = \mathrm{Cov}\{Z(\mathbf{t}), Z(\mathbf{t}+\mathbf{h})\} = E[Z(\mathbf{t})Z(\mathbf{t}+\mathbf{h})] - E[Z(\mathbf{t})]E[Z(\mathbf{t}+\mathbf{h})] \tag{2.29}$$

The covariance decreases with separation distance, and becomes zero at separation distances large enough so that realisations of $Z(\mathbf{t})$ are unrelated. In geostatistics, the variogram function in (2.29) is more commonly used as a measure of spatial continuity [111]. The variogram function can be deduced from a covariance function by the formula

$$v(\mathbf{h}) = C(\mathbf{0}) - C(\mathbf{h}) \tag{2.30}$$

but the reverse is not true, because the variogram is not necessarily bounded. The variogram is clearly the opposite of the covariance function, in that it generally increases with separation distance until a further increase no longer causes a corresponding increase in the average squared difference between pairs of values. At this point, known as the *range*, the variogram reaches a plateau. The value associated with the plateau the variogram reaches at the range is called the *sill*. In terms of the covariance function the *sill* for the process $Z(\mathbf{t})$ is given by $C(\mathbf{0})$, and is thus denoted by σ_Z^2 [25, 74].

The scale of variation

The variogram model is useful for analysing the spatial variation in the random process $Z(\mathbf{t})$. Cressie [25] presents the following decomposition of the random process $Z(\mathbf{t})$:

$$Z(\mathbf{t}) = \mu(\mathbf{t}) + W(\mathbf{t}) + \eta(\mathbf{t}) + \epsilon(\mathbf{t}) \tag{2.31}$$

where

- $\mu = E(Z)$ is a deterministic mean structure known as large scale variation.
- $W(\mathbf{t})$ is a zero mean, $\mathcal{L}^2$ continuous, intrinsically stationary process whose variogram range is larger than $\min\{||\mathbf{t}_i - \mathbf{t}_j|| : 1 \leq i < j \leq n\}$, the minimum separation distance. This process accounts for the spatial dependence in the data values.
- $\eta(\mathbf{t})$ is a zero mean, intrinsically stationary process whose variogram range is smaller than $\min\{||\mathbf{t}_i - \mathbf{t}_j|| : 1 \leq i < j \leq n\}$
- $\epsilon(\mathbf{t})$ is a zero-mean random noise process attributed to measurement error.

The assumption of intrinsic stationarity implies that the process $Z(\mathbf{t})$ is spatially continuous, with zero semivariance at zero separation distance. In reality, sample values separated by extremely small distances may be quite dissimilar. Cressie [25] attributes this to a combination of the sampling error $\epsilon(\mathbf{t})$ and discontinuity in the microscale process $\eta(\mathbf{t})$. It is often reasonable to expect the microscale variation in the physical process to be spatially coherent, but little can be known about it as the data observations are spread too far apart to represent it. If the microscale process is continuous, then the errors in the approximation of g will be correlated at close separations. Due to the earth sciences origin of geostatistics, the discontinuity at the origin is termed the nugget effect. The nugget value c_0 is given by

$$c_0 = \lim_{||\mathbf{h}|| \to 0} v(\mathbf{h}) = c_{ME} + c_{MS} \tag{2.32}$$

where c_{ME} is the variance of the noise process $\epsilon(\mathbf{t})$ and c_{MS} is the nugget effect of the microscale process. The variogram model clearly assumes that the nugget variance is constant across the data network, an assumption that the thin plate smoothing spline model avoids by incorporating the matrix V. A typical variogram for the process $Z(\mathbf{t})$ is depicted in Figure 2.4. As stated above, little is known about the process $\eta(\mathbf{t})$, and so in practice the variogram at separation distances closer than the minimum distance between data points is usually extrapolated through the vertical axis. The processes shown

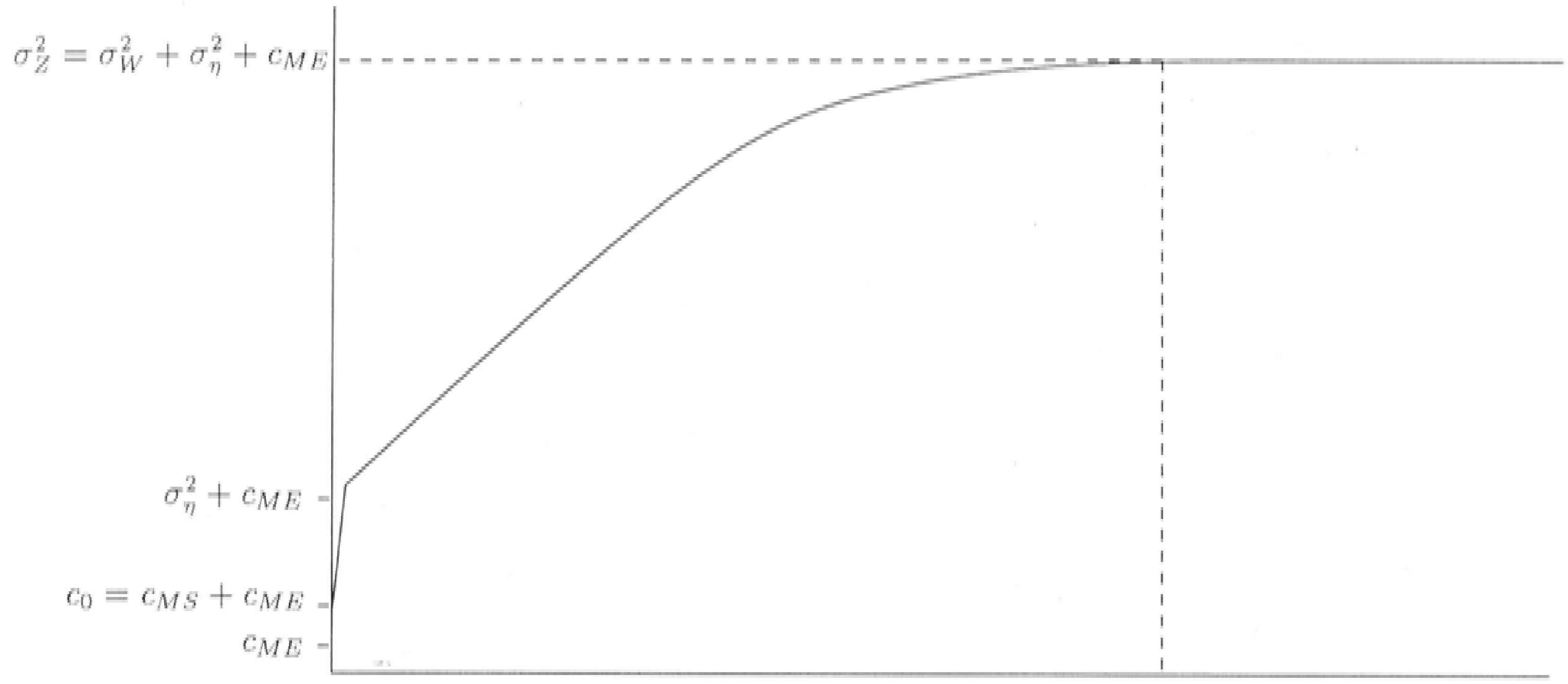


Figure 2.4: Variogram components.

in Figure 2.4 are also implicitly present in the smoothing spline model in equation (2.10). Wahba [113] has shown that thin plate smoothing splines and a particular type of kriging, known as universal kriging, are formally equivalent. Variograms and other covariance structures are not explicitly used in the thin plate spline fitting process. However, these structures, created from observed data values, are a useful method of visualising the spatial dependence in the data.

Chapter 3

Discretisation of the univariate smoothing spline equations

There are a number of possible approaches to discretising equation (2.9) to obtain a series of equations suitable for numerical solution. A standard method would be piecewise constant discretisation. However smooth basis elements offer a better structure for approximating smooth functions, particularly when used in conjunction with the multigrid method of numerical solution, as will be discussed in later chapters. This study has adopted the B-spline framework for approximating the thin plate smoothing spline function [27]. The simplest first order B-spline representation is to represent functions by piecewise constants. Such functions do not have continuous derivatives, but derivatives can be approximately represented by taking finite differences of the piecewise constant values. Third order B-splines represent functions by piecewise quadratic elements. These functions have continuous first derivatives and piecewise constant second derivatives. This study has developed procedures for discretising the thin plate smoothing spline problem using both piecewise constants and quadratic elements, or third order B-splines. These formulations are discussed below. A comparison of the results for both forms of discretisation is discussed in the Part 2 of this thesis.

While this project is directed towards smooth bivariate spatial interpolation, a thorough analysis of univariate interpolation was first conducted. It was found that the simplicity of the univariate case allowed empirical analysis to be conducted in a transparent and detailed fashion. This was ideal given that existing theory relating to the methods used in this project is not comprehensive, and user experimentation is required to understand the algorithms and optimise their performance. Note, however, that $O(n)$ algorithms, where n is the number of data points, already exist for calculating analytic univariate thin plate smoothing splines [69].

3.1 Discretisation with piecewise constants

Consider an approximation $f(x)$ to the natural cubic spline minimising expression (2.9). In the case of piecewise constants, $f(x)$ will be a discrete approximation represented on a grid labelled as in Figure 3.1 where $\gamma_0, \ldots, \gamma_N$ are knot points and $f_0, \ldots, f_{N-1}$ are the values of the discretised function in each grid cell.

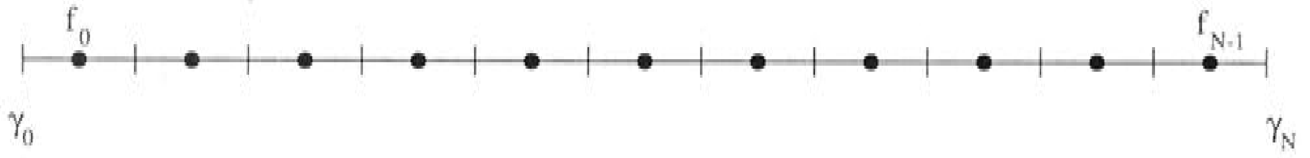


Figure 3.1: The unidimensional grid.

There are $N+1$ grid points or 'knot points' and N grid cells. The integral term in equation (2.10) was approximated across the N grid cells using second differences to give

$$\lambda \int_{-\infty}^{\infty} (f''(x))^2 \, dx = \lambda \frac{1}{h^3} \sum_{I=1}^{N-2} (f_{I+1} - 2f_I + f_{I-1})^2 \tag{3.1}$$

Note that the discretised integral only covers the interval $[\gamma_0, \gamma_N]$, not the infinite line. This does not affect the system, due to the natural boundary conditions in (2.6). Using (3.1), the minimisation problem can be expressed in fully discretised form as

$$\text{Minimise} \qquad ||P\mathbf{f} - \mathbf{z}||^2 + \frac{\lambda}{h^3} ||Q\mathbf{f}||^2 \tag{3.2}$$

where $\mathbf{f}$ is a vector of length N containing the values of the fitted function f across the grid cells. The vector $\mathbf{z}$ contains values of the data points. It has components z_i, $i = 1, \ldots, n$. The matrix Q is an $(N-2) \times N$ matrix of the form

$$\begin{pmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & & \ddots & & & \\ & & & & 1 & -2 & 1 \end{pmatrix} \tag{3.3}$$

This matrix operates on $\mathbf{f}$ to calculate finite difference second derivatives. The matrix P selects from $\mathbf{f}$ the grid values at grid cells containing data points, so

$$[P\mathbf{f}]_i = f_I \tag{3.4}$$

where f_I is the value of $\mathbf{f}$ in the I^{th} grid cell, which contains data point i. Each row of P has exactly one non-zero element. If data point i is in gridcell I, the element $[P]_{iI}$ will be 1. If there is no data point in grid cell I the I^{th} column of P will be zero.

Differentiating expression (3.2) with respect to the vector $\mathbf{f}$ gives and equating to zero for minimisation gives the system

$$(P^TP + \frac{\lambda}{h^3}Q^TQ)\mathbf{f} = P^T\mathbf{z} \tag{3.5}$$

The matrix P^TP is the $N \times N$ diagonal matrix

$$\begin{pmatrix} \ddots & & \\ & m_{II} & \\ & & \ddots \end{pmatrix} \tag{3.6}$$

where m_{II} is the number of data points in the I^{th} grid cell. This can be seen by considering the example of three data points in the first grid cell, which makes the first three elements of the first column of the matrix P equal to 1 and the remaining elements in the first column equal to zero. Thus $[P^TP]_{11}$ equals 3. The right hand side 'data' term $P^T\mathbf{z}$ effectively sums the values of the data points occurring in the same grid cell. The i^{th} entry of vector $P^T\mathbf{z}$ is zero if there are no data points in the I^{th} grid cell. Thus data points occurring in the same grid cell are effectively averaged, with the sums on the right and side and the counts in the diagonal matrix on the left hand side. The matrix

Q^TQ is a symmetric, non-negative definite $N \times N$ matrix of the form

$$\begin{pmatrix} 1 & -2 & 1 & & & & & \\ -2 & 5 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & 1 & & & \\ & 1 & -4 & 6 & -4 & 1 & & \\ & & & & \ddots & & & \\ & & & & 1 & -4 & 5 & -2 \\ & & & & & 1 & -2 & 1 \end{pmatrix} \tag{3.7}$$

It can be seen that, assuming there are no data points in the first and last grid cells, the second differences at the endpoints are set to zero. This is a result of the minimisation condition enforcing zero second derivatives at the boundary. The rank deficiency of Q^TQ is 2. The addition of the matrix P^TP prevents this from being a rank deficient system. However, the system will approach rank deficiency if the value of λ/h^3 is large.

3.2 Discretisation with quadratic B-splines

3.2.1 B-spline formulation

B-spline definition

As an alternative to piecewise constants, spline functions known as quadratic, or third order B-splines, were used as the building blocks to approximate the function $f_\lambda(x)$. The B-spline framework was developed by Schoenberg [100] and is extensively documented by de Boor [27]. It incorporates piecewise constants as a special case. B-splines allow efficient and flexible representation of spline functions, and are used in a variety of multivariate function estimation problems [3, 34, 52, 81]. A practical introduction to B-splines is given in the following paragraphs.

In one dimension, the r^{th} normalised B-spline of degree k, and 'order' $k+1$, with knots $\gamma_r, \ldots, \gamma_{r+k+1}$ is defined as

$$B_{r,k+1}(x) = (\gamma_{r+k+1} - \gamma_r)\, \triangle_t^{k+1}\, (\gamma_r, \ldots, \gamma_{r+k+1})(t-x)_+^k \tag{3.8}$$

where $\triangle_t^{k+1}$ is the k^{th} divided difference operator. The above expression takes the k^{th} divided difference of $(t-x)_+^k$ evaluated at the points $\gamma_r, \ldots, \gamma_{r+k+1}$. The subscript t is used to indicate that the divided difference of the function $(t-x)_+^k$ of two variables is to be taken by fixing x and considering $(t-x)_+^k$ to be a function of t alone [27].

The rationale behind definition (3.8) given in de Boor [27]. The essential element of this definition is the truncated power function, which is itself a simple spline function [27]. It is given by

$$(t-x)_+^k = \begin{cases} (t-x)^k & \text{if } t \geq x \\ 0 & \text{if } t < x \end{cases} \tag{3.9}$$

The fact that the k^{th} derivative of this function is zero on the left side of the discontinuity means that the B-spline (3.8) 'has small support' ie

$$B_{r,k+1}(x) = 0 \text{ for } x \notin [\gamma_r, \gamma_{r+k+1}] \tag{3.10}$$

[27]. As an example of how this arises, consider the case of second order B-splines, where $k=1$. This gives

$$B_{r,2}(x) = (\gamma_{r+2} - \gamma_r)\, \triangle_t^2\, (\gamma_r, \gamma_{r+1}, \gamma_{r+2})(t-x)_+ \tag{3.11}$$

Letting $\gamma_r = 3$, with knot intervals of length 1, gives

$$2\, \triangle_t^2\, (3,4,5)(t-x)_+ \tag{3.12}$$

As $t-x$ is linear, the second difference $\triangle_t^2$ is always zero, except inside the interval $[3,5]$, where the discontinuity comes into effect. This can be seen by setting $x=4$, which gives

$$B_{r,2}(4) = 2(0+0+5-4) = 2 \tag{3.13}$$

This is not zero because $(3-4)_+$ is zero, as t is less than x.

It is also shown in de Boor [27] that a B-spline is always non-negative. Thus a visual concept of B-splines can be developed, looking at first, second and third order B-splines, as seen in Figure 3.2. It is shown in de Boor [27] that

B-splines are spline functions ie.

$$B_{i,k+1} \in \mathcal{S}_{k+1,\gamma}, \qquad \text{all } i \tag{3.14}$$

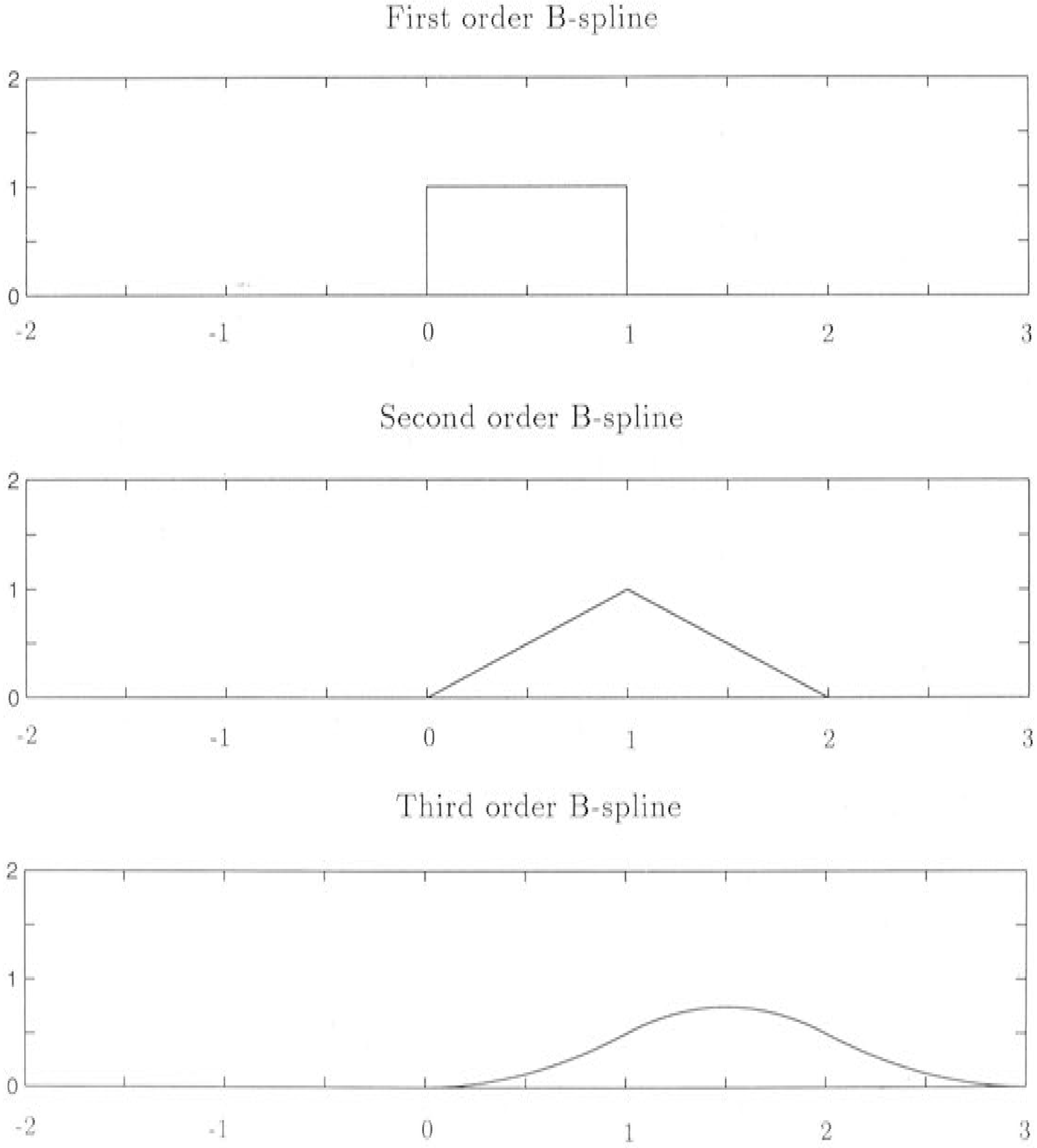


Figure 3.2: B-splines of varying orders.

Quadratic B-splines splines were selected for this analysis for practical reasons. Firstly, third order is the minimum order required for a visually 'smooth' function, because its first derivatives are continuous. It is desirable to choose functions of small order, to maintain small compact support. This allows the associated system of equations to be solved simply and efficiently by numerical methods. It is also shown in Marsden [89] that quadratic splines produce better fits to continuous functions than do cubic splines. Further favourable properties of quadratic B-splines are discussed in the following sections.

Fortunately, the formula for quadratic B-splines functions does not have to be derived from equation (3.8), thanks to a recursive formulation given in de Boor [27]. This is given by

$$\begin{aligned}\frac{B_{r,k+1}(x)}{\gamma_{r+k+1}-\gamma_r} &= \frac{x-\gamma_r}{(\gamma_{r+k}-\gamma_r)(\gamma_{r+k+1}-\gamma_r)}B_{r,k}(x)\\ &+\frac{\gamma_{r+k+1}-x}{(\gamma_{r+k+1}-\gamma_{r+1})(\gamma_{r+k+1}-\gamma_r)}B_{r+1,k}(x)\end{aligned} \tag{3.15}$$

Thus the expression for third order, or quadratic B-splines, can be derived from first order B-splines. A first order B-spline is simply

$$B_{r,1}(x) = \begin{cases} 1 & \text{if } \gamma_r \leq x \leq \gamma_{r+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

Using equation (3.15), the expression for third order B-splines is

$$\begin{aligned}B_{r,3} = &\frac{x-\gamma_r}{\gamma_{r+2}-\gamma_r}\left[\frac{x-\gamma_r}{\gamma_{r+1}-\gamma_r}B_{r,1}(x)+\frac{\gamma_{r+1}-x}{\gamma_{r+3}-\gamma_{r+1}}B_{r+1,1}(x)\right]\\ +&\frac{\gamma_{r+3}-x}{\gamma_{r+3}-\gamma_{r+1}}\left[\frac{x-\gamma_{r+1}}{\gamma_{r+2}-\gamma_{r+1}}B_{r+1,1}(x)+\frac{\gamma_{r+3}-x}{\gamma_{r+3}-\gamma_{r+2}}B_{r+2,1}(x)\right]\end{aligned} \tag{3.17}$$

This can be clarified by considering the polynomial piece over each knot interval of the quadratic B-spline basis element. Assume we have equally spaced knot intervals of length h. We can then represent the quadratic B-spline in terms of $w_r = (x-\gamma_r)/h$, the proportion of the distance along the r^{th} knot interval.

For the basis element $B_{r,3}(x)$, the polynomial pieces are

$$B_{r,3}(x) = \begin{cases} \frac{w_r^2}{2} & \gamma_r \leq x < \gamma_{r+1} \\ w_{r+1}(1 - w_{r+1}) + \frac{1}{2} & \gamma_{r+1} \leq x < \gamma_{r+2} \\ \frac{(1-w_{r+2})^2}{2} & \gamma_{r+2} \leq x < \gamma_{r+3} \end{cases} \tag{3.18}$$

These polynomial pieces are depicted in figure 3.3.

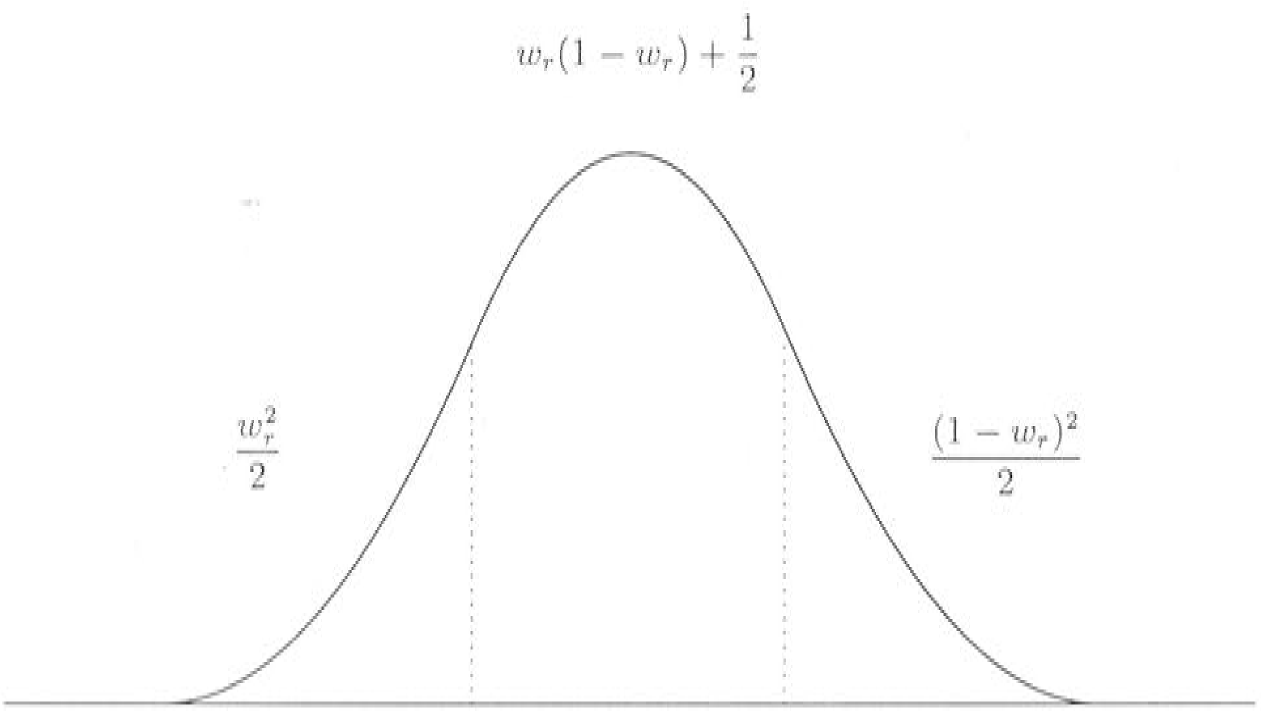


Figure 3.3: Polynomial pieces of a quadratic B-spline, with w_r ranging from 0 to 1 on each knot interval.

B-splines as a vector space basis

The spline vector space was discussed in Chapter 2. It is shown in de Boor [27] that a sequence of B-splines $B_{0,k+1}, \dots, B_{N+k-1,k+1}$ is a basis for the space $\mathcal{S}_{k,\gamma}$, defined in Chapter 2. De Boor [27] presents the definition of a spline function in terms of B-splines, stating that a spline function of order $k + 1$ with knot sequence γ is any linear combination of B-splines of order $k + 1$ for the knot sequence γ. The advantage of representing splines in terms of B-spline basis elements is that the spline can be constructed over all knot intervals by simply calculating the coefficients of the basis vectors for the spline space. The basis vectors are themselves splines and have the desired continuity properties. This ensures that all the other vectors of the space, which are linear combinations of the basis, have the same properties.

Any spline function of degree k defined on $[\gamma_k, \gamma_{N+k}]$ can be represented as

$$f(x) = \sum_{r=0}^{N+k-1} \alpha_r B_{r,k+1}(x) \tag{3.19}$$

where α_r are called the B-spline coefficients of $f(x)$. There are $N+1$ knot intervals and N spans. Note that the B-spline basis is normalised, so

$$\sum_{r=0}^{N+k-1} B_{r,k+1}(x) = 1 \tag{3.20}$$

A piecewise constant representation corresponds to a construction of $f(x)$ using B-splines of order 1. The B-spline coefficients correspond to the values of the piecewise constant function on the knot interval $[\gamma_r, \gamma_{r+1}]$. Clearly the dimension of the space of first order B-splines is N, while the dimension of the space of third order B-splines is $N+2$. This means that, in the case of third order B-splines, there are more basis elements than spans. In order to represent a third order spline in terms of a quadratic spline basis we included two extra basis elements, centered a distance $h/2$ outside the first and last spans, as can be seen from Figure 3.4. Note that $f(x)$ is not defined on the exterior regions $[\gamma_0, \gamma_2]$ and $[\gamma_{N+2}, \gamma_{N+4}]$.

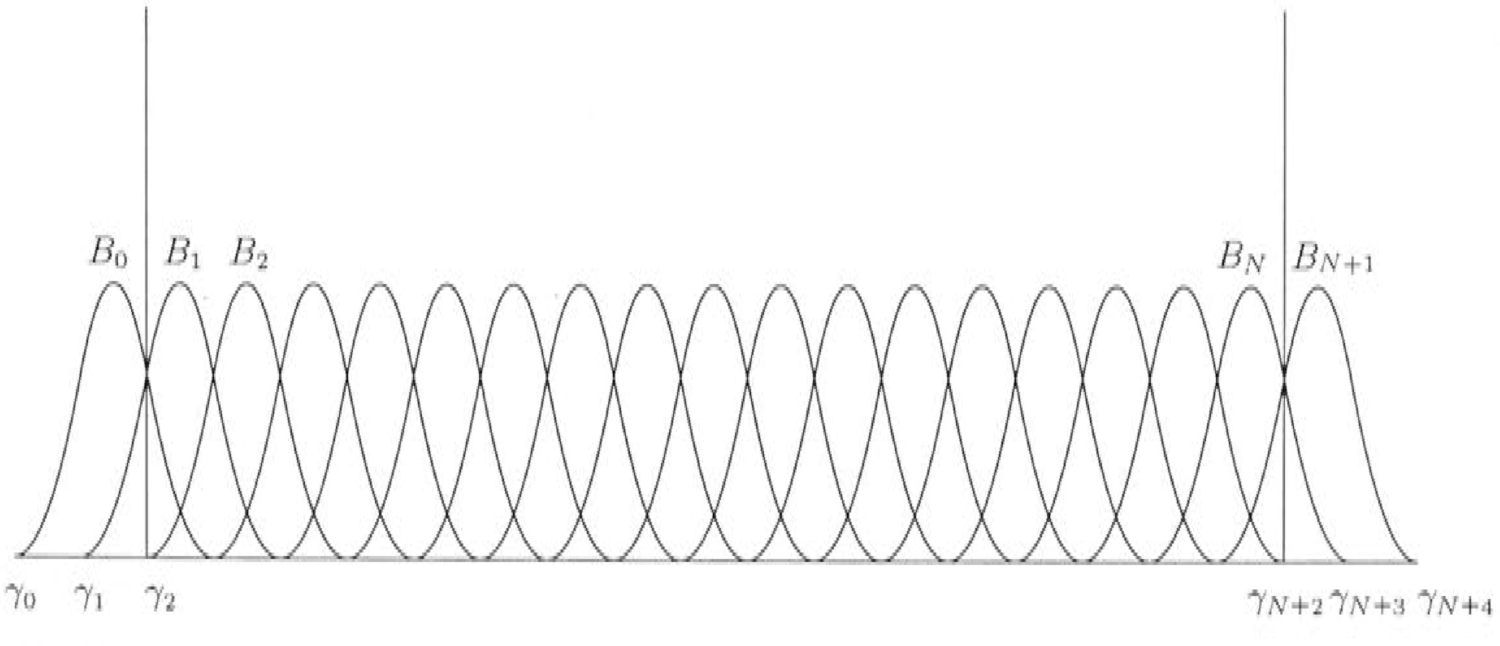


Figure 3.4: Positions of quadratic B-spline basis elements B_r on the unidimensional grid.

3.2.2 B-spline approximation of smoothing splines

Starting with the minimisation problem (2.9) a quadratic B-spline approximation $f(x)$ is formulated as

$$f(x) = \sum_{r=0}^{N+1} \alpha_r B_{r,3}(x) \tag{3.21}$$

Thus f is a spline function composed of quadratic B-splines, and is defined on the interval $[\gamma_2, \gamma_{N+2}]$. We substitute the approximation f into the minimisation expression (2.9) and choose the coefficients α_r to minimise. This requires differentiating (3.21). The derivative of a B-spline $B_{r,k}(x)$ is given in page 138 of De Boor [27] as

$$\frac{dB_{r,k}(x)}{dx} = (k-1)\left(\frac{-B_{r+1,k-1}(x)}{\gamma_{r+k} - \gamma_{r+1}} + \frac{B_{r,k-1}(x)}{\gamma_{r+k-1} - \gamma_r}\right) \tag{3.22}$$

which implies that

$$\frac{d}{dx}\left(\sum_{r=0}^{N+1} \alpha_r B_{r,k+1}\right) = \sum_{r=1}^{N+1} \frac{k(\alpha_r - \alpha_{r-1})}{\gamma_{r+k} - \gamma_r} B_{r,k-1}(x) \tag{3.23}$$

[27]. It can be seen from this expression that the first derivative of a spline function f can be found simply by differencing its B-spline coefficients. For quadratic splines with knots equally spaced at intervals of length h this gives

$$f'(x) = \sum_{r=1}^{N+1} \frac{(\alpha_r - \alpha_{r-1})}{h} B_{r,2}(x) \tag{3.24}$$

$$f''(x) = \sum_{r=2}^{N+1} \frac{\alpha_r - 2\alpha_{r-1} + \alpha_{r-2}}{h^2} B_{r,1}(x) \tag{3.25}$$

Thus the second derivative of a quadratic B-spline is given by taking the second difference of its coefficients. The simplicity of this formulation further motivates the choice of quadratic B-splines for use in this study. The roughness penalty term of the minimisation problem in equation (2.9) is given by

$$\lambda \int (f''(x))^2 \, dx = \lambda h \sum_{r=2}^{N+1} \left(\frac{\alpha_r - 2\alpha_{r-1} + \alpha_{r-2}}{h^2} \right)^2 = \frac{\lambda}{h^3} \sum_{r=2}^{N+1} (\alpha_r - 2\alpha_{r-1} + \alpha_{r-2})^2 \tag{3.26}$$

Equation (3.26) is the same as equation (3.1), with the $\mathbf{f}$ values in the roughness penalty now substituted for $\boldsymbol{\alpha}$ values, where $\boldsymbol{\alpha}$ is a vector containing the $N+2$ quadratic B-spline coefficients α_r. The minimisation problem is therefore given by

$$||P\boldsymbol{\alpha} - \mathbf{z}||^2 + \frac{\lambda}{h^3}||Q\boldsymbol{\alpha}||^2 \tag{3.27}$$

The matrix Q is once again given by (3.3). The matrix P operates on $\boldsymbol{\alpha}$ to calculate values of $f(x)$ at data point locations i and is given by

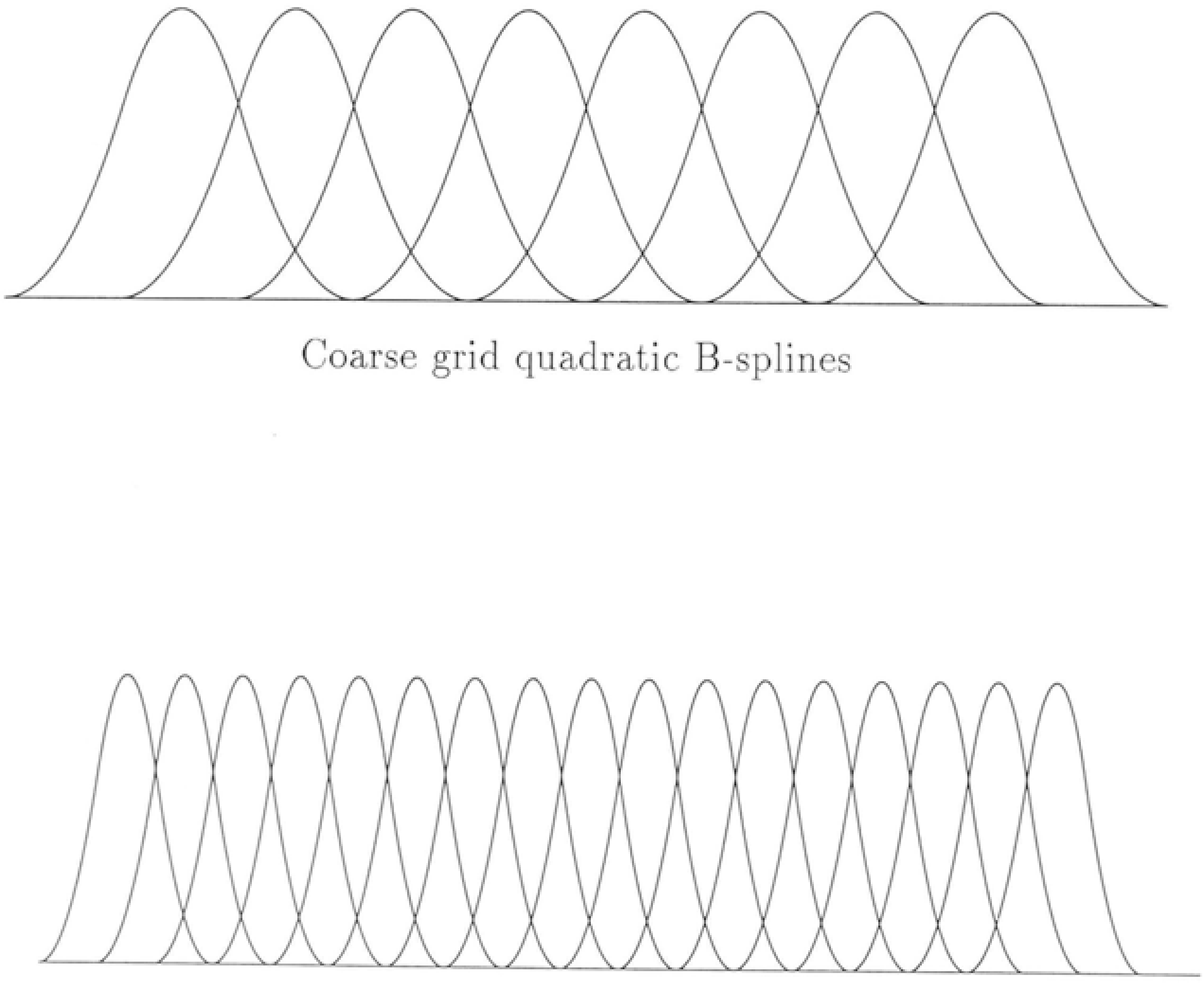
$$[P]_{ir} = B_{r,3}(x_i) \tag{3.28}$$

There are at most 3 non zero B-splines in any knot interval, as can be seen from Figure 3.4, so the matrix P has no more than 3 non zero entries in each row. Differentiating (3.27) and equating to zero once again gives

$$(P^T P + \frac{\lambda}{h^3} Q^T Q)\boldsymbol{\alpha} = P^T \mathbf{z} \tag{3.29}$$

The matrix $P^T P$ is 5 banded, which requires more storage than the diagonal systems arising from first order B-spline discretisations. However, the results presented in Chapter 11 show that the ability of quadratic B-splines to produce smooth functions at very coarse discretisations is essential to the development of an efficient algorithm for the purposes of this study.

Solution of equation (3.29) gives coefficients for B-splines with a support width of $3h$. Thus on the coarse grid there will be few B-splines of relatively large support, and on the fine grid there will be more B-splines with smaller support, as demonstrated in Figure 3.5. In the next chapter, multigrid methods of solving discretised systems such as those mentioned above will be discussed.



Coarse grid quadratic B-splines

Fine grid quadratic B-splines

Figure 3.5: Quadratic B-spline basis elements on coarse and fine grids.

Chapter 4

Multigrid methods

The basic idea behind multigrid methods is simply to perform the process of iterative solution of a system of equations on several grids of varying resolution, rather than just iterating on the resolution at which the solution is desired. This initial concept is central to the algorithm developed in this study. A review of multigrid methods is therefore given in this chapter, at a basic level, to demonstrate the value of multigrid principles to the algorithm presented in this project.

4.1 Classical iterative methods

A basic analysis of classical iterative methods was useful in understanding the multigrid schemes involved in this study and interpreting their behaviour. Classical iterative methods include variations of the Jacobi method, the Gauss-Seidel method, and successive overrelaxation (SOR) [78]. These methods are discussed in detail in Young [117], Hageman and Young [47] and Briggs [20]. Classical iterative methods, often called basic iterative methods, are used to perform 'smoothing' iterations during the multigrid process. Iteration using basic iterative methods is often termed 'relaxation'. In particular, this study examined the weighted Jacobi method and the SOR method. Aspects of these methods that are relevent to this study are discussed below.

4.1.1 The weighted Jacobi method

Assume we wish to solve the linear system

$$A\mathbf{f} = \mathbf{z} \tag{4.1}$$

where A is an $N \times N$ matrix, and the $N \times 1$ vector $\mathbf{f}$ is the solution for the $N \times 1$ right hand side vector $\mathbf{z}$. Classical iterative methods typically begin by partitioning the matrix A as follows

$$A = D + L + U \tag{4.2}$$

where L is the lower triangular part of A, U is the upper triangular part of A, and D is a diagonal matrix containing the diagonal elements of A.
The Jacobi iteration is given by

$$D\mathbf{f}^{k+1} = (-L - U)\mathbf{f}^k + \mathbf{z} \tag{4.3}$$

where $\mathbf{f}^k$ is the approximation to $\mathbf{f}$ after k Jacobi iterations. The updates $\mathbf{f}^{k+1}$ are easy to calculate because D is easy to invert.
The performance of the Jacobi iteration can be enhanced by taking a weighted average of the current approximation and its update [20]. Set

$$D\mathbf{f}^* = (-L - U)\mathbf{f}^k + \mathbf{z} \tag{4.4}$$

The weighted Jacobi iteration is then given by the convex combination

$$\mathbf{f}^{k+1} = (1 - \omega)\mathbf{f}^k + \omega\mathbf{f}^* \tag{4.5}$$

where ω is a real number chosen by the user. When $\omega = 1$ the weighted Jacobi method reduces to the Jacobi method.

4.1.2 The SOR method

The SOR iteration is given by

$$(D + \omega L)\mathbf{f}^{k+1} = [(1 - \omega)D - \omega U]\,\mathbf{f}^k + (1 - \omega)\mathbf{z} \tag{4.6}$$

The only difference between this iteration and the weighted Jacobi iteration is that, by incorporating L into the left hand side, it uses the updated components ${f_i}^{k+1}$ in the calculation of the components $\mathbf{f}^{k+1}$ that have yet to be updated. This can be seen by writing the SOR iteration in terms of the vector components as follows

$$f_i^{k+1} = \omega \frac{z_i}{d_{ii}} - \omega \left(\sum_{j=1}^{i-1} \frac{l_{ij}}{d_{ii}} f_j^{k+1} + \sum_{j=i+1}^{N} \frac{u_{ij}}{d_{ii}} f_j^k \right) + (1-\omega) f_i^k \tag{4.7}$$

and comparing this to the weighted Jacobi iteration

$$f_i^{k+1} = \omega \frac{z_i}{d_{ii}} - \omega \left(\sum_{j=1}^{i-1} \frac{l_{ij}}{d_{ii}} f_j^{k} + \sum_{j=i+1}^{N} \frac{u_{ij}}{d_{ii}} f_j^k \right) + (1-\omega) f_i^k \tag{4.8}$$

where $1 \leq i \leq N$. SOR is a more efficient method than weighted Jacobi, given that its updates use more recently updated information, but both were examined because the Jacobi iteration is easier to analyse.

4.1.3 Convergence

The weighted Jacobi and SOR iterations can be expressed as

$$\mathbf{f}^{k+1} = G\mathbf{f}^k + K\mathbf{z} \tag{4.9}$$

where G is known as the 'iteration matrix'. In the case of weighted Jacobi

$$G = (1-\omega)I + \omega D^{-1}(-L-U) \tag{4.10}$$

and

$$K = \omega D^{-1} \tag{4.11}$$

For SOR

$$G = (D+\omega L)^{-1}\left[(1-\omega)D - \omega U\right] \tag{4.12}$$

and

$$K = (D+\omega L)^{-1}\omega \tag{4.13}$$

It is easy to show that the iterations (4.9) will only converge if $S(G) < 1$ where $S(G)$ is the spectral radius, or the eigenvalue of maximum magnitude, of G [104]. If $\mathbf{e}^k$ is the error $\mathbf{f} - \mathbf{f}^k$ after k iterations, then

$$\mathbf{e}^{k+1} = G\mathbf{e}^k \tag{4.14}$$

and

$$\mathbf{e}^k = (G)^k\mathbf{e}^0 \tag{4.15}$$

so

$$\|\mathbf{e}^k\| = \|G\|^k\|\mathbf{e}^0\| \tag{4.16}$$

It is stated in Hageman and Young [47] that $S(G) < 1$ for the Jacobi iteration if the matrix A is symmetric positive definite, and is irreducible with weak diagonal dominance. The weighted Jacobi iteration will converge under these conditions with the additional requirement that $0 < w \leq 1$. The SOR method converges provided that A is symmetric positive definite, and $0 < \omega < 2$.

4.2 The multigrid method

The literature that is most relevant to this study strongly indicates that multigrid techniques are well suited to the algorithm proposed. As discussed in Chapter 1, Hutchinson [58] and Altas [2] found multigrid to be a simple, efficient strategy for solving systems similar to the one considered here. The following review was conducted to give an understanding of the multigrid process in the context of the thin plate smoothing spline problem.

4.2.1 Multigrid context

With the growth in computational mathematics during recent decades, there has been intensive research into the problem of numerically solving discretised linear systems. Typically, the algorithms fall into two classes: direct and iterative methods [103]. Direct methods, such as Gaussian elimination, fast Fourier transforms and cyclic reduction, calculate the solution exactly to within machine precision. These methods are very efficient for certain types of problems, such as separable, self adjoint boundary value problems [103]. They are also not dramatically affected by the conditioning of the system. However, they are computationally intensive, and require excessive amounts of computer memory when applied to large, dense problems [103].

Iterative methods are more widely applicable than direct methods. The class of iterative methods consists of the basic iterative methods discussed in section 4.1, as well as other more advanced methods including alternating directions implicit (ADI), Chebyshev semi-iterative methods, conjugate gradient methods and multigrid methods [78]. Iterative methods approximately solve the discretised system by starting from an initial guess and, using some computational cycle, obtaining increasingly accurate approximations until a desired accuracy is achieved [103]. Iterative methods are often better suited to computer solution than direct methods as they consist of a repetition of simple steps [103]. Moreover, they can take advantage of the sparse nature of finite difference systems of equations, and may require no computer memory in addition to the storage of the discretised domain. They are also generally more efficient than direct methods for multidimensional problems [103].

Multigrid methods were developed after study of the basic iterative methods revealed that, while these methods are very useful, they have two fundamental deficiencies that are manifestations of the same underlying property [103]. Firstly, the convergence factor, $S(G)$, is $1 - O(h^2)$ or $1 - O(h)$ where h is the size of the grid or mesh [103, 41]. Hence any attempt to improve accuracy by refining the grid will cause convergence to deteriorate [20]. Secondly, a related difficulty is that the smooth (or long frequency) modes of the error are very slowly reduced, so relaxation quickly stalls once high frequency modes are eliminated. Recognition of these problems lead to the development of multigrid. The multigrid process uses relaxation to obtain smooth errors on an initial grid, then calculates corrections to the approximate solution by relaxing on grids of varying coarseness. Or, in the case of this study, relaxation is performed on the coarsest grid to obtain a smooth initial guess, and then the grids are successively refined to develop fine scale structure. An optimal multigrid scheme has a convergence rate that is bounded away from 1, and is independent of the mesh size h, so we can increase the number of unknowns and not change the convergence rate. As a consequence an acceptable approximation of the discrete problem can be obtained at the expense of computational work proportional to the number of unknowns [46]. It has been shown in practice that suitable multigrid methods are at least competitive with direct fast solvers [17].

4.2.2 History of multigrid development

For about one century the Jacobi and Gauss-Seidel methods were the only tools for solving small linear systems iteratively [46]. Then, with the development of the conjugate gradient method by Hestenes and Stiefel [56], the idea of correcting blocks of unknowns instead of single unknowns was introduced. The first development of a correction process involving two grids was made by Fedorenko [32]. For Poisson's equation on a regular grid, Fedorenko [32] proved that the procedure reduced the residuals by a factor ϵ in $O(N|log\epsilon|)$ operations, where N is the number of grid points. This is asymptotically the optimal result, in that the algorithm is $O(N)$, and therefore scalable, for large ϵ. Bakhvalov [5] later generalised this result to any second order elliptic PDE with continuous coefficients. In 1973 the first full multigrid algorithms and results were published by Brandt [15]. However, the convergence properties of multigrid methods were still poorly understood. In 1975, Hackbusch [45] began to systematise convergence analyses of general multigrid methods. Subsequent developments in multigrid convergence theory have clarified a number of the uncertainties surrounding this issue, but further developments are required before convergence theories can match the convergence rates found in practice.

4.2.3 The multigrid principle

The multigrid principle is based on the recognition that a continuous problem can be discretised on a grid of any coarseness for numerical solution. Different discretisations have different accuracy and different convergence properties, both of which depend on the scale of the solution structure. Multigrid takes advantage of the fact that the discretisations of the same continuous problem are related [16].

Consider the system

$$A\mathrm{f} = \mathrm{z} \tag{4.17}$$

discussed in section 4.1. Briggs [20] establishes that, if A in this case is the finite difference matrix for the following boundary value problem

$$-f''(x)+\sigma f(x) = z(x) \qquad f(0) = f(1) = 0, \qquad 0 < x < 1, \qquad \sigma \geq 0 \quad (4.18)$$

then the eigenvectors of A are the same as the eigenvectors of G, the iteration matrix for the weighted Jacobi relaxation method. The error $\mathbf{e}^{(k)}$ can then be represented by an eigenvector expansion such as the following:

$$\mathbf{e}^{(k)} = \sum_{i=1}^{N} c_i \lambda_i^k \mathbf{w}_i \tag{4.19}$$

where c_i are the expansion coefficients, λ_i are the eigenvalues of G and $\mathbf{w}_i$ are the eigenvectors of A. The $\mathbf{w}_i$ are known as the 'modes' of the error. The j^{th} component of the i^{th} eigenvector is given by

$$w_{ij} = sin\left(\frac{ji\pi}{N}\right) \tag{4.20}$$

Thus the first mode $\mathbf{w}_1$ is a smooth sine curve, and the modes become increasingly oscillatory as i increases (see Figure 4.1).

Briggs [20] shows that

$$\lambda_1 \approx 1 - \frac{\omega^2 h^2 \pi^2}{2} \tag{4.21}$$

where ω is the relaxation factor for the weighted Jacobi method. This implies that the eigenvalue associated with the smoothest mode will always be close to 1. Thus no value of w will reduce the smooth components of the error effectively. However, if $w = 2/3$, the modes corresponding to $N/2 \leq i \leq N$ can be reduced by a factor of at least $1/3$ with each relaxation [20]. Arguments such as the above can also be developed for other relaxation methods and different model problems [20].

The next key concept behind the multigrid principle relates to the transfer of the error $\mathbf{e}^{(k)}$ to a coarser grid. It is demonstrated in Briggs [20] and Jespersen [78] that, for the 'smooth' modes $1 \leq i < N/2$, the mode becomes more oscillatory when passing from a fine grid to a coarse grid. This is demonstrated in Figure 4.2. The essential principle of multigrid is therefore to iterate or 'relax' on the fine grid to eliminate oscillatory components, then move to a coarse grid to make smooth components more oscillatory and continue relaxation. A

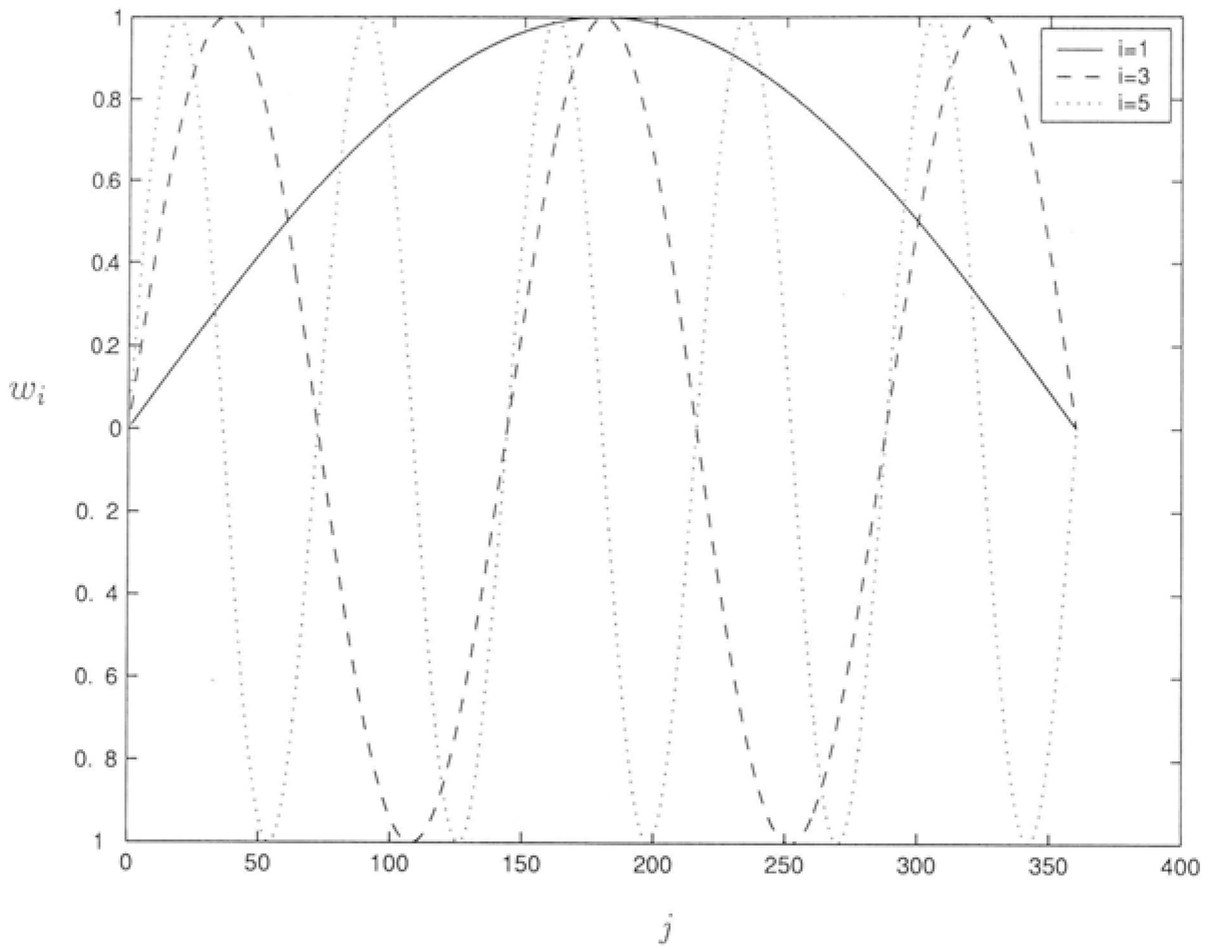


Figure 4.1: The modes $\mathbf{w}_i$.

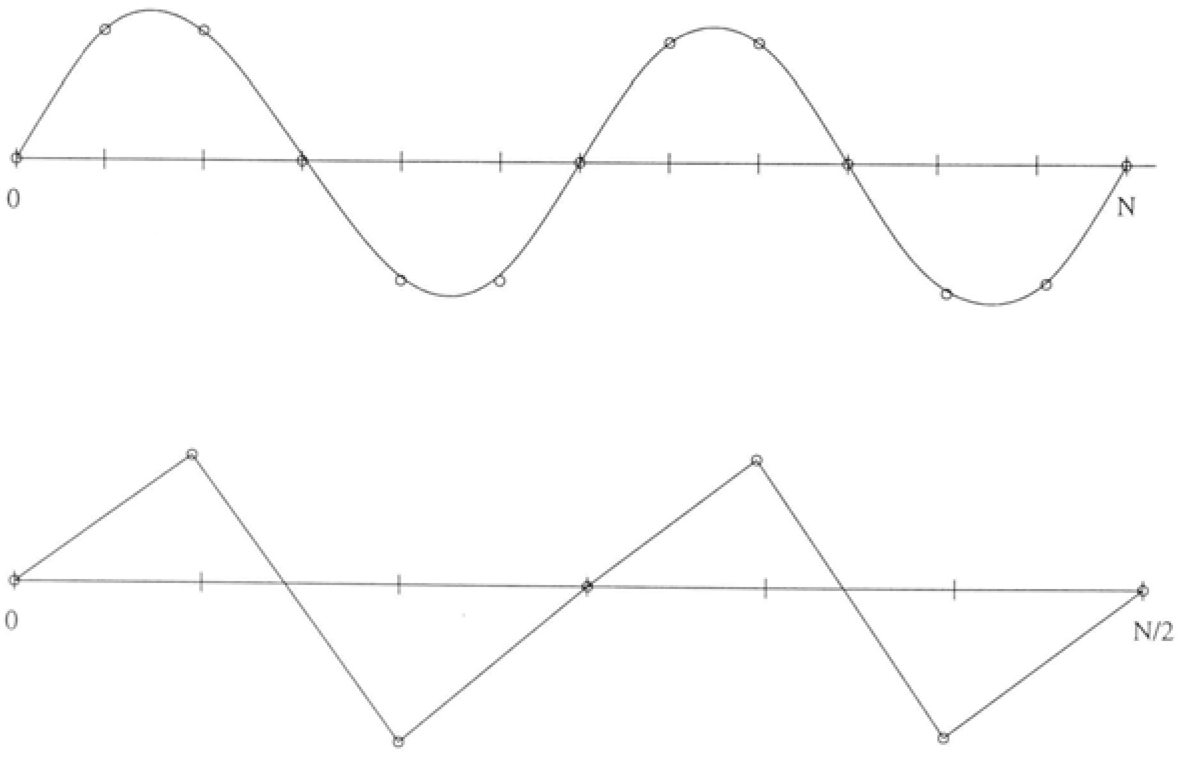


Figure 4.2: Representation of a given mode on a fine grid and a coarse grid (adapted from Briggs[20]).

smooth correction to the original approximation is then transferred back to the fine grid, leaving only the oscillatory modes to be reduced at the fine scale.

4.2.4 Elements of multigrid

As discussed above, the multigrid solution process involves relaxation, and transfer of the solution process between coarse and fine grid levels. This section will summarise the technical details of these processes and describe their role in the multigrid algorithm.

The residual equation

The first step in moving from a fine grid to a coarse grid involves the residual equation. If $\mathbf{f}_l$ is an approximation to the exact solution $\mathbf{f}$ then the error $\mathbf{e}_l = \mathbf{f} - \mathbf{f}_l$ satisfies the residual equation $A_l\mathbf{e}_l = \mathbf{z}_l - A_l\mathbf{f}_l = \mathbf{r}_l$. The subscript l denotes discretisation on grid level l, $l = 0, 1, 2, \ldots, L$, where a larger value of l corresponds to a coarser grid, and L is the level of the coarsest grid visited. Relaxing on the residual equation with an initial guess $\mathbf{e}_l = 0$ is equivalent to relaxing on the original equation with an arbitrary initial guess $\mathbf{f}_l$ [20]. For a particular grid level l, a simple form of multigrid, known as the coarse grid correction, proceeds through the following steps:

1. Relax on $A_l\mathbf{f}_l = \mathbf{z}_l$
2. Compute the residual $\mathbf{r}_l = \mathbf{z}_l - A_l\mathbf{f}_l$
3. Transfer the residual $\mathbf{r}_l$ to a coarser grid
4. Solve the residual equation directly on the coarse grid ie. $\mathbf{e}_{l+1} = A_{l+1}^{-1}\mathbf{r}_{l+1}$
5. Transfer the solution $\mathbf{e}_{l+1}$ to the fine grid
6. Correct the fine grid approximation $\mathbf{f}_l = \mathbf{f}_l + \mathbf{e}_l$
7. Relax on the fine grid

This procedure is repeated until a desired tolerance is reached, which might be the discretisation error. It uses the solution on the coarse grid to correct the fine grid solution. The coarse grid correction 'solves' for the smooth components of the solution, leaving only the oscillatory components to be obtained on the fine grid. Note that the residual equation can be solved directly on the coarse grid, however for large problems this can be quite expensive computationally. Also, the above procedure does not specify how the information is transferred between the grids. This is discussed below.

Prolongation

Transfer of approximations from a coarse grid to a fine grid is known as prolongation. A common method of prolongation is by a linear interpolation operator. The linear interpolation operator $T_{l+1,1}$ maps from grid $l+1$ to grid l and is of the form

$$T_{l+1,1} = \frac{1}{2} \begin{pmatrix} 1 & & & \\ 2 & & & \\ 1 & 1 & & \\ & 2 & & \\ & 1 & \ddots & \\ & & & 1 \\ & & & 2 \\ & & & 1 \end{pmatrix} \tag{4.22}$$

If there are $N/2$ grid cells on grid level $l+1$ then $T_{l+1,1}\mathbf{f}_{l+1}$ gives the following expression for the components of $\mathbf{f}_l$, ignoring the end conditions

$$f_{l,2i} = f_{l+1,i} \tag{4.23}$$

$$f_{l,2i+1} = \frac{1}{2}(f_{l+1,i} + f_{l+1,i+1}), \qquad 1 \leq i \leq N/2 - 1 \tag{4.24}$$

Clearly, linear interpolation will be more accurate when $\mathbf{f}_l$ is smooth. Other forms of interpolation, such as quadratic interpolation, may improve interpolation accuracy [46].

Restriction

The second class of intergrid transfer functions involves moving vectors from fine grids to coarse grids. They are known as restriction operators, denoted by $T_{l,2}$. A common form of restriction is called full weighting, where $T_{l,2}$ is of the form

$$T_{l,2} = \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 & & & & \\ & 1 & 2 & 1 & & & \\ & & & \ddots & & & \\ & & & & 1 & 2 & 1 \end{pmatrix} \tag{4.25}$$

[20]. $T_{l-1,2}\mathbf{f}_{l-1}$ gives the following expression for the components of $\mathbf{f}_l$

$$f_{l,i} = \frac{1}{4}(f_{l-1,2i-1} + 2f_{l-1,2i} + f_{l-1,2i+1}), \qquad 1 \leq i \leq \frac{N}{2} \tag{4.26}$$

There are, however, many possible choices of restriction operators. It is theoretically convenient to have $T_{l,2}^T = T_{l,1}$, although in practice similar results are observed when $T_{l,2}$ is the adjoint of a prolongation operator different from $T_{l,1}$ [46]. Some guidelines for choosing appropriate intergrid transfer operators are discussed in section 4.2.5.

It is shown in Wesseling [115] that, for a two grid multigrid procedure, if $\mathbf{r}_l$ is the residual $\mathbf{z}_l - \mathbf{A}_l\mathbf{f}_l$ after the coarse grid correction,

$$T_{l,2}\mathbf{r}_l = 0 \tag{4.27}$$

This relation implies that, on the fine grid, the residual after a coarse grid correction is 'rough', in that it consists only of oscillatory components. This explains why multigrid convergence is independent of h [115].

The coarse grid operator

There is some flexibility in what is used for A_{l+1} ,the operator A on grid level $l+1$. There are two common alternatives for the coarse grid operator. The first is the discretisation coarse grid operator, that is simply the operator that results from discretising the problem on the coarse grid [115]. Secondly there is the Galerkin coarse grid operator

$$A_{l+1} = T_{l,2}A_lT_{l,1} \tag{4.28}$$

where $T_{l,2}^T = T_{l,1}$. This alternative has some theoretical advantages in calculation of relative consistency [46] and convergence analysis [115]. In practice, however, similar results are achieved for both operators and the Galerkin form generally does not warrant the extra computation involved [46]. It is further stated in Brandt [17] that in principle, the coarse grid operator may be any reasonable difference operator approximating the fine grid operator.

The algorithm

There are a number of multigrid algorithms which combine the above elements. This review will discuss two multigrid algorithms known as v-cycle and nested grid, which form the basic building blocks for more complex multigrid algorithms.

V-cycle

The v-cycle algorithm is an extension of the coarse grid correction, or two-grid algorithm, discussed in section 4.2.4. The v-cycle algorithm is

```
for $l = 1$ to $L - 1$
    $\mathbf{f}_l = S_l{}^{v_1}(\mathbf{f}_l, \mathbf{z}_l)$
    $\mathbf{r}_l = \mathbf{z}_l - A_l\mathbf{f}_l$
    $\mathbf{z}_{l+1} = T_{l,2}\mathbf{r}_l$
end
$\mathbf{f}_L = S_L{}^{v_1}(\mathbf{f}_L, \mathbf{z}_L)$
for $l = L - 1$ to 1
    $\mathbf{f}_l = \mathbf{f}_l + T_{l+1,1}\mathbf{f}_{l+1}$
    $\mathbf{f}_l = S_l{}^{v_2}(\mathbf{f}_l, \mathbf{z}_l)$
end
$\mathbf{f}_1 = S_1{}^{v_2}(\mathbf{f}_1, \mathbf{z}_1)$
```

where v_1 is the number of 'pre-smoothing' operations, or the number of relaxations per grid level before the correction process begins (the first half of the cycle) and v_2 is the number of 'post-smoothing' operations, or the number of relaxations per grid level during the correction process, L is the number of grid levels, and S is the iteration matrix for the basic iterative method chosen. In multigrid terminology, relaxation on a given grid level is called 'smoothing' and S is called the smoothing iteration matrix. The notation $S^{v_1}(\mathbf{f}, \mathbf{z})$ corresponds to performing v_1 smoothing iterations on a generic system $A\mathbf{f} = \mathbf{z}$.

The 'multigrid schedule' for this algorithm, which dictates the order in which the grids are visited, is shown in Figure 4.3. V-cycle is used to replace the two grid algorithm not to improve convergence, but to avoid expensive computation of the exact solution of the coarse grid equation [46]. The idea is that

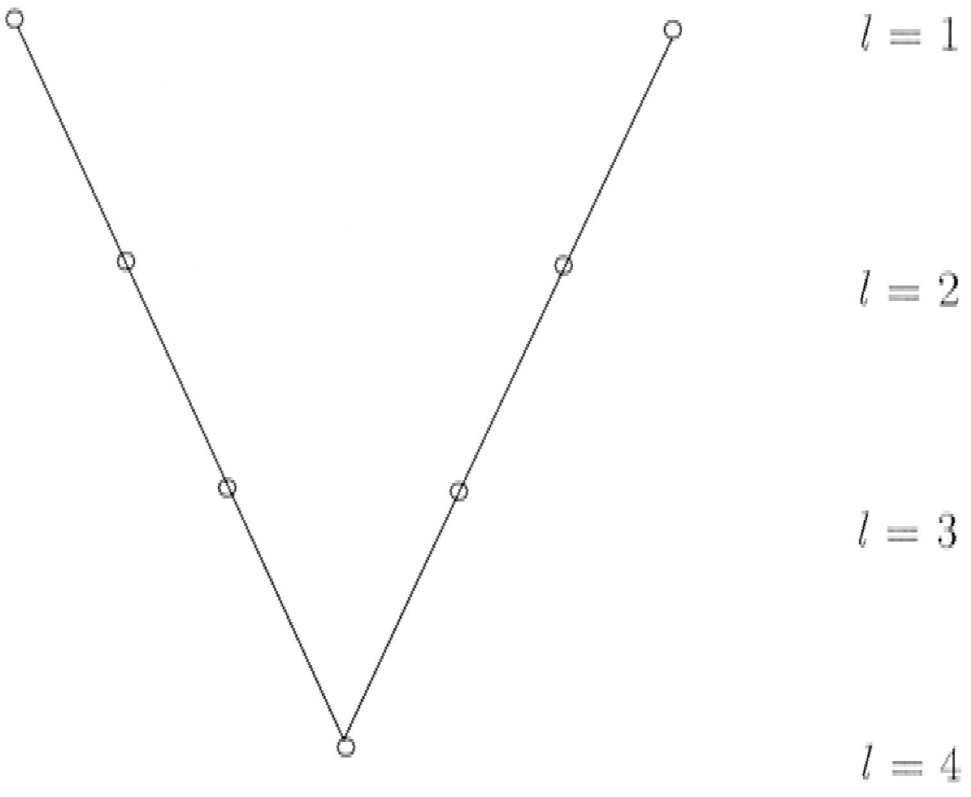


Figure 4.3: Grid schedule for the v-cycle algorithm.

the coarse grid correction takes care of the smooth error components on the current fine grid. Therefore, by adding relaxation at all levels, all components of the error are eventually acted on and quickly removed [20].

Nested grid

Nested grid is one of the simplest and earliest forms of multigrid. It does not involve coarse grid correction or the residual equation. The idea of nested iteration is to provide a good starting guess by means of iterating on a coarser grid. The algorithm can be described as follows

```
for $l = L$ to 2
        $\mathbf{f}_l = S_l{}^{v_1}(\mathbf{f}_l, \mathbf{z}_l)$
        $\mathbf{f}_{l-1} = T_{l,1}\mathbf{f}_l$
end
$\mathbf{f}_1 = S_1{}^{v_1}(\mathbf{f}_1, \mathbf{z}_1)$
```

The multigrid schedule for nested grid is shown in Figure 4.4. Unlike v-cycle,

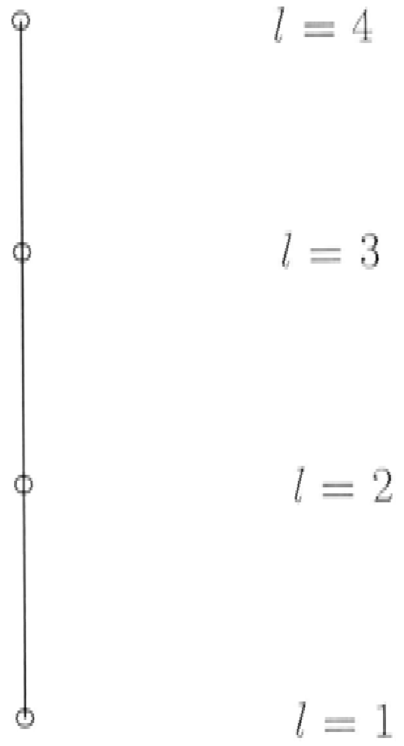


Figure 4.4: Grid schedule for the nested grid algorithm.

nested grid is not a cyclic algorithm, so some experimentation is involved in setting v_1 to achieve an appropriate level of accuracy. Nested grid operates under the simple principle that relaxation on a number of grid levels will eventually eliminate all components of the error. The coarse grids will remove smooth components and the finer grids will deal with oscillatory components.

The smoothing rate

On the basis of these arguments, a criterion for measuring the effectiveness of a multigrid method, known as the smoothing rate, was developed. One measure of the smoothing rate ρ is defined as

$$\rho = \sup_{l \geq 1} max\{|\lambda_j^l|^{v_1} : j = \psi_l, ..., N_l\} \tag{4.29}$$

where ψ_l is the value of j corresponding to the start of the domain of oscillatory modes on grid level l, N_l is the number of grid cells on level l and λ_j^l are the eigenvalues of the smoothing iteration matrix G on grid level l, denoted G_l [46]. This is the convergence factor for the relaxation method over the domain of the oscillatory modes, or the worst factor by which all high frequency error components are reduced per cycle. The smooth components of the error are ignored in the smoothing factor, because they are annihilated by iteration on coarse grids [115].

4.2.5 Guidelines for optimising the multigrid algorithm

From the above discussion it is clear that there are a lot of 'parameters' in the multigrid process that must be set by the practitioner, ranging from the type of algorithm chosen to the number of relaxations performed at each grid level [103]. Some theoretical principles that help guide the choice of multigrid parameters are outlined below. Despite these guidelines, multigrid is far from being a fixed method, and some experimentation is desirable [103].

a) Let m_p be the order of the prolongation operator and m_r be the order of the restriction operator, where the order is defined as the order of the polynomial that would be exactly approximated by the given interpolation technique. For differential equations of order m, the following relation should hold:

$$m_r + m_p > m \tag{4.30}$$

For example, piecewise linear interpolation and full weighting restriction gives $m_r = m_p = 2$, which is adequate for a differential equation of no more than third order. The necessity of the above conditions has been shown by Hemker [55]. In the case of two grid multigrid, the explanation relates to ensuring that the coarse grid correction does not amplify the oscillatory part of the error in the process of annhilating the smooth part [115]. Wesseling [115] attributed the failure of a v-cycle algorithm to a breach of this relation.

b) The optimal number of smoothing operations per grid level is 3, or 2 for very efficient cycles [17].

c) The optimal coarsening ratio, or the ratio of the number of grid points on grid level l to the number of grid points on grid level $l+1$, is normally 2. Large ratios will not save significantly more work, but will significantly degrade the smoothing rates [17].

d) Basic relaxation methods with fast overall convergence rates do not necessarily have low smoothing rates. Jacobi relaxation often has superior smoothing properties to Gauss-Seidel, even it is slower to converge as a basic iterative method. In the case of SOR, overrelaxation results in a better convergence rate but underrelaxation improves the smoothing rate, so it is often preferred in multigrid schemes [103].

Despite these rather abstract guidelines there is still a lot of flexibility in the multigrid process. Some of the other parameters which must be chosen by the practitioner include the type of cycle, the number of grid levels, the type of discretisation, and the type of initial guess.

4.2.6 Multigrid convergence

The general consensus in literature on multigrid convergence is that the convergence theories are often too abstract to provide constructive criteria for the development of optimal methods for concrete situations [17]. Briggs and McCormick [19] explain that the analysis of any multigrid algorithm can be quite complex since it must account for many factors, including relaxation and its ordering, prolongation and restriction. In an attempt to overcome this complexity, the theories are constructed on an abstract level, and have significant uncertainty attached to the parameters involved in theoretical calculations. As a consequence the theories tend to be pessimistic, predicting convergence rates that are significantly slower than those observed in practice [115, 17, 6]. Details on the theory of multigrid convergence can be found in Hackbusch [46], Wesseling [115] and Bramble [14]. The theory is based on two sufficient conditions known as the smoothing property and the approximation property [115].

Brandt [17] suggests a practical technique called local mode analysis to predict the speed of multigrid convergence. In the context of local mode analysis, Brandt [17] conceptualises the multigrid principle by emphasising that reduction of the oscillatory error components is essentially a local task and can thus be efficiently performed by relaxation, which is a local process. Through the use of coarse grids, multigrid methods use relaxation essentially as a means of attenuating the oscillatory modes [19]. Therefore, the overall convergence factor for a good multigrid scheme is close to the convergence factor of relaxation restricted to the oscillatory modes [19]. This is simply the smoothing rate. The argument that multigrid convergence is independent of h follows from the fact that the reduction of the oscillatory modes is independent of h and thus the smoothing rate is independent of h [19]. Brandt claims that local mode analysis always gives reliable estimates of the overall convergence rate of the multigrid method [78].

Chapter 5

Prolongation and restriction of univariate quadratic B-splines

5.1 Hierarchial B-splines

Using B-splines for approximating the solution of the discretised smoothing spline system has some advantages from a multigrid perspective. Using the hierarchial B-spline framework, intergrid transfer can be made with greater efficiency and accuracy in comparison to the standard multigrid prolongation and restriction techniques [34, 80]. A hierarchial spline space can be defined as a linear span of B-splines with nested knot sequences [80]. Consider the knot sequences

$$\gamma_{r,l} = a + r2^l h, \qquad r = 0, \ldots, N_l + k + 2, \qquad l = 0, 1, 2 \ldots \tag{5.1}$$

where a is the first knot in the sequence, l is the grid level, h is the width of the knot intervals on the finest level and N_l is the number of spans on level l. The width of the knot intervals doubles as l is incremented. The B-spline $B^l_{r,k}$ of level l is the B-spline to the knot sequence $\gamma_{r,l}, \ldots, \gamma_{r+k+1,l}$ with $supp\ B^l_{r,k} = [\gamma_{r,l}, \gamma_{r+k+1,l}]$. Consider the spaces $\mathcal{S}_l = span\{B^l_{r,k}\} = \{s^l \in \mathcal{S}_l : s^l = \sum_{r=1}^{d_l} \alpha_r B^l_{r,k}; \alpha_r \in \mathcal{R}\}$, where d_l is the dimension of level l. These spaces form a sequence of nested subspaces such that

$$\mathcal{S}_l \subset \mathcal{S}_{l-1} \subset, \ldots, \mathcal{S}_1 \subset \mathcal{S}_0 \tag{5.2}$$

This means that any element in $\mathcal{S}_l$ can be written as a linear combination of the basis vectors $B_{r,k}^{l-1}$ of $\mathcal{S}_{l-1}$, although this representation would be redundant since the dimension of $\mathcal{S}_l$ is half that of $\mathcal{S}_{l-1}$. B-splines are therefore 'refinable' [34], in that each one can be re-expressed as a linear combination of one or more 'smaller' basis functions. In the multigrid context, this means that standard intergrid transfer operators are not required. In the case of nested grid, refinement occurs by simply expressing each coarse grid basis element in terms of the fine grid basis, as discussed below.

5.2 Prolongation

Prolongation of the quadratic B-spline solution estimate by refining the basis elements does not change the solution estimate. The process involves representing the solution estimate on grid l in terms of the basis vectors on grid $l-1$. The coarse grid basis element can be expressed as a combination of 4 fine grid basis elements as follows.

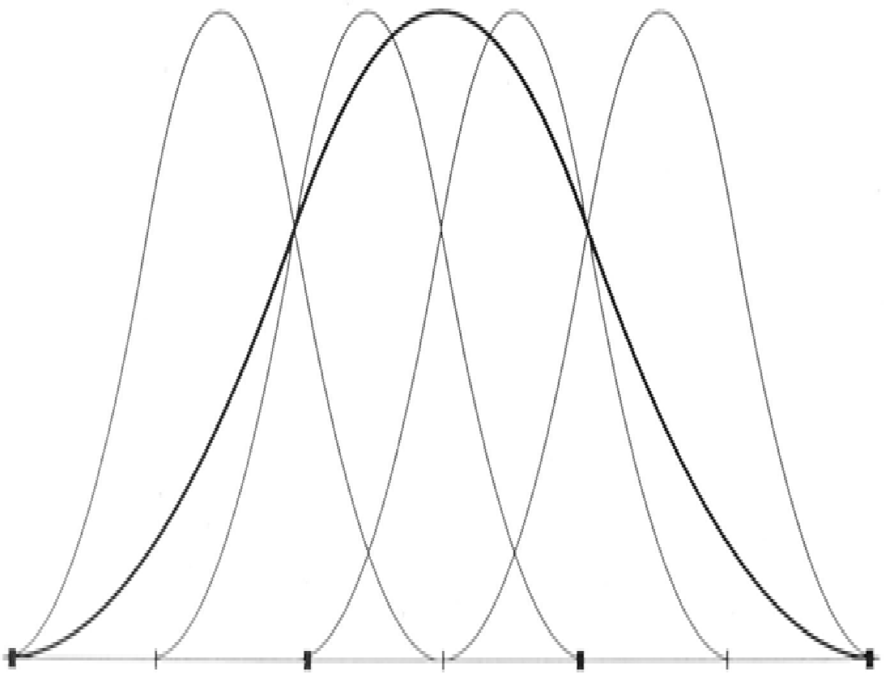


Figure 5.1: Refinement of a quadratic B-spline basis element.

$$B_{l+1,q}(x) = \sum_{r=2q-2}^{2q+1} \beta_r B_{l,r}(x) \tag{5.3}$$

where $B_{l+1,q}(x)$ is the q^{th} coarse grid basis element and $B_{l,r}(x)$ is the r^{th} fine grid element. This concept is illustrated in Figure 5.1. The coefficients β_r are chosen to obey this equality. This is done by setting 4 conditions requiring that the second derivative of $\sum_{r=2q-2}^{2q+1} \beta_r B_{l,r}(x)$ equals that of $B_{l+1,q}(x)$, and that the values of these two functions are equal at the knot intervals of the coarse grid function. Setting $q = 1$, the conditions can be stated as

$$\beta_0 - 2\beta_1 + \beta_2 = \beta_1 - 2\beta_2 + \beta_3 \tag{5.4}$$

$$\frac{1}{2}\beta_0 + \frac{1}{2}\beta_1 = \frac{1}{2} \tag{5.5}$$

$$\frac{1}{2}\beta_1 + \frac{1}{2}\beta_2 = \frac{3}{4} \tag{5.6}$$

$$\frac{1}{2}\beta_2 + \frac{1}{2}\beta_3 = \frac{1}{2} \tag{5.7}$$

This uses the fact that the values of a quadratic B-spline basis element at the middle two knot intervals within its support are 1/2 and 1/2, and the value at the centre point is 3/4. This gives $\boldsymbol{\beta}^T = (1/4, 3/4, 3/4, 1/4)$. From this, the coarse grid quadratic B-spline solution can be 'converted' to a fine grid solution using the above relationship to change the basis. Let a_r be the B-spline coefficients on grid level l and b_q be the B-spline coefficients on grid level $l+1$. Substituting (5.3) into

$$\sum_{r=0}^{N_l+1} a_r B_{l,r}(x) = \sum_{q=0}^{N_{l+1}+1} b_q B_{l+1,q}(x) \tag{5.8}$$

gives

$$\mathrm{a} = P_2 \mathrm{b} \tag{5.9}$$

where **a** and **b** are the vectors of coefficients $(a_0, \ldots, a_{N_l+1})^T$ and $(b_0, \ldots, b_{N_{l+1}+1})^T$ respectively, and P_2 is an $(N_l + 2) \times (N_{l+1} + 2)$ matrix of the form

$$\frac{1}{4}\begin{pmatrix} 3 & 1 & & & & & \\ 1 & 3 & & & & & \\ & 3 & 1 & & & & \\ & 1 & 3 & & & & \\ & & & \ddots & & & \\ & & & & & 3 & 1 \\ & & & & & 1 & 3 \end{pmatrix} \tag{5.10}$$

5.3 Restriction

The B-spline framework also allows the development of a specific restriction method involving a least squares fit of the fine grid solution to the coarse grid. This method decomposes the solution into components that correspond to the grid scales at which they can be represented. In comparison to standard multigrid techniques, this allows a clearer understanding of how the solution changes as it passes from a fine grid to a coarse grid. The least squares objective is to minimise the $\mathcal{L}^2$ norm of the difference between the coarse grid and fine grid functions ie.

$$\text{Minimise} \quad \int_a^b \left(\sum_{q=0}^{N_{l+1}+1} b_q B_{l+1,q}(x) - \sum_{r=0}^{N_l+1} a_r B_{l,r}(x) \right)^2 dx \tag{5.11}$$

Differentiating this expression with respect to the k^{th} coarse grid B-spline coefficient b_k gives

$$\int_a^b 2 \left(\sum_{q=0}^{N_{l+1}+1} b_q B_{l+1,q}(x) - \sum_{r=0}^{N_l+1} a_r B_{l,r}(x) \right) B_{l+1,k}(x) dx \tag{5.12}$$

Equating to zero for minimisation with respect to all coefficients b_k gives

$$\sum_{q=0}^{N_{l+1}+1} b_q \int_a^b B_{l+1,q}(x) B_{l+1,k}(x) dx = \sum_{r=0}^{N_l+1} a_r \int_a^b B_{l,r}(x) B_{l+1,k}(x) dx \tag{5.13}$$

To solve for $\mathbf{b}$, the vector of coarse grid coefficients, the equations for $k = 0, \ldots, N_{l+1} + 1$ can be written as

$$X\mathbf{b} = Y\mathbf{a} \tag{5.14}$$

where

$$[X]_{qk} = \int_a^b B_{l+1,q}(x)B_{l+1,k}(x)dx \tag{5.15}$$

$$[Y]_{rk} = \int_a^b B_{l,r}(x)B_{l+1,k}(x)dx \tag{5.16}$$

Integrals of this form often arise in applications where B-splines are used as basis functions, such as the finite element method and least squares fitting [110]. Vermeulen et al. [110] present a general method, based on integration by parts, for integrating

$$\int_a^b \left(\sum_r E_r B_{r,k,\mathbf{x}}(t)\right)\left(\sum_q F_q B_{q,l,\mathbf{y}}(t)\right) dt \tag{5.17}$$

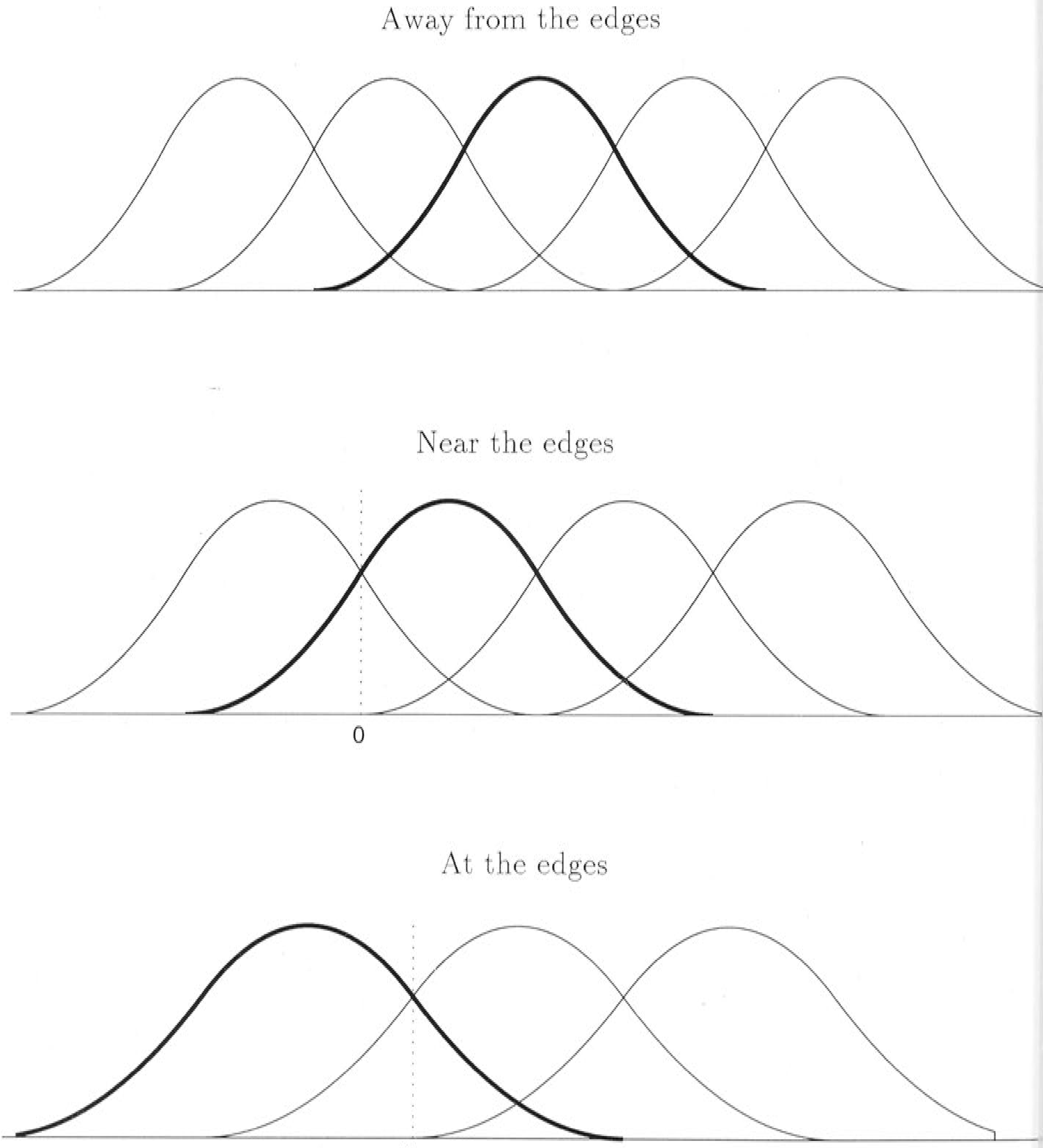
where $B_{r,k}$ is the r^{th} B-spline of order k defined over the knots $\mathbf{x}$. Fortunately the matrices X and Y are simple cases. Calculation of these matrices is discussed in the following sections.

5.3.1 The matrix X

Figure 5.2 shows that all basis elements $B_{l+1,r}$, except the first two and the last two on the grid, overlap with only 5 others, which means that the matrix X is 5-banded. It is also shown that the first and last basis elements overlap with 3 other basis elements and the second and second last basis elements overlap with 4 other basis elements.

To calculate the value of the above integrals, the polynomial representation of a quadratic B-spline in equation (3.18) is used. For simplicity of notation we write $u = w_r$. Values of the integrated products in the matrix X for elements away from the endpoints are

$$\int_0^1 B_{l+1,k-2}(u)B_{l+1,k}(u)du = \int_0^1 \frac{u^2}{2}(1-u)du = \frac{1}{120} \tag{5.18}$$



Away from the edges

Near the edges

At the edges

Figure 5.2: Overlaps of coarse grid basis elements with surrounding coarse grid basis elements.

$$\begin{aligned}\int_0^1 B_{l+1,k-1}(u)B_{l+1,k}(u)du &= \int_0^1 \frac{u^2}{2}\left(u(1-u)+\frac{1}{2}\right)du \\ &+ \int_0^1 \frac{(1-u)^2}{2}\left(u(1-u)+\frac{1}{2}\right)du \\ &= \frac{13}{120}+\frac{13}{120}=\frac{26}{120}\end{aligned} \tag{5.19}$$

$$\begin{aligned}\int_0^1 B_{l+1,k}(u)B_{l+1,k}(u)du &= \int_0^1 \left(\frac{u^2}{2}\right)^2 du \\ +\int_0^1 \left(u(1-u)+\frac{1}{2}\right)^2 du &+ \int_0^1 \left(\frac{(1-u)^2}{2}\right)^2 du \\ &= \frac{6}{120}+\frac{54}{120}+\frac{6}{120}=\frac{66}{120}\end{aligned} \tag{5.20}$$

$$\int_0^1 B_{l+1,k+1}(u)B_{l+1,k}(u)du = \int B_{l+1,k-1}(u)B_{l+1,k}(u)du = \frac{26}{120} \tag{5.21}$$

$$\int_0^1 B_{l+1,k+2}(u)B_{l+1,k}(u)du = \int B_{l+1,k-2}(u)B_{l+1,k}(u)du = \frac{1}{120} \tag{5.22}$$

For $B_{l+1,1}(x)$, integration goes only to the grid boundary, as shown in Figure 5.2. The values of the above integrals for these basis elements are

$$\int_0^1 B_{l+1,0}(u)B_{l+1,1}(u)du = \int_0^1 \frac{(1-u)^2}{2}\left(u(1-u)+\frac{1}{2}\right) = \frac{13}{120} \tag{5.23}$$

$$\begin{aligned}\int_0^1 B_{l+1,1}(u)B_{l+1,1}(u)du &= \int_0^1 \left(u(u-1)+\frac{1}{2}\right)^2 + \left(\frac{(1-u)^2}{2}\right)^2 du \\ &= \frac{54}{120}+\frac{6}{120}=\frac{60}{120}\end{aligned} \tag{5.24}$$

$$\begin{aligned}\int_0^1 B_{l+1,2}(u)B_{l+1,1}(u)du &= \int_0^1 \frac{u^2}{2}\left(u(u-1)+\frac{1}{2}\right)+ \\ &= \frac{(1-u)^2}{2}\left(u(u-1)+\frac{1}{2}\right)du = \frac{26}{120}\end{aligned} \tag{5.25}$$

$$\int_0^1 B_{l+1,3}(u)B_{l+1,1}(u)du = \int_0^1 \frac{(1-u)^2}{2}\frac{u^2}{2}du = \frac{1}{120} \tag{5.26}$$

The identical situation occurs for the basis element $B_{l+1,N_{l+1}}(u)$. For $B_{l+1,0}(u)$ the integrals are

$$\int_0^1 B_{l+1,0}(u)B_{l+1,0}(u)du = \int_0^1 \left(\frac{(1-u)^2}{2}\right)^2 du = \frac{6}{120} \tag{5.27}$$

$$\int_0^1 B_{l+1,1}(u)B_{l+1,0}(u)du = \int_0^1 \frac{(1-u)^2}{2}\left(u(1-u)+\frac{1}{2}\right)du = \frac{13}{120} \tag{5.28}$$

$$\int_0^1 B_{l+1,2}(u)B_{l+1,0}(u)du = \int_0^1 \frac{(1-u)^2}{2}\frac{u^2}{2}du = \frac{1}{120} \tag{5.29}$$

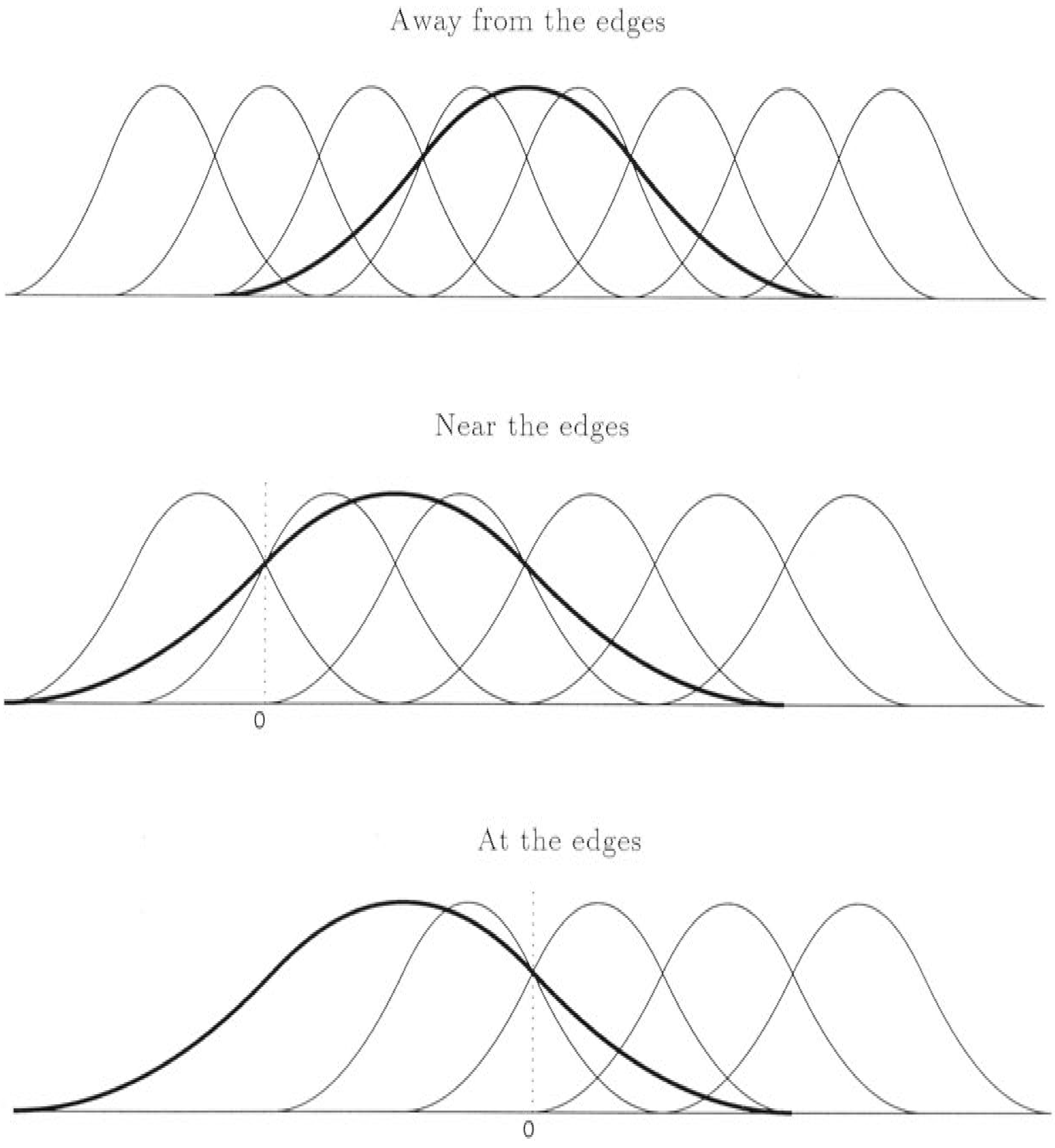
The situation is again the same for $B_{l+1,N_{l+1}+1}(x)$.

Given our scaling u, we have to multiply all the above integrals by h. Thus the matrix X is an $(N_{l+1}+2)\times(N_{l+1}+2)$ matrix of the form

$$\frac{h}{120}\begin{pmatrix} 6 & 13 & 1 & & & & \\ 13 & 60 & 26 & 1 & & & \\ 1 & 26 & 66 & 26 & 1 & & \\ & 1 & 26 & 66 & 26 & 1 & \\ & & & & \ddots & & \\ & & & 1 & 26 & 60 & 13 \\ & & & & 1 & 13 & 6 \end{pmatrix}$$

5.3.2 The matrix Y

Figure 5.3 shows that basis elements $B_{l+1,r}$, except the first two and the last two on the grid, overlap with 8 elements from $\mathcal{S}_l$. The matrix Y is therefore 8-banded. To calculate the integrated products in this matrix, the difference in the width of the knot intervals must be accounted for. Given that the knot intervals in S_l are half the width of those in S_{l+1}, the basis element from $\mathcal{S}_{l+1}$ must be represented as shown in Figure 5.4. The total range of integration is now twice that considered in section 5.3.1, because we take the fine grid knot intervals to be 1. Now the elements of Y away from the endpoints are


Away from the edges

Near the edges

At the edges

Figure 5.3: Overlap of coarse grid basis elements with surrounding fine grid basis elements.

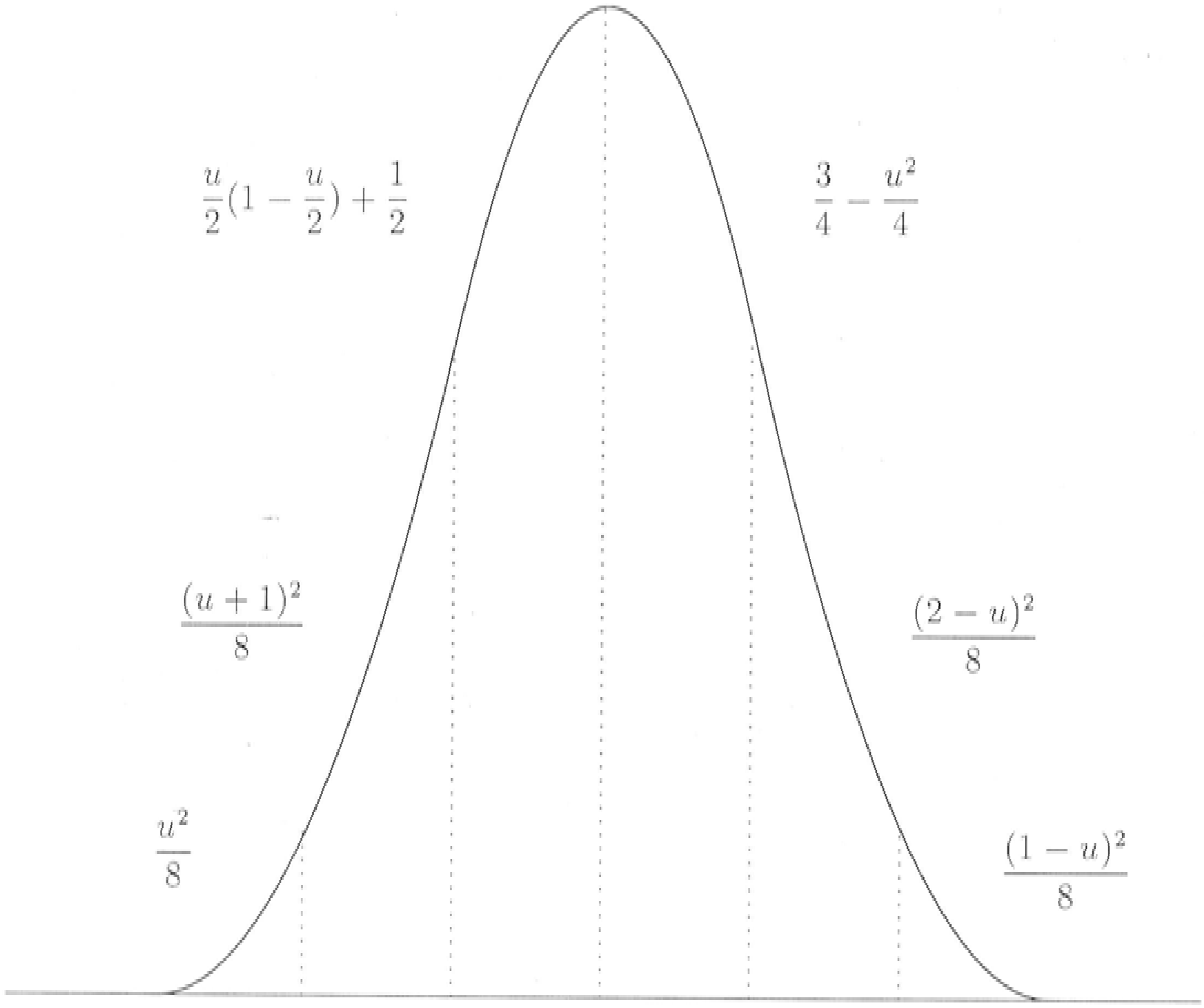


Figure 5.4: Polynomial pieces of a quadratic B-spline, divided up into fine grid knot intervals, with u ranging from 0 to 1 on each knot interval.

$$\int_0^1 B_{l,2k-2}(u)B_{l+1,k}(u)du = \frac{1}{4}\int_0^1 \frac{(1-u)^2}{2}\frac{u^2}{2}du = \frac{1}{480} \tag{5.30}$$

$$\begin{aligned}\int_0^1 B_{l,2k-1}(u)B_{l+1,k}(u)du &= \int_0^1 \left(u(1-u)+\frac{1}{2}\right)\frac{u^2}{8}du + \\ &= \int_0^1 \frac{(1-u)^2}{2}\frac{(u+1)^2}{8}du = \frac{29}{480}\end{aligned} \tag{5.31}$$

$$\int_0^1 B_{l,2k}(u)B_{l+1,k}(u)du = \int_0^1 \frac{u^2}{2}\frac{u^2}{8}du + \int_0^1 \left(u(1-u)+\frac{1}{2}\right)\frac{(u+1)^2}{8}du + \int_0^1 \frac{(1-u)^2}{2}\left(\frac{u}{2}(1-\frac{u}{2})+\frac{1}{2}\right)du = \frac{147}{480}$$

$$\int_0^1 B_{l,2k+1}(u)B_{l+1,k}(u)du = \int_0^1 \frac{u^2}{2}\frac{(u+1)^2}{8}du + \int_0^1 \left(u(1-u)+\frac{1}{2}\right)\left(\frac{u}{2}(1-\frac{u}{2})+\frac{1}{2}\right)du + \int_0^1 \frac{(1-u)^2}{2}\left(\frac{3}{4}-\frac{u^2}{4}\right)du = \frac{303}{480}$$

$$\int_0^1 B_{l,2k+2}(u)B_{l+1,k}(u)du = \int_0^1 B_{l,2k+1}(u)B_{l+1,k}(u)du = \frac{303}{480} \tag{5.32}$$

$$\int_0^1 B_{l,2k+3}(u)B_{l+1,k}(u)du = \int_0^1 B_{l,2k}(u)B_{l+1,k}(u)du = \frac{147}{480} \tag{5.33}$$

$$\int_0^1 B_{l,2k+4}(u)B_{l+1,k}(u)du = \int_0^1 B_{l,2k-1}(u)B_{l+1,k}(u)du = \frac{29}{480} \tag{5.34}$$

$$\int_0^1 B_{l,2k+5}(u)B_{l+1,k}(u)du = \int_0^1 B_{l,2k-2}(u)B_{l+1,k}(u)du = \frac{1}{480} \tag{5.35}$$

For $B_{l+1,1}(x)$ the integrals are

$$\int_0^1 B_{l,0}(u)B_{l+1,1}(u)du = \int_0^1 \frac{(1-u)^2}{2}\left(\frac{u}{2}(1-\frac{u}{2})+\frac{1}{2}\right)du = \frac{48}{480} \tag{5.36}$$

$$\int_0^1 B_{l,1}(u)B_{l+1,1}(u)du = \int_0^1 \left(u(1-u)+\frac{1}{2}\right)\left(\frac{u}{2}(1-\frac{u}{2})+\frac{1}{2}\right)du + \int_0^1 \frac{(1-u)^2}{2}\left(\frac{3}{4}-\frac{u^2}{4}\right)du = \frac{272}{480} \tag{5.37}$$

$$\int_0^1 B_{l,2}(u)B_{l+1,1}(u)du = \int_0^1 B_{l,2k+1}(u)B_{l+1,k}(u)du = \frac{303}{480} \tag{5.38}$$

$$\int_0^1 B_{l,3}(u)B_{l+1,1}(u)du = \int_0^1 B_{l,2k}(u)B_{l+1,k}(u)du = \frac{147}{480} \tag{5.39}$$

$$\int_0^1 B_{l,4}(u)B_{l+1,1}(u)du = \int_0^1 B_{l,2k-1}(u)B_{l+1,k}(u)du = \frac{29}{480} \tag{5.40}$$

$$\int_0^1 B_{l,5}(u)B_{l+1,1}(u)du = \int_0^1 B_{l,2k-2}(u)B_{l+1,k}(u)du = \frac{1}{480} \tag{5.41}$$

For $B_{l+1,0}(x)$ they are

$$\int_0^1 B_{l,0}(u)B_{l+1,0}(u)du = \int_0^1 \frac{(1-u)^2}{2}\frac{(2-u)^2}{8}du = \frac{31}{480} \tag{5.42}$$

$$\begin{aligned}\int_0^1 B_{l,1}(u)B_{l+1,0}(u)du = &\int_0^1 \left(u(1-u)+\frac{1}{2}\right)\frac{(2-u)^2}{8}du \\ &+\int_0^1 \frac{(1-u)^2}{2}\frac{(1-u)^2}{8}du = \frac{99}{480}\end{aligned} \tag{5.43}$$

$$\begin{aligned}\int_0^1 B_{l,2}(u)B_{l+1,0}(u)du &= \int_0^1 \frac{u^2}{2}\frac{(2-u)^2}{8}du + \\ &= \int_0^1 \left(u(1-u)+\frac{1}{2}\right)\frac{(1-u)^2}{8}du = \frac{29}{480}\end{aligned} \tag{5.44}$$

$$\int_0^1 B_{l,3}(u)B_{l+1,0}(u)du = \int_0^1 \frac{u^2}{2}\left(\frac{u}{2}(1-\frac{u}{2})+\frac{1}{2}\right)du = \frac{1}{480} \tag{5.45}$$

Due to doubling of the total range of integration, we have to divide all these integrals by 2 to stay consistent with our transformation $u = w_r$. The matrix Y is therefore

$$\frac{h}{960}\begin{pmatrix} 31 & 99 & 29 & 1 & & & & & \\ 48 & 272 & 303 & 147 & 29 & 1 & & & \\ 1 & 29 & 147 & 303 & 303 & 147 & 29 & 1 & \\ & 1 & 29 & 147 & 303 & 303 & 147 & 29 & 1 \\ & & & & & & \ddots & & \\ & & & 1 & 29 & 147 & 303 & 272 & 48 \\ & & & & & 1 & 29 & 99 & 31 \end{pmatrix}$$

In fact, both matrices X and Y can be derived by the representation in equation (5.3). The matrix X_l, containing the inner products of the elements $B_{l,r}$, can be transformed into the matrix X_{l+1} using the relation in equation (5.3) to turn fine grid quadratic B-spline elements into coarse grid elements. Consider the product $P_2^T X_l P_2$, given by

$$\begin{pmatrix} 3 & 1 & & & \\ 1 & 3 & 3 & 1 & \\ & & \ddots & & \\ & 1 & 3 & 3 & 1 \\ & & & 1 & 3 \end{pmatrix} \begin{pmatrix} 6 & 13 & 1 & & & & \\ 13 & 60 & 26 & 1 & & & \\ 1 & 26 & 66 & 26 & 1 & & \\ & 1 & 26 & 66 & 26 & 1 & \\ & & & & \ddots & & \\ & & & 1 & 26 & 60 & 13 \\ & & & & 1 & 13 & 6 \end{pmatrix} \begin{pmatrix} 3 & 1 & & \\ 1 & 3 & & \\ & 3 & 1 & \\ & 1 & 3 & \\ & & \ddots & \\ & & 3 & 1 \\ & & 1 & 3 \end{pmatrix}$$

The matrices P_2^T and P_2 contain the coefficients β_i, and therefore convert the products $B_{l,r}B_{l,k}$ to coarse grid inner products $B_{l+1,q}B_{l+1,k}$. Multiplying the first two matrices of the above product gives $P_2^T X_l$, which contains the inner products $B_{l+1,q}B_{l,k}$. This produces the matrix Y, in a much more efficient way than that presented at the start of this section. Multiplying all three matrices gives the matrix X_{l+1}.

5.3.3 The restricted solution

It would be expected that a function f_l that is in the space S_l and is also in the subset $S_{l+1} \subset S_l$ would be the same before and after restriction from level l to level $l+1$. This means that a function that is prolongated and then restricted should not change. If the function is not in $S_{l+1} \subset S_l$ then f_l will contain components that cannot be represented on the coarser grid. These components can be obtained by calculating the orthogonal distance between the solution estimate on grid l and its least squares fit to grid $l+1$, denoted by g_l. This is given by

$$g_l = \sum_{i=0}^{N_l+1} \zeta_i B_{l,i}(x) \tag{5.46}$$

where $\boldsymbol{\zeta} = (\zeta_0, \ldots, \zeta_{N_l+1})^{\mathrm{T}}$ are the B-spline coefficients, given by

$$\boldsymbol{\zeta} = \mathbf{a} - P_2\mathbf{b} \tag{5.47}$$

recalling that $\mathbf{b}$ is given by equation (5.14). The $\mathcal{L}^2$ norm of g_l is given by

$$||g_l|| = \sqrt{h\boldsymbol{\zeta}^{\mathbf{T}}\mathbf{X}\boldsymbol{\zeta}} \tag{5.48}$$

This norm is a useful measure of the effect the information lost when a function transferred from a fine grid to a coarser grid. The results in Part II will show this to be an effective criterion for deciding the grid resolution for representation of the final solution.

Chapter 6

Discretisation of the bivariate thin plate smoothing spline equations

The techniques for discretising univariate smoothing spline systems using the B-spline framework can be generalised to the bivariate case. As discussed in Chapter 2, the exact solution to the bivariate minimisation problem

$$\text{Minimise} \quad \frac{1}{n}(\mathbf{z}-\mathbf{f})^T(\mathbf{z}-\mathbf{f}) + \lambda \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2 dxdy \qquad (6.1)$$

is given by a thin plate smoothing spline. The bivariate thin plate smoothing spline solution was approximated using tensor product quadratic B-splines composed of the quadratic B-splines used in the univariate problem. Tensor product quadratic B-splines are conforming finite elements, in that they are in the space $\mathcal{X}$ in which the bivariate thin plate spline problem is solved [13].

Tensor product splines are commonly used to approximate bivariate functions, and fit smooth surfaces to data observations [42, 3, 33, 94, 81, 21, 44]. De Boor [27] states that tensor product methods should be used where applicable because they are extremely efficient compared to other surface approximation techniques. They are, however, generally only suited to rectangular domains, which is the case considered in this study.

6.1 Tensor product splines

The theory of univariate B-splines can be generalised to bivariate splines using tensor products. The tensor product $w(x, y)$ of two functions $u(x)$ and $v(y)$ is given by

$$w(x,y) = u(x)v(y) \qquad x, y \in \mathcal{R} \tag{6.2}$$

[27]. To construct a bivariate B-spline, or tensor product spline, the tensor product of unidimensional quadratic B-splines can be used as the basis element. Thus a tensor product spline is given by

$$f(x,y) = \sum_{I=0}^{N+1}\sum_{J=0}^{M+1} \alpha_{IJ} B_I(x) B_J(y) \tag{6.3}$$

$f(x, y)$ is defined on a finite region $[a, b] \times [c, d]$, shown in Figure 6.1 . There are two independent, strictly increasing knot sequences:

$$\gamma_I, \qquad I = 0, \ldots, N+4, \qquad \text{where } \gamma_2 = a \text{ and } \gamma_{N+2} = b \tag{6.4}$$

$$\mu_J, \qquad J = 0, \ldots, M+4, \qquad \text{where } \mu_2 = c \text{ and } \mu_{N+2} = d \tag{6.5}$$

Thinking in univariate terms, γ is the knot sequence for the quadratic B-spline elements $B_I(x)$ and μ is the knot sequence for the quadratic B-spline elements $B_J(y)$. The coefficient α_{IJ} corresponds to the basis element centered in the middle of grid cell IJ.

A visual depiction of the bivariate basis element, $B_I(x)B_J(y)$ is given in Figure 6.2. Clearly, $B_I(x)B_J(y) = 0$ if $x \notin [\gamma_I, \gamma_{I+3}]$ or $y \notin [\mu_J, \mu_{J+3}]$. The basis is still normalised, so that,

$$\sum_{I=0}^{N+1}\sum_{J=0}^{M+1} B_I(x)B_J(y) = \sum_{I=0}^{N+1} B_I(x) \sum_{J=0}^{M+1} B_J(y) = 1 \tag{6.6}$$

The properties of quadratic tensor product splines are analogous to those of the univariate quadratic B-splines splines previously discussed. They are continuous, and have continuous first partial derivatives $\partial s(x,y)/\partial x$, $\partial s(x,y)/\partial y$. The dimension of the tensor product space is obtained by multiplying the dimensions of the two univariate spaces in the tensor product [27]. In the case of quadratic tensor product splines, this is $(N+2) \times (M+2)$. Note that, as in

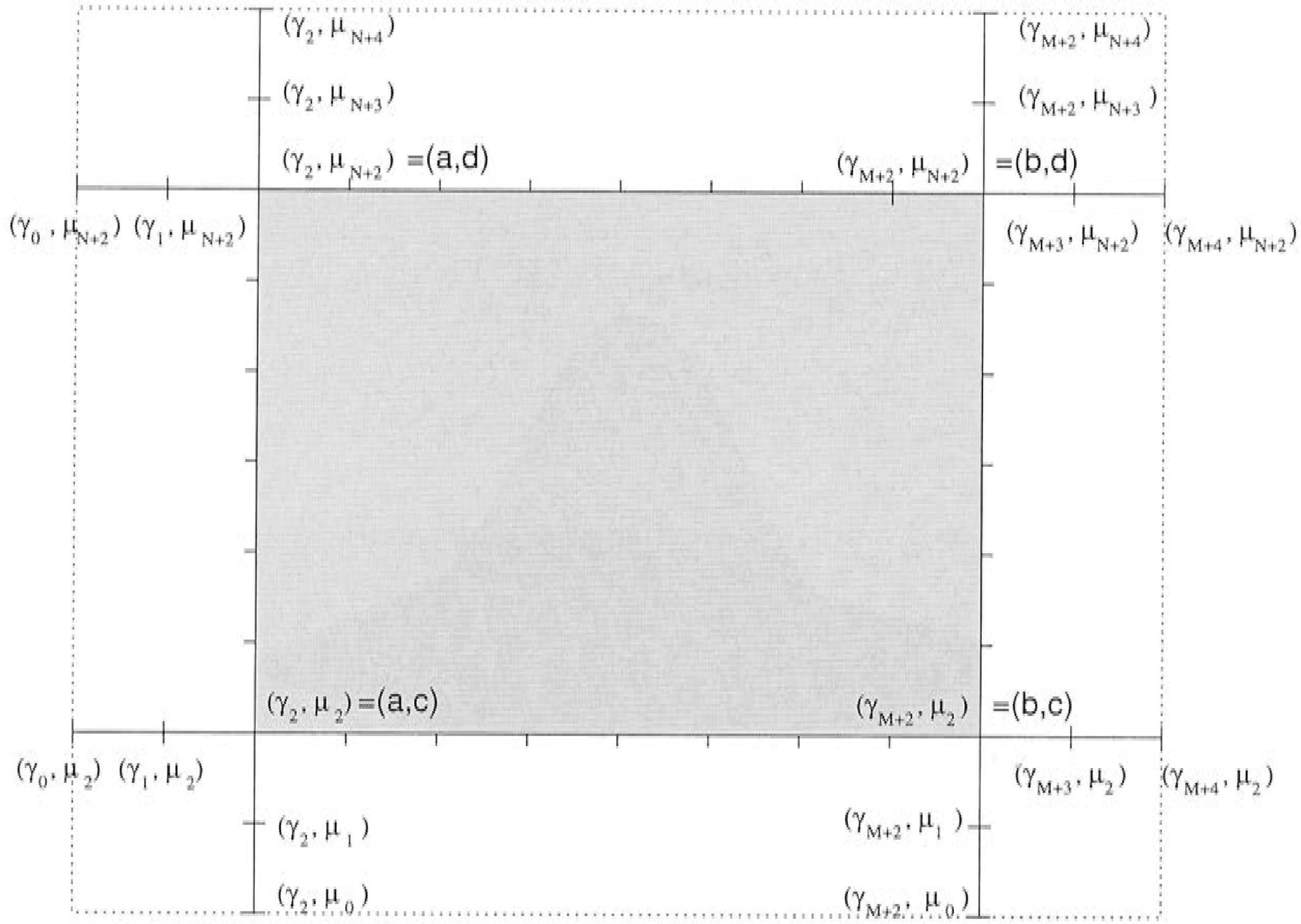

(γ_2, μ_{N+4})
$(\gamma_{M+2}, \mu_{N+4})$
(γ_2, μ_{N+3})
$(\gamma_{M+2}, \mu_{N+3})$
(γ_2, μ_{N+2}) =(a,d)
$(\gamma_{M+2}, \mu_{N+2})$ =(b,d)
(γ_0, μ_{N+2}) (γ_1, μ_{N+2})
$(\gamma_{M+3}, \mu_{N+2})$ $(\gamma_{M+4}, \mu_{N+2})$
(γ_2, μ_2)=(a,c)
(γ_{M+2}, μ_2) =(b,c)
(γ_0, μ_2) (γ_1, μ_2)
(γ_{M+3}, μ_2) (γ_{M+4}, μ_2)
(γ_2, μ_1)
(γ_{M+2}, μ_1)
(γ_2, μ_0)
(γ_{M+2}, μ_0)

Figure 6.1: Two dimensional grid for the bivariate smoothing spline.

the univariate case, the tensor product spline is not defined on the $2 \times 4NM$ 'exterior' knot intervals shown in Figure 6.1. The region of definition covers the $N \times M$ spans within the shaded region.

6.2 Roughness penalty calculation

The discretisation process in 2 dimensions once again involves substituting the discretised approximation to the thin plate smoothing spline into the minimisation expression (2.11). A more involved process must be followed to calculate the discretised bivariate roughness penalty. As in the unidimensional case, the expression for $f(x, y)$ in equation (6.3), is substituted into the roughness penalty expression

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2 dxdy \tag{6.7}$$

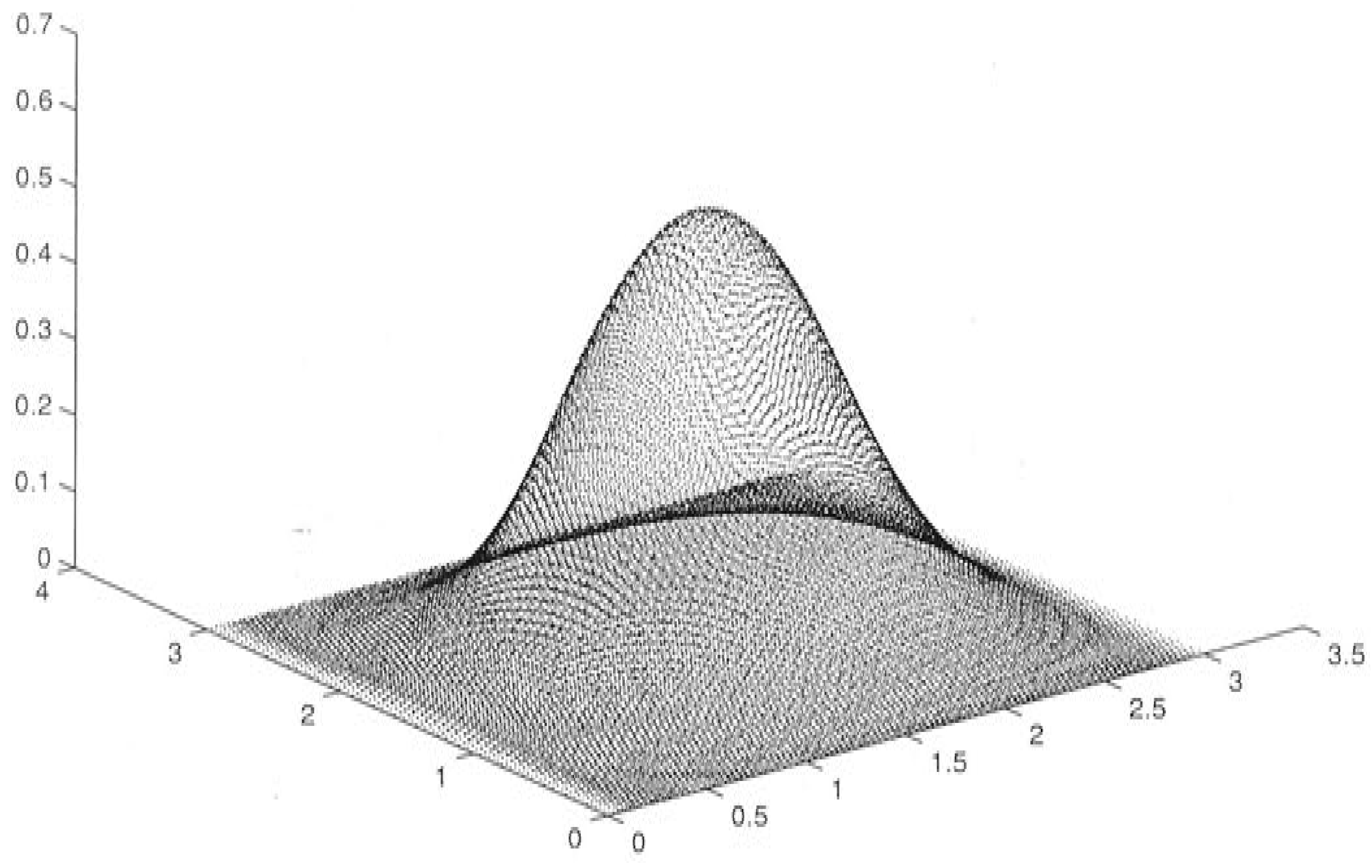


Figure 6.2: Bivariate quadratic B-spline basis element.

Once again, given that our discretised approximation is defined only on a set region, we no longer integrate over the whole 2 dimensional plane. Roughness is minimised only on the defined region. Denoting the basis element $B_I(x)B_J(y)$ as B_{IJ}, the substitution process gives firstly

$$\sum_{IJ} \alpha_{IJ} \int_a^b \int_c^d f_{xx} B_{IJxx} dx dy + 2 \sum_{IJ} \alpha_{IJ} \int_a^b \int_c^d f_{xy} B_{IJxy} dx dy$$
$$+ \sum_{IJ} \alpha_{IJ} \int_a^b \int_c^d f_{yy} B_{IJyy} dx dy$$

and then

$$\sum_{IJ}\sum_{kl}\alpha_{IJ}\alpha_{kl}\Big(\int_a^b\int_c^d B_{IJxx}B_{klxx}dxdy + 2\int_a^b\int_c^d B_{IJxy}B_{klxy}dxdy \\ + \int_a^b\int_c^d B_{IJyy}B_{klyy}dxdy\Big) \tag{6.8}$$

$$\begin{aligned} &= \boldsymbol{\alpha}^T\left[\int_a^b\int_c^d B_{IJxx}B_{klxx}dxdy\right]\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^T\left[\int_a^b\int_c^d B_{IJxy}B_{klxy}dxdy\right]\boldsymbol{\alpha} \\ &+ \boldsymbol{\alpha}^T\left[\int_a^b\int_c^d B_{IJyy}B_{klyy}dxdy\right]\boldsymbol{\alpha}\end{aligned}$$

$$= \boldsymbol{\alpha}^T Z_1\boldsymbol{\alpha} + \boldsymbol{\alpha}^T Z_2\boldsymbol{\alpha} + \boldsymbol{\alpha}^T Z_3\boldsymbol{\alpha} \tag{6.9}$$

$$= \boldsymbol{\alpha}^T Z\boldsymbol{\alpha} \tag{6.10}$$

where Z_1, Z_2, Z_3 and Z are matrices of dimension $(M+2)(N+2)\times(M+2)(N+2)$. We now calculate these matrices.

Firstly, consider $\boldsymbol{\alpha}^T Z_1\boldsymbol{\alpha}$, the first term of (6.9). The elements of a row of Z_1 are $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} B_{IJxx}B_{klxx}dxdy$, for $k = 0,\ldots,N+1$ and $l = 0,\ldots,M+1$. We need to consider all non-zero overlaps with the element B_{IJ}. Figure 6.3 shows the centre position of all basis elements that overlap with B_{IJ}.

Now we write $\int_a^b\int_c^d B_{IJxx}B_{klxx}dxdy$ as

$$\int_a^b\int_c^d B_I''(x)B_k''(x)B_J(y)B_l(y)dxdy = \left(\int_a^b B_I''(x)B_k''(x)dx\right)\left(\int_a^b B_J(y)B_l(y)dy\right) \tag{6.11}$$

so we can work with the unidimensional components. Figure 6.3 corresponds to the 5×5 matrix

$$\begin{pmatrix} \int_c^d B_JB_{l+2}dy \\ \int_c^d B_JB_{l+1}dy \\ \int_c^d B_JB_{l}dy \\ \int_c^d B_JB_{l-1}dy \\ \int_c^d B_JB_{l-2}dy \end{pmatrix} \left(\int_a^b B_I''B_{k-2}''dx \quad \int_a^b B_I''B_{k-1}''dx \quad \int_a^b B_I''B_{k}''dx \quad \int_a^b B_I''B_{k+1}''dx \quad \int_a^b B_I''B_{k+2}''dx\right) \tag{6.12}$$

We now follow a similar procedure to that described in Chapter 5, whereby

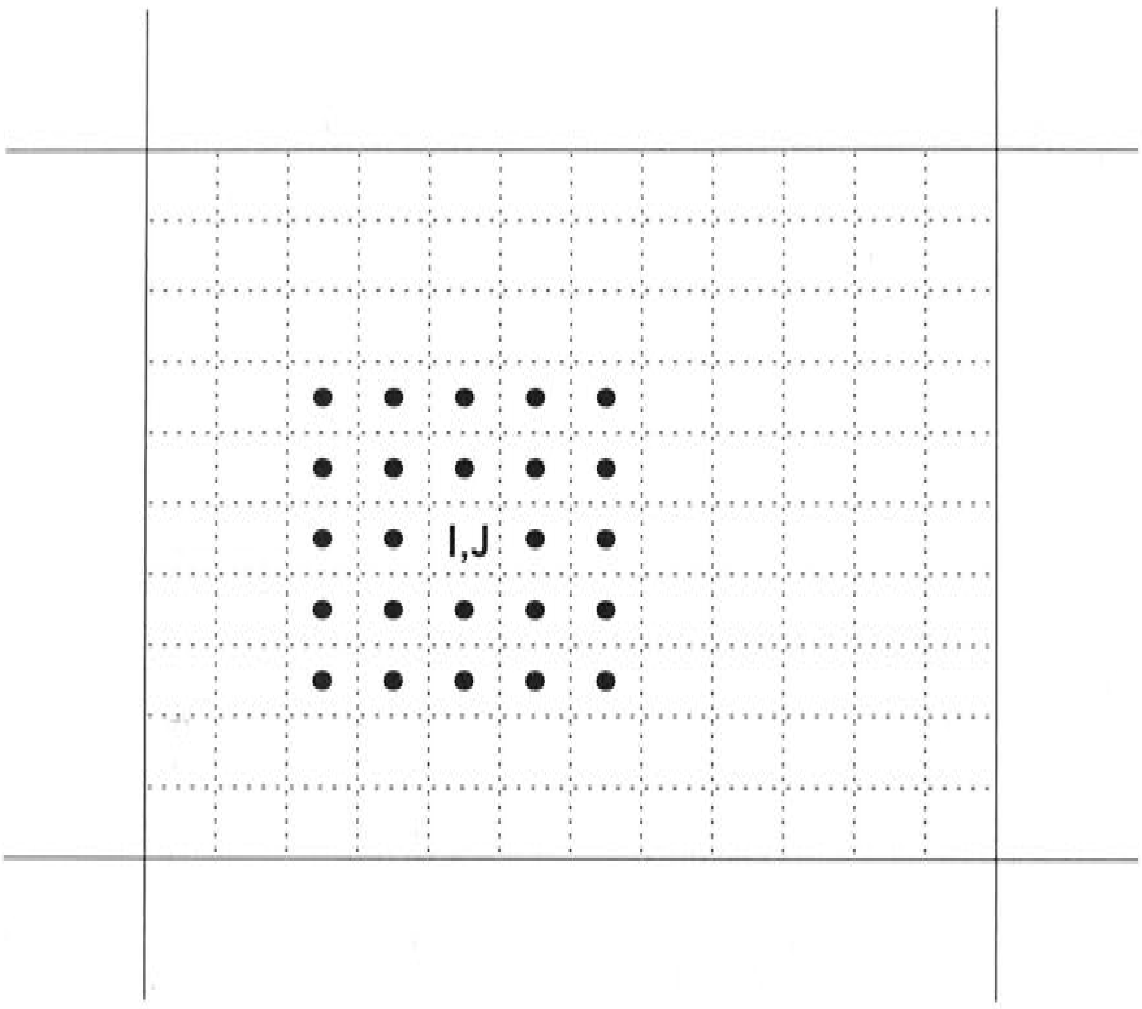


Figure 6.3: Centre positions of basis elements that overlap with B_{IJ}.

the non-zero overlap between the unidimensional basis element $B_I(x)$ or $B_J(y)$ and other basis elements $B_k(x)$ or $B_l(y)$ is calculated by integrating from 0 to 1 across each knot interval within the overlapping region. We again use the transformation $u = w_r$ for knot intervals in the x direction, and $v = w_r$ for knot intervals in the y direction. Values of $\int_0^1 B_I(u)B_k(u)dx$ or $\int_0^1 B_I(v)B_k(v)dy$ can be obtained from Chapter 5, but now we need corresponding values for the first and second derivatives of the basis elements as well. The derivatives of the quadratic B-spline are shown in Figure 6.4, once again using the polynomial expressions for each knot interval given in equation (3.18). Calculation of the integrated products $\int_0^1 B_I''(u)B_k''(u)du$ away from the endpoints gives

$$\int_0^1 B_{k-2}''(u)B_k''(u)du = 1 \times 1 = 1 \tag{6.13}$$

$$\int_0^1 B_{k-1}''(u)B_k''(u)du = -2 \times 1 + 1 \times -2 = -4 \tag{6.14}$$

$$\int_0^1 B_k''(u)B_k''(u)du = 1 \times 1 + -2 \times -2 + 1 \times 1 = 6 \tag{6.15}$$

$$\int_0^1 B''_{k+1}(u)B''_k(u)du = \int_0^1 B_{k-1}(u)B_k(u)du = -4 \tag{6.16}$$

$$\int_0^1 B''_{k+2}(u)B''_k(u)du = \int_0^1 B_{k-2}(u)B_k(u)du = 1 \tag{6.17}$$

Non-zero values of $\left(\int_0^1 B_J(v)B_l(v)dv\right)\left(\int_0^1 B''_I(u)B''_k(u)du\right)$ for all I, J, k and l away from the endpoints are therefore given by

$$\frac{1}{120}\begin{pmatrix} 1 \\ 26 \\ 66 \\ 26 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & -4 & 6 & -4 & 1 \end{pmatrix} \tag{6.18}$$

where the numbers in the first vector were obtained in Chapter 5. Multiplying this product gives

$$\frac{1}{120}\begin{pmatrix} 1 & -4 & 6 & -4 & 1 \\ 26 & -104 & 156 & -104 & 26 \\ 66 & -264 & 396 & -264 & 66 \\ 26 & -104 & 156 & -104 & 26 \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix} \tag{6.19}$$

The second term of expression (6.9), corresponding to the matrix Z_2, is

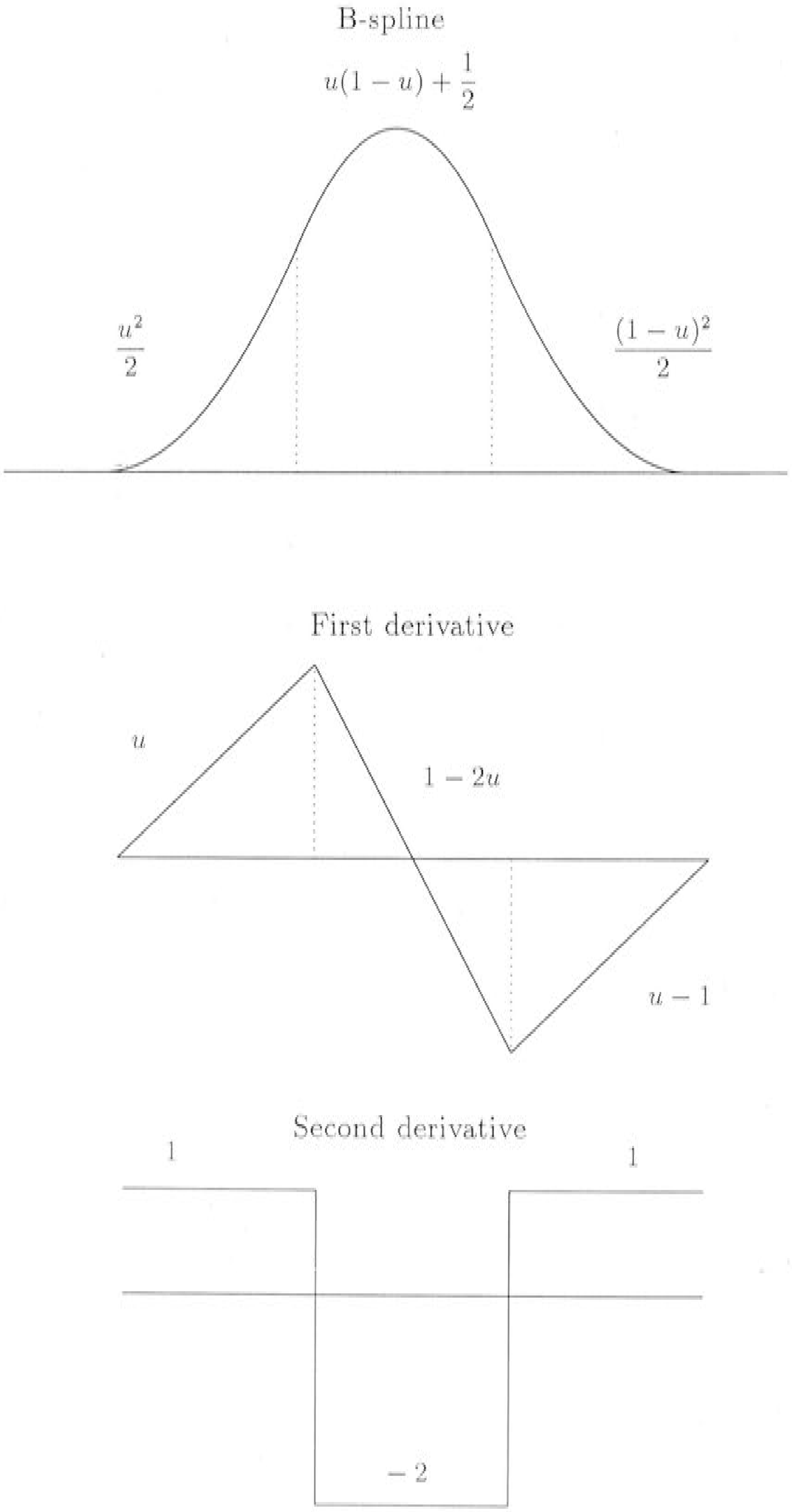
$$\boldsymbol{\alpha}^T \left[\int_a^b \int_c^d B_{IJxy} B_{klxy}\right] \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \left(\int_a^b B'_I(x)B'_k(x)dx\right)\left(\int_c^d B'_J(y)B'_l(y)dy\right)\boldsymbol{\alpha} \tag{6.20}$$

Using the polynomial expressions in Figure 6.4, the products $\int_0^1 B'_I(u)B'_k(u)du$ away from edges and corners are

$$\int_0^1 B'_{k-2}(u)B'_k(u)du = \int_0^1 u(u-1)du = -\frac{1}{6} \tag{6.21}$$

$$\int_0^1 B'_{k-1}(u)B'_k(u)du = \int_0^1 u(1-2u) + (u-1)(1-2u)du = -\frac{1}{3} \tag{6.22}$$

$$\int_0^1 B'_k(u)B'_k(u)du = \int_0^1 u^2 + (1-2u)^2 + (u-1)^2 du = 1 \tag{6.23}$$


B-spline

First derivative

Second derivative

Figure 6.4: Polynomial pieces for a quadratic B-spline and its first and second derivatives, with u ranging from 0 to 1 over each knot interval.

$$\int_0^1 B'_{k+1}(u)B'_k(u)du = \int_0^1 B'_{k-1}(u)B'_k(u)du = -\frac{1}{3} \tag{6.24}$$

$$\int_0^1 B'_{k+2}(u)B'_k(u)du = \int_0^1 B'_{k-2}(u)B'_k(u)du = -\frac{1}{6} \tag{6.25}$$

The values of $\left(\int_0^1 B'_J(v)B'_l(v)dv\right)\left(\int_0^1 B'_I(u)B'_k(u)du\right)$ are therefore

$$\frac{1}{36}\begin{pmatrix} 1 \\ -2 \\ 6 \\ -2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & 6 & -2 & 1 \end{pmatrix} = \frac{1}{36}\begin{pmatrix} 1 & 2 & -6 & 2 & 1 \\ 2 & 4 & -12 & 4 & 2 \\ -6 & -12 & 36 & -12 & -6 \\ 2 & 4 & -12 & 4 & 2 \\ 1 & 2 & -6 & 2 & 1 \end{pmatrix} \tag{6.26}$$

For the final term of expression (6.9), corresponding to the matrix Z_3, the calculation is similar to that for the first term. The values of $\left(\int_0^1 B''_J(v)B''_l(v)dv\right)\left(\int_0^1 B_I(u)B_k(u)du\right)$ are given by the transpose of (6.19), ie.

$$\frac{1}{120}\begin{pmatrix} 1 \\ -4 \\ 6 \\ -4 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 26 & 66 & 26 & 1 \end{pmatrix} \tag{6.27}$$

which gives

$$\frac{1}{120}\begin{pmatrix} 1 & 26 & 66 & 26 & 1 \\ -4 & -104 & -264 & -104 & -4 \\ 6 & 156 & 396 & 156 & 6 \\ -4 & -104 & -264 & -104 & -4 \\ 1 & 26 & 66 & 26 & 1 \end{pmatrix} \tag{6.28}$$

The total of the values for all three terms is

$$\frac{1}{360}\begin{pmatrix} 26 & 106 & 96 & 106 & 26 \\ 106 & -544 & -564 & -544 & 106 \\ 96 & -564 & 3096 & -564 & 96 \\ 106 & -544 & -564 & -544 & 106 \\ 26 & 106 & 96 & 106 & 26 \end{pmatrix} \tag{6.29}$$

The above calculations apply to basis elements that are away from the edges of the region in Figure 6.1 , and are not influenced by edge effects. Separate calculations are required at the endpoints and near the endpoints, as for the unidimensional case discussed in Chapter 1. For the bivariate case, this corresponds to a number of different possible positions, as shown in Figure 6.5. The following procedures demonstrate that there is high symmetry in the left, right, top and bottom positions. It turns out the matrices for the right, top and bottom positions are simple rearrangements of the numbers for the left bottom positions B-F, shown in Figure 6.6.

B. At the corners

At the endpoints, the first and last basis elements overlap with 3 other basis elements, as discussed in Chapter 5. At the corners both sequences $B_I(x)$ and $B_J(y)$ are at the end points. For the first term of expression (6.9), the values of $\int_0^1 B_J(v)B_l(v)dv$ at the end points were calculated in Chapter 5. The values

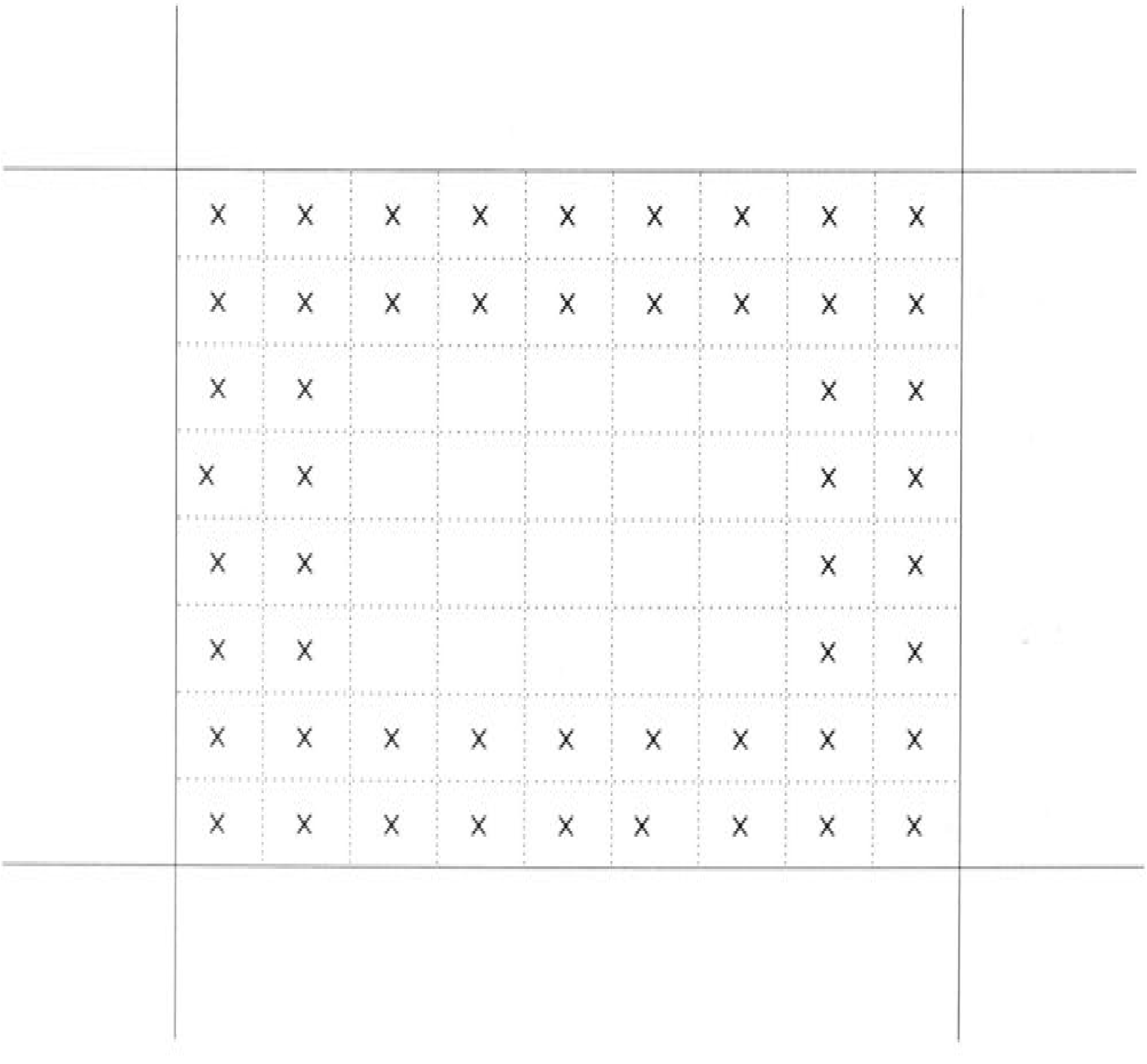


Figure 6.5: Centre points of basis elements in positions at or near the edges.

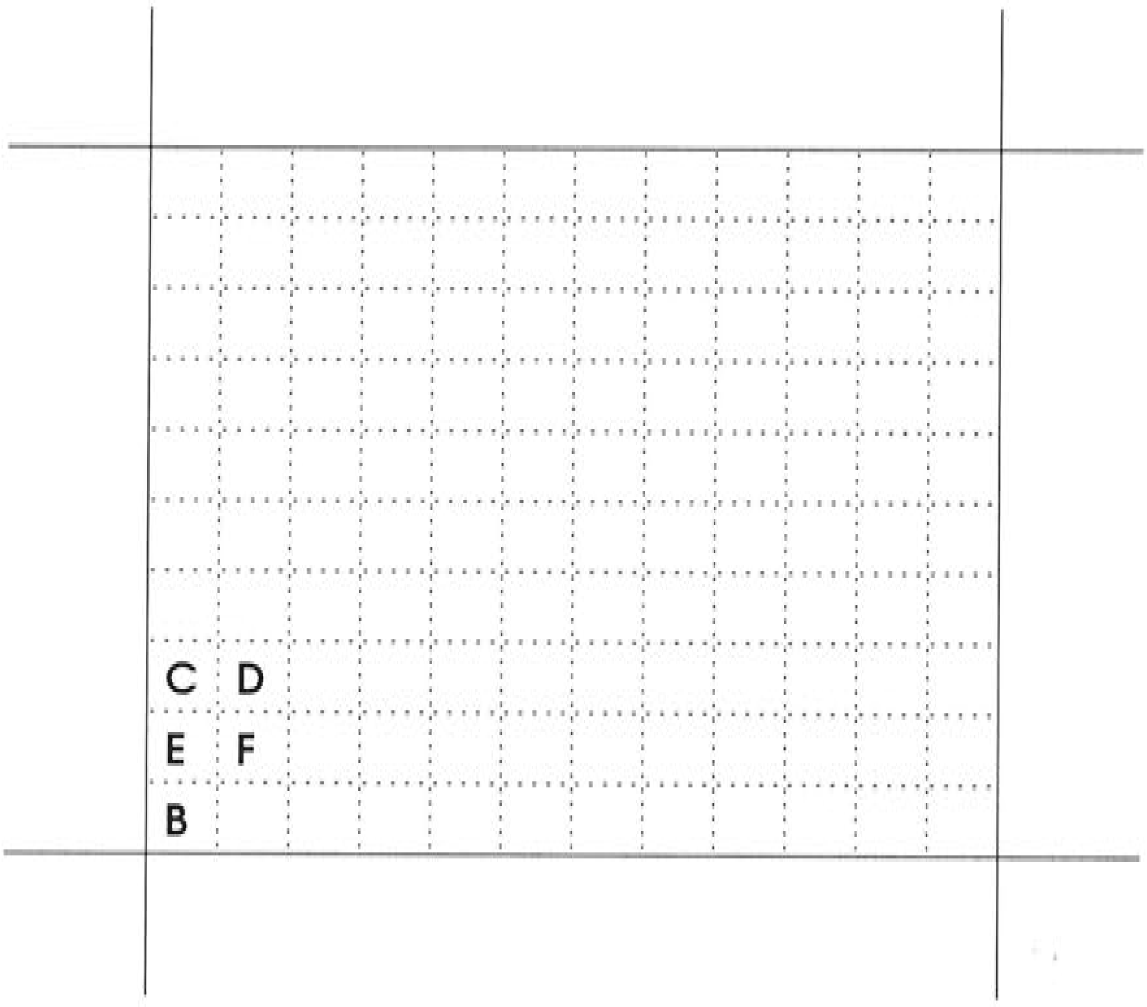


Figure 6.6: Edge and near edge positions for the lower left corner.

of $\int_0^1 B_I''(u)B_k''(u)du$ at the endpoints are given by

$$\int_0^1 B_0''(u)B_0''(u)du = 1 \times 1 = 1 \tag{6.30}$$

$$\int_0^1 B_1''(u)B_0''(u)du = 1 \times -2 = -2 \tag{6.31}$$

$$\int_0^1 B_2''(u)B_0''(u)du = 1 \times 1 = 1 \tag{6.32}$$

Values of $\left(\int_0^1 B_J(v)B_l(v)dv\right)\left(\int_0^1 B_I''(u)B_k''(u)du\right)$ at the left bottom corner are given by

$$\frac{1}{120}\begin{pmatrix} 0 \\ 0 \\ 6 \\ 13 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & -2 & 1 \end{pmatrix} = \frac{1}{120}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & -12 & 6 \\ 0 & 0 & 13 & -26 & 13 \\ 0 & 0 & 1 & -2 & 1 \end{pmatrix} \tag{6.33}$$

where the zeros correspond to quadratic B-spline elements whose support lies outside the defined region.

For the lower right corner, the only change is a shift in the position of the zeros.

$$\frac{1}{120}\begin{pmatrix} 0 \\ 0 \\ 6 \\ 13 \\ 1 \end{pmatrix}\begin{pmatrix} 1 & -2 & 1 & 0 & 0 \end{pmatrix} = \frac{1}{120}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 6 & -12 & 6 & 0 & 0 \\ 13 & -26 & 13 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \end{pmatrix} \tag{6.34}$$

Similarly, values for the upper left corner are

$$\frac{1}{120}\begin{pmatrix} 0 & 0 & 6 & -12 & 6 \\ 0 & 0 & 13 & -26 & 13 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.35}$$

and the upper right corner values are

$$\frac{1}{120}\begin{pmatrix} 6 & -12 & 6 & 0 & 0 \\ 13 & -26 & 13 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.36}$$

For the second term of expression (6.9), calculations of $\int_0^1 B_l'(u)B_k'(u)du$ at the endpoints are required. These are as follows.

$$\int_0^1 B_0'(u)B_0'(u)du = \int_0^1 (u-1)^2 du = \frac{1}{3} \tag{6.37}$$

$$\int_0^1 B_1'(u)B_0'(u)du = \int_0^1 (u-1)(1-2u)du = -\frac{1}{6} \tag{6.38}$$

$$\int_0^1 B_2'(u)B_0'(u)du = \int_0^1 u(1-u)du = -\frac{1}{6} \tag{6.39}$$

The values of $\left(\int_0^1 B_l'(v)B'(v)dv\right)\left(\int_0^1 B_I'(u)B_k'(u)du\right)$ in the lower left corner are therefore

$$\frac{1}{36}\begin{pmatrix} 0 \\ 0 \\ 2 \\ -1 \\ -1 \end{pmatrix}\begin{pmatrix} 0 & 0 & 2 & -1 & -1 \end{pmatrix} \tag{6.40}$$

which gives

$$\frac{1}{36}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & -2 & -2 \\ 0 & 0 & -2 & 1 & 1 \\ 0 & 0 & -2 & 1 & 1 \end{pmatrix} \tag{6.41}$$

Values for the other corners are given by shifting the zeros in the same manner as in (6.34), (6.35) and (6.36). The final term, $\left(\int_0^1 B_J''(v)B_l''(v)dv\right)\left(\int_0^1 B_I(u)B_k(u)du\right)$ is again just the transpose of the matrices for $\left(\int_0^1 B_J(v)B_l(v)dv\right)\left(\int_0^1 B_I''(u)B_k''(u)du\right)$.

The total of the values for all three terms for the left bottom corner is

$$\frac{1}{360}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 116 & -37 & -19 \\ 0 & 0 & -37 & -136 & 53 \\ 0 & 0 & -19 & 53 & 26 \end{pmatrix} \tag{6.42}$$

The total values for the other corners are the same but, as the above procedure shows, the position of the zeros differs for each template.

C. Edges, away from corners

At the edges, away from corners, one quadratic B-spline sequence is at an end point and the other is unchanged by edge effects. Using the numbers calculated above, the values of $\left(\int_0^1 B_J(v)B_l(v)dv\right)\left(\int_0^1 B_I''(u)B_k''(u)du\right)$ at the left edge are

$$\frac{1}{120}\begin{pmatrix} 1 \\ 26 \\ 66 \\ 26 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & -2 & 1 \end{pmatrix} \tag{6.43}$$

which gives

$$\frac{1}{120}\begin{pmatrix} 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 26 & -52 & 26 \\ 0 & 0 & 66 & -132 & 66 \\ 0 & 0 & 26 & -52 & 26 \\ 0 & 0 & 1 & -2 & 1 \end{pmatrix} \tag{6.44}$$

Using the same process involved in changing corners, the right edge is given by

$$\frac{1}{120}\begin{pmatrix} 1 & -2 & 1 & 0 & 0 \\ 26 & -52 & 26 & 0 & 0 \\ 66 & -132 & 66 & 0 & 0 \\ 26 & -52 & 26 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \end{pmatrix} \tag{6.45}$$

The top edge is clearly the transpose of the matrix for the right edge, and the bottom edge is the transpose of the matrix for the left edge. For $\left(\int_0^1 B_l'(v)B'(v)dv\right)\left(\int_0^1 B_I'(u)B_k'(u)du\right)$, the left edge is

$$\frac{1}{36}\begin{pmatrix} 1 \\ -2 \\ 6 \\ -2 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 2 & -1 & -1 \end{pmatrix} \tag{6.46}$$

which gives

$$\frac{1}{36}\begin{pmatrix} 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & -4 & 2 & 2 \\ 0 & 0 & 12 & -6 & -6 \\ 0 & 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -1 & -1 \end{pmatrix} \tag{6.47}$$

The right, top and bottom edges are obtained in the same fashion as for the first term. The matrix for $\left(\int_0^1 B_J(v)B_l(v)dv\right)\left(\int_0^1 B_I''(u)B_k''(u)du\right)$ is no longer the transpose of the matrix for the first term, because the quadratic B-spline series in the x direction is in a different position to the series in the y direction. The matrix at the left edge is

$$\frac{1}{120}\begin{pmatrix} 1 \\ -4 \\ 6 \\ -4 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 6 & 13 & 1 \end{pmatrix} \tag{6.48}$$

which gives

$$\frac{1}{120}\begin{pmatrix} 0 & 0 & 6 & 13 & 1 \\ 0 & 0 & -24 & 52 & -4 \\ 0 & 0 & 36 & 78 & 6 \\ 0 & 0 & -24 & -52 & -4 \\ 0 & 0 & 6 & 13 & 1 \end{pmatrix} \tag{6.49}$$

and the other edges are once again calculated as above.

Values for the total, again in the left bottom position, are

$$\frac{1}{360}\begin{pmatrix} 0 & 0 & 26 & 53 & -19 \\ 0 & 0 & 106 & -272 & -74 \\ 0 & 0 & 96 & -282 & 546 \\ 0 & 0 & 106 & -272 & -74 \\ 0 & 0 & 26 & 53 & -19 \end{pmatrix} \tag{6.50}$$

D. Near edges, away from corners

Near the endpoints, the second and second last basis elements overlap with 4 other basis elements, as discussed in Chapter 5. Values of $\int_0^1 B_I(v)B_k(v)dv$ near the endpoints were calculated in Chapter 5. For $\int_0^1 B_I''(u)B_k''(u)du$ the corresponding calculations are

$$\int_0^1 B_0''(u)B_1''(u)du = 1 \times -2 = -2 \tag{6.51}$$

$$\int_0^1 B_1''(u)B_1''(u)du = -2 \times -2 + 1 \times 1 = 5 \tag{6.52}$$

$$\int_0^1 B_2''(u)B_1''(u)du = 1 \times -2 + -2 \times 1 = -4 \tag{6.53}$$

$$\int_0^1 B_3''(u)B_1''(u)du = 1 \times 1 = 1 \tag{6.54}$$

Away from the corners, only one quadratic B-spline sequence is influenced by edge effects. The matrix for $\left(\int_0^1 B_J(v)B_l(v)dv\right)\left(\int_0^1 B_I''(u)B_k''(u)du\right)$ near the left edge is thus given by

$$\frac{1}{120}\begin{pmatrix} 1 \\ 26 \\ 66 \\ 26 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & -2 & 5 & -4 & 1 \end{pmatrix} = \frac{1}{120}\begin{pmatrix} 0 & -2 & 5 & -4 & 1 \\ 0 & 52 & 130 & -104 & 26 \\ 0 & -132 & 330 & -264 & 66 \\ 0 & -52 & 130 & -104 & 26 \\ 0 & -2 & 5 & -4 & 1 \end{pmatrix} \tag{6.55}$$

The near right edge matrix was calculated to be

$$\frac{1}{120}\begin{pmatrix} -2 & 5 & -4 & 1 & 0 \\ 52 & 130 & -104 & 26 & 0 \\ -132 & 330 & -264 & 66 & 0 \\ -52 & 130 & -104 & 26 & 0 \\ -2 & 5 & -4 & 1 & 0 \end{pmatrix} \tag{6.56}$$

The matrix for near the bottom edge is clearly the transpose of the matrix corresponding to the near left edge position. Similarly, the near top edge matrix is the transpose of the matrix for the near right edge position.

To calculate the matrices for second term of expression (6.9), values of $\int_0^1 B_I'(u)B_k'(u)du$ must once again be calculated. The calculations for $\int_0^1 B_I'(u)B_k'(u)du$ near the endpoints are

$$\int_0^1 B_0'(u)B_1'(u)du = \int_0^1 (1-2u)(u-1)du = -\frac{1}{6} \tag{6.57}$$

$$\int_0^1 B_1'(u)B_1'(u)du = \int_0^1 (1-2u)^2 + (u-1)^2 du = \frac{2}{3} \tag{6.58}$$

$$\int_0^1 B_2'(u)B_1'(u)du = \int_0^1 u(1-2u)+(1-2u)(u-1)du = -\frac{1}{3} \tag{6.59}$$

$$\int_0^1 B_3'(u)B_1'(u)du = \int_0^1 u(u-1) = -\frac{1}{6} \tag{6.60}$$

The matrix for $\left(\int_0^1 B_l'(v)B'(v)dv\right)\left(\int_0^1 B_I'(u)B_k'(u)du\right)$ near the left edge is

$$\frac{1}{36}\begin{pmatrix} 1 \\ -2 \\ 6 \\ -2 \\ 1 \end{pmatrix}\begin{pmatrix} 0 & -1 & 4 & -2 & 1 \end{pmatrix} = \frac{1}{36}\begin{pmatrix} 0 & -1 & 4 & -2 & 1 \\ 0 & 2 & -8 & 4 & 2 \\ 0 & -6 & 24 & -12 & -6 \\ 0 & 2 & -8 & 4 & 2 \\ 0 & -1 & 4 & -2 & -1 \end{pmatrix} \tag{6.61}$$

The matrices for positions near the right, top and bottom edges are then derived in the same manner as for the first term. The matrices for $\left(\int_0^1 B_J''(v)B_l''(v)dv\right)\left(\int_0^1 B_I(u)B_k(u)du\right)$ near the left edge are given by

$$\frac{1}{120}\begin{pmatrix} 1 \\ -4 \\ 6 \\ -4 \\ 1 \end{pmatrix}\begin{pmatrix} 0 & 13 & 60 & 26 & 1 \end{pmatrix} = \frac{1}{120}\begin{pmatrix} 0 & 13 & 60 & 26 & 1 \\ 0 & -52 & -240 & -104 & -4 \\ 0 & 78 & 360 & 156 & 6 \\ 0 & -52 & -240 & -104 & -4 \\ 0 & 13 & 60 & 26 & 1 \end{pmatrix} \tag{6.62}$$

and the other positions are again calculated by shifting the position of the zeroes as above.

The total of the values for all three terms near the left edge is

$$\frac{1}{360}\begin{pmatrix} 0 & 53 & 115 & 106 & 26 \\ 0 & -272 & -490 & -544 & 106 \\ 0 & -282 & 2550 & -564 & 96 \\ 0 & -272 & -490 & -544 & 106 \\ 0 & 53 & 115 & 106 & 26 \end{pmatrix} \tag{6.63}$$

E. At one edge and near one edge

Using the numbers obtained above, the matrix for

$\left(\int_0^1 B_J(v)B_l(v)dv\right)\left(\int_0^1 B_I''(u)B_k''(u)du\right)$ at the left bottom position is

$$\frac{1}{120}\begin{pmatrix}0\\0\\1\\-2\\1\end{pmatrix}\begin{pmatrix}0 & 13 & 60 & 26 & 1\end{pmatrix} = \frac{1}{120}\begin{pmatrix}0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\\0 & 13 & 60 & 26 & 1\\0 & -26 & -120 & -52 & -2\\0 & 13 & 60 & 26 & 1\end{pmatrix} \tag{6.64}$$

The matrix for the right bottom position is once again given by shifting the columns of zeros, as follows.

$$\frac{1}{120}\begin{pmatrix}0\\0\\1\\-2\\1\end{pmatrix}\begin{pmatrix}13 & 60 & 26 & 1 & 0\end{pmatrix} = \frac{1}{120}\begin{pmatrix}0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\\13 & 60 & 26 & 1 & 0\\-26 & -120 & -52 & -2 & 0\\13 & 60 & 26 & 1 & 0\end{pmatrix} \tag{6.65}$$

For the left top position, we have

$$\frac{1}{120}\begin{pmatrix}0\\0\\1\\-2\\1\end{pmatrix}\begin{pmatrix}13 & 60 & 26 & 1 & 0\end{pmatrix} = \frac{1}{120}\begin{pmatrix}0 & 13 & 60 & 26 & 1\\0 & -26 & -120 & -52 & -2\\0 & 13 & 60 & 26 & 1\\0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\end{pmatrix} \tag{6.66}$$

and for the right top position,

$$\frac{1}{120}\begin{pmatrix}1\\-2\\1\\0\\0\end{pmatrix}\begin{pmatrix}13 & 60 & 26 & 1 & 0\end{pmatrix} = \frac{1}{120}\begin{pmatrix}13 & 60 & 26 & 1 & 0\\-26 & -120 & -52 & -2 & 0\\13 & 60 & 26 & 1 & 0\\0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\end{pmatrix} \tag{6.67}$$

Moving on to the second term, the values of

$\left(\int_0^1 B_l'(v)B'(v)dv\right)\left(\int_0^1 B_I'(u)B_k'(u)du\right)$ in the left bottom position are

$$\frac{1}{36}\begin{pmatrix}0\\0\\2\\-1\\-1\end{pmatrix}\begin{pmatrix}0 & -1 & 4 & -2 & 1\end{pmatrix} = \frac{1}{36}\begin{pmatrix}0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\\0 & -2 & 8 & -4 & 2\\0 & 1 & -4 & 2 & -1\\0 & 1 & -4 & 2 & -1\end{pmatrix} \tag{6.68}$$

Values for the second term at the right, top and bottom positions are obtained in the same way as for the first term.

Values for $\left(\int_0^1 B_J''(v)B_l''(v)dv\right)\left(\int_0^1 B_I(u)B_k(u)du\right)$ at the left bottom edge are

$$\frac{1}{120}\begin{pmatrix}0\\0\\6\\13\\1\end{pmatrix}\begin{pmatrix}0 & -2 & 5 & -4 & 1\end{pmatrix} = \begin{pmatrix}0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\\0 & -12 & 30 & -24 & 6\\0 & -26 & 65 & -52 & 13\\0 & -2 & 5 & -4 & 1\end{pmatrix} \tag{6.69}$$

and the other positions are also obtained in the same way as for the other two terms. For the left bottom position, the total is

$$\frac{1}{360}\begin{pmatrix}0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0\\0 & -37 & 430 & -74 & -19\\0 & -136 & -245 & -272 & 53\\0 & 53 & 115 & 106 & 26\end{pmatrix} \tag{6.70}$$

F. Near the corners

Calculations near the corners are similar to those at the corners, but we use the near endpoint values, rather than the values at the endpoints. So for $\left(\int_0^1 B_J(v)B_l(v)dv\right)\left(\int_0^1 B_I''(u)B_k''(u)du\right)$, in the near left bottom corner position,we have

$$\frac{1}{120}\begin{pmatrix}0\\13\\60\\26\\1\end{pmatrix}\begin{pmatrix}0 & -2 & 5 & -4 & 1\end{pmatrix} = \frac{1}{120}\begin{pmatrix}0 & 0 & 0 & 0 & 0\\0 & -26 & 65 & -52 & 13\\0 & -120 & 300 & -240 & 60\\0 & -52 & 130 & -104 & 26\\0 & -2 & 5 & -4 & 1\end{pmatrix} \tag{6.71}$$

Near the right bottom corner, we have

$$\frac{1}{120}\begin{pmatrix}0\\13\\60\\26\\1\end{pmatrix}\begin{pmatrix}-2 & 5 & -4 & 1 & 0\end{pmatrix} = \frac{1}{120}\begin{pmatrix}0 & 0 & 0 & 0 & 0\\-26 & 65 & -52 & 13 & 0\\-120 & 300 & -240 & 60 & 0\\-52 & 130 & -104 & 26 & 0\\-2 & 5 & -4 & 1 & 0\end{pmatrix} \tag{6.72}$$

For the right top corner, the matrix is

$$\frac{1}{120}\begin{pmatrix}-26 & 65 & -52 & 13 & 0\\-120 & 300 & -240 & 60 & 0\\-52 & 130 & -104 & 26 & 0\\-2 & 5 & -4 & 1 & 0\\0 & 0 & 0 & 0 & 0\end{pmatrix} \tag{6.73}$$

and for the left top corner the matrix is

$$\frac{1}{120}\begin{pmatrix}0 & -26 & 65 & -52 & 13\\0 & -120 & 300 & -240 & 60\\0 & -52 & 130 & -104 & 26\\0 & -2 & 5 & -4 & 1\\0 & 0 & 0 & 0 & 0\end{pmatrix} \tag{6.74}$$

The matrix for $\left(\int_0^1 B_l'(v)B'(v)dv\right)\left(\int_0^1 B_I'(u)B_k'(u)du\right)$ is

$$\frac{1}{36}\begin{pmatrix} 0 \\ -1 \\ 4 \\ -2 \\ 1 \end{pmatrix}\begin{pmatrix} 0 & -1 & 4 & -2 & 1 \end{pmatrix} = \frac{1}{36}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 2 & -1 \\ 0 & -4 & 16 & -8 & 4 \\ 0 & 2 & -8 & 4 & -2 \\ 0 & -1 & 4 & -2 & 1 \end{pmatrix} \tag{6.75}$$

The other positions are again calculated in the same way as they were for the first term.

Values for the third term, $\left(\int_0^1 B_J''(v)B_l''(v)dv\right)\left(\int_0^1 B_I(u)B_k(u)du\right)$, are given by the transpose of the matrices for the first term, because both the $B_I(x)$ and $B_J(y)$ series are in the same position.

The total of the values for all three terms is

$$\frac{1}{360}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 53 & 115 & 106 & 26 \\ 0 & -272 & 490 & -544 & 106 \\ 0 & -245 & 2120 & -490 & 115 \\ 0 & -136 & -245 & -272 & 53 \end{pmatrix} \tag{6.76}$$

6.3 The total roughness penalty

As explained in section 6.2, the above templates give the rows of the roughness penalty matrix Z. The elements in matrix (6.12) appear in row I, J in the order

$$B_{IJ}B_{k-2,l-2}, B_{IJ}B_{k-1,l-2}, \ldots, B_{IJ}B_{k-2,l-1}, \ldots\ldots, B_{IJ}B_{k-2,l+2}, \ldots, B_{IJ}B_{k+2,l+2} \tag{6.77}$$

Using the templates for the total derivative calculations, given at the bottom of each of the above sections, the full roughness penalty matrix Z is of the form shown in Figure 6.7. We have to divide the matrix by h^2 due to our scalings $u = w_r/h$ and $v = w_r/h$. The plot in Figure 6.7 shows that the matrix Z is 'block-banded', in that it has 5 bands surrounding the main diagonal, with each band consisting of blocks of 5 banded matrices. Each block is symmetric, as is the total matrix Z. Its repetitive nature stems from the fact that it is the sum of three tensor product matrices. The dimensions of Z are $(M+2)(N+2) \times (M+2)(N+2)$.

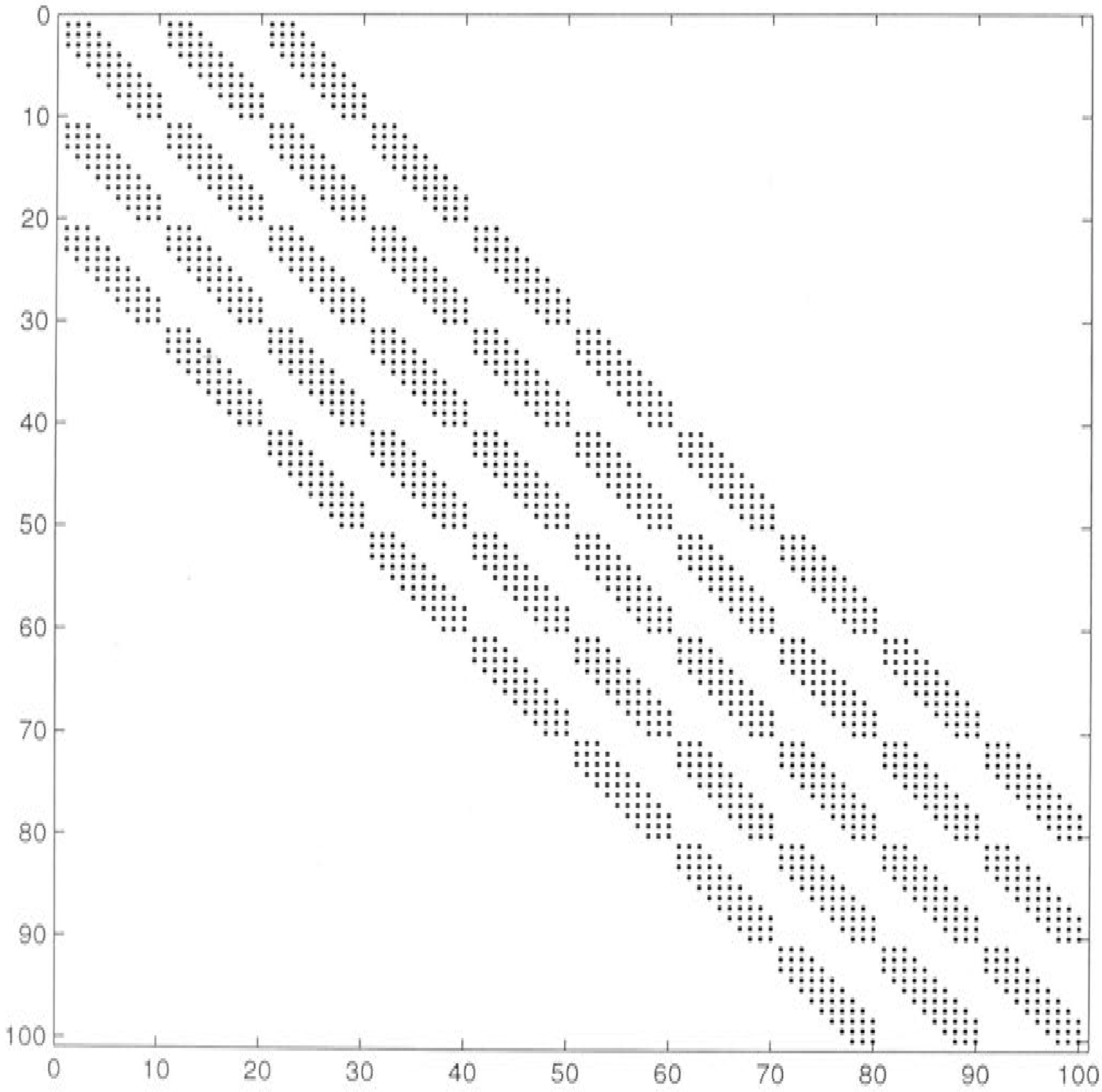


Figure 6.7: Position of non-zero entries in the roughness penalty matrix Z.

We now have the discretised minimisation problem

$$\text{Minimise } \frac{1}{n}\|P\boldsymbol{\alpha} - \mathbf{z}\|^2 + \frac{\lambda}{h^2}\boldsymbol{\alpha}^T Z \boldsymbol{\alpha} \tag{6.78}$$

where the matrix P stores the value of the basis element $B_I(x)B_J(y)$ at the data point locations (x_i, y_i), so that

$$P_{i,ind} = B_I(x_i)B_J(y_i) \tag{6.79}$$

where $ind = (N+2)J + I$. Differentiating and equating to zero gives

$$(P^T P + \frac{\lambda}{h^2} Z)\boldsymbol{\alpha} = P^T \mathbf{z} \tag{6.80}$$

This system is of the same form as equations (3.5) and (3.29). There are a couple of points to note about this discretised system. Firstly, these equations minimise roughness only over the defined region, where as analytic bivariate splines consider the infinite plane. Localising the region of integration was not an issue for the univariate case because the second derivative reduced to zero outside the interval covering the data points. For bivariate thin plate splines the second derivative does not reduce to zero, so there is a fundamental difference between the analytic solution and the discretised solution for the bivariate case.

Secondly, note that, for a finite difference discretisation of the bivariate thin plate smoothing spline equations, the roughness penalty matrix is the same as that produced from a finite difference discretisation of the bivariate biharmonic equation [18]. The bivariate biharmonic equation is given by

$$\begin{aligned} &\frac{\partial^4 f}{\partial x^4} + 2\frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4} = z(x,y), && (x,y) \in \Omega \\ &f = g_1(x,y), \qquad \frac{\partial f}{\partial n} = g_2(x,y), && (x,y) \in \partial\Omega \end{aligned} \tag{6.81}$$

where $z(x,y)$ is some function of x and y, Ω is a closed convex domain in two dimensions and $\partial\Omega$ is its boundary. The functions g_1 and g_2 are boundary conditions, where n is the unit normal to the boundary $\partial\Omega$. For regions containing no data points, the bivariate thin plate smoothing spline system reduces to (6.81). We can therefore use current knowledge on the numerical solution of the biharmonic equation to interpret the results of algorithms developed in this study.

The biharmonic equation is known to be poorly conditioned, and therefore slow to solve iteratively, on fine resolutions. Many numerical techniques exhibited convergence when applied to the biharmonic equation, even when the system is converted to a system of second order equations [2]. The experiments conducted by Brandt [16] found relatively poor convergence rates for the biharmonic equation using multigrid algorithms. Braess [12] related the poor convergence to the high condition number of fourth order problems.

Chapter 7

Prolongation and restriction of bivariate quadratic B-splines

Prolongation and restriction processes for the bivariate case involves the unidimensional processes discussed in Chapter 5. These processes act on both quadratic B-spline series in the x direction and the quadratic B-spline series in the y direction. Any prolongaton or restriction that occurs in one direction will not affect the bivariate basis element in the other direction. This means that the result of prolongation is independent of the order in which the two univariate prolongation operations are performed.

7.1 Prolongation

The process of prolongating the bivariate solution estimate to a finer grid is shown in Figure 7.1. The rows are prolongated first, giving an intermediate set of coefficients corresponding to the refinement of the resolution in the x direction. The procedure is completed by refining the columns.

Pease [95] demonstrates a general method which can be used to look at the two univariate operations simultaneously. We can apply the process in Pease [95] by considering a bivariate quadratic spline that is zero on the entire grid except for one basis element that has a coefficient of 1.

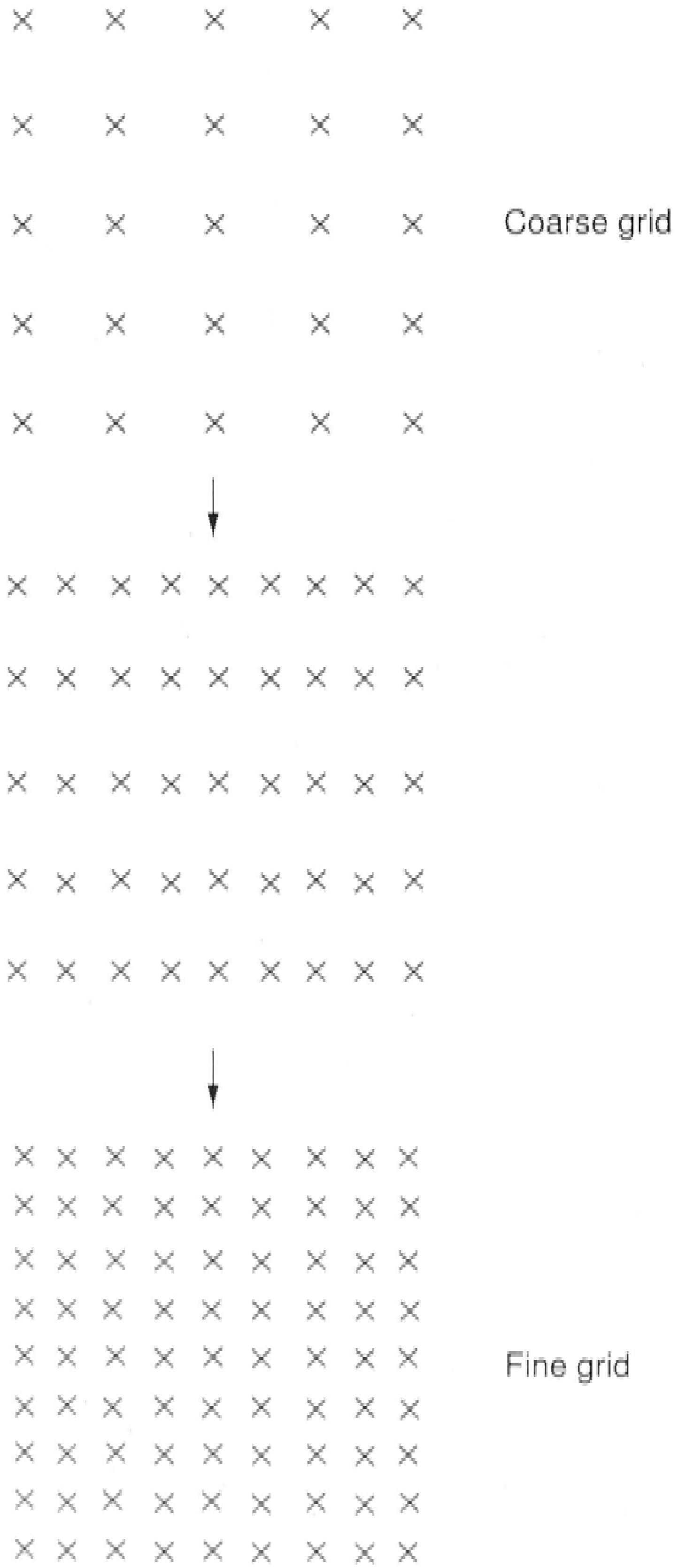


Figure 7.1: The process of prolongating biquadratic B-splines, where the × symbol corresponds to the centre point of a biquadratic B-spline basis element.

This spline is

$$s(x,y) = \sum_{I=0}^{M_{l+1}+1} \sum_{J=0}^{N_{l+1}+1} b_{IJ} B_{l+1,I}(x) B_{l+1,J}(y) \tag{7.1}$$

where

$$\begin{aligned} b_{IJ} &= 0 \qquad I \neq K, J \neq S \\ b_{IJ} &= 1 \qquad I = K, J = S \end{aligned} \tag{7.2}$$

so the spline $s(x,y)$ is

$$s(x,y) = B_K(x) B_S(y) \tag{7.3}$$

To prolongate $s(x,y)$ down to grid l we only have to prolongate one basis element, which involves prolongating each univariate basis element in the tensor product. The two univariate processes are

$$\begin{aligned} \boldsymbol{\mu} &= P\boldsymbol{\nu} \\ \boldsymbol{\phi} &= Q\boldsymbol{\varphi} \end{aligned} \tag{7.4}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are the coefficients for the B-spline series in the x direction, for the fine grid and the coarse grid respectively, $\boldsymbol{\phi}$ and $\boldsymbol{\varphi}$ are the corresponding coefficients for the B-spline series in the y direction, and P and Q are prolongation matrices for each direction. The vectors $\boldsymbol{\nu}$ and $\boldsymbol{\varphi}$ clearly have ν_K and φ_S as their only non-zero elements. For notation simplicity, set the lengths of the vectors $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, $\boldsymbol{\phi}$ and $\boldsymbol{\varphi}$ to n_1, n_2, m_1 and m_2 respectively.

We know that the non-zero part of $s(x,y)$ on grid l will be

$$(\frac{1}{4}B_{l,K-2} + \frac{3}{4}B_{l,K-1} + \frac{3}{4}B_{l,K} + \frac{1}{4}B_{l,K+1})(\frac{1}{4}B_{l,S-2} + \frac{3}{4}B_{l,S-1} + \frac{3}{4}B_{l,S} + \frac{1}{4}B_{l,S+1}) \tag{7.5}$$

The fine grid coefficients for $s(x,y)$, denoted by a_{ij}, are therefore

$$\begin{pmatrix} \vdots \\ a_{2K-2,2S-2} \\ a_{2K-1,2S-2} \\ a_{2K,2S-2} \\ a_{2K+1,2S-2} \\ \vdots \\ a_{2K-2,2S-1} \\ a_{2K-1,2S-1} \\ a_{2K,2S-1} \\ a_{2K+1,2S-1} \\ \vdots \\ \text{etc} \end{pmatrix} = \begin{pmatrix} \vdots \\ \frac{1}{4}\times\frac{1}{4} \\ \frac{3}{4}\times\frac{1}{4} \\ \frac{3}{4}\times\frac{1}{4} \\ \frac{1}{4}\times\frac{1}{4} \\ \vdots \\ \frac{1}{4}\times\frac{3}{4} \\ \frac{3}{4}\times\frac{3}{4} \\ \frac{3}{4}\times\frac{3}{4} \\ \frac{1}{4}\times\frac{3}{4} \\ \vdots \\ \text{etc} \end{pmatrix}$$

Knowing this, we write **a** and **b** in terms of products of the univariate coefficients $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, $\boldsymbol{\phi}$ and $\boldsymbol{\varphi}$.

$$\mathbf{a} = \begin{pmatrix} \mu_1\phi_1 \\ \mu_2\phi_1 \\ \vdots \\ \vdots \\ \vdots \\ \mu_{n_1}\phi_{m_1} \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} \nu_1\varphi_1 \\ \nu_2\varphi_1 \\ \vdots \\ \vdots \\ \vdots \\ \nu_{n_2}\varphi_{m_2} \end{pmatrix}$$

Now to find the relationship between these two vectors, look at the individual components

$$\begin{aligned} \mu_1\phi_1 &= (p_{11}\nu_1 + \cdots + p_{1n_2}\nu_{n_2})(q_{11}\varphi_1 + \ldots q_{1m_2}\varphi_{m_2}) \\ &= p_{11}q_{11}\nu_1\varphi_1 + p_{12}q_{11}\nu_2\varphi_1 + \cdots + p_{11}q_{12}\nu_1\varphi_2 + \ldots \text{etc} \\ &= q_{11}[p_{11}\ p_{12} \ldots p_{1n_2}]\mathbf{b}_1 + q_{12}[p_{11}\ p_{12} \ldots p_{1m_2}]\mathbf{b}_2 + \ldots \text{etc} \end{aligned} \tag{7.6}$$

where $\mathbf{b}_i = [\nu_1\varphi_i\ \nu_2\varphi_i \ldots \nu_{n_2}\varphi_i]^T$, the coefficients corresponding to the i^{th} row of the coarse grid. We now see a relationship between **a** and **b** which can be written symbolically as

$$\mathbf{a} = \begin{pmatrix} q_{11}P & q_{12}P & \dots & q_{1m_2}P \\ q_{21}P & q_{22}P & \dots & q_{2m_2}P \\ \vdots & & & \\ q_{m_11}P & \dots & \dots & q_{m_1m_2}P \end{pmatrix} \mathbf{b} \tag{7.7}$$

This matrix is a Kronecker product, also called a direct product or a tensor product [95, 85]. It is denoted

$$Q \otimes P \tag{7.8}$$

so we can write

$$\mathbf{a} = Q \otimes P\mathbf{b} \tag{7.9}$$

In general, the Kronecker product of an $(m \times n)$ matrix $A = [a_{ij}]$ and a $(p \times q)$ matrix $B = [b_ij]$ is an $(mp \times nq)$ matrix given by

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & & a_{mn}B \end{pmatrix}$$

[85].

Now that we have considered the simple case of a spline with $b_{KS} = 1$ being the only non-zero coefficient, we generalise the result to any spline. This follows by considering the spline as a linear combination of single basis elements and the linearity of the transformation represented by the Kronecker product matrix. Clearly in the case of prolongation the Kronecker product corresponds to prolongating first the rows of coefficients then the columns. Each row, $\mathbf{b}_i$, is operated on by the prolongation matrix P. The prolongated rows are then added together in an appropriate linear combination to prolongate the columns. The Kronecker product matrix for prolongating biquadratic B-splines is depicted in Figure 7.2.

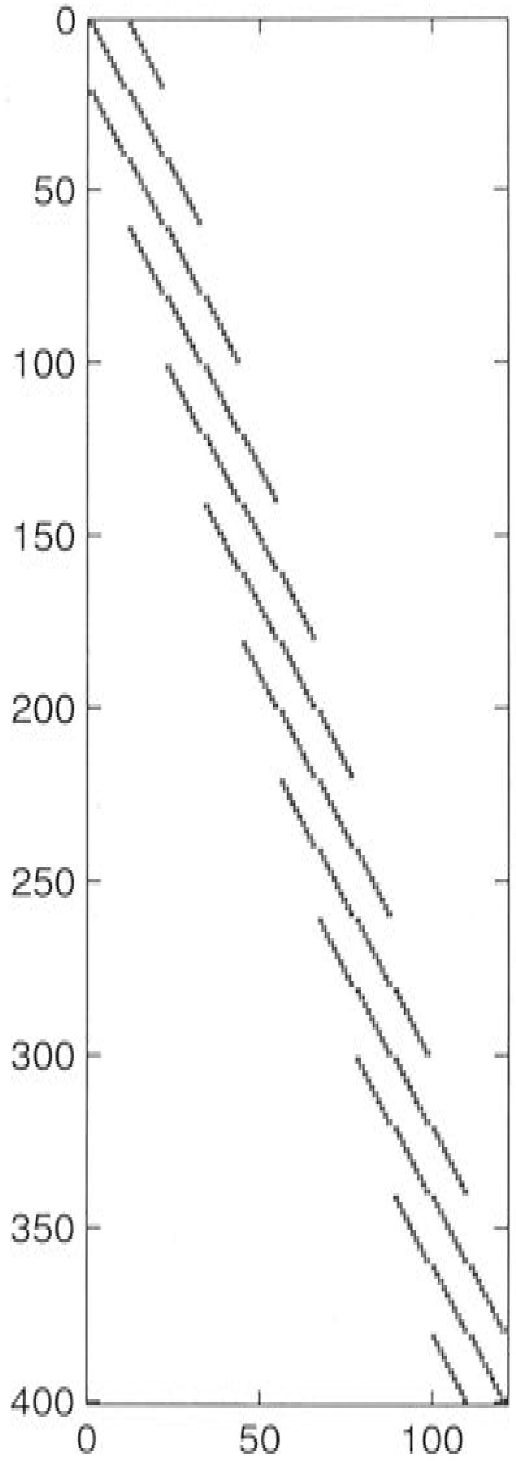


Figure 7.2: Position of the non-zero entries in the tensor product matrix for prolongation of a bivariate quadratic B-spline.

7.2 Restriction

The bivariate restriction process is analogous to the prolongation process above. The bivariate restriction minimisation problem is

$$\begin{aligned}\text{Minimise} \iint \Bigg(& \sum_{I=0}^{M_{l+1}+1} \sum_{J=0}^{N_{l+1}+1} b_{IJ} B_{l+1,I}(x) B_{l+1,J}(y) \\ & - \sum_{i=0}^{M_l+1} \sum_{j=0}^{N_l+1} a_{ij} B_{l,i}(x) B_{l,j}(y) \Bigg)^2 dx dy\end{aligned} \tag{7.10}$$

Differentiating with respect to coefficients b_{KS} gives

$$\begin{aligned}\sum_{I=0}^{M_{l+1}+1}\sum_{J=0}^{N_{l+1}+1} b_{IJ} \iint B_{l+1,I}(x)B_{l+1,J}(y)B_{l+1,K}(x)B_{l+1,S}(y)dxdy \\ = \sum_{i=0}^{N_l+1}\sum_{j=0}^{M_l+1} a_{ij} \iint B_{l,i}(x)B_{l,j}(y)B_{l+1,K}(x)B_{l+1,S}(y)dxdy\end{aligned} \tag{7.11}$$

Separating out the integrals gives

$$\begin{aligned}\sum_{I=0}^{M_{l+1}+1}\sum_{J=0}^{N_{l+1}+1} b_{IJ} \int B_{l+1,I}(x)B_{l+1,K}(x)dx \int B_{l+1,J}(y)B_{l+1,S}(y)dy \\ = \sum_{i=0}^{M_l+1}\sum_{j=0}^{N_l+1} a_{ij} \int B_{l,i}(x)B_{l+1,K}(x)dx \int B_{l,j}(y)B_{l+1,S}(y)dy\end{aligned} \tag{7.12}$$

Looking at the left hand side, we can see that $\int B_{l+1,J}(y)B_{l+1,S}(y)dy$ will stay constant for each row, and on the right hand side $\int B_{l,j}(y)B_{l+1,S}(y)dy$ will stay constant for each row. We can write (7.12) as

$$X_2 \otimes X_1\mathbf{b} = Y_2 \otimes Y_1\mathbf{a} \tag{7.13}$$

where X_1 and X_2 are restriction matrices X, discussed in Chapter 5, for the x and y directions respectively and Y_1 and Y_2 are matrices Y for each direction.

Both these Kronecker product matrices are depicted in Figures 7.3 and 7.4. These operations can also be seen to act first on the rows and secondly on the columns. On the right hand side, for example, the matrix Y_1 acts on each row, then the rows are multiplied by elements of the matrix Y_2 and added together to restrict the columns. This process is shown in Figure 7.5.

The least squares restriction has been used in this study to determine the difference between fine grid and coarse grid bivariate quadratic B-spline functions. For the bivariate case the orthogonal distance between the solution estimate

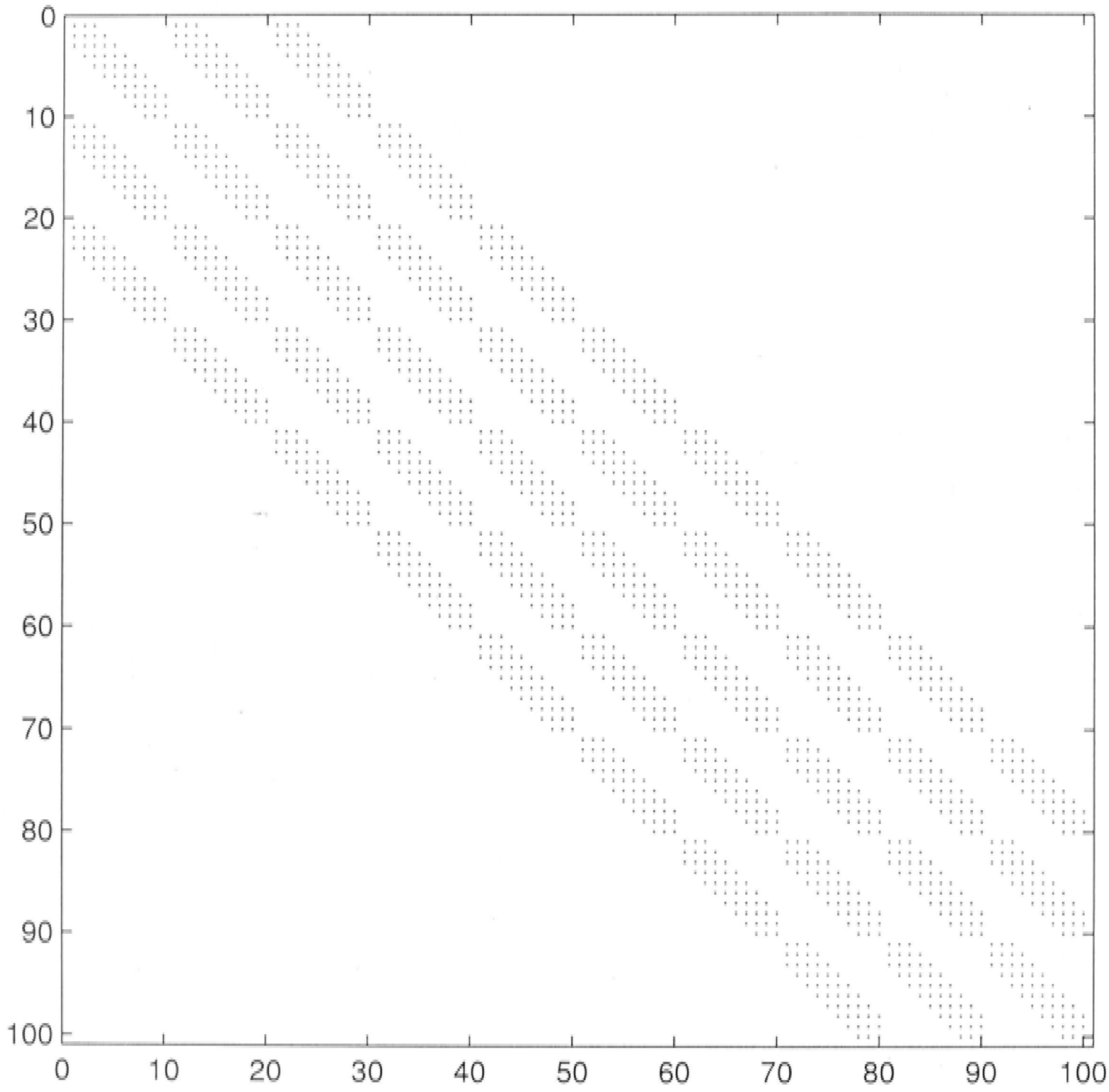


Figure 7.3: Positions of non-zero entries in the tensor product matrix $X_2 \otimes X_1$.

on grid l and its least squares fit to grid $l+1$ is

$$g_l = \sum_{i=0}^{M_l+1} \sum_{j=0}^{N_l+1} \zeta_{ij} B_{l,i}(x) B_{l,j}(y) \tag{7.14}$$

where

$$\boldsymbol{\zeta} = \mathbf{a} - P \otimes Q\mathbf{b} \tag{7.15}$$

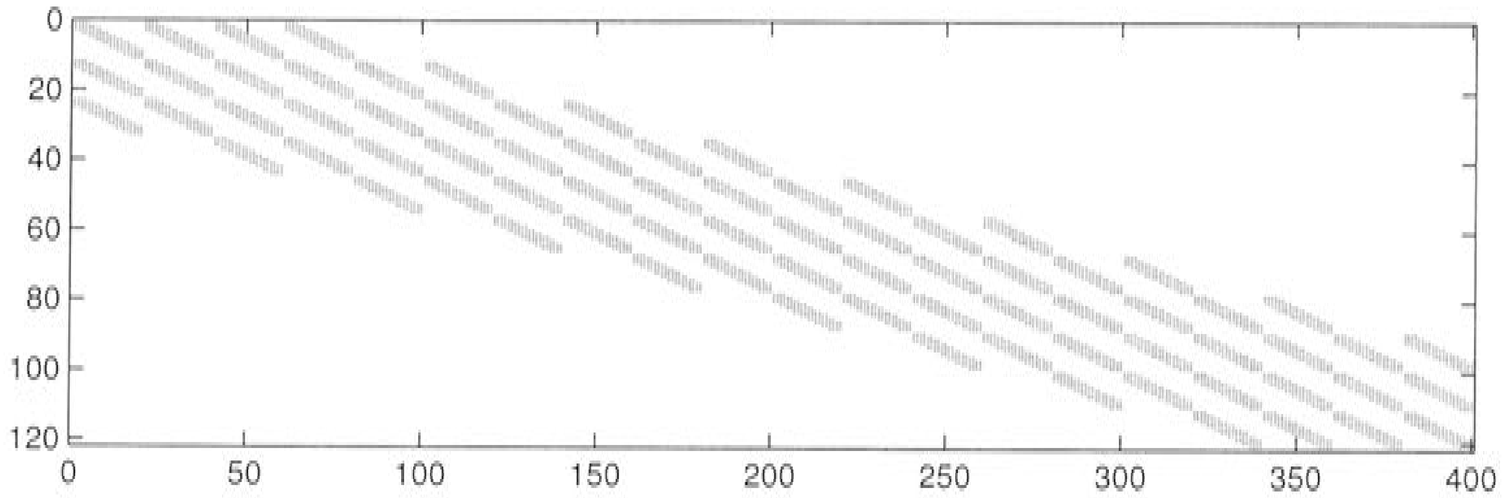


Figure 7.4: Positions of non-zero entries in the tensor product matrix $Y_2 \otimes Y_1$.

recalling that $\mathbf{b}$ is given by (7.13). The norm of this function is

$$\|g_l\| = \sqrt{h^2 \zeta^T X_2 \otimes X_1 \zeta} \tag{7.16}$$

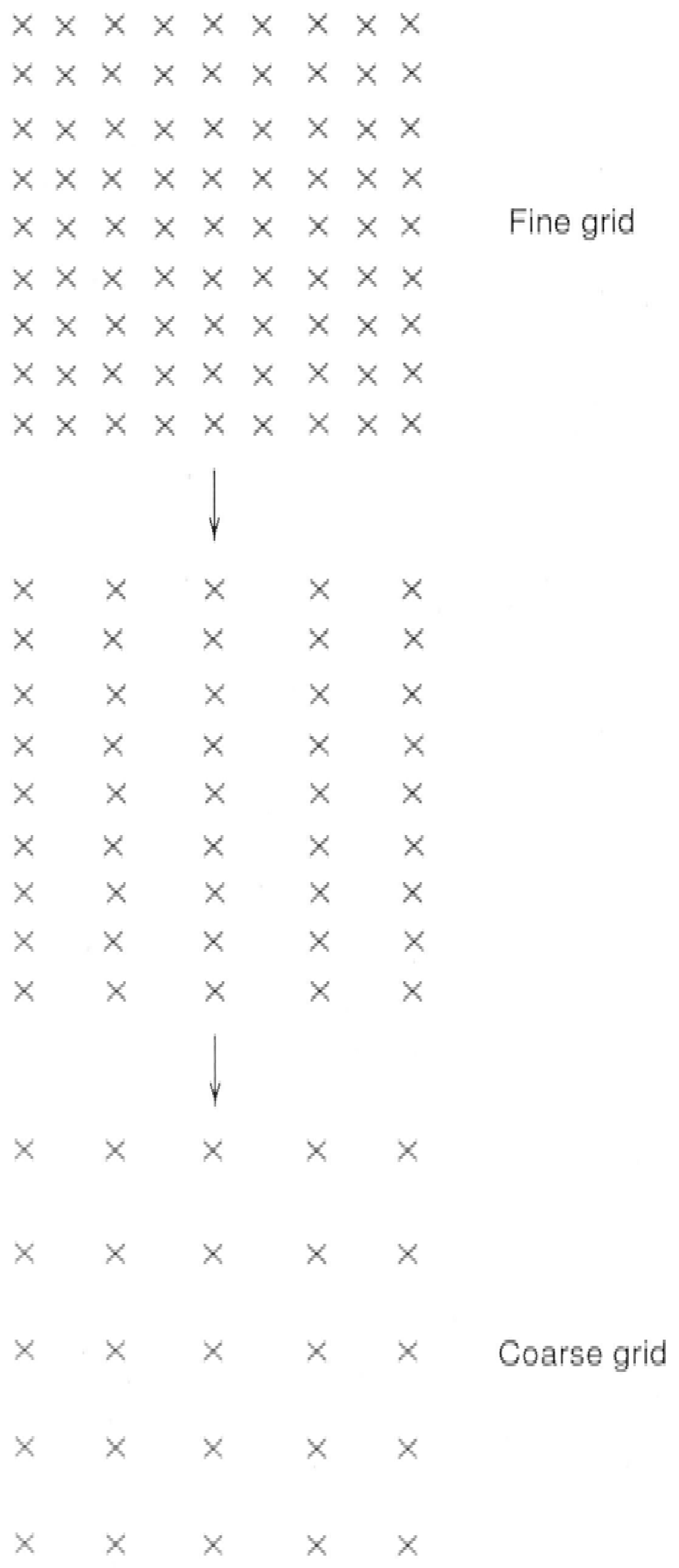


Figure 7.5: The process of restricting a biquadratic B-spline.

Chapter 8

Optimising the smoothing parameter

The techniques used in this study for optimising the parameter λ for the univariate discretised system in equations (3.5) and (3.29) or the bivariate discretised system in equation (6.80) are based on an adaptive iterative strategy described in Hutchinson [67]. The process involves double iteration to attain increasingly accurate estimates of both the solution and the smoothing parameter. The method uses the nested grid method to iteratively solve the system whilst periodically updating the estimate of λ using the current solution estimate. This is designed to be more efficient than converging completely to the solution for a given estimate of λ before obtaining a more accurate update. However, it does compromise the multigrid process by repeatedly altering the discretised system, changing the target solution and adding new error components to the existing solution estimate.

The Hutchinson [67] study optimised the smoothness by iteratively updating the smoothing parameter λ to deliver a user specified residual sum of squares. This method is modified here to optimise the smoothness by minimising GCV. Both of these iterative processes are outlined in the following paragraphs.

8.1 The OPTRSS algorithm

As the name suggests, the OPTRSS algorithm optimises the smoothing parameter to yield a prescribed residual sum of squares from the data. The algorithm, detailed in Hutchinson [67], uses an iterative Newton procedure to produce smoothing parameter estimates that converge to the prescribed residual sum of squares. Such a method is useful in terrain modelling where some information about data errors is known [58]. The OPTRSS algorithm was used in this study to understand and test the double iteration framework before applying it to minimisation of the GCV, given that it is a simpler, well-tested algorithm. The OPTRSS procedure is based on constructing a Newton increment in the smoothing parameter, denoted $\delta\theta$, from the residual sum of squares as follows:

$$\delta\theta = (S - R)/\frac{dR}{d\theta} \tag{8.1}$$

where S is the prescribed residual sum of squares, R is the residual sum of squares for the current solution estimate and $\theta = ln\lambda$. This scheme is used to obtain updates to the θ estimate that lead to convergence of R to S. The logarithm of the smoothing parameter is used to ensure that the smoothing parameter estimate does not become negative in the initial stages of the iterative procedure.

The Newton scheme requires calculation of $dR/d\theta$. From Chapters 3 and 6, covering both piecewise constant and quadratic B-spline discretisation for the univariate case, and biquadratic B-spline discretisation for the bivariate case, we write the generic system of discretised thin plate smoothing spline equations as

$$(P^TP + \lambda B)\mathbf{u} = P^T\mathbf{z} \tag{8.2}$$

where $\mathbf{z}$ is the vector of data values. Both P and $\mathbf{u}$ depend on whether piecewise constants or quadratic B-splines are used to approximate the thin plate spline solution f. For piecewise constant discretisation, $\mathbf{u}$ is a vector of coefficients for first order B-splines as in equation (3.5), where $u_i = f_i$. For univariate quadratic B-spline discretisation $\mathbf{u}$ stores the coefficients in equation (3.29), so $u_i = \alpha_i$. For the bivariate case, $u_i = \alpha_{IJ}$, as in equation (6.80). The vector $P\mathbf{u}$ gives the values of the function f at the data point locations. The matrix P is given by (3.4) for piecewise constants, (3.28) for univariate quadratic B-splines and (6.79) for biquadratic B-splines. The matrix B is the generic roughness penalty matrix, which is given by Q^TQ/h^3 for the univariate case and Z/h^2 for the bivariate case. Now we can write the residual sum of squares for any of the cases above as

$$R = \|P\mathbf{u} - \mathbf{z}\|^2 = \mathbf{u}^T P^T P\mathbf{u} - 2\mathbf{u}^T P\mathbf{z} + \mathbf{z}^T\mathbf{z} \tag{8.3}$$

Applying the chain rule to equation (8.3) gives

$$\frac{dR}{d\lambda} = \frac{dR}{d\mathbf{u}} \cdot \frac{d\mathbf{u}}{d\lambda} = 2(\mathbf{v}^1)^T\frac{d\mathbf{u}}{d\lambda} \tag{8.4}$$

where

$$\mathbf{v}^1 = P^T P\mathbf{u} - P^T\mathbf{z} = P^T(P\mathbf{u} - \mathbf{z}) \tag{8.5}$$

Differentiation of the discretised system in (8.2) with respect to λ gives

$$(P^T P + \lambda B)\frac{d\mathbf{u}}{d\lambda} = -B\mathbf{u} = \mathbf{v}^1/\lambda \tag{8.6}$$

Converting to differentiation with respect to θ gives

$$(P^T P + \lambda B)\frac{d\mathbf{u}}{d\theta} = -\lambda B\mathbf{u} = \mathbf{v}^1 \tag{8.7}$$

given that

$$\frac{d\lambda}{d\theta} = e^\theta = \lambda \tag{8.8}$$

Thus $d\mathbf{u}/d\theta$ satisfies the same system of equations as $\mathbf{u}$, but with the data vector $P^T\mathbf{z}$ replaced by the vector $\mathbf{v}^1$. It is therefore possible to simultaneously solve both (8.2) and (8.7) numerically with a relatively low storage requirement.

Having established the Newton method procedure, the OPTRSS algorithm is given by modifying the nested grid algorithm given in section 4.2.4. Convergence of the smoothing parameter estimate on a given grid was determined by a criterion Q, and the resolution refinement process was terminated by a criterion D. Assuming that we start on a grid of coarseness 2^l and refine the resolution by a factor of 2, the algorithm can be written as follows:

```
while $D > tol1$
    $q = 0$
    while $Q > tol2$
        $\mathbf{u}_l(\theta_q) = S_l^{v_1}(\mathbf{u}_l(\theta_q), \mathbf{z}_l)$
        $\mathbf{u}'_l(\theta_q) = S_l^{v_1}(\mathbf{u}'_l(\theta_q), \mathbf{v}_l)$
        $\delta\theta = (S - R)/\frac{dR}{d\theta}$
        $\theta_{q+1} = \theta_q + \delta\theta$
        $q = q + 1$
    end
    $\mathbf{u}_{l-1} = T_l\mathbf{u}_l$
    $\mathbf{u}'_{l-1} = T_l\mathbf{u}'_l$
    $l = l - 1$
end
$q = 0$
while $Q > tol2$
    $\mathbf{u}_l = S_l^{v_1}(\mathbf{u}_l, \mathbf{z}_l)$
    $\mathbf{u}'_l = S_l^{v_1}(\mathbf{u}'_l, \mathbf{v}_l)$
    $q = q + 1$
end
```

The process in the above algorithm is to perform v_1 smoothing iterations on the discretised system on grid level l, to solve for $\mathbf{u}_l$, the solution on level l, and its derivative with respect to θ, $\mathbf{u}'_l$. The estimate of $\mathbf{u}'_l$ is then used to update θ_q, the q^{th} update of θ, to converge to the smoothing parameter corresponding to the prescribed residual S. The procedure is repeated until the θ updates converge, as determined by the criterion Q. The value of Q was set to the sum of the absolute differences between 4 consecutive θ updates. The appropriateness of this tolerance is discussed in Chapter 12. When the θ updates converge, the estimates of the discretised solution and its derivative are prolongated to grid level $l-1$. The prolongation operator is denoted T_l. For the piecewise constant univariate discretised system, the prolongation operator T_l is the linear interpolation operator given in (4.22), and for the quadratic B-spline univariate discretised system the prolongation operator corresponds to hierarchial B-spline refinement and is given by (5.10). For the biquadratic

B-spline discretised system, T_l is the tensor product prolongation in (7.9). This process continues until the criterion D prevents further grid refinement. The value of D is given by

$$D = \frac{\|g_l\|}{\|f_l\|} \tag{8.9}$$

where g_l comes from the least squares restriction discussed in section 5.3.3. The value of $\|g_l\|$, normalised by $\|f_l\|$, is a measure of the fine scale information contributed by the finer grid. If this measure was less than an experimentally determined tolerance, it was considered unnecessary to consider refinement. This is discussed further in Chapter 11. However, during the initial testing phases reported in Chapters 10 and 11 the number of θ updates and the number of grids was often allowed to go beyond the tolerance to observe the behaviour of the algorithm.

8.2 The MINGCV algorithm

The objective of this study was to develop a procedure to iteratively solve for minimum GCV bivariate finite element thin plate smoothing splines. The minimum GCV criterion is used to optimise the predictive accuracy of the fitted surface, as discussed in Chapters 1 and 2. The algorithm developed during this study to converge to a minimum GCV solution is termed the MINGCV algorithm. The MINGCV algorithm has the same design as the OPTRSS algorithm, but λ is optimised to achieve a minimum GCV rather than a prescribed residual sum of squares. This is done using the following second order taylor series expansion

$$GCV(\theta) = GCV(\theta_q) + \frac{dGCV(\theta_q)}{d\theta}(\theta - \theta_q) + \frac{d^2GCV(\theta_q)}{d\theta^2}\frac{(\theta - \theta_q)^2}{2} \tag{8.10}$$

Differentiating with respect to θ gives

$$\frac{dGCV(\theta)}{d\theta} = b + c\theta \tag{8.11}$$

where

$$b = -\frac{dGCV(\theta_q)}{d\theta} + \frac{d^2GCV(\theta_q)}{d\theta^2}\theta_q \tag{8.12}$$

$$c = \frac{d^2GCV(\theta_q)}{d\theta^2} \tag{8.13}$$

Thus the value of θ that minimises GCV is estimated by

$$\theta = -b/c \tag{8.14}$$

This procedure requires estimates for the first and second derivatives of $GCV(\theta_p)$. The expression for the GCV in equation (2.23) can be expressed as

$$\frac{nR}{Tr^2} \tag{8.15}$$

where $Tr = tr(I - A)$ and A is the influence matrix defined in (2.22). Differentiating equation (2.23) gives

$$\frac{dGCV}{d\theta} = \frac{n}{Tr^4}\left(\frac{dR}{d\theta}Tr^2 - 2Tr\frac{dTr}{d\theta}R\right) \tag{8.16}$$

$$\begin{aligned}\frac{d^2GCV}{d\theta^2} = \frac{n}{Tr^4}&\left(\frac{d^2R}{d\theta^2}Tr^2 - \left(2Tr\frac{dTr}{d\theta} + 2\left(\frac{dTr}{d\theta}\right)^2\right)R\right)\\ &- \frac{4n}{Tr^5}\frac{dTr}{d\theta}\left(\frac{dR}{d\theta}Tr^2 - 2Tr\frac{dTr}{d\theta}R\right)\end{aligned} \tag{8.17}$$

These derivatives require an estimate of the second derivative of R in addition to the first derivative required by the OPTRSS algorithm. This is obtained by differentiating equation (8.4) to give

$$\frac{d^2R}{d\theta^2} = 2(\mathbf{v}^1)^T\frac{d^2\mathbf{u}}{d^2\theta} + 2P^TP\left(\frac{d\mathbf{u}}{d\theta}\right)^2 \tag{8.18}$$

An estimate of $d^2\mathbf{u}/d\theta^2$ is obtained by differentiating equation (8.7) to give

$$(P^TP + \lambda B)\frac{d^2\mathbf{u}}{d\theta^2} = -\mathbf{v}^1 + 2P^TP\frac{d\mathbf{u}}{d\theta} = \mathbf{v}^2 \tag{8.19}$$

Thus $d^2\mathbf{u}/d\theta^2$ is obtained by solving the same system of equations as those that give $d\mathbf{u}/d\theta$ and $\mathbf{u}$.

The equations for the GCV and its derivatives also involve Tr, $dTr/d\theta$ and

$d^2Tr/d\theta^2$. Calculation of the trace and its derivatives requires some knowledge of the influence matrix A. Given that this matrix is not involved in the previous calculations of the discretised system, Hutchinson [59] proposes a computationally efficient approximate method that yields a stochastic estimate of Tr. This method is discussed below.

8.2.1 A stochastic estimate for the trace of the influence matrix

The stochastic estimator developed by Hutchinson [59] is motivated by the relation

$$tr(A) = \sum_{i=1}^{n} \mathbf{e}_i^T A \mathbf{e}_i \tag{8.20}$$

where $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ are the n linearly independent columns of the $n \times n$ identity matrix. This summation is numerically impractical, since it requires n solutions $\{A\mathbf{e}_1, \ldots, A\mathbf{e}_\mathrm{n}\}$ of the smoothing spline. Hutchinson [59] therefore suggests replacing the n linearly independent vectors by a single vector with stochastically independent entries. It is shown in Hutchinson [59] that

$$E(\mathbf{t}^T A\mathbf{t}) = \sigma^2 trA \tag{8.21}$$

where $\mathbf{t} = (t_1, \ldots, t_n)^T$ is a vector of n independent samples from a random variable T with mean zero and variance σ^2. This follows immediately from the expansion

$$\mathbf{t}^T A\mathbf{t} = \sum_{i,j} t_i a_{ij} t_j \tag{8.22}$$

The value of $E(T^2)$ is σ^2, and the value of $E(T)E(T)$ is zero, so the only non-zero terms in the sum are those in which $i = j$.

Thus $\mathbf{t}^T A\mathbf{t}$ is an unbiased estimator of trA if $\sigma^2 = 1$. This relation has been recognised by both Girard [36] and Hutchinson [59]. Girard [36] takes T to be a standard normal random variable. Hutchinson [59] proposes a minimum variance estimator, obtained by chosing $\mathbf{t}$ to be a vector of n independent samples from the discrete random variable T which takes the values 1,-1 each with probability 1/2 [59]. This estimator has been employed in a wide range of numerical problems, including calculation of the trace and determinant of symmetric positive definite matrices, and general Tikhonov regularization problems and associated applications, as documented by [38, 37, 4] and [114]. Hutchinson [59] also shows that, with this choice of $\mathbf{t}$, the relative standard error of $\mathbf{t}^T(I-A)\mathbf{t}$ in estimating $tr(I-A)$ is bounded by

$$(2/n)^{1/2} \tag{8.23}$$

It is further shown in [59] that this is a conservative bound, so the actual standard error could be expected to be considerably less.
An estimate of Tr at any point in the iterative procedure can be obtained by applying the MINGCV procedure to the equation

$$(P^TP+\lambda B)\mathbf{b} = P^T\mathbf{t} = \mathbf{w}^0 \tag{8.24}$$

Equation (8.24) is the same as the discretised system for the smoothing spline, but it has $P^T\mathbf{t}$ as the right hand side instead of $P^T\mathbf{z}$. The vector $A\mathbf{t}$ is then estimated by $P\mathbf{b}$. With this implicit estimate of the influence matrix, the trace estimate is given by

$$trA = \mathbf{t}\hat{\mathbf{t}} \tag{8.25}$$

where $\hat{\mathbf{t}} = P\mathbf{b}$. The first and second derivatives of the vector $\mathbf{b}$ are given by

$$(P^TP+\lambda B)\frac{d\mathbf{b}}{d\theta} = \mathbf{w}^1 \tag{8.26}$$

where

$$\mathbf{w}^1 = P^TP\mathbf{b} - P^T\mathbf{t} \tag{8.27}$$

and

$$(P^TP+\lambda B)\frac{d^2\mathbf{b}}{d\theta^2} = \mathbf{w}^2 \tag{8.28}$$

where

$$\mathbf{w}^2 = -\mathbf{w}^1 + 2P^TP\frac{d\mathbf{b}}{d\theta} \tag{8.29}$$

The first and second derivatives of Tr are then calculated by

$$\frac{dTr}{d\theta} = -\frac{d\hat{\mathbf{t}}}{d\theta}\mathbf{t} \tag{8.30}$$

$$\frac{d^2Tr}{d\theta^2} = -\frac{d^2\hat{\mathbf{t}}}{d\theta^2}\mathbf{t} \tag{8.31}$$

8.2.2 The algorithm

The MINGCV algorithm is given on the next page, where $\mathbf{v}_l^0$ is $P^T\mathbf{z}_l$ and $\mathbf{u}^{(m)}$ is the m^{th} derivative of $\mathbf{u}$ with respect to θ. The MINGCV algorithm proceeds in a similar manner to the OPTRSS algorithm. This time, smoothing iterations are performed to solve for m derivatives of the solution estimate and n derivatives of the vector $\mathbf{b}$. These estimates are all needed to update θ_q to converge to the minimum GCV smoothing parameter. All these vectors need to be prolongated when refinement occurs.

8.2.3 Differentiation of Tr

The above process for obtaining the derivatives of Tr can be made considerably more efficient by differentiating with respect to λ instead of θ. Equations (8.26) and (8.28) can be written as

$$(P^TP + \lambda B)\frac{d\mathbf{b}}{d\lambda} = -B\mathbf{b} = \mathbf{w}^1 \tag{8.32}$$

$$(P^TP + \lambda B)\frac{d^2\mathbf{b}}{d\lambda^2} = -2B\frac{d\mathbf{b}}{d\lambda} = \mathbf{w}^2 \tag{8.33}$$

Note that the right hand side of expressions (8.32) and (8.33) has been expressed in terms of the solution vector $\mathbf{b}$ rather than the data vector $\mathbf{t}$. Rearranging (8.33) gives

$$\frac{d\mathbf{b}}{d\lambda} = -(P^TP + \lambda B)^{-1}B\mathbf{b} \tag{8.34}$$

Differentiating $\mathbf{t}^TP\mathbf{b}$ with respect to λ and using (8.34) and (8.24) gives

$$\frac{dTr}{d\lambda} = \mathbf{b}^TB\mathbf{b} \tag{8.35}$$

and

$$\frac{dTr}{d\theta} = \lambda\mathbf{b}^TB\mathbf{b} \tag{8.36}$$

Differentiating again gives

$$\frac{d^2Tr}{d\theta^2} = \lambda^2\frac{d^2Tr}{d\lambda^2} + \lambda\frac{dTr}{d\lambda} = -2\lambda^2(B\mathbf{b})^T\frac{d\mathbf{b}}{d\lambda} + \lambda\mathbf{b}^TB\mathbf{b} \tag{8.37}$$

```
while $D > tol1$
    $q = 0$
    while $Q > tol2$
        $\mathbf{u}_l(\theta_q) = S_l^{v_1}(\mathbf{u}_l(\theta_q), \mathbf{v}_l^0)$
        $\mathbf{u}_l'(\theta_q) = S_l^{v_1}(\mathbf{u}_l'(\theta_q), \mathbf{v}_l^1)$
        $\mathbf{u}_l''(\theta_q) = S_l^{v_1}(\mathbf{u}_l''(\theta_q), \mathbf{v}_l^2)$

        for $n = 0$ to $2$
            $\mathbf{b}_l^{(n)}(\theta_q) = S_l^{v_1}(\mathbf{b}_l^{(n)}(\theta_q), \mathbf{w}_l^n)$

            $\mathbf{b}_l^{(n)}(\theta_q) = S_l^{v_1}(\mathbf{b}_l^{(n)}(\theta_q), \mathbf{w}_l^n)$
        end

        $\theta_{q+1} = -\frac{b}{2c}$
        $q = q + 1$
    end
    for $m = 0$ to $2$
        $\mathbf{u}_{l-1}^{(m)} = T_l \mathbf{u}_l^{(m)}$
    end
    for $n = 0$ to $2$
        $\mathbf{b}_{l-1}^{(n)} = T_l \mathbf{b}_l^{(n)}$
    end
    $l = l - 1$
end
$q = 0$
while $Q > tol2$
    for $m = 0$ to $2$
        $\mathbf{u}_1^{(m)}(\theta_q) = S_l^{v_1}(\mathbf{u}_l^{(m)}(\theta_q), \mathbf{v}_l^m)$
    end
    for $n = 0$ to $2$
        $\mathbf{b}_1^{(m)}(\theta_q) = S_l^{v_1}(\mathbf{b}_l^{(n)}(\theta_q), \mathbf{w}_l^n)$
    end
    $\theta_{q+1} = -\frac{b}{2c}$
    $q = q + 1$
end
```

Note that the expression for $dTr/d\theta$ does not involve $d\mathbf{b}/d\lambda$ and $d^2Tr/d\theta^2$ only requires the first derivative of $\mathbf{b}$ with respect to λ. This cuts v_1 smoothing iterations out of each θ update procedure in the MINGCV algorithm. The modified MINGCV algorithm is given on the next page.

8.2.4 An alternative for the bivariate case

Most of the exploratory analysis for the MINGCV algorithm was done using the univariate processes. The design of the bivariate algorithm was therefore more structured, and it was decided to use a consistent formulation for the above systems of equations. Given the above results for Tr and its derivatives, it was decided to differentiate both the vectors $\mathbf{u}$ and $\mathbf{b}$ with respect to λ. This gives the following equations for the bivariate solution and its derivatives.

$$(P^TP + \lambda B)\mathbf{u} = P^T\mathbf{z} \tag{8.38}$$

$$(P^TP + \lambda B)\frac{d\mathbf{u}}{d\lambda} = -B\mathbf{u} \tag{8.39}$$

$$(P^TP + \lambda B)\frac{d^2\mathbf{u}}{d\lambda^2} = -2B\frac{d\mathbf{u}}{d\lambda} \tag{8.40}$$

Note that the right hand sides for (8.39) and 8.40 have now been expressed in terms of the solution vector $\mathbf{u}$ rather than the data vector $\mathbf{z}$. This was done to be consistent with the equations for $\mathbf{b}$. The equations for the derivatives of R in terms of $d\mathbf{u}/d\lambda$ and $d^2\mathbf{u}/d\lambda^2$ are

$$\frac{dR}{d\theta} = -2\frac{\lambda^2}{h^2}\mathbf{u}^TB\frac{d\mathbf{u}}{d\lambda} \tag{8.41}$$

$$\frac{d^2R}{d\theta^2} = -4\lambda^2\mathbf{u}B\frac{d\mathbf{u}}{d\lambda} - 2\lambda^3(\frac{d\mathbf{u}}{d\lambda})^TB\frac{d\mathbf{u}}{d\lambda} - 2\lambda^3\mathbf{u}^TB\frac{d^2\mathbf{u}}{d\lambda^2} \tag{8.42}$$

```
while $D > tol1$
    $q = 0$
    while $Q > tol2$
        for $m = 0$ to 2
            $\mathbf{u}_l^{(m)}(\theta_q) = S_l^{v_1}(\mathbf{u}_l^{(m)}(\theta_q), \mathbf{v}_l^m)$
        end
        for $n = 0$ to 1
            $\mathbf{b}_l^{(m)}(\theta_q) = S_l^{v_1}(\mathbf{b}_l^{(n)}(\theta_q), \mathbf{w}_l^n)$
        end
        $\theta_{q+1} = -\frac{b}{2c}$
        $q = q + 1$
    end
    for $m = 0$ to 2
        $\mathbf{u}_{l-1}^{(m)} = T_l \mathbf{u}_l^{(m)}$
    end
    for $n = 0$ to 1
        $\mathbf{b}_{l-1}^{(n)} = T_l \mathbf{b}_l^{(n)}$
    end
    $l = l - 1$
end
$q = 0$
while $Q > tol2$
    for $m = 0$ to 2
        $\mathbf{u}_1^{(m)}(\theta_q) = S_l^{v_1}(\mathbf{u}_l^{(m)}(\theta_q), \mathbf{v}_l^m)$
    end
    for $n = 0$ to 1
        $\mathbf{b}_1^{(m)}(\theta_q) = S_l^{v_1}(\mathbf{b}_l^{(n)}(\theta_q), \mathbf{w}_l^n)$
    end
    $\theta_{q+1} = -\frac{b}{2c}$
    $q = q + 1$
end
```

8.2.5 Variation in solution characteristics with changes in λ

There are some basic relationships between the statistics and parameters described in the previous sections that are useful in analysing the output of the MINGCV algorithm. Standard output descriptors include the smoothing parameter λ, the residual sum of squares R, the GCV and the signal, or $tr(A)$. These quantities are produced by the programs in ANUSPLIN [68], that calculates analytic minimum GCV thin plate smoothing splines. It was useful to this study to consider the variation in R, the GCV and the signal with changes in λ.

The typical structure of the GCV curve is shown in Figure 8.1 [68]. The GCV normally has a unique local minimum value for a smoothing parameter value $\lambda = \lambda_m$, unless there are significant errors in the data or the model has been significantly misspecified [68]. Using the GCV curve, the expected shapes of the curves for $dGCV/d\lambda$ and $d^2GCV/d\lambda^2$ were derived, as shown in Figure 8.2. A close up of the interval containing the λ values close in magnitude to λ_m is shown in Figure 8.3. The curves show that, for the region in the vicinity of

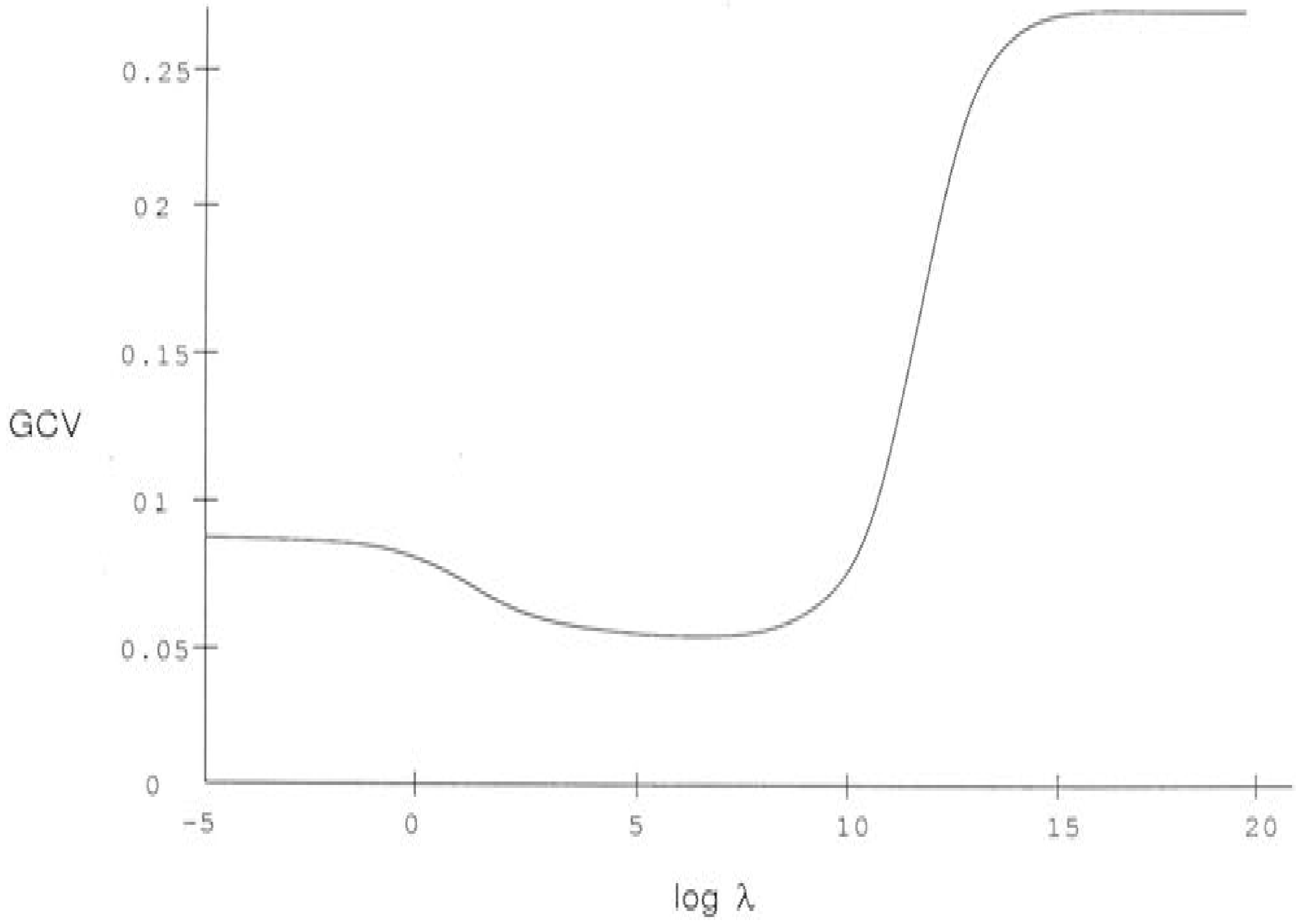


Figure 8.1: Plot of the GCV as a function of the logarithm of the smoothing parameter.

λ_m, the value of $dGCV/d\lambda$ should be negative if $\lambda < \lambda_m$, positive if $\lambda > \lambda_m$, and zero at the minimum. Near the endpoints of the GCV curve $dGCV/d\lambda$ will clearly approach zero, at a slow rate. The second derivative, d^2GCV/λ^2, should always be positive near λ_m, although at the endpoints its value may be negative or zero.

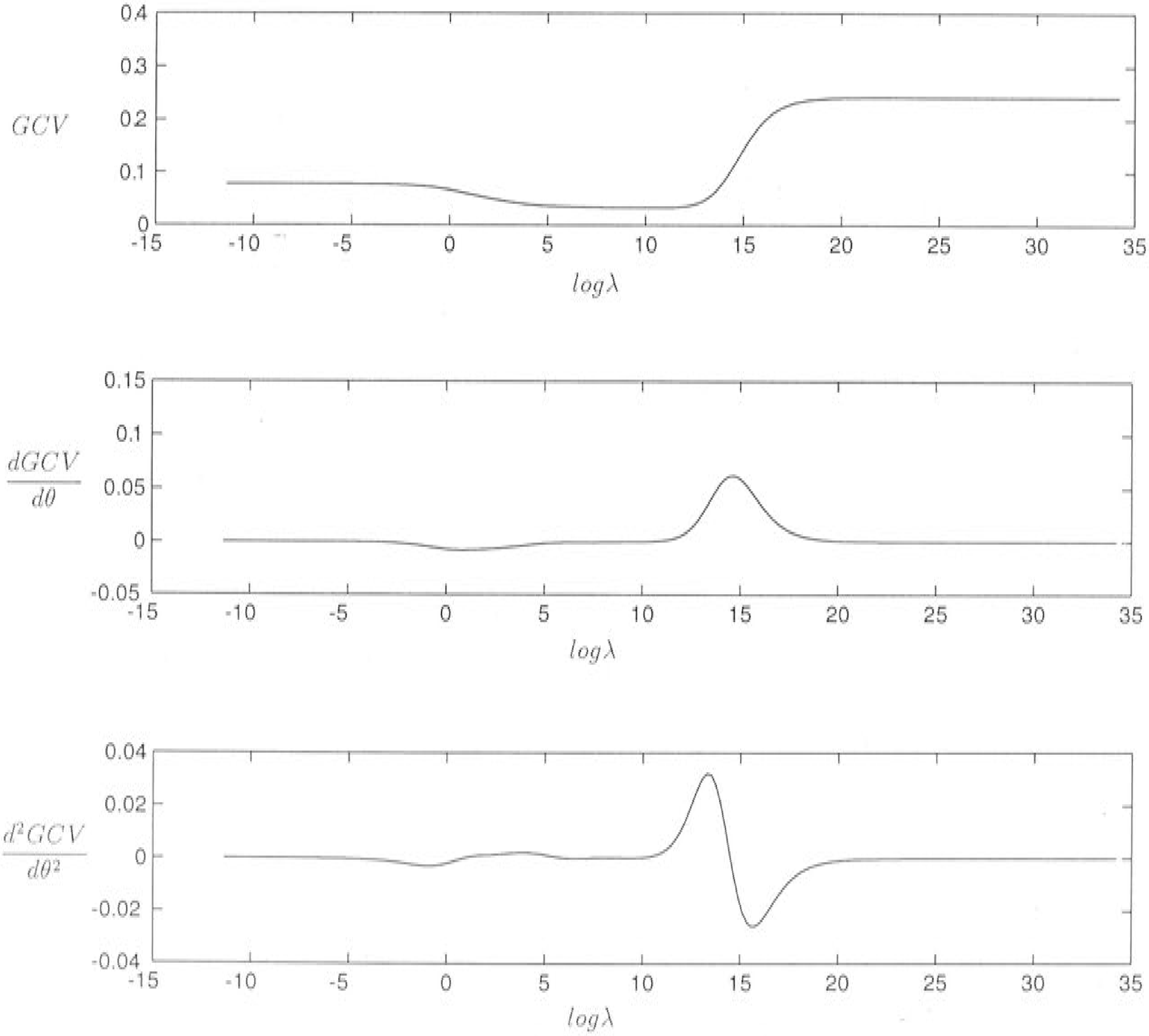


Figure 8.2: Plot of the GCV and its derivatives as a function of the logarithm of the smoothing parameter.

The curves for R are shown in Figure 8.4. As the smoothing parameter increases, the fitted curve becomes smoother and the data are not fitted as closely, so R increases. The value of $dR/d\lambda$ should therefore always be positive, leveling off at the endpoints. The value of $d^2R/d\lambda^2$ can be positive, zero or negative. The signal and its derivatives behave in the reverse manner,

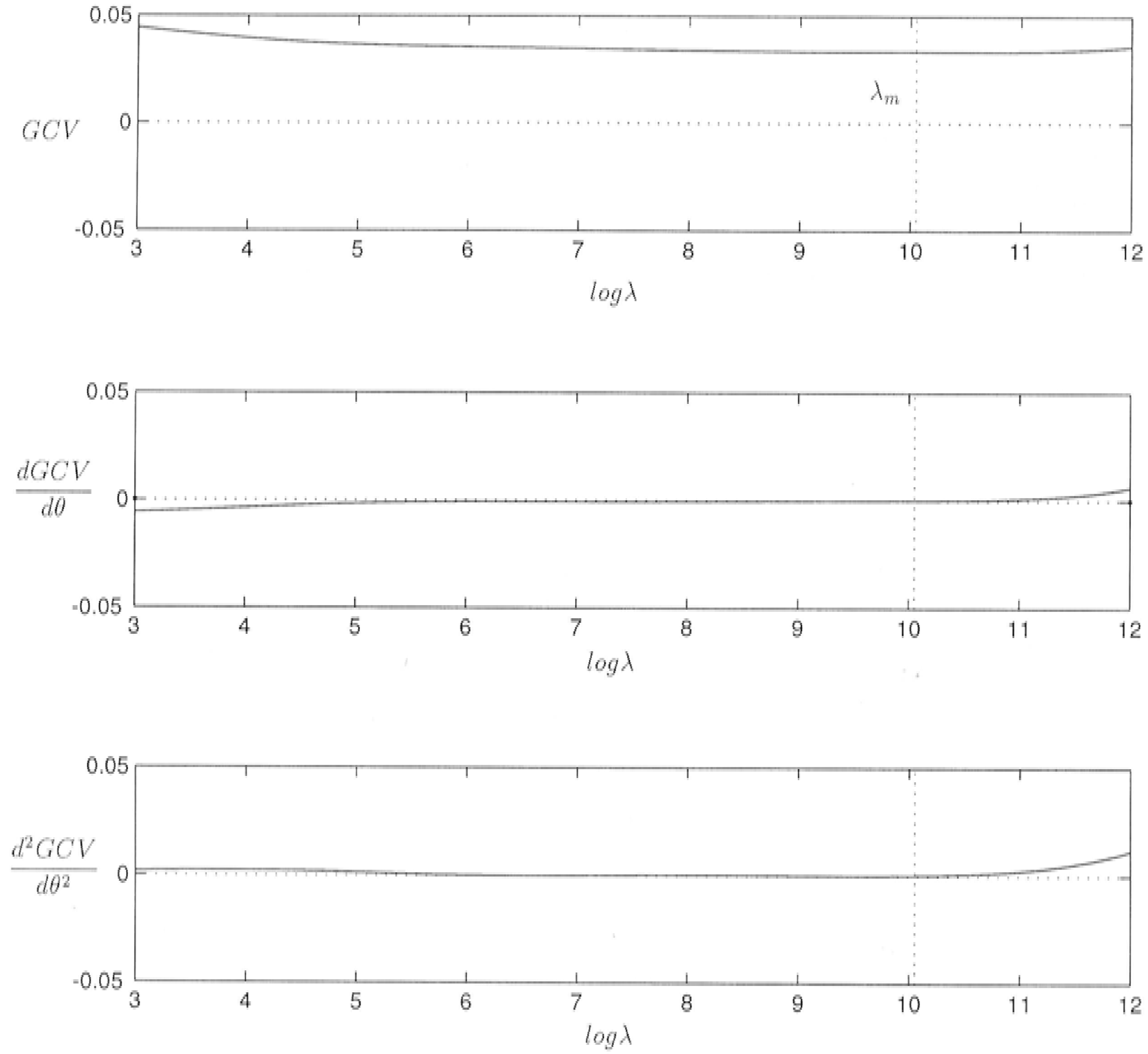


Figure 8.3: A closeup of Figure 8.2, showing the region surrounding the minimum GCV.

as shown in Figure 8.5. As the smoothing parameter decreases the structure of the fitted curve becomes more complicated and the effective number of parameters increases. A common feature of the patterns observed above is the inconsistency of the relationship at the endpoints of the curve, as the processes approaches exact interpolation or flat plane regression. The curve either changes erratically or weakens to no relationship. It is therefore expected that the values of R, the GCV and the signal will not give accurate predictions of the minimum GCV value at the end regions of the curve.

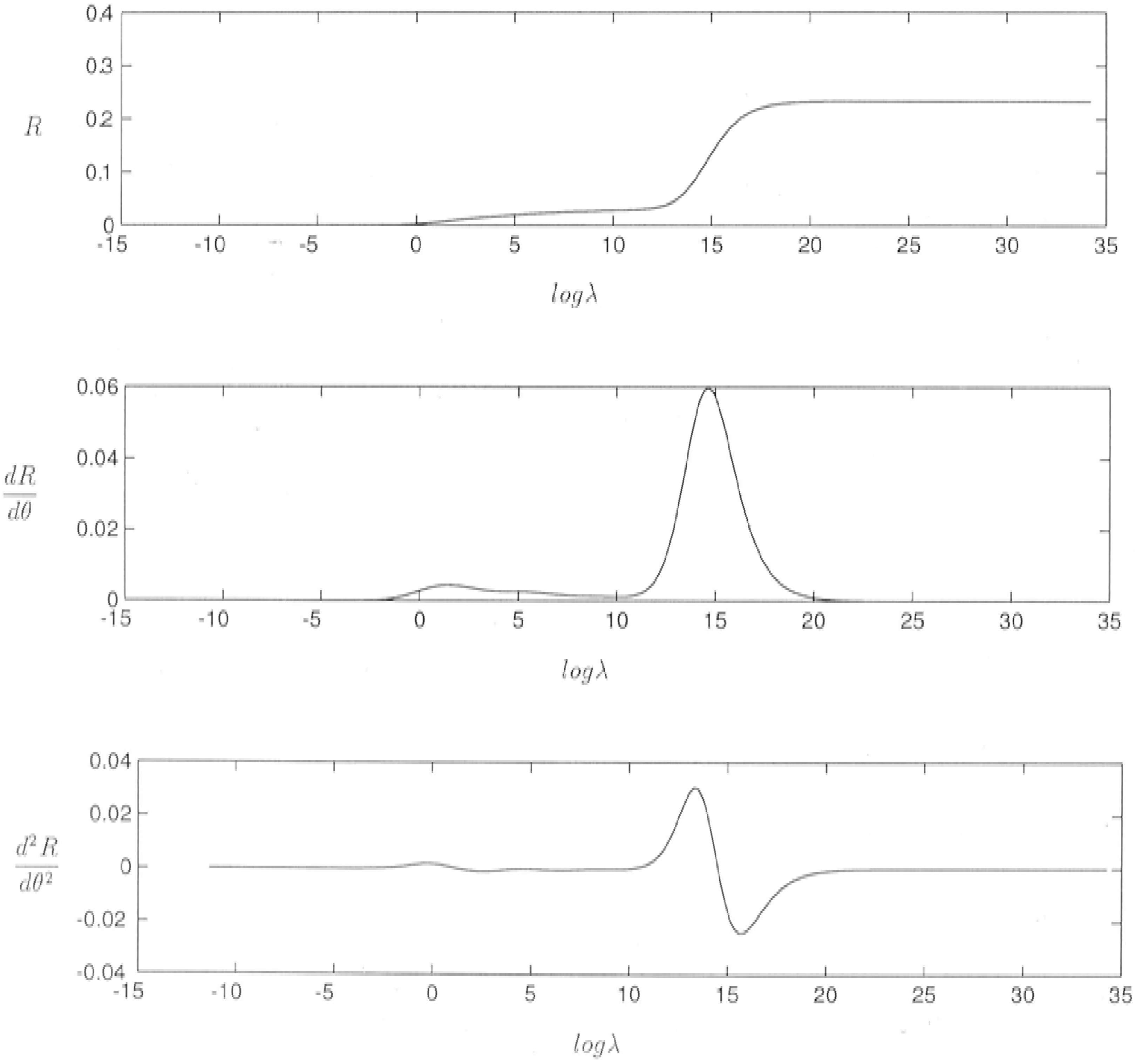


Figure 8.4: Plot of R and its derivatives as a function of the logarithm of the smoothing parameter.

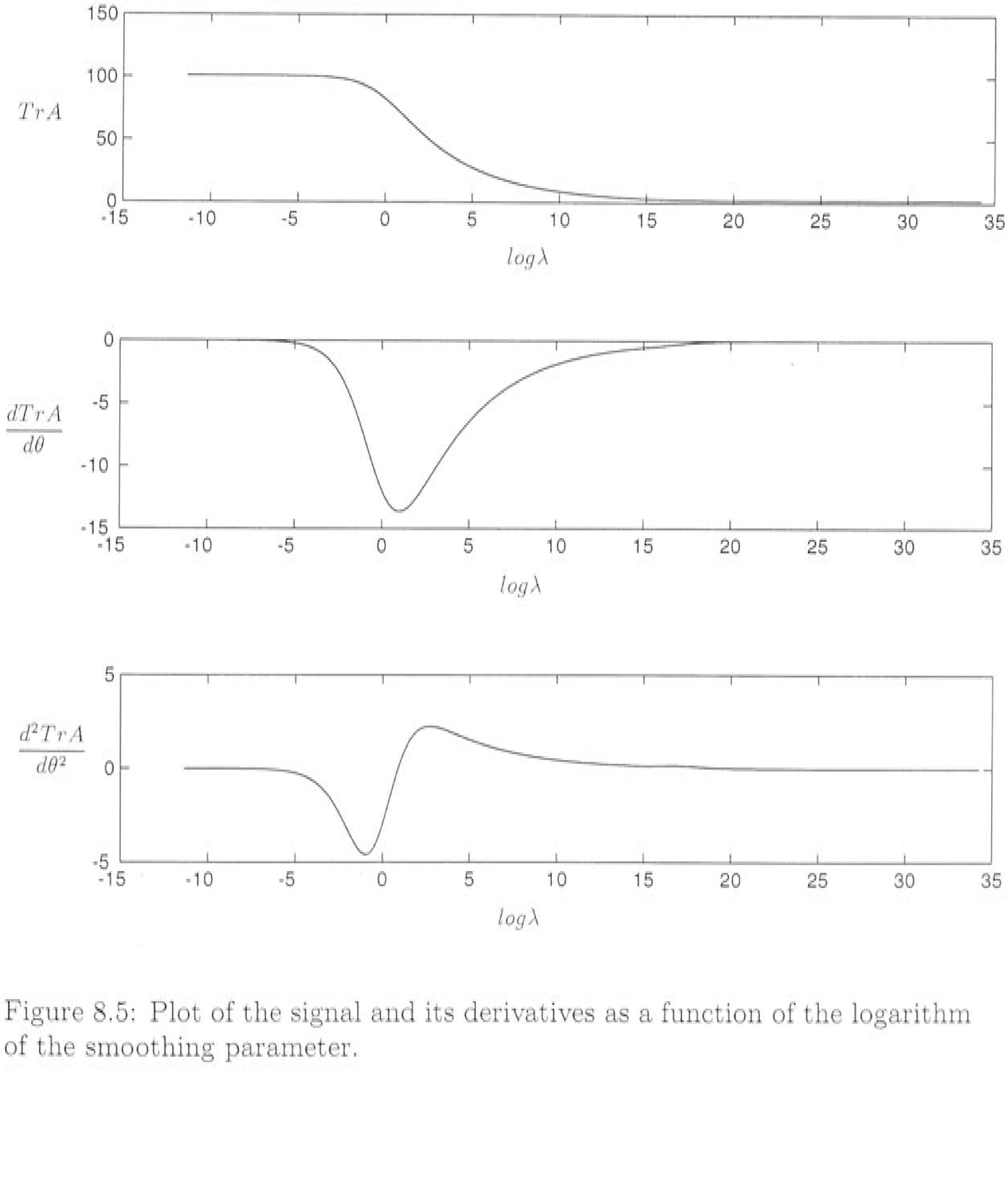


Figure 8.5: Plot of the signal and its derivatives as a function of the logarithm of the smoothing parameter.

Part II

Results

Chapter 9

Testing of multigrid algorithms

As discussed in Chapter 1, the initial method of discretisation was to use a piecewise constant approximation, with a constant value in each grid cell. Preliminary analysis of the v-cycle and nested grid multigrid schemes applied to the univariate piecewise constant smoothing spline system in equation (3.5) was conducted. These multigrid algorithms were investigated here on the basis of the review in Chapter 4. The nested grid and v-cycle algorithms are described in section 4.2.4. The aim was to assess the performance of multigrid as a method of iteratively solving the univariate piecewise constant smoothing spline problem without the additional complications associated with optimising smoothness. The optimal, or minimum GCV, smoothing parameter λ was therefore determined separately using the analytic procedures incorporated in the program ANUSPLIN [68]. The ANUSPLIN program calculates analytic thin plate spline fits to data, with the option of minimising GCV, and produces diagnostic statistics, such as the GCV, the residual sum of squares R, and the signal, or $tr(A)$.

The analyses in this chapter were designed to identify the key influences on the performance of multigrid algorithms when applied to piecewise constant smoothing splines, and to assist in the design of an optimal multigrid algorithm for solution of this system. It was emphasised in Chapter 4 that, while multigrid literature provides guidance and recommendations, multigrid schemes are very flexible in their design, allowing the user to experiment with different algorithms and settings to optimise convergence properties. The initial settings chosen for nested grid and v-cycle in this analysis are summarised in Table

1. The sensitivity of the multigrid algorithm to changes in these settings is discussed in this chapter.

Setting	**V-cycle**	**Nested grid**
No. of grid levels (L)	6	6
No. of smoothing iterations (v_1) per grid level	3	10
Coarsening ratio	2	2
Type of smoothing iteration	SOR	SOR
Relaxation parameter (ω)	1.6	1.6
Coarse grid operator	discretisation	discretisation
Type of prolongation	linear interpolation	linear interpolation
Type of restriction	Full weighting	Full weighting
No. of grid points on the fine grid	361	361
Initial guess ($\mathbf{u}_0$)	zero	zero
Data set	sine.dat	sine.dat

Table 9.1: Initial multigrid settings.

The significance of these settings has been discussed in Chapter 4, however some further points require clarification. Several settings, including the number of grid levels, the type of smoothing iteration, the coarse grid operator, the number of grid points on the fine grid and the initial guess were set arbitrarily and optimised experimentally. The relaxation parameter for SOR smoothing was also optimised experimentally for both algorithms, as is commonly done for multigrid algorithms [2]. It is stated in Brandt [16] that the smoothing factor is generally insensitive to the relaxation parameter, and its value is not significantly increased if ω is not optimal. Interestingly, the value of $\omega = 1.6$ was also obtained experimentally for SOR by Hutchinson [58] and Atlas [2].
For v-cycle, it is recommended by Brandt [17] to choose 3 presmoothing and 3 postsmoothing iterations. Nested grid, however, is not a cyclic algorithm, so the number of smoothing iterations is determined by the desired accuracy. The results in this chapter indicate that there is an advantage in having more smoothing iterations on some grids than on others.
With regard to the transfer operators, both are second order, and are standard choices for multigrid algorithms [20]. The order of these transfer operators is slightly too small to be consistent with the recommendation in equation (4.30), as the smoothing spline equations are fourth order. Preliminary investigation

showed that this did not significantly influence the results presented here.

The following sections describe the process undertaken to understand multigrid and optimise its performance. To make the analysis as transparent as possible, only the univariate case was considered and the initial data set was designed to be small and simple. This allowed the analytic solution to be calculated easily using ANUSPLIN. The analytic solution has been used extensively throughout this study to assess the accuracy of the solution estimates, estimates of the GCV, R and the signal, and to provide an optimal smoothing parameter.

For the analyses described below, the error is measured by the deviation between the analytic solution determined from ANUSPLIN and the solution estimate. Deviations from the analytic values of the GCV and the signal due to the piecewise constant discretisation were found to be negligible in comparison to other sources of error. The dominant influence on the accuracy of the these statistics for the data sets considered in this chapter was found to be stochastic error in the estimate of the trace of the influence matrix. This can be seen from the following analysis.

9.1 Performance of v-cycle and nested grid

The v-cycle and nested grid algorithms described in Chapter 4, with the settings from Table 9.1, were applied to equation (3.5) using data set sine.dat. This data set, shown in Figure 9.1, consists of 101 noisy data points randomly perturbed from a single sine curve by values from a zero mean normal variable with standard deviation 0.2. The data points are equally spaced at intervals of 3.6. The optimal smoothing parameter for this data set, obtained from ANUSPLIN, was found to be 25500. This value is large enough to cause the second term of equation (3.5) to dominate the system. Both multigrid algorithms produced accurate estimates of the analytic solution, as shown in Figures 9.2 and 9.3. The analytic solution shows good agreement with the true sine curve from which the data was obtained, demonstrating the appropriateness of the cubic smoothing spline for this data set. Note that, for the plots of the piecewise constant discretisation in the results chapters of this thesis, the centre points of each grid cell are joined by straight lines. This was deemed to be an appropriate representation of piecewise constant discretistations of smoothing splines.

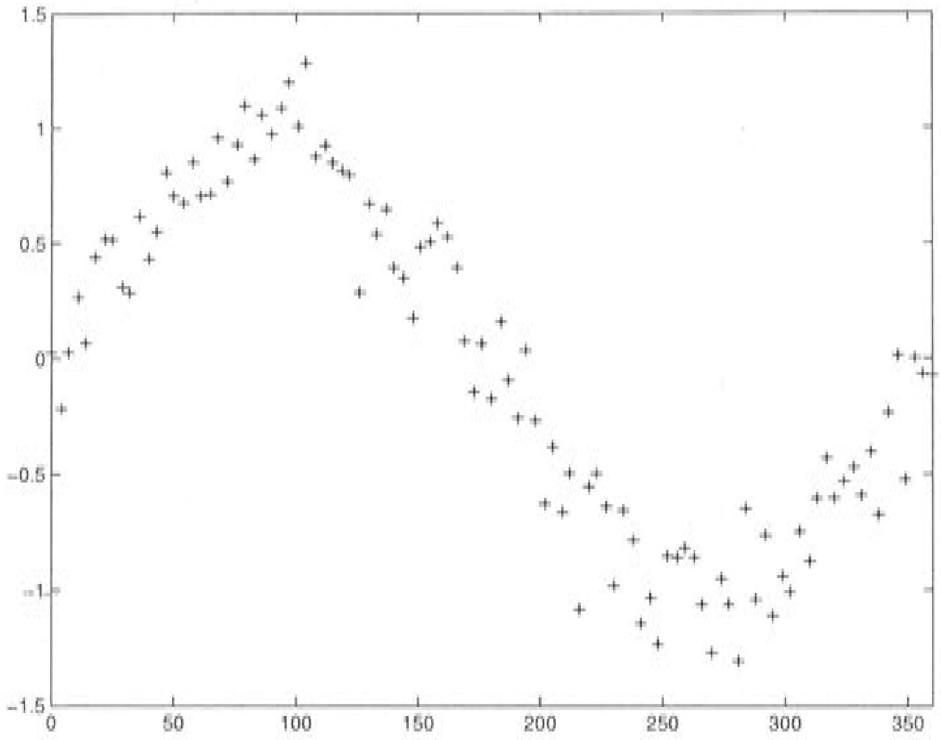


Figure 9.1: The data set sine.dat.

Accurate estimates of the GCV, R and signal corresponding to the discretised solution were also obtained, as shown in Table 9.2. These estimates were obtained from equation (8.15) using the stochastic estimate of Tr in equation (8.21). The low signal reflects the smooth nature of the underlying process, and explains the large smoothing parameter. Figure 9.4 shows how the smoothing parameter varies with the signal for this data set. If, rather than minimising the GCV, a smoothing parameter of 50 is enforced, the solution in Figure 9.5 is produced, which is clearly a much poorer reflection of the underlying process from which the data was created.

	Analytic Solution	Nested Grid	V-cycle
Signal	8.4	9.8	9.1
Generalised Cross Validation	0.034	0.035	0.034
Residual Sum of Squares	2.85	2.86	2.83

Table 9.2: A comparison of output statistics for nested grid, v-cycle and the analytic solution.

The stochastic error in the estimate of the trace of the influence matrix, $tr(A)$, was found to have a considerable influence on other solution characteristics,

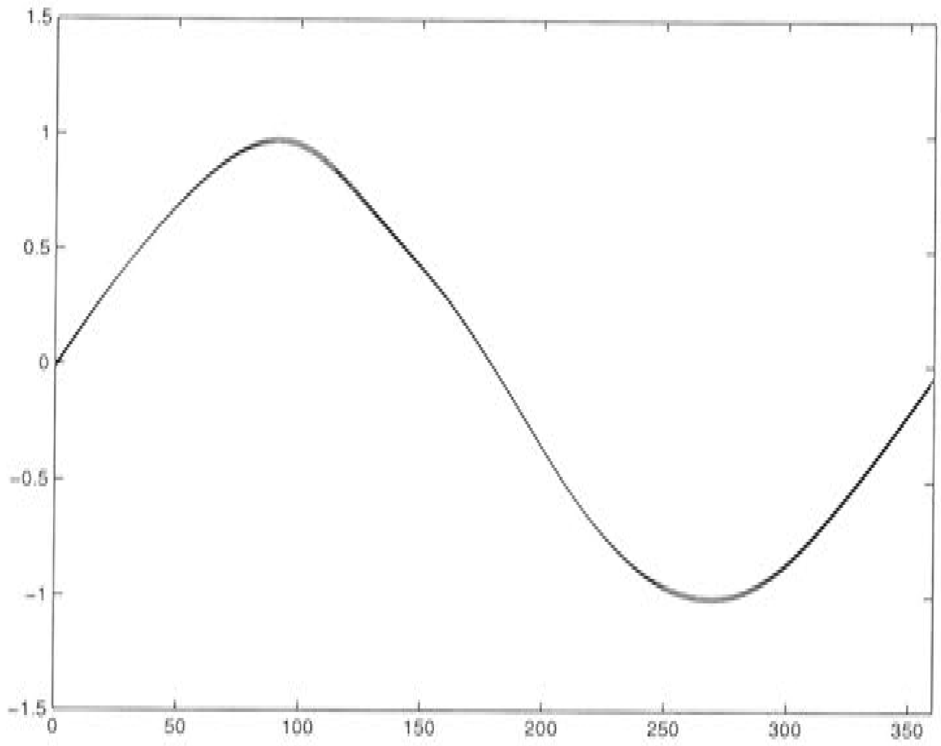


Figure 9.2: The analytic solution and the piecewise constant approximation obtained using the v-cycle algorithm.

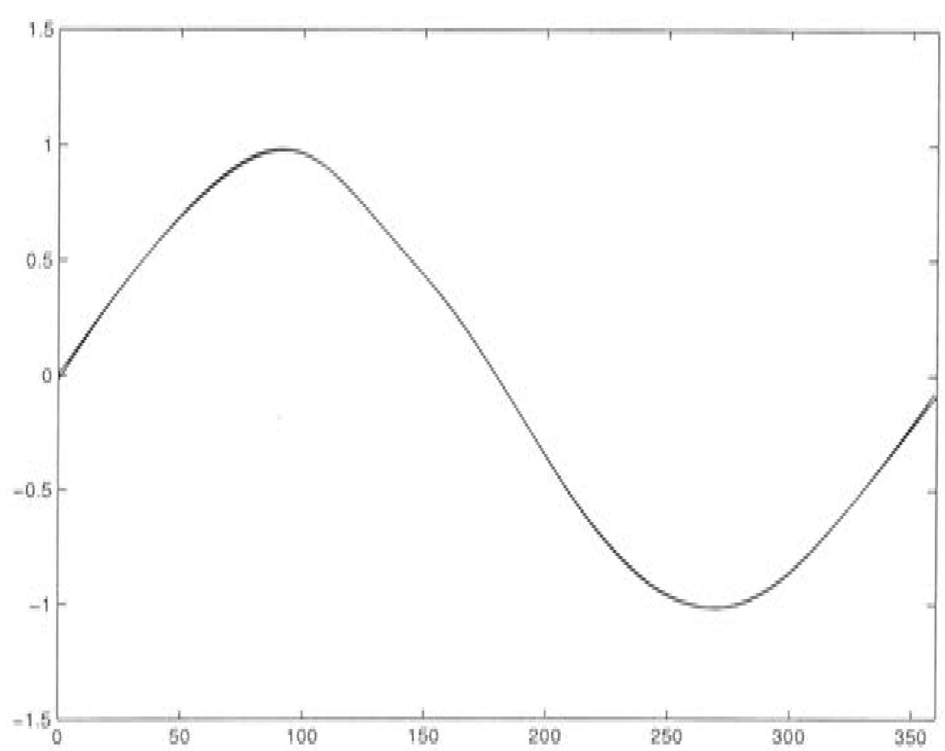


Figure 9.3: The analytic solution and the piecewise constant approximation obtained using the nested grid algorithm.

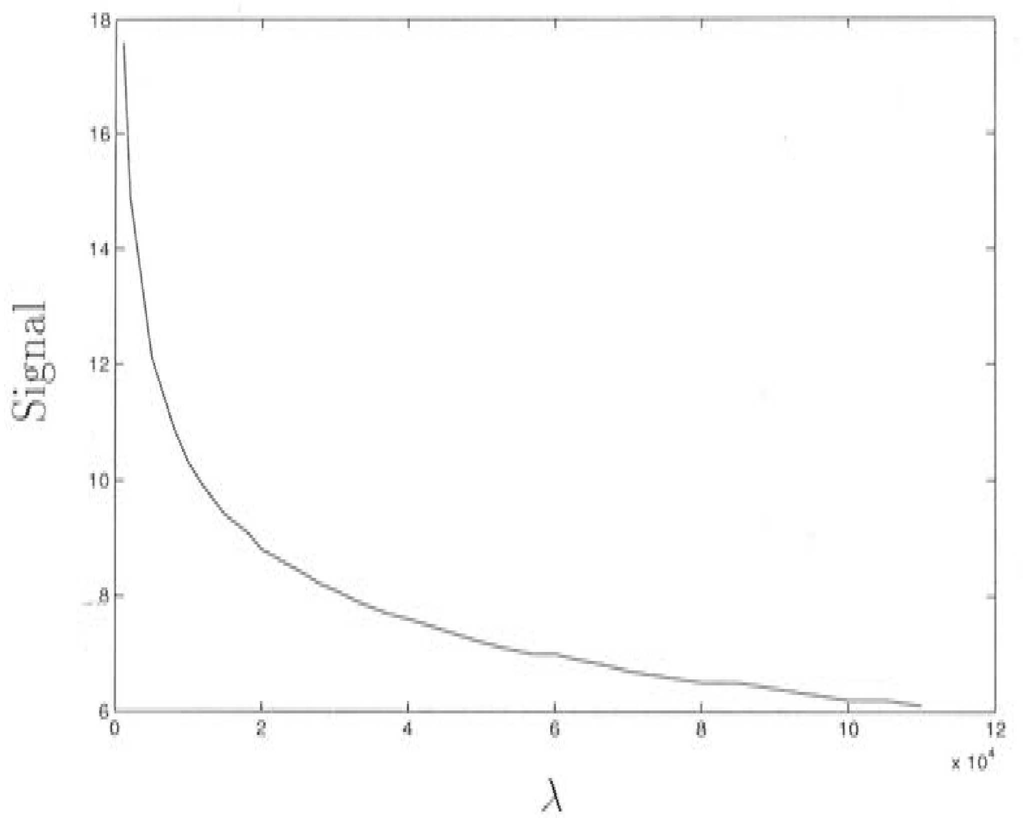


Figure 9.4: The signal versus the smoothing parameter for the data set sine.dat.

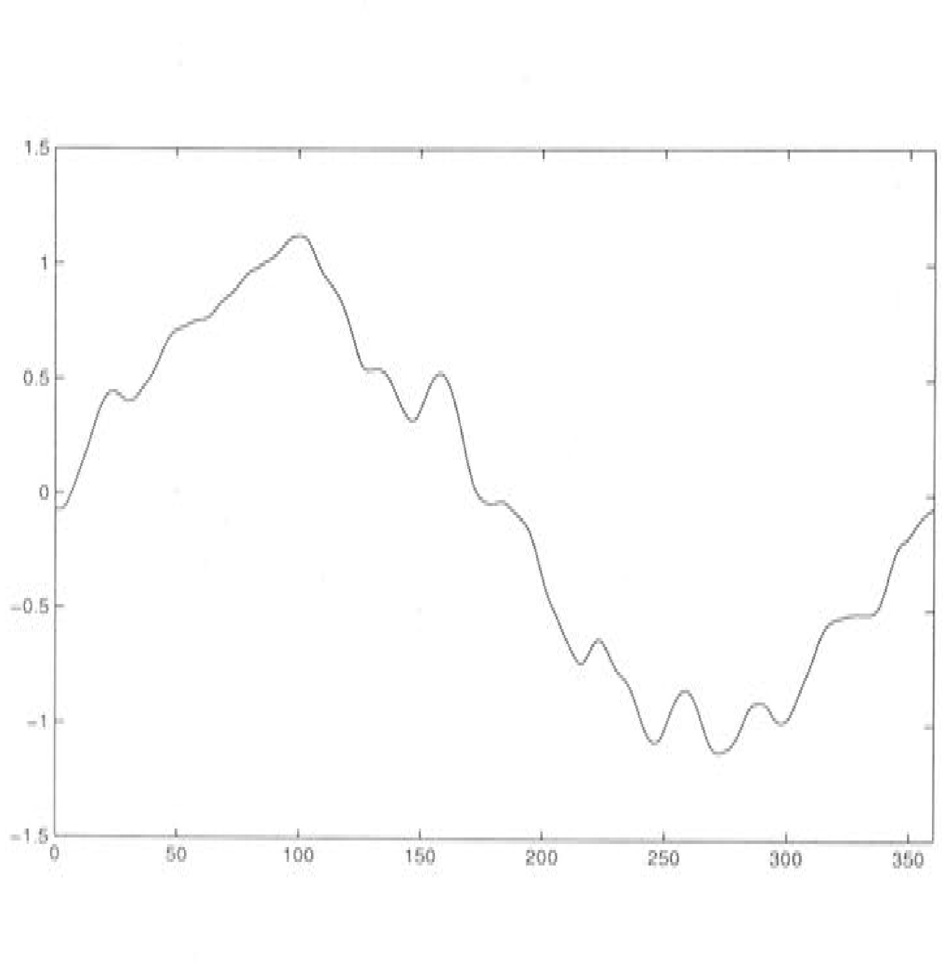


Figure 9.5: The smoothing spline solution corresponding to a smoothing parameter of $\lambda = 50$.

including the GCV. A sample of 50 spline fits using 50 different random vectors gave the results in Figure 9.6, where we now present Tr, or $tr(I-A)$, instead of $tr(A)$. The extent of the stochastic variability induced in the GCV estimates becomes especially noticeable after considering the analyses in Chapters 11 and 12, which show that the GCV is usually extremely stable.

According to formula (8.23) the relative standard error in the Tr estimate for this problem is bounded by $(2/n)^{1/2} = 0.14$, giving an absolute standard error of 12.97. The standard error of the sample in Figure 9.6 was found to be 3.08 which is significantly lower than this upper bound, demonstrating that the bound is conservative. Hutchinson [59] also presents a more accurate calculation of the bound on the standard error. This makes reasonable assumptions on the distribution of the magnitude of the eigenvalues of the influence matrix A, and involves making integral approximations for Tr and Tr^2. Using these calculations, the bound on the standard error of Tr for this case was calculated to be 2.97. This is an accurate estimate of the observed standard error. This standard error must be considered as a significant source of inaccuracy in the solution for this data set. However, for large data sets the error in the Tr estimation will be insignificant.

To compare the stochastic error with the discretisation error in the trace, the trace of $I - A$ for the piecewise constant system was calculated as

$$tr(I - P(P^TP + \frac{\lambda}{h^3}Q^TQ)^{-1}P^T) \tag{9.1}$$

This formula was derived by recalling from section 2.2.1 that $A\mathbf{z} = P\mathbf{f}$ and using the expression for $\mathbf{f}$ given by the discretised system (3.5) ie.

$$\begin{aligned}(I-A)\mathbf{z} &= \mathbf{z} - P\mathbf{f} \\ &= \mathbf{z} - P(P^TP + \frac{\lambda}{h^3}Q^TQ)^{-1}P^T\mathbf{z} \\ &= (I - P(P^TP + \frac{\lambda}{h^3}Q^TQ)^{-1}P^T)\mathbf{z} \\ \therefore tr(I-A) &= tr(I - P(P^TP + \frac{\lambda}{h^3}Q^TQ)^{-1}P^T)\end{aligned} \tag{9.2}$$

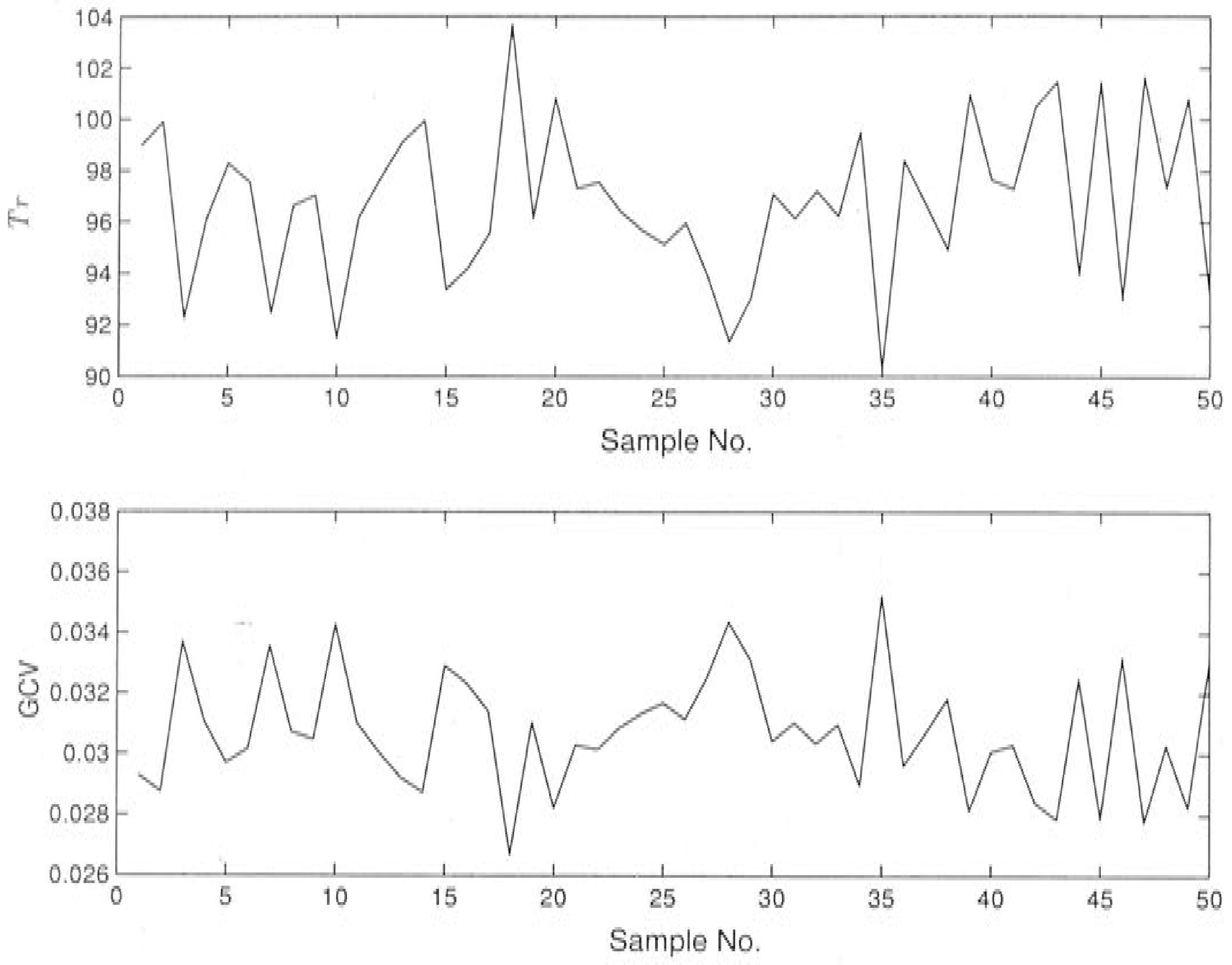


Figure 9.6: Values of $tr(I - A)$ and the GCV for different random vectors $\mathbf{t}$.

The multigrid algorithms would converge to this trace in the absence of stochastic error. Using this formula, the Tr was calculated to be 92.6. The discretisation error, calculated as the magnitude of the difference between the analytic trace obtained from ANUSPLIN, and the trace obtained from equation (9.2), is therefore zero to 3 significant figures. This is insignificant in comparison to the stochastic error.

The GCV, R and signal estimates were also evaluated after each SOR iteration for the nested grid algorithm. These results are shown in Tables 9.3, 9.4 and 9.5. The estimates on the coarse grid are initially quite variable, but they settle down and converge to 4 decimal places on the 2 finest grids. Interestingly, the Tr estimates on the coarsest grid are not much different to those on the fine grid, indicating that little fine scale structure has developed as a result of the finer discretisation. Also, the trace estimates change very little after grid number 3. This implies that further iteration of the solution on grids finer than the resolution of grid number 3 makes little change to the solution.

grid	iteration number									
number	1	2	3	4	5	6	7	8	9	10
1	95.22	95.21	95.21	95.21	95.21	95.21	95.21	95.21	95.21	95.21
2	95.20	95.20	95.21	95.20	95.20	95.21	95.21	95.21	95.21	95.21
3	95.08	95.12	95.12	95.12	95.14	95.14	95.16	95.16	95.16	95.17
4	95.27	95.24	95.19	95.08	94.93	94.81	94.78	94.75	94.74	94.71
5	95.00	95.14	94.30	93.65	93.86	94.13	94.37	94.41	94.41	94.42
6	92.27	96.10	94.84	95.10	95.12	95.04	95.14	95.03	95.13	95.05

Table 9.3: $tr(I - A)$ values after each iteration, for the nested grid algorithm.

grid	iteration number									
number	1	2	3	4	5	6	7	8	9	10
1	2.84	2.84	2.84	2.84	2.84	2.84	2.84	2.84	2.84	2.84
2	2.88	2.88	2.88	2.88	2.88	2.88	2.88	2.88	2.88	2.88
3	2.98	2.97	2.97	2.97	2.97	2.97	2.97	2.97	2.97	2.97
4	2.78	2.83	2.80	2.82	2.80	2.81	2.80	2.81	2.80	2.80
5	2.87	2.93	2.79	2.90	2.81	2.85	2.84	2.84	2.84	2.83
6	13.54	5.75	3.94	3.54	3.55	3.48	3.49	3.48	3.48	3.48

Table 9.4: R values after each iteration, for the nested grid algorithm.

grid	iteration number									
number	1	2	3	4	5	6	7	8	9	10
1	0.0316	0.0316	0.0316	0.0316	0.0316	0.0316	0.0316	0.0316	0.0316	0.0316
2	0.0320	0.0321	0.0320	0.0321	0.0320	0.0321	0.0320	0.0321	0.0320	0.0320
3	0.0333	0.0331	0.0332	0.0331	0.0332	0.0331	0.0331	0.0331	0.0331	0.0331
4	0.0310	0.0316	0.0313	0.0316	0.0314	0.0315	0.0315	0.0316	0.0315	0.0315
5	0.0321	0.0327	0.0317	0.0334	0.0323	0.0325	0.0323	0.0322	0.0322	0.0321
6	0.1606	0.0629	0.0442	0.0395	0.0396	0.0389	0.0389	0.0390	0.0389	0.0389

Table 9.5: GCV values after each iteration, for the nested grid algorithm.

It was found that v-cycle required 1724 SOR iterations to converge to the set tolerance, the level of accuracy reached by nested grid in 60 SOR iterations. The nested grid scheme was clearly a successful algorithm. V-cycle, however, reduced the error by an average of only 8.1% per cycle, a much slower performance than that of an optimal multigrid algorithm. It was therefore necessary to further investigate the behaviour of the v-cycle in order to understand its poor performance. As discussed in Chapter 4, multigrid theory does not facilitate straightforward analysis and evaluation of multigrid practice. The following sections therefore experimentally explain the observed behaviour.

9.2 Application of the multigrid principle

As described in section 4.2, the multigrid principle makes iteration highly efficient at reducing all components of the error by transferring the problem to and from grids of varying coarseness. To verify that this phenomenon is occuring in the case of the v-cycle, the error after k cycles was represented as the following expansion of the eigenvectors of the matrix $P^TP + \frac{\lambda}{h^3}Q^TQ$.

$$\mathbf{e}^{(k)} = \sum_{i=1}^{N} c_i \mathbf{w}_i \tag{9.3}$$

where $\mathbf{w}_i$ are the eigenvectors of the matrix $P^TP + \frac{\lambda}{h^3}Q^TQ$ and c_i are the coefficients. The effectiveness of v-cycle at eliminating these modes is shown in Figure 9.7. The power of multigrid is clearly seen by comparing this behaviour with the poor result given by basic SOR iteration, shown in Figure 9.8. The ineffectiveness of basic iteration implies that the dominant modes of the error are highly smooth. Figure 9.9 shows that this is indeed the case. This is a direct reflection of the smoothness of the solution. These results confirm that the multigrid procedures are effective in targeting components of the error that could not be reduced by basic iteration. The slow convergence of the SOR iteration on the fine grid could be expected considering the eigenvalues for this smoothing spline problem. The spectral radii of the smoothing iteration matrices for each grid are shown in Table 9.6. Corresponding condition

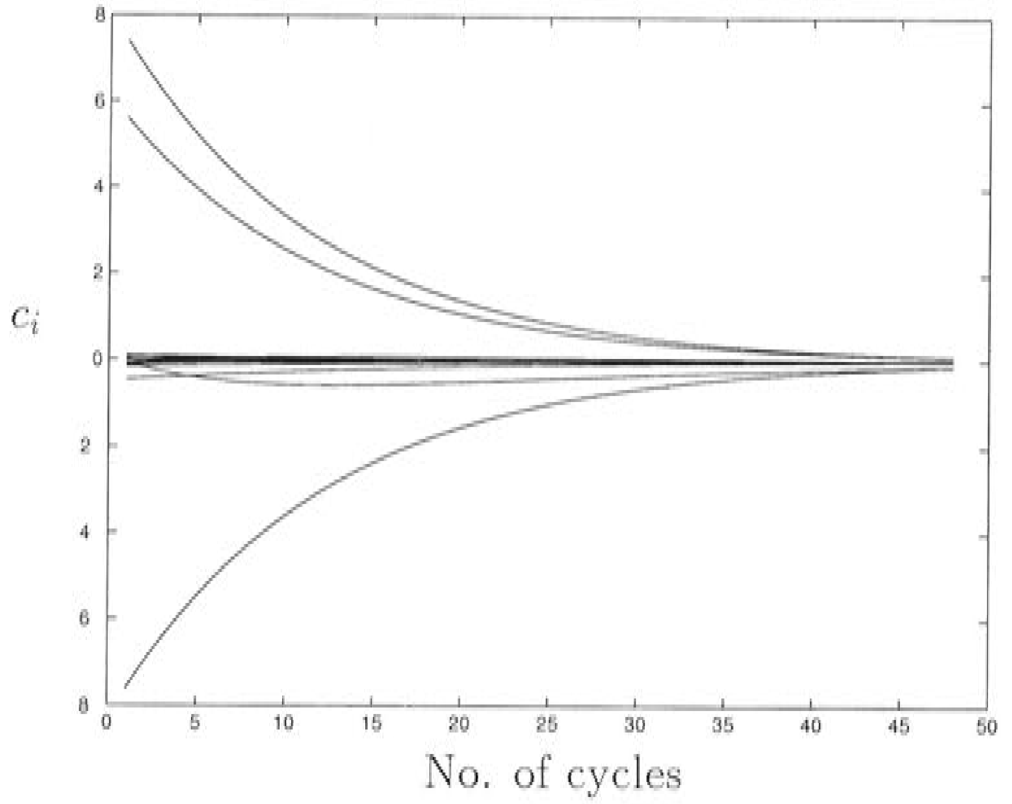


Figure 9.7: The coefficient for each error mode VS the number of cycles for the v-cycle algorithm, for the data set sine.dat.

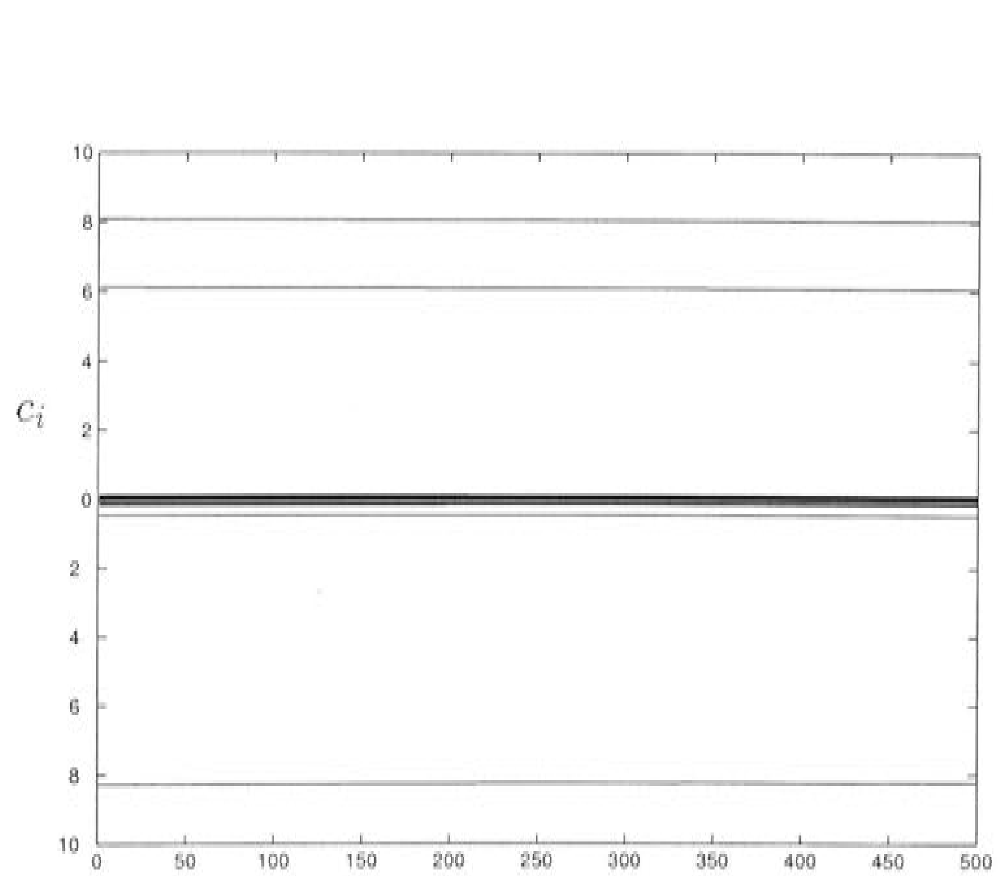


Figure 9.8: The coefficient for each error mode VS the number of SOR iterations for SOR iteration on the fine grid, for the data set sine.dat.

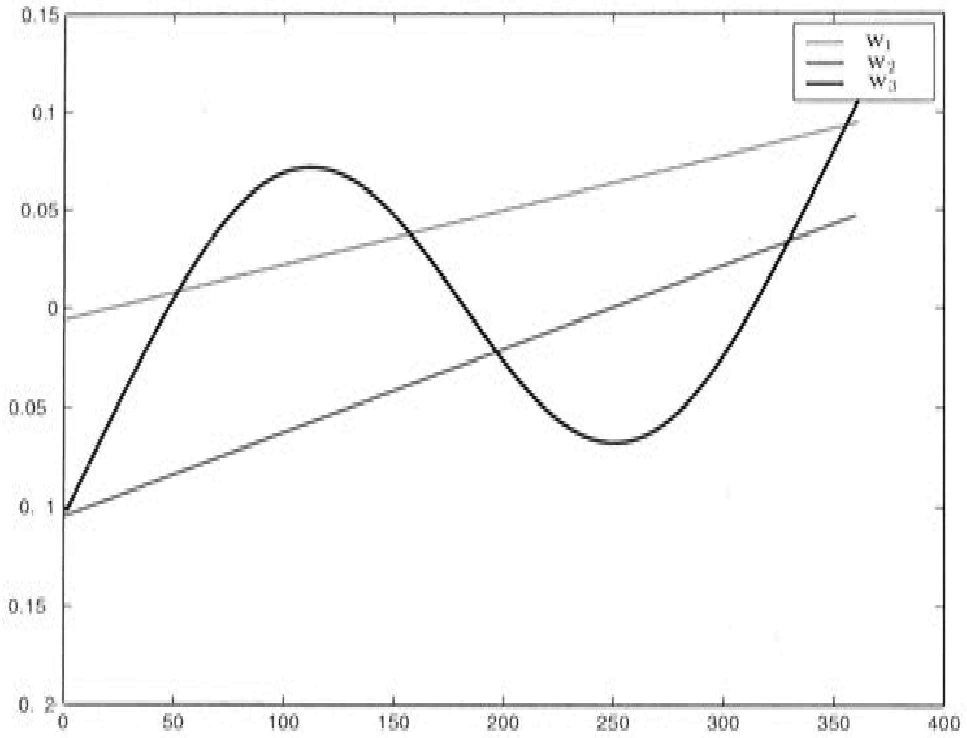


Figure 9.9: Modes of $\mathbf{e}^{(0)}$ corresponding to the 3 highest coefficients c_i for the data set sine.dat.

numbers are shown in Table 9.7. The spectral radius of 1 on the fine grid demonstrates the inability of basic iteration to eliminate smooth modes. The coarser the grid, the more efficient a basic iterative method will be.

For multigrid algorithms, the smoothing factor is used as a performance measure, as discussed in Chapter 4. Here it was suggested that fast iterative methods do not necessarily make good smoothers, and that the weighted Jacobi iterative method may be a better smoother than SOR. It was found that this was not the case for the smoothing spline problem. When weighted Jacobi was used as the smoother in the v-cycle algorithm it required 4125 iterations to converge to the tolerance, more than twice as slow as SOR. When weighted Jacobi was used in the nested grid algorithm, 300 iterations were required for convergence to the same tolerance as that achieved in 60 iterations using SOR.

The poor conditioning of this system on the fine grid is a result of the high smoothing parameter, which causes the rank deficient matrix Q^TQ to dominate the system. Conditioning improved significantly on coarser grids, as a result of increasing h. However, in the case of the v-cycle algorithm, the data are only accessed on the fine grid. An initial hypothesis was therefore that, for v-cycle, iterations on coarser grids were ineffective because they relied on information from the fine grid, were relaxation is very slow to yield accurate approximations.

Grid number	Spectral radius of SOR iteration matrix
1	1.000
2	0.998
3	0.996
4	0.974
5	0.889
6	0.736

Table 9.6: Spectral radius of SOR iteration matrix for the data set sine.dat.

Grid number	Condition number
1	1 486 000
2	91 745
3	6321
4	796
5	99.71
6	56.22

Table 9.7: Condition number of the matrix $P^TP + \frac{\lambda}{h^3}Q^TQ$ for the data set sine.dat.

Equation (3.5) can be rearranged to give the following expression for the fitted values

$$f_g = \frac{\sum_{g^{th}cell} z_g - \frac{\lambda}{h^3}c_g}{m_g + \frac{\lambda}{h^3}a_g} \tag{9.4}$$

where f_g is the value of $\mathbf{f}$ in the g^{th} grid cell, m_g are the diagonal elements of the diagonal matrix P^TP, a_g are the diagonal elements of the matrix Q^TQ and c_g are the sums of the contributions from the off-diagonal elements of the g^{th} row of Q^TQ acting on the vector $\mathbf{f}$ ie. $f_{g-2} - 4f_{g-1} - 4f_{g+1} + f_{g+2}$.

As h becomes large and the number of data points per grid cell increases, the function values approach the local averages of the data points in each grid cell. The similarity is shown in Figure 9.10. Thus on the coarse grid the solution is contained almost entirely in the data points, and little smoothing is required, so the system is well conditioned. The nested grid algorithm quickly attains this solution on the first grid, and has a good solution estimate to pass down to finer grids. Figures 9.11 - 9.16 show that, in the case of nested grid, little contribution is required from the fine grids as the solution for this data set is estimated accurately by iteration on coarser grids. This progression from coarse to fine grids demonstrates that the grids chosen in this analysis are unnecessarily fine, as the solution for this data set can be represented with far fewer grid points. Clearly there has been little alteration in the solution after grid number 4. The inclusion of unnecessarily fine grids in the multigrid algorithms has caused a deterioration in their performance, particularly in the case of the v-cycle algorithm.

In order to further understand how poor conditioning affects the v-cycle algorithm, and verify that the transfer procedures were working correctly, the algorithm was modified to solve the equations directly on the coarsest grid. This is a common version of the v-cycle algorithm [115]. The number of grids was set to 2, and increased to 4 and then 6 grids. The results are shown in Table 9.8. Convergence was fast for the two-grid algorithm, and deteriorated as the number of grids was increased. When 6 grids were used the convergence rate was no improvement on that obtained when the equations were not solved directly.

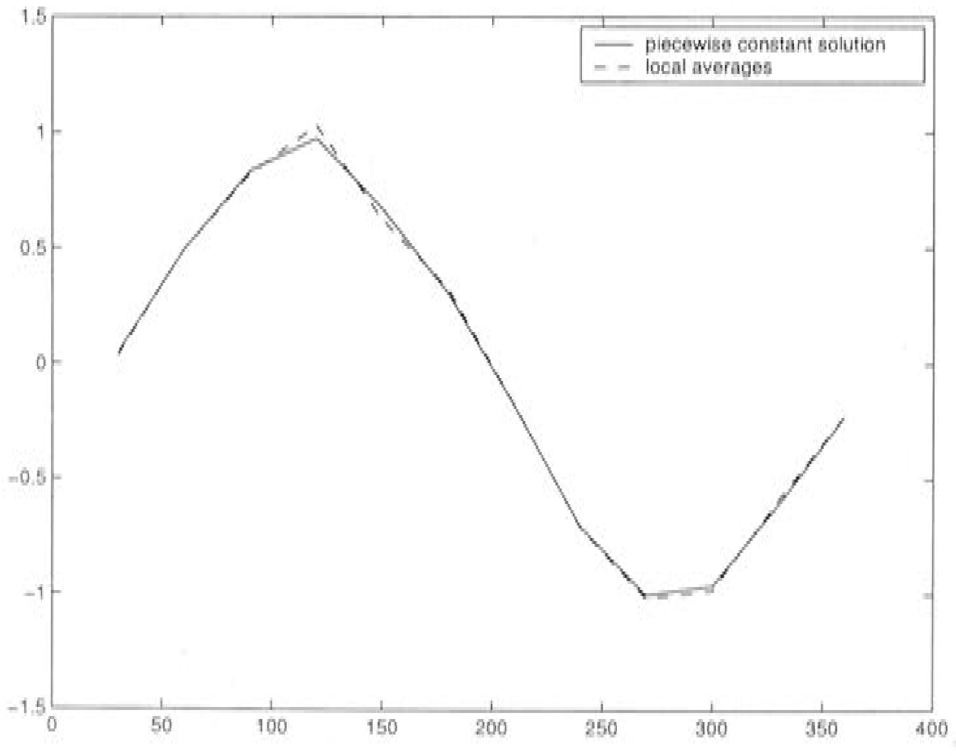


Figure 9.10: The piecewise constant solution on the coarsest grid and the local averages of the data points in each grid cell, for the data set sine.dat.

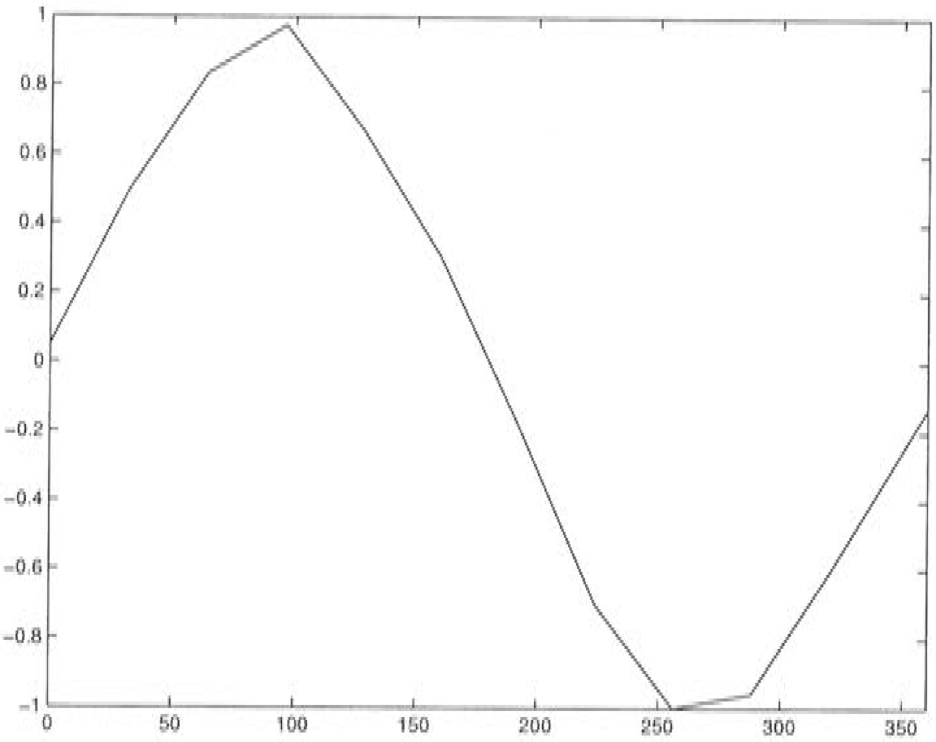


Figure 9.11: The piecewise constant solution on grid no. 6.

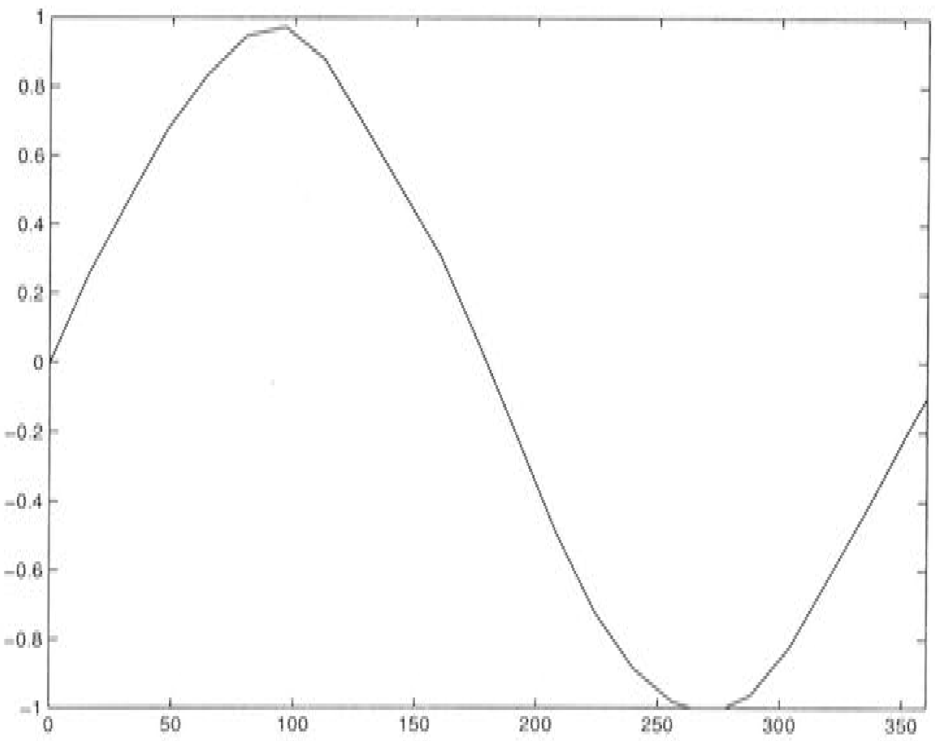


Figure 9.12: The piecewise constant solution on grid no. 5.

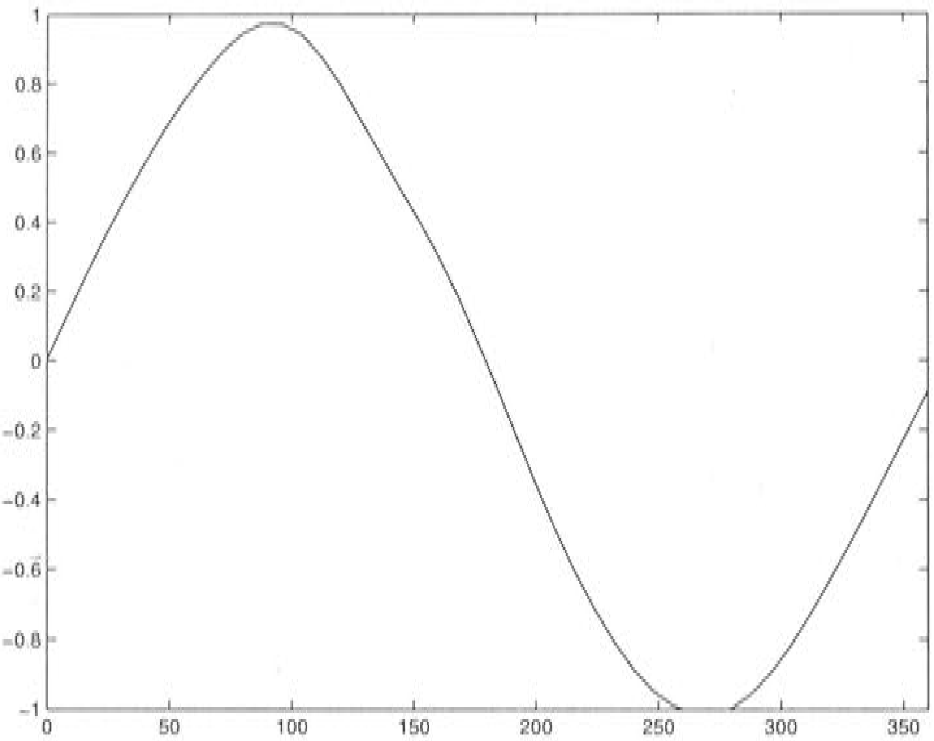


Figure 9.13: The piecewise constant solution on grid no. 4.

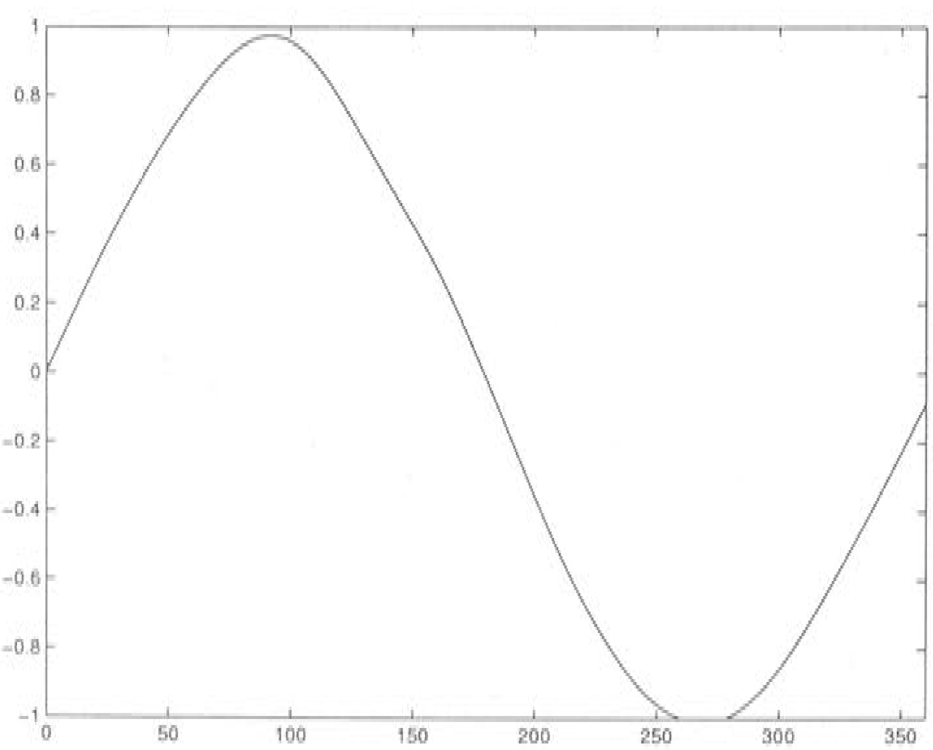


Figure 9.14: The piecewise constant solution on grid no. 3.

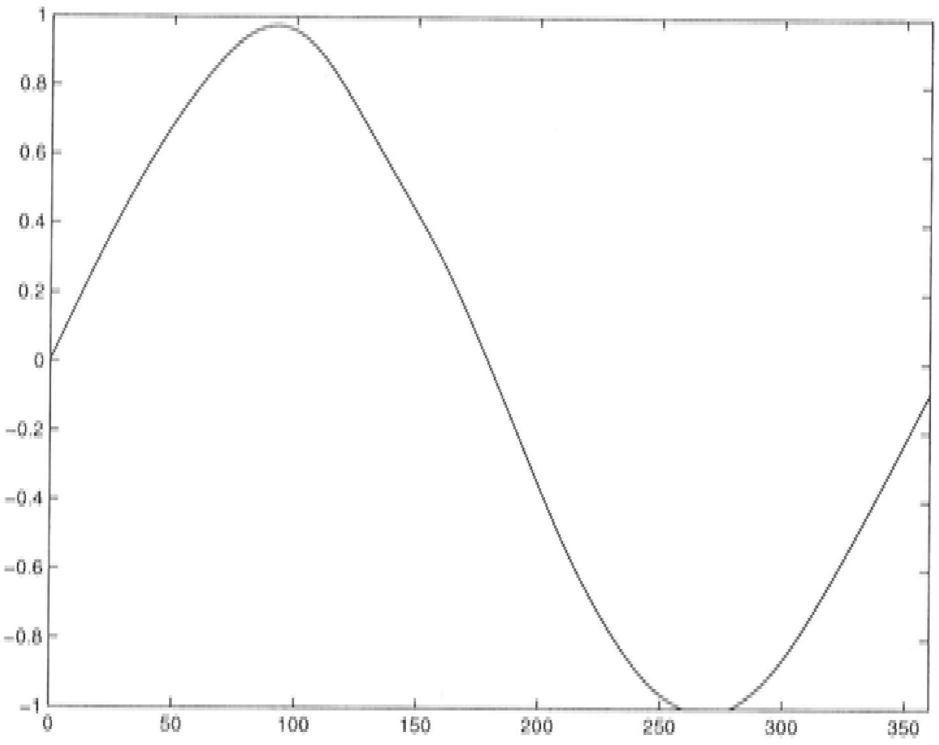


Figure 9.15: The piecewise constant solution on grid no. 2.

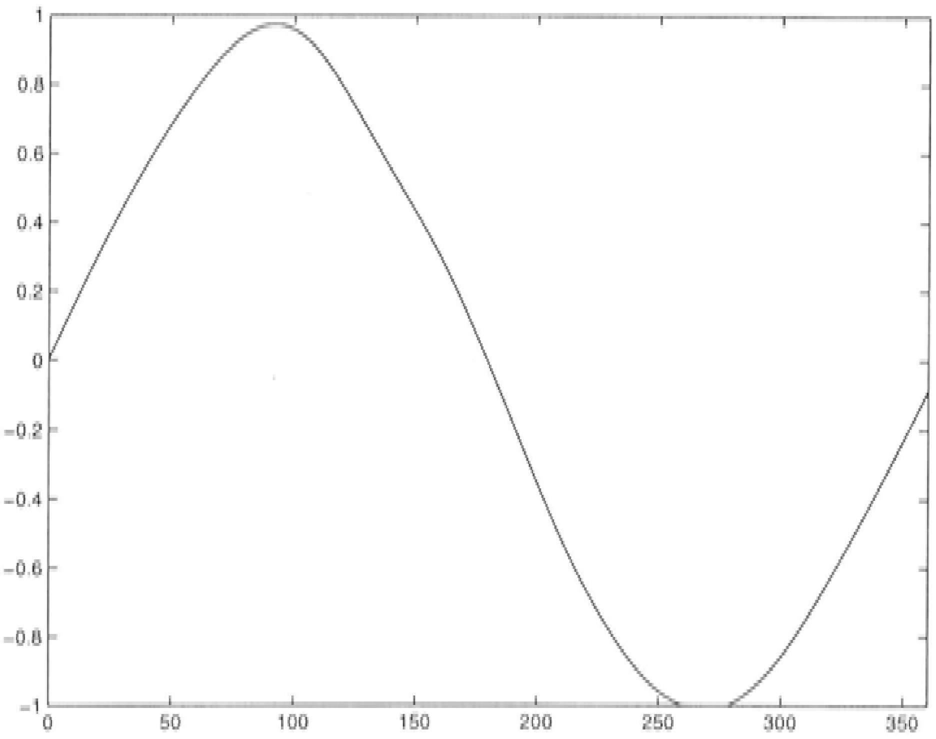


Figure 9.16: The piecewise constant solution on grid no. 1.

These results are not surprising, since direct solution on the second grid gives a very accurate initial guess on the fine grid. There is clearly an efficiency gradient in the SOR process, meaning that if the solution is not solved directly on grids close to the final grid, high accuracy cannot be achieved because the SOR process becomes more inefficient as the grids get finer. The fast performance of the v-cycle for schemes with fewer grid levels implies that the v-cycle procedure, particularly the transfer processes, are functioning properly.

No. of grid levels	SOR iterations required to reach the tolerance
2	231
3	627
6	1853

Table 9.8: Direct solution on coarsest grid, with different numbers of grid levels in the v-cycle algorithm.

9.3 Different data sets

Given that many of aspects of multigrid performance observed above are a direct result of the nature of the data set, it was important to investigate the performance of both algorithms on other data sets. Considering the above results, it would be expected that a larger data set would lessen the impact of the term Q^TQ and improve the conditioning of the system. This should result in faster convergence of the v-cycle. A data set with the same properties as sine.dat was constructed, but 360 points were generated instead of 101. This data set, 360.dat, is shown in Figure 9.17.

Using ANUSPLIN, the smoothing parameter was calculated to be 110 000. For this data set, the v-cycle algorithm converged to the tolerance in 2211 SOR iterations, slower than for sine.dat. This result is shown in Figure 9.18. The slow convergence indicates that conditioning has not improved by increasing the number of data points. However comparison is complicated by the minimum GCV smoothing parameter, which increased significantly with the addition of more data points even though the underlying process from which the data were generated did not change. When the analysis for 360dat.mat

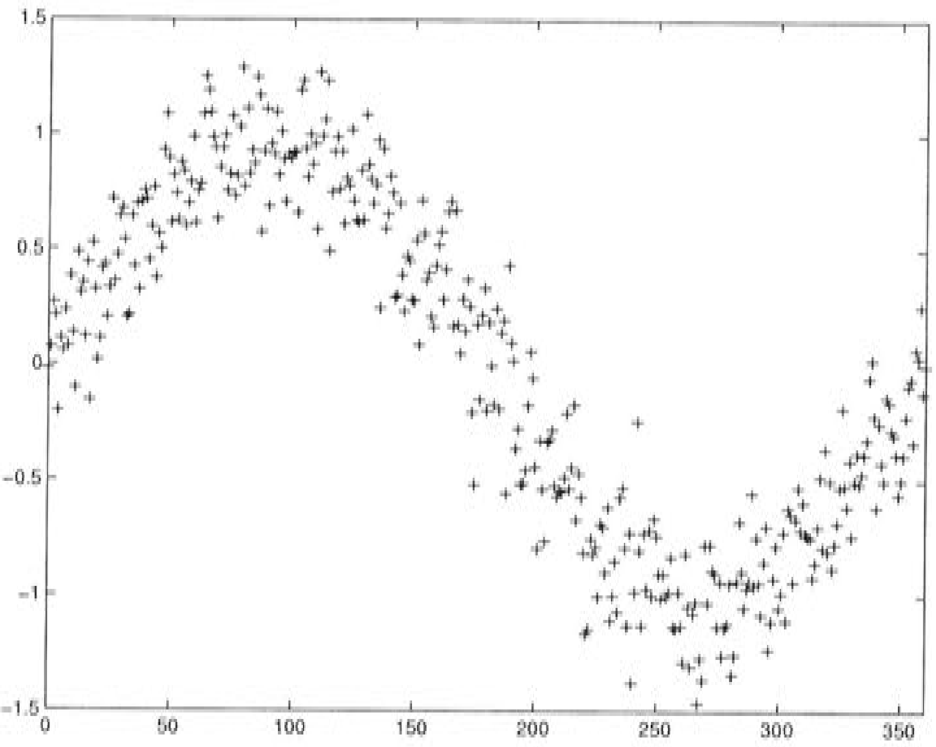


Figure 9.17: The data set 360.dat.

was rerun with the smoothing parameter set to 25500, v-cycle converged in 1254 iterations, faster than for sine.dat. This was the result expected.
The condition number of the system should also be reduced by increasing the fine scale structure of the data generation process, which should lower the smoothing parameter. This would be expected to improve the convergence of the v-cycle, but may require more iterations on the finer grids of the nested grid algorithms as these grids are important in developing the fine scale structure. In order to test these assertions, the 101 point data set bumpy.dat, shown in Figure 9.19, was constructed from the function

$$\sin 2\pi x/180 + 0.5 \cos 4\pi x/180 \tag{9.5}$$

Random noise of 0.2 was again added. The smoothing parameter was calculated using ANUSPLIN to be 733, reflecting the fine scale patterns in the data. Comparisons of the piecewise constant approximation with the analytic solution is shown in Figure 9.20. The v-cycle algorithm converged in 666 SOR iterations, over 50% faster than for sine.dat. The nested grid algorithm required 210 SOR iterations, approximately 3 times as many as required for sine.dat. Most of these iterations were performed on finer grids, further reducing the computational speed of the algorithm. In fact, the two coarsest grids

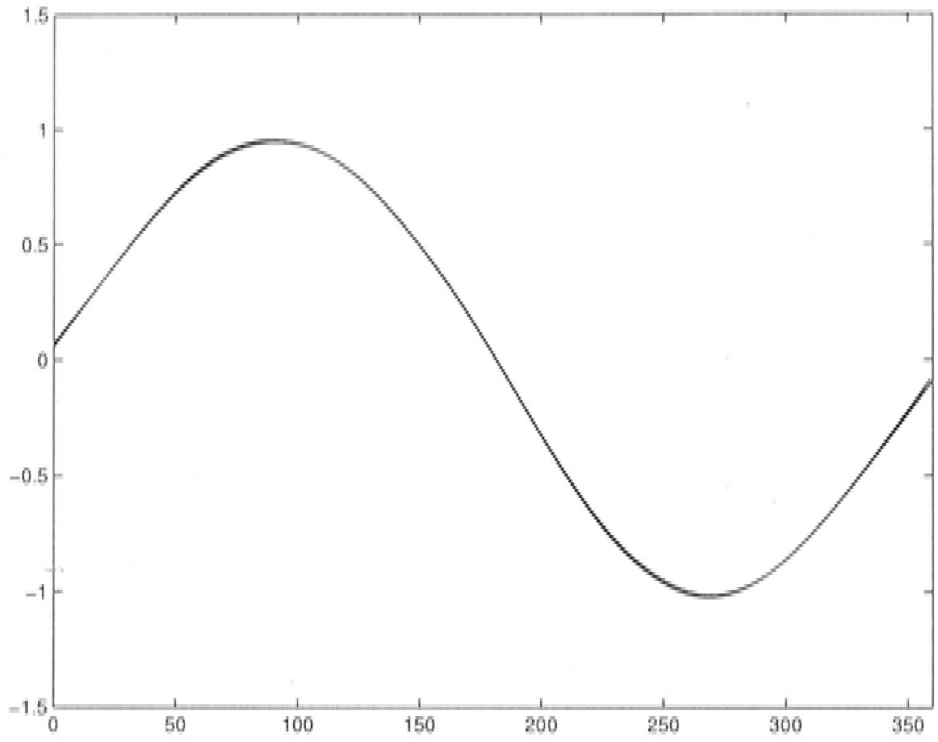


Figure 9.18: The analytic solution and the piecewise constant approximation obtained using the v-cycle algorithm, for the data set 360.dat.

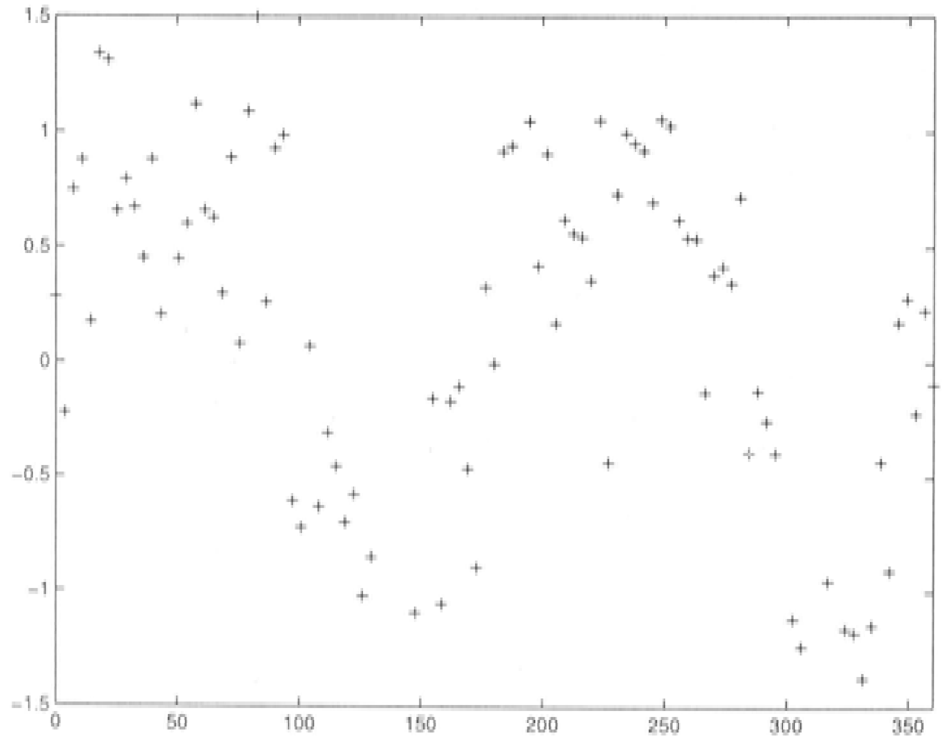


Figure 9.19: The data set bumpy.dat.

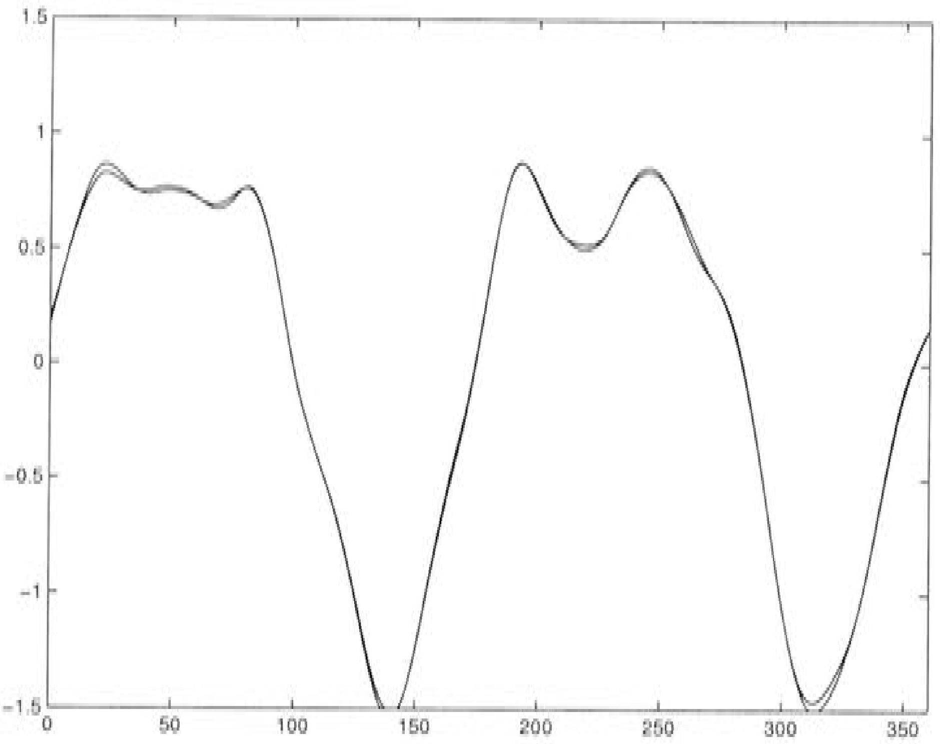


Figure 9.20: The analytic solution and the piecewise constant approximation obtained using the v-cycle algorithm, for the data set bumpy.dat.

had only 5 iterations per grid where as all other grids had 50 iterations. These results further emphasise the importance of the scale of the data generation process. To give an optimal performance the multigrid algorithms have to be tailored to suit the scale of the solution, so the dominant modes of the error can be targeted effectively on grids of suitable coarseness.

The results presented thus far show that the above numerical methods for estimating piecewise constant thin plate smoothing splines collapse for extremely large values fo the smoothing parameter. In the case of data that represents zero mean random noise or a global linear trend, ANUSPLIN reduces to linear regression and estimates the smoothing parameter to be a number approaching infinity. Clearly this corresponds to the matrix $P^TP + \frac{\lambda}{h^3}Q^TQ$ having an infinite condition number, rendering numerical solution by the above methods impossible.

9.4 Conclusion

The analyses in this chapter demonstrate the value of empirical testing in the design of a multigrid algorithm for a given system of equations. The performance of the algorithm clearly depends on the characteristics of the problem to be solved. In this study, the scale of the solution was found to be a major issue influencing performance, in that representation of a smooth problem on fine grids lead to systems that were poorly conditioned. This problem was overcome by choosing the nested grid algorithm, which incorporated a grid schedule that began on a coarse grid where the system was well conditioned. The nested grid results showed that there was little difference between solution representations on fine grids for the smooth data set. It was therefore only necessary to solving the smoothing spline problem on coarse grids for this data set. The incorporation of unnecessarily fine grids slowed down the solution process considerably.

Stochastic error in the estimate of the trace of the influence matrix was found to be the dominant source of error in the approximation of the signal and the GCV corresponding to the discretised solution. Discretisation error in these statistics was found to be negligible in comparison to the stochastic error. However, the magnitude of the stochastic error greatly reduces as the size of the data set increases [59]. This source of error should therefore not significantly influence the performance of the algorithm developed during this study when it is applied to the large data sets for which it was intended. The findings from the procedures reported in this chapter were used as a platform on which to base further investigation of the algorithms developed during this study.

A test of the v-cycle algorithm

The copy of this thesis originally submitted for examination had a sign error in the pseudo-code for the v-cycle algorithm, given in Chapter 3. This sign error did not occur in the code developed to implement this algorithm. The table below shows the performance of the v-cycle algorithm for systems of the form $y'' + ay = z$. This system is identical to the univariate smoothing spline system with one data point in each grid cell, except that the boundary conditions, smoothing parameter $1/a$ and right hand side z have been varied. The results show that the performance of the v-cycle algorithm is slowed down by the conditioning of the system, not by errors in the code.

problem number	model problem	a	boundary conditions	ω	smoothing method	average error reduction rate (%)
1	$y'' + ay = sinx$	0	dirichlet	1.6	SOR	34.75
2	$y'' + ay = sinx$	2	dirichlet	1.6	SOR	23.14
3	$y'' + ay = sinx$	2	neumann	1.6	SOR	74.87
4	$y'' + ay = sinx$	0.1	neumann	1.6	SOR	99.99
5	$y'' + ay = sinx$	0	dirichlet	0.5	Jacobi	80.5
6	$y'' + ay = sinx$	2	dirichlet	0.5	Jacobi	73.5
7	$y'' + ay = sinx$	2	neumann	0.5	Jacobi	98.53
8	$y'' + ay = sinx$	0.1	neumann	0.5	Jacobi	100
9	$y'' + ay = sin254\pi x$	2	neumann	1.6	SOR	99.96

Table 1: Results for each model problem, with different v-cycle specifications. ω is the relaxation parameter used in the SOR and (weighted) Jacobi relaxation methods. The average error reduction rate is the amount by which the difference between the solution estimate and the analytic solution is reduced with each cycle.

Chapter 10

Minimising GCV for the univariate piecewise constant smoothing spline system

10.1 Performance of the OPTRSS algorithm

After determining that nested grid was the appropriate choice of multigrid algorithm for use in this analysis, the OPTRSS algorithm, described in section 8.1, was implemented. As discussed in Chapter 8, the OPTRSS algorithm was constructed as a preliminary step towards minimising GCV. The objective was to test the underlying adaptive iterative framework. This was intended to be the basis of the MINGCV algorithm. These processes were carried out in the early stages of the univariate analysis and therefore used only the piecewise constant discretisation of the univariate smoothing spline problem, given in equation (3.5).

As with the analyses in Chapter 9, there are a number of initial settings for the OPTRSS algorithm that were prescribed to intuitively sensible values, and then optimised experimentally during the course of the investigation. The initial settings in Table 9.1 were again used for the nested grid framework. As discussed in section 8.1, the convergence criteria in the OPTRSS algorithm, determining the resolution of the final grid and the number of updates performed on each grid, were not used during this testing phase. Instead, the number of grids, the final grid resolution, and the number of updates per grid were prescribed in the same manner as for the analysis in the previous chapter. The initial value of θ was set to the minimum GCV value obtained from ANUSPLIN. The θ estimate then was updated every 3 SOR iterations using the Newton method in equation (8.1), and 10 updates were performed on a given grid before prolongation, giving a total of 30 SOR iterations per grid. Thus for the purposes of this chapter, the OPTRSS algorithm from section 8.1 is simplified to

```
for $l = 6$ to 2
    for $q = 0$ to 9
        $\mathbf{u}_l(\theta_q) = S_l{}^3(\mathbf{u}_l(\theta_q), \mathbf{z}_l)$
        $\mathbf{u}'_l(\theta_q) = S_l{}^3(\mathbf{u}'_l(\theta_q), \mathbf{v}_l)$
        $\delta\theta = (S - R)/\frac{dR}{d\theta}$
        $\theta_{q+1} = \theta_q + \delta\theta$
    end
    $\mathbf{u}_{l-1} = T_l\mathbf{u}_l$
    $\mathbf{u}'_{l-1} = T_l\mathbf{u}'_l$
end
$l = 1$
for $q = 0$ to 9
    $\mathbf{u}_l = S_l{}^3(\mathbf{u}_l, \mathbf{z}_l)$
    $\mathbf{u}'_l = S_l{}^3(\mathbf{u}'_l, \mathbf{v}_l)$
end
```

For the initial implementation of the OPTRSS algorithm, a different value of the prescribed residual sum of squares S was set for each grid. The S values were set by fixing the smoothing parameter to the minimum GCV value obtained from ANUSPLIN and solving the piecewise constant system on each grid. The values of S on each grid are therefore the values of the residual sum of squares corresponding to the solution to the piecewise constant discretised system on each grid, with a fixed smoothing parameter. Prescribing the S values in this way is the simplest test of the iterative framework, because the solution on grid $l+1$ is an accurate approximation to the solution on grid l. The prescribed S values on each grid are shown in Table 10.1, with corresponding stochastic estimates of the signals and GCV values. The smoothing parameter is fixed on all grids to the analytic minimum GCV value of $\theta = 10.14$. There is considerable variation in the statistics in Table 10.1 between the grids, demonstrating that the fundamental characteristics of the solution change when it is represented on different grids, even though the smoothing parameter is the same.

It might be expected that the residual sum of squares would decrease and the signal would increase as the grids get finer and the solution develops fine scale structure. However, after the transition from coarsest grid, there are no consistent trends in the solution characteristics. The statistics imply that representation on finer grids has not induced strong changes in the smoothing spline solution for this data set.

grid no.	S	GCV	signal
1	2.8208	0.0313	5.5915
2	2.8577	0.0317	5.5842
3	2.9489	0.0327	5.6123
4	2.7817	0.0313	6.2487
5	2.8376	0.0322	6.7007
6	3.4815	0.0389	5.9134

Table 10.1: Prescribed residuals on each grid level, with GCV and signal values.

The results of running the OPTRSS algorithm with the prescribed S values in Table 10.1 are shown in section A.1 with summary results in Table 10.2. In the tables in this chapter, the algorithm is said to have converged on a particular grid if the θ updates converge to 2 decimal places. To fully appreciate the convergence behaviour of the algorithm it is necessary to examine the more detailed results in the appendices. On grid number 6, the θ estimate settled down and actually agreed to almost 4 decimal places after 10 updates. As the grids got finer the θ estimate became less likely to converge. On the 3 finest grids the estimate shows no sign of converging, although it changes very little with each update. The superior performance of the algorithm on the coarse grids compared to the finer grids would be expected, given the results of the analyses in Chapter 9 which showed that the conditioning of the smoothing spline system deteriorates as the grid resolution is refined.

Grid no.	No. of	Converged	Converged	Converged
	updates	θ value	$S - R$ value	$dR/d\theta$ value
6	6	10.14	0.000	0.068
5	8	10.14	0.000	0.14
4		non	convergence	
3		non	convergence	
2		non	convergence	
1		non	convergence	

Table 10.2: Results generated by the OPTRSS algorithm for the data set sine.dat.

A clearer picture of the behaviour of the θ estimate can be shown by plotting its behaviour throughout the solution process, as seen in Figures 10.1- 10.6. On the coarsest grid, convergence is almost direct. On grid number 5, θ oscillates towards the optimal value. This means that the solution estimate, and estimates of $d\mathbf{u}/d\theta$ and $dR/d\theta$ are not reacting to changes in θ as quickly as on the coarse grid. As a result, the process becomes poorly synchronised. The estimates of the solution and its derivatives reflect old θ updates more than recent ones. The θ estimate 'overshoots' the optimal value that would be obtained from Newton's method if the derivative estimates were accurate for that θ value. The finer the grid, the more of a problem this becomes.

Clearly on the finest grid solution estimates are so slow to reflect changing θ that its value keeps increasing, and will probably continue to do so until θ is so large that it will be impossible to recover by updating on any grid. This is demonstrated by looking at the $S - R$ values in Table A.1.6, which shows that the residual sum of squares on the fine grid is very slow to increase to the prescribed value. Techniques to overcome this problem are discussed in following chapters.

Further insights were gained by varying some of the initial settings chosen. Those factors that were found to have a considerable influence on the behaviour of the algorithm are discussed below.

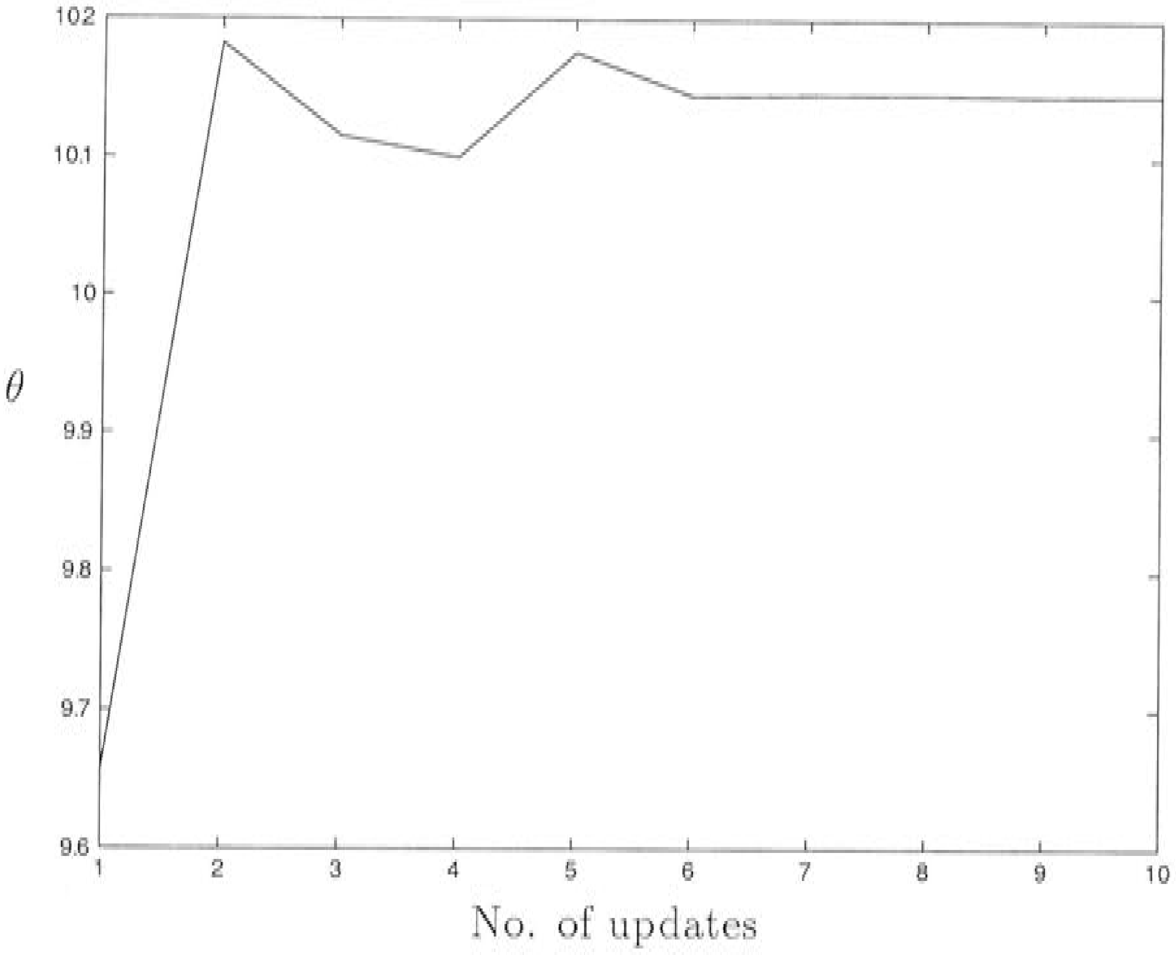


Figure 10.1: Successive smoothing parameter updates on grid no. 6 generated by the OPTRSS algorithm, for the data set sine.dat.

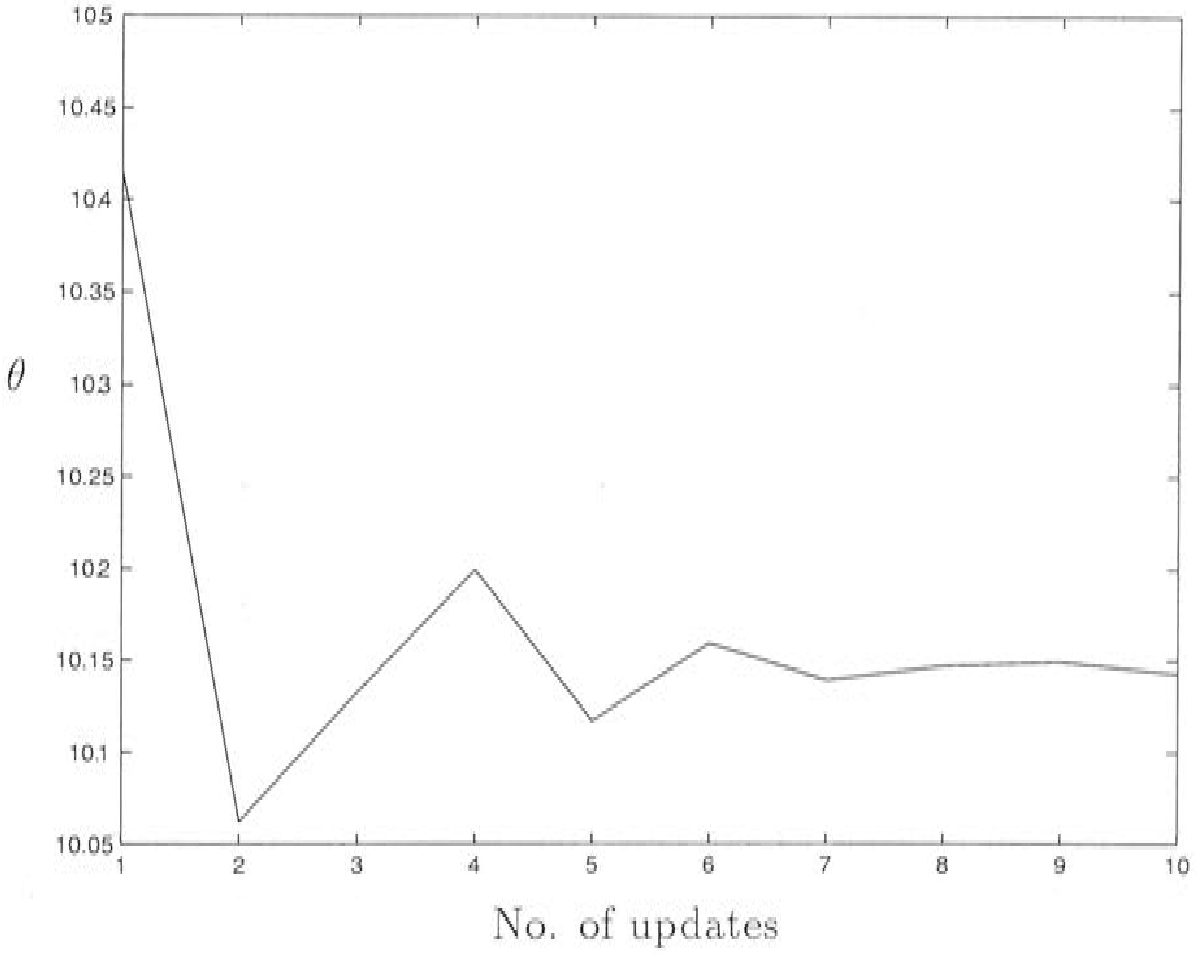


Figure 10.2: Successive smoothing parameter updates on grid no. 5 generated by the OPTRSS algorithm, for the data set sine.dat.

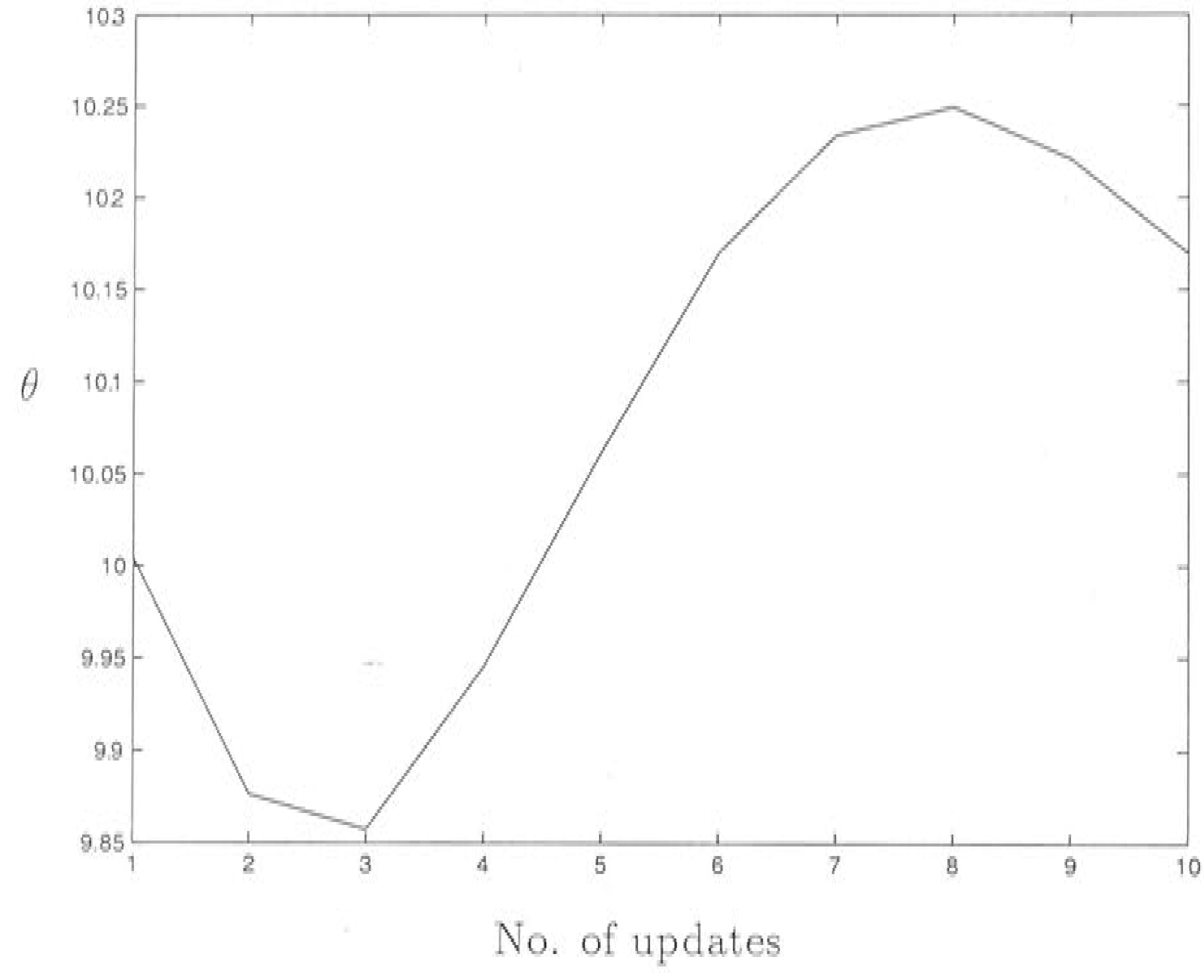


Figure 10.3: Successive smoothing parameter updates on grid no. 4 generated by the OPTRSS algorithm, for the data set sine.dat.

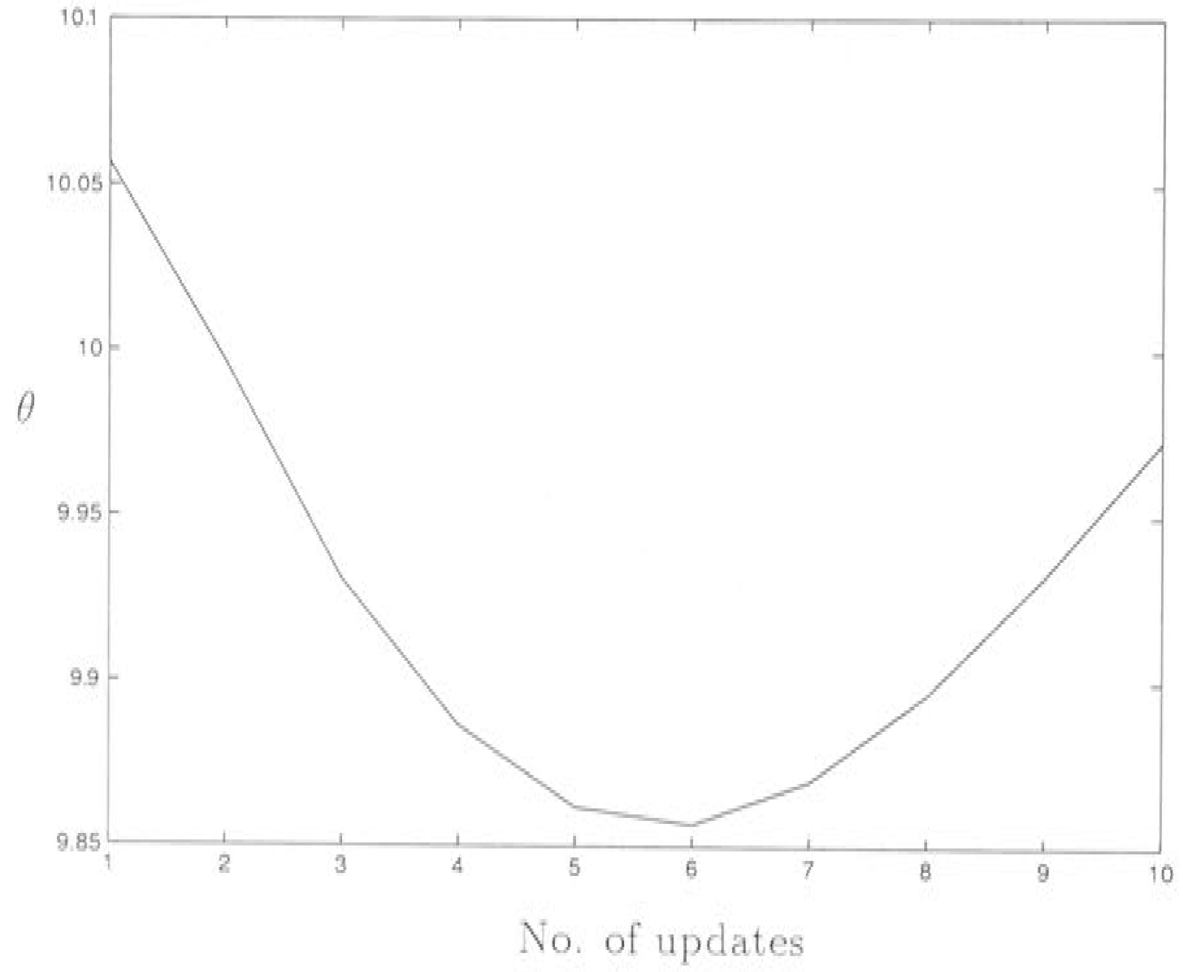


Figure 10.4: Successive smoothing parameter updates on grid no. 3 generated by the OPTRSS algorithm, for the data set sine.dat.

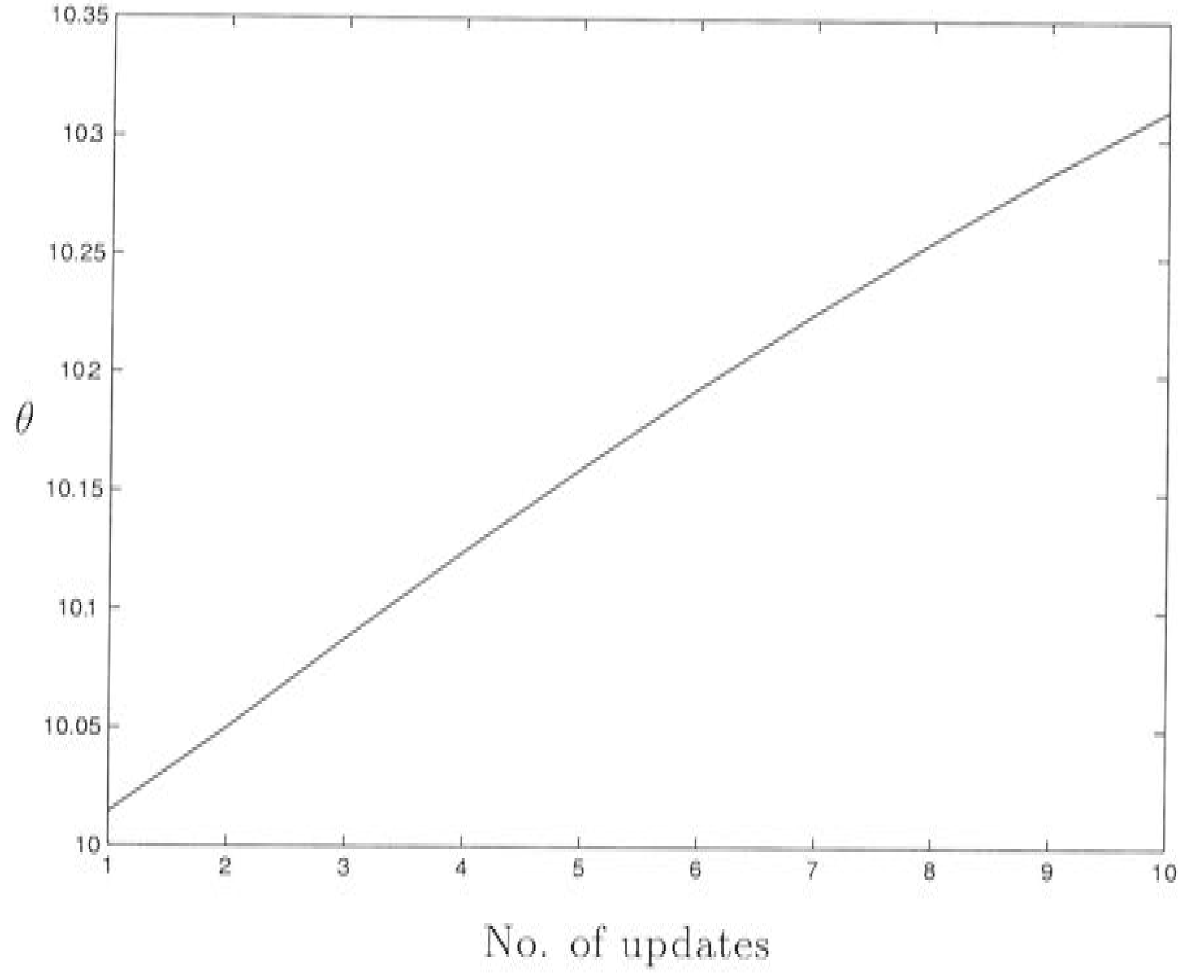


Figure 10.5: Successive smoothing parameter updates on grid no. 2 generated by the OPTRSS algorithm, for the data set sine.dat.

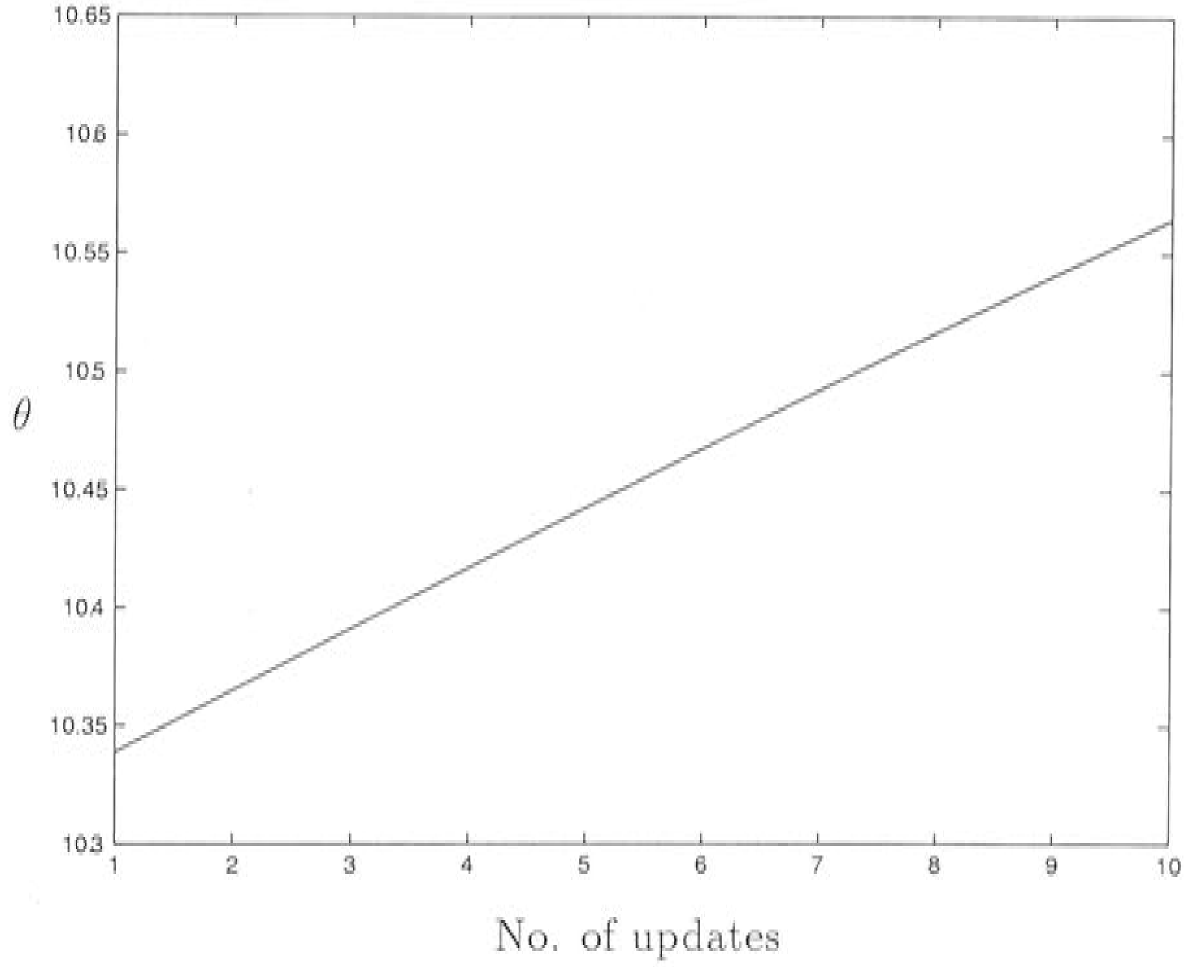


Figure 10.6: Successive smoothing parameter updates on grid no. 1 generated by the OPTRSS algorithm, for the data set sine.dat.

10.1.1 The starting value of θ

Experiments varying the starting value θ_0 on the coarsest grid showed that some care must be taken when choosing the initial estimate. In the case of this problem, the analytic value of the smoothing parameter is known to be 25500, which corresponds to a θ value of 10.1464. Starting values in a range two orders of magnitude smaller than the analytic smoothing parameter, 500 and 100, were chosen. The results for $\lambda_0 = 500$ are shown in section A.2. The θ estimates are clearly much more unstable than those in section A.2. When the starting value was set to 100, the procedure diverged even faster. However, it was found that if starting values around 2 orders of magnitude larger are chosen ($\lambda_0 = 5000000$), the estimates were much more stable (section A.3). Smaller values of θ are likely to cause instability because they result in very small values of λ/h^3, particularly on the coarse grid. This will cause the system of equations (3.5) to become dominated by the P^TP term, making the solution and its derivatives relatively insensitive to changes in θ. This means that θ can deviate from the optimum value by a large amount before the derivatives adjust in response to this change.

These results show that it is clearly advantageous to chose a starting value of θ in the smooth end of the spectrum. This makes intuitive sense as the nested grid algorithm ascertains the smooth, broadscale trends first by starting on a very coarse grid, then gradually builds up fine scale structure in the solution estimate as the process is transferred to finer grids. The obvious question is how to calculate this starting estimate of θ. It was not an issue in this case because the full spectrum of λ values and corresponding signal values could be calculated analytically, as was done in Figure 9.4. However, one method is to set the starting value of λ, given by λ_0, as

$$10 = \frac{6\lambda_0}{h^3} \tag{10.1}$$

This is designed to set the two terms in the left hand side of equation (3.5) to have approximately equal influence, assuming there are around 10 data points in each grid cell on the coarsest grid. This avoids putting particular emphasis put on smoothing or close fitting of the data.

10.1.2 The prescribed residual sum of squares

The above analysis prescribed different values of the residual sum of squares for different grids in such a way that the optimal value of λ would not change from grid to grid. It was not the original intention of the algorithm to prescribe S values in this manner, as information about the effect of discretisation on grids of differing coarseness is obviously unknown in practice. The algorithm was therefore modified to prescribe an S value of 2.8208 on all grids. This was the value obtained by solving the piecewise constant smoothing spline system on the fine grid with the smoothing parameter set to the minimum GCV value calculated from ANUSPLIN. The Newton method should therefore converge to this smoothing parameter.

The effect of this modification was quite disastrous for the algorithm, as indicated by the results in section A.4. The immediate divergence of θ to negative values occurred because the prescribed R was too small to be achieved on the coarse grid. The smallest possible residual sum of squares on the coarse grid can be calculated by prescribing a smoothing parameter of zero. The resulting R was found to be 3.4201 on the coarse grid, larger than the fine grid value of 2.8208. The θ estimate quickly became very small on the coarse grid in an attempt to achieve the fine grid S value, and eventually become infinitesimal. This emphasises the fact that exact interpolation cannot be achieved on grids too coarse to allow each data point to have a separate grid cell. In the case of this problem, there is more than one data point per grid cell until grid 2 because the data occur at regularly spaced intervals of 3.6.

One way to combat this problem is to put a lower limit on the θ update, so it does not diverge past the point of recovery on grids for which the prescribed residual sum of squares is too small. A condition was set so that, if a θ update was lower than a certain threshold, the update would be set to that threshhold. Threshold values were chosen to give a value of λ/h^3 that would not cause equation (3.5) to be dominated by one particular term. Requiring a lower limit on λ/h^3 allows the minimum θ value to decrease as the resolution is refined, as λ/h^3 will be larger on finer grids for a given θ because h decreases. The results for a chosen threshold value of $\lambda/h^3 = 0.5$ are shown in Table 10.3 and section A.5.

Clearly the θ estimate is fixed at the threshold value for all coarse grid up-

Grid no.	No. of updates	Converged θ value	Converged $S - R$ value	Converged $dR/d\theta$ value
6	1	9.70	-0.64	0.046
5	9	10.02	0.000	0.14
4		non	convergence	
3		non	convergence	
2		non	convergence	
1		non	convergence	

Table 10.3: Results generated by the OPTRSS algorithm, with a lower threshold on λ updates of $\lambda/h^3 = 0.5$, for the data set sine.dat.

dates. Extreme negative updates have been avoided, stabilising the algorithm somewhat. The algorithm does converge on grid number 5, although the θ estimates in section A.5 are still not as stable as those produced by the procedure in section 10.1. Convergence on the grid number 5 is slower, and the estimates clearly diverge on grids 1 and 2 to a greater extent than in section A.1. These problems are a result of the fact that, in order to achieve the same prescribed residual sum of squares on all grids, the required smoothing parameter must change from grid to grid. This means that the solution must change on every grid to adjust for the new θ required to meet the prescribed S value. The analyses in Chapter 9 demonstrated the slow rate at which solution estimates change on fine grids, a manifestation of the poor conditioning of the system. It is clear that, in this case, the solution on finer grids cannot be corrected for the changing θ estimate. The θ estimate keeps increasing due to the continual discrepancy between the prescribed residual sum of squares and the residual sum of squares of the current solution. The problem of 'synchronising' the θ update with the changing solution is fundamental to the use of these techniques on systems that are not well conditioned. Further methods of dealing with this phenomenon are discussed in Chapter 11.

10.1.3 The number of iterations per update

The number of SOR iterations performed between updates of θ is fundamental to the synchronisation issues previously discussed. It might be expected that the process of updating θ could be stabilised if the number of iterations per update was increased to allow the solution to change in response to the previous update. This was confirmed by studying the rate of convergence on grid number 4 when the number of iterations per update was increased from 3 to 10. Convergence of θ to four decimal places was achieved in a total of 180 SOR iterations with 10 iterations per update, where as a total of 306 SOR iterations were required with 3 iterations per update. These results suggest that the process is at its most efficient with fewer iterations per update on coarse grids and more on fine grids. This is a logical conclusion considering that on the coarse grid the solution can be changed by the required amount after each update in a very short time, and further iterations are wasteful.

10.1.4 The effect of fixing the smoothing parameter to a lower value

Many characteristics of the above results depended heavily on the scale of variation of the chosen solution. It was insightful to run the procedure on the data set sine.dat with a prescribed smoothing parameter of $\lambda = 5$ or $\theta = 1.61$. The analytic spline solution corresponding to a smoothing parameter of 5 is shown in Figure 10.7. While this solution is not optimal, it has much greater fine scale structure and is therefore be better suited to the scale of the grids chosen for this algorithm.

The procedure used at the start of this chapter, where S values on each grid were prescribed to correspond to a smoothing parameter that did not vary between the grids, was again employed here, where λ was fixed at 5. The lower threshold on θ updates given by $\lambda/h^3 = 0.5$ was again emplaced. This was necessary to maintain the stability of the algorithm on coarse grids, where exact interpolation was required to estimate the spline solution corresponding to this lower smoothing parameter. The results are shown in Table 10.4 and section A.6.

On finer grids the θ estimates show a lot more stability than the results for the higher smoothing parameter. On the 2 coarsest grids, the results show the forcing of the θ estimate to its lower threshold. Convergence to 4 decimal places was achieved on grid numbers 4 and 3. However, convergence was quite slow on the finest grid. Figures 10.8 and 10.9 show that the grids in this algorithm are again too fine compared to the scale of the solution. The solution clearly changes very little between grid numbers 2 and 1, implying that grid number 2 is fine enough to represent the variability in the solution. This raises the

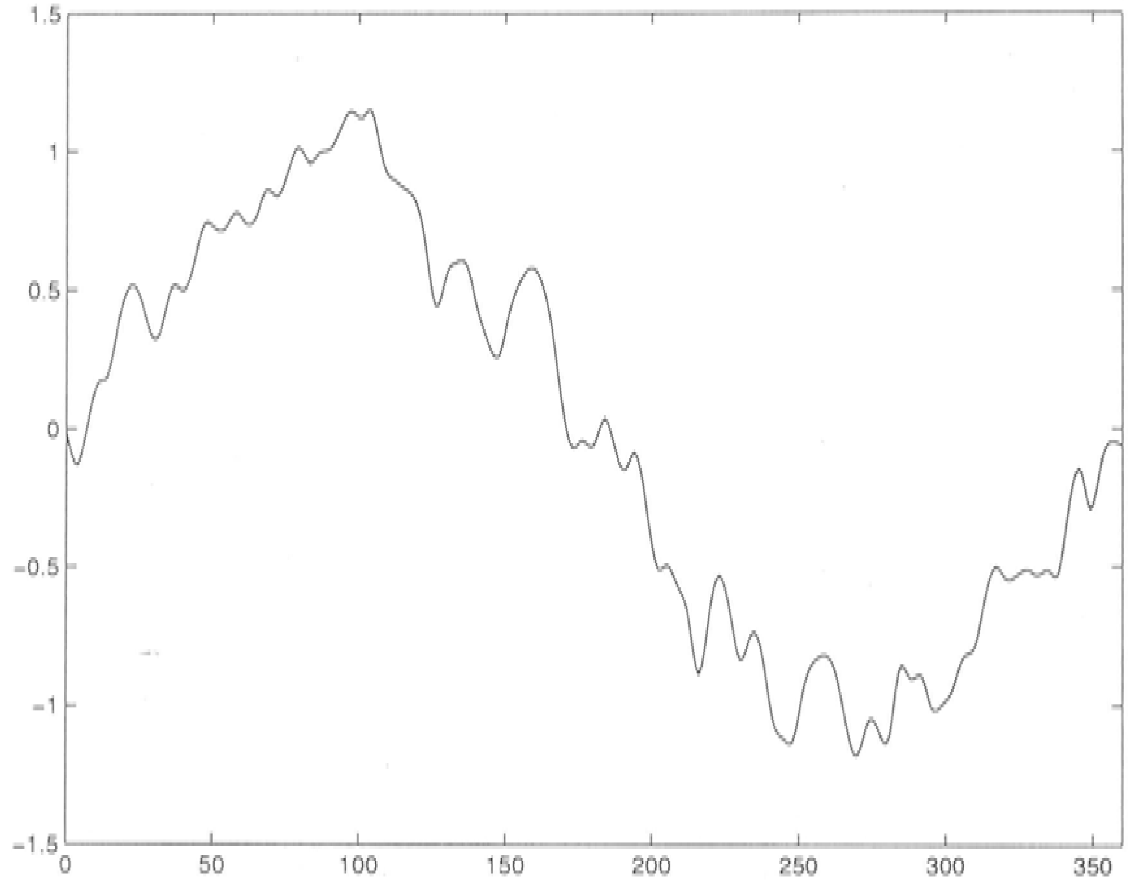


Figure 10.7: Smoothing spline solution for the data set sine.dat, with a fixed smoothing parameter of 5.

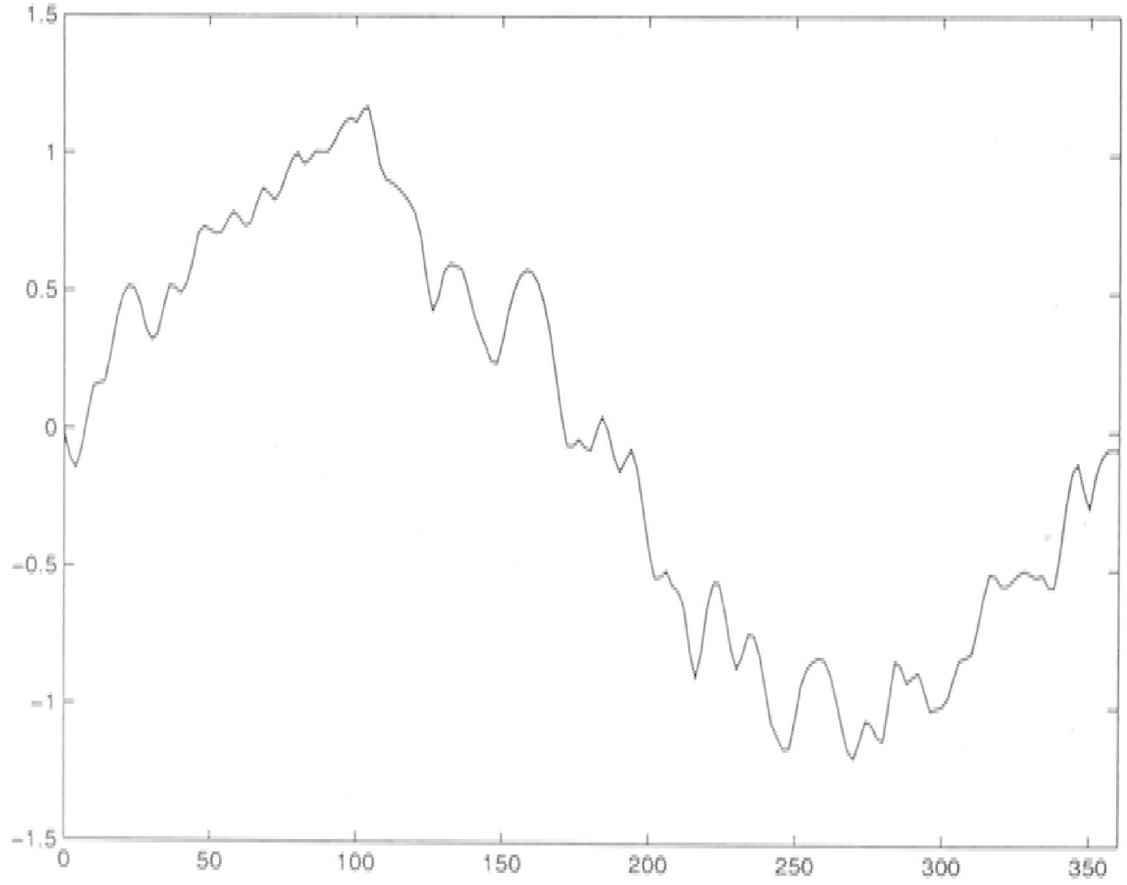


Figure 10.8: OPTRSS solution on grid no. 2 for a fixed smoothing parameter of 5, for the data set sine.dat.

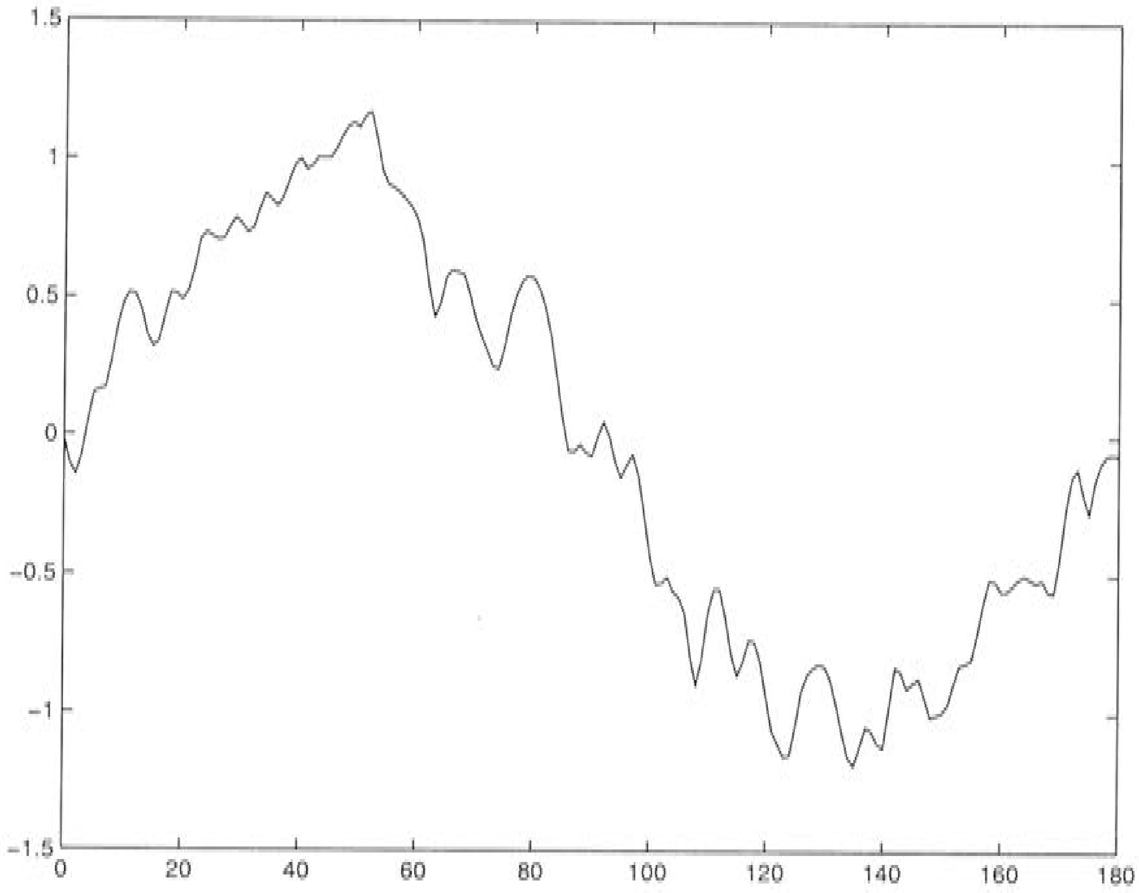


Figure 10.9: OPTRSS solution on grid no. 1 for a fixed smoothing parameter of 5, for the data set sine.dat.

Grid no.	No. of updates	Converged θ value	Converged $S-R$ value	Converged $dR/d\theta$ value
6	0	5.09	0.000	0.00
5	4	3.02	0.000	0.00
4	7	1.61	0.000	0.00
3	4	1.61	0.000	0.33
2	8	1.61	0.000	0.44
1	14	1.61	0.000	0.44

Table 10.4: Results of the OPTRSS algorithm, with residuals prescribed to correspond to a fixed smoothing parameter of 5, for the data set sine.dat.

fundamental question of when to stop refining the grids, an issue which is addressed in following chapters.

10.2 Performance of the MINGCV algorithm

This section describes the results produced by the MINGCV algorithm described in Chapter 8, applied to the piecewise constant discretised system. The results of this process led to the incorporation of the quadratic B-spline framework. The initial settings for the following analysis were the same as those used for the OPTRSS algorithm, as discussed in the second paragraph of this chapter, except that 20 iterations were performed per grid instead of 10 to give more information about the behaviour of the θ estimation process. The simplified MINGCV algorithm used in this chapter is shown on the next page.

Details of the results of the MINGCV algorithm are shown in Table 10.5 and section A.7. The patterns are similar to those observed for the OPTRSS algorithm, which is not surprising considering that the two algorithms are very similar in structure. Convergence to 4 decimal places was relatively fast on the two coarsest grids. This is also reflected in the $dGCV/d\theta$ estimates, which quickly approach zero. The minimum GCV estimate of θ was different on these grids, demonstrating that discretising on grids of different coarseness affects the measured smoothness of the solution. Convergence deteriorated on the finer grids. On fine grids the solution is clearly non-responsive to changes in θ, to an apparently worse degree than that demonstrated by the OPTRSS algorithm. There is reason to expect the OPTRSS method to be more robust given that the Newton correction to θ in equation (8.1) is based on the difference between the current residual sum of squares and the prescribed residual sum of squares. It therefore always has the right sign, provided θ values are not in the extreme range of Figure 8.1. The correction to θ for the MINGCV algorithm, given in equation (8.14), is dependent on a greater number of estimated values, and is not so robust in sign or magnitude.

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6	8	10.10	3.48	6.0	0.0389
5	9	8.54	2.59	10.2	0.0317
4		non	convergence		
3		non	convergence		
2		non	convergence		
1		non	convergence		

Table 10.5: Results generated by the MINGCV algorithm, for the data set sine.dat.

```
for $l = 6$ to 2
    for $q = 0$ to 19
        for $m = 0$ to 2
            $\mathbf{u}_l^{(m)}(\theta_q) = S_l^{v_1}(\mathbf{u}_l^{(m)}(\theta_q), \mathbf{v}_l^m)$
        end
        for $n = 0$ to 1
            $\mathbf{b}_l^{(m)}(\theta_q) = S_l^{v_1}(\mathbf{b}_l^{(n)}(\theta_q), \mathbf{w}_l^n)$
        end
        $\theta_{q+1} = -\frac{b}{2c}$
        $q = q + 1$
    end
    for $m = 0$ to 2
        $\mathbf{u}_{l-1}^{(m)} = T_l \mathbf{u}_l^{(m)}$
    end
    for $n = 0$ to 1
        $\mathbf{b}_{l-1}^{(n)} = T_l \mathbf{b}_l^{(n)}$
    end
end
$l = 1$
for $q = 0$ to 19
    for $m = 0$ to 2
        $\mathbf{u}_1^{(m)}(\theta_q) = S_l^{v_1}(\mathbf{u}_l^{(m)}(\theta_q), \mathbf{v}_l^m)$
    end
    for $n = 0$ to 1
        $\mathbf{b}_1^{(m)}(\theta_q) = S_l^{v_1}(\mathbf{b}_l^{(n)}(\theta_q), \mathbf{w}_l^n)$
    end
    $\theta_{q+1} = -\frac{b}{2c}$
end
```

An initial test of whether the algorithm is working correctly is to check whether the results are consistent with the analytic trends in Figures 8.1- 8.5. According to Figures 8.4 and 8.5, $dR/d\theta$ and $dTr/d\theta$ estimates should always be positive, provided θ is not in the extreme low or high regions. The results in section A.7 show that this is always the case on the three coarsest grids. On finer grids the results deviate from analytic behaviour. According to Figure 8.3, the $dGCV/d\theta$

estimate can be negative or positive, depending on whether θ is greater than or less than the value corresponding to the minimum GCV. However, the $d^2GCV/d\theta^2$ estimate should not be negative unless θ is in the extreme high or low ranges. If the $d^2GCV/d\theta^2$ estimate is negative, the MINGCV algorithm will be unable to converge to a minimum GCV. The estimates of $d^2GCV/d\theta^2$ in section A.7 are always positive on the three coarsest grids, which is an indication that the algorithm is stable on these grids. The process clearly did not converge on the finer grids.

The plots of the θ estimate in Figures 10.10 - 10.15 give insight into the behaviour of the algorithm. The θ estimate clearly converges on grid numbers 6 and 5, but looks unlikely to do so on grid number 4, and clearly diverges on the finer grids. There could be a number of reasons for this. Based on the results of the OPTRSS algorithm, it seems logical that the inefficiency of basic iteration is responsible for the divergence. On fine grids, the system is poorly conditioned and SOR is very slow to adjust to the required changes in the smoothness of the solution. This appears to be the problem on grid number 3, where the inaccuracy of the solution is reflected in the $dGCV/d\theta$ value. As θ decreases to values clearly below the optimal level, $dGCV/d\theta$ decreases in magnitude and eventually becomes negative is response to the small θ value, but it is very slow to do so. The θ estimate then increases to values that are too large, and $dGCV/d\theta$ does not respond in time, creating a pattern of oscillatory divergence. Eventually θ becomes extreme and $d^2GCV/d\theta^2$ becomes negative, which destroys any chance of convergence.

Further investigation of the GCV revealed another possible reason for the divergence observed in Tables 10.5 and section A.7. Table 10.6 shows GCV values corresponding to the piecewise constant solution for the 3 coarsest grids, for different smoothing parameter values. The GCV trends deviate from the analytic pattern in Figure 8.3. On grid number 6 there is a clear minimum at 25500 as would be expected, but the GCV also appears to have a local minimum at 70000, an anomaly that did not occur in the analytic case. The results in section A.7 are consistent with this scenario, converging to a θ value corresponding to $\lambda = 24416$ on grid number 6. On grid number 5 there appears to be a local minimum at $\lambda = 5000$ and again at $\lambda = 30000$ and $\lambda = 500000$. The MINGCV algorithm clearly targets the smaller optimum. However, on grid number 4, where there are minima at $\lambda = 5000$ and $\lambda = 500000$, the

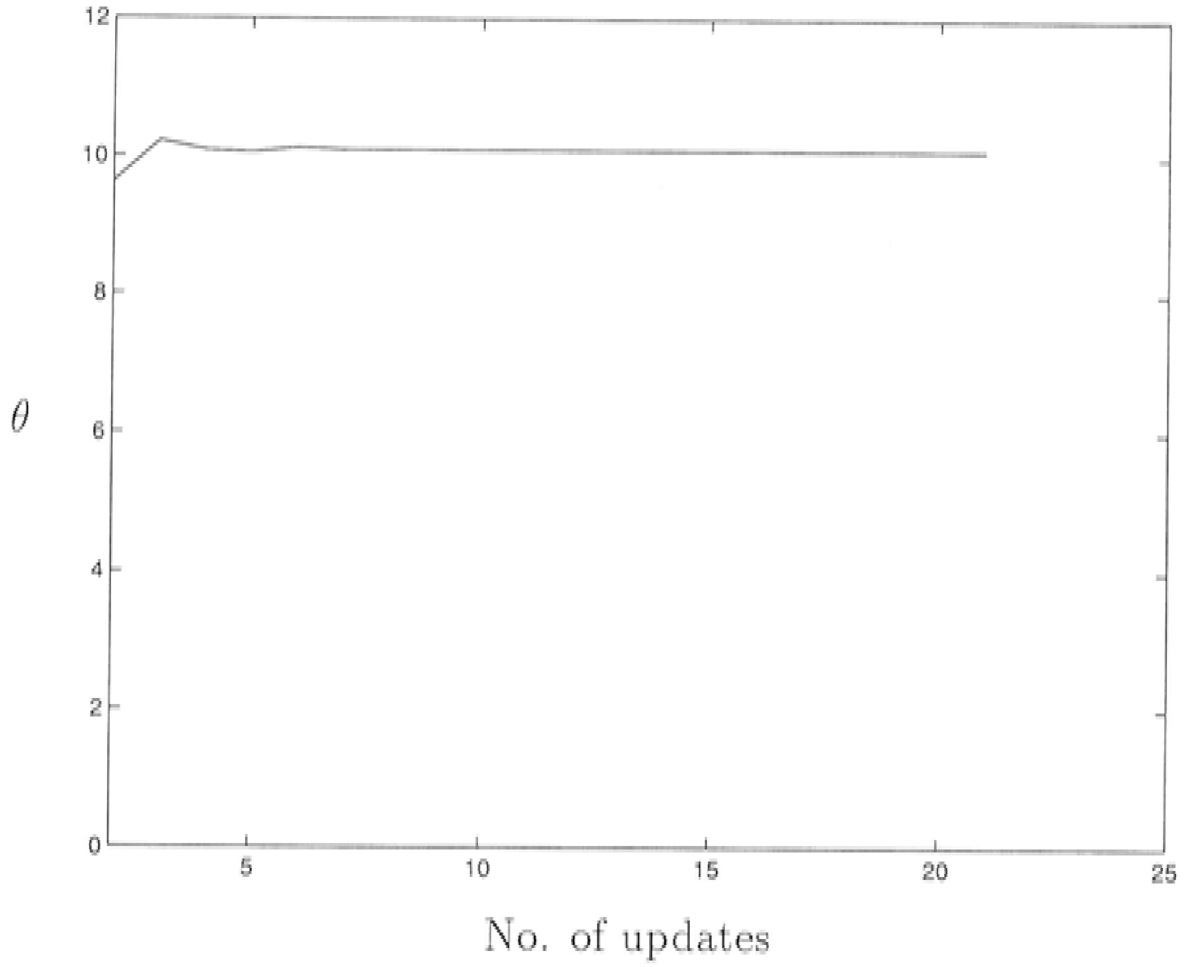


Figure 10.10: Successive smoothing parameter updates on grid 6 generated by the MINGCV algorithm, for the data set sine.dat.

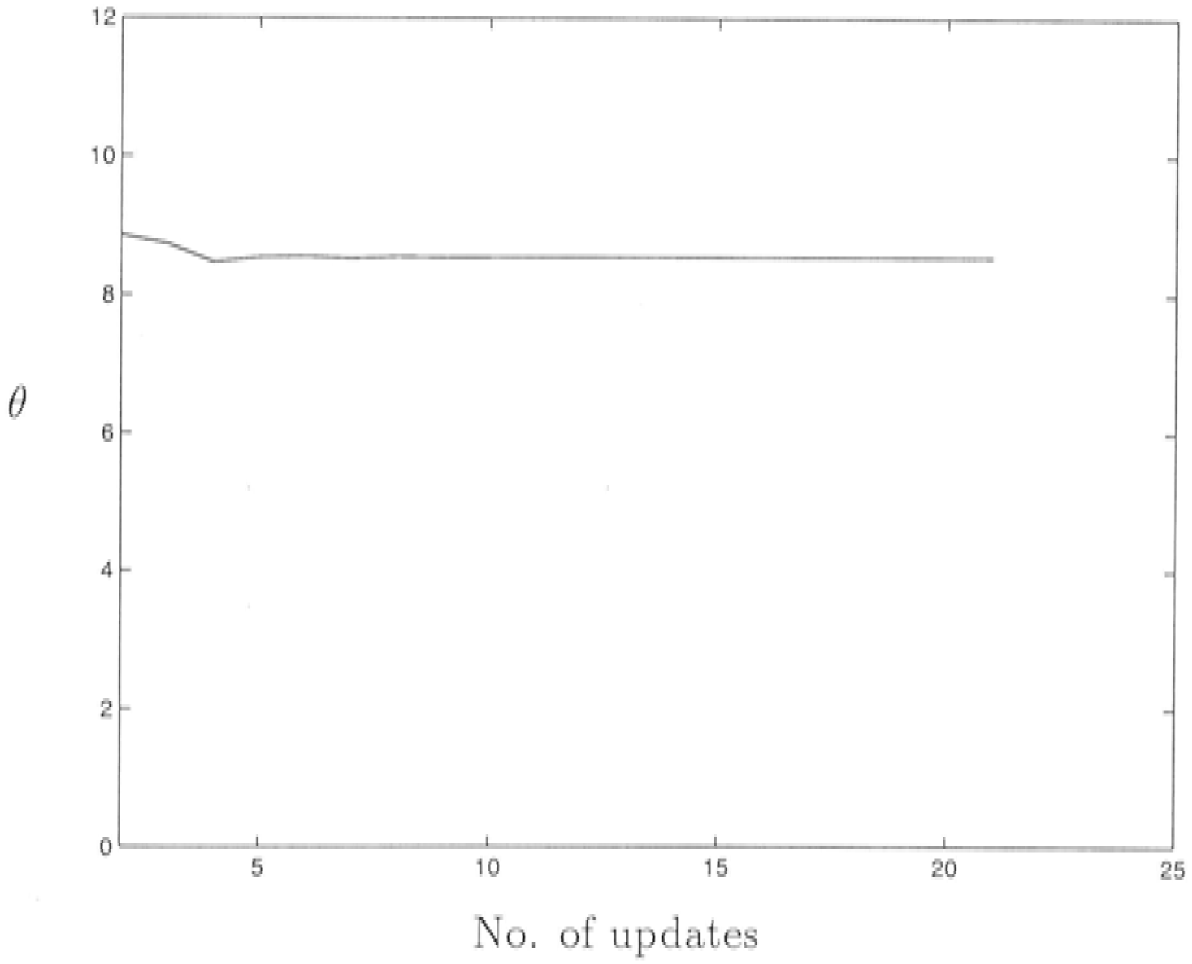


Figure 10.11: Successive smoothing parameter updates on grid 5 generated by the MINGCV algorithm, for the data set sine.dat.

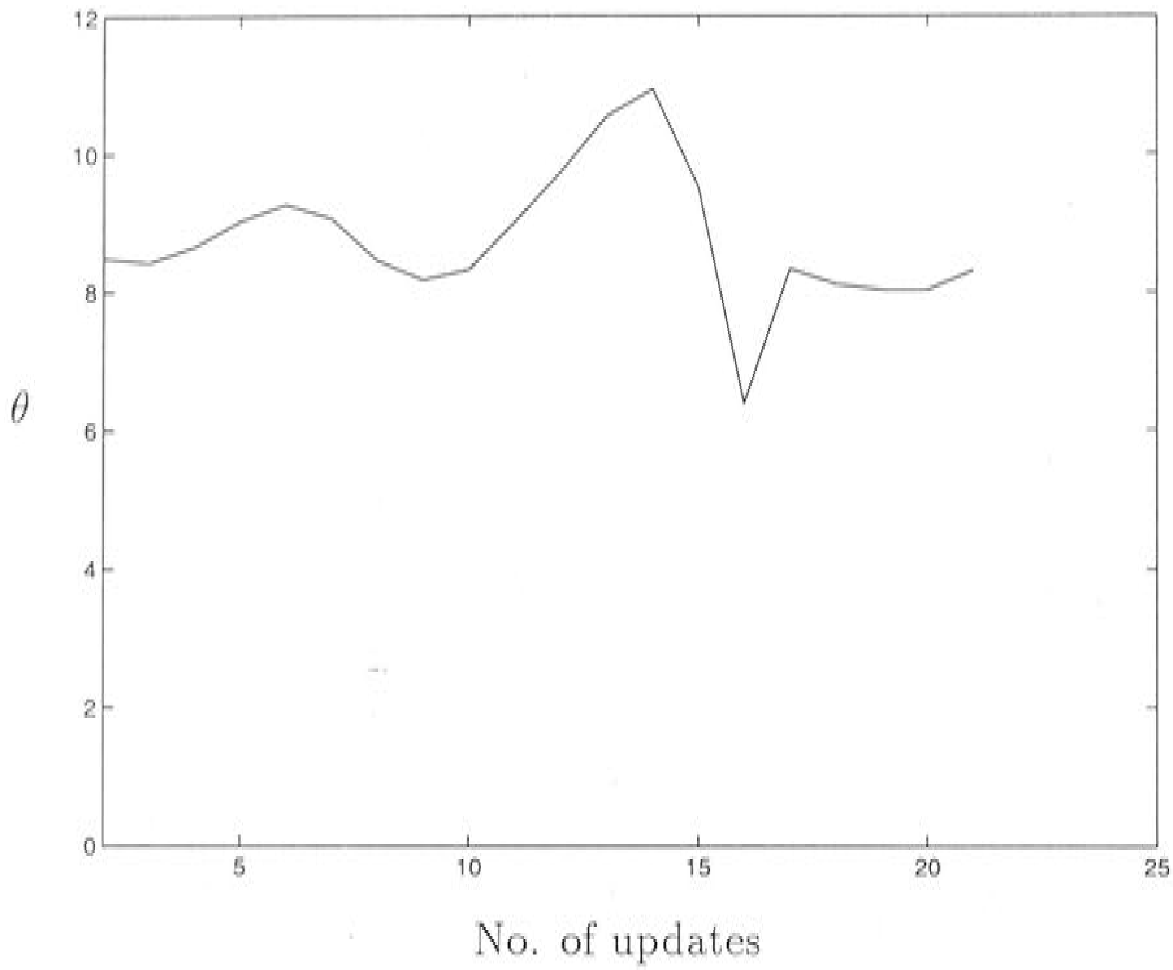


Figure 10.12: Successive smoothing parameter updates on grid 4 generated by the MINGCV algorithm, for the data set sine.dat.

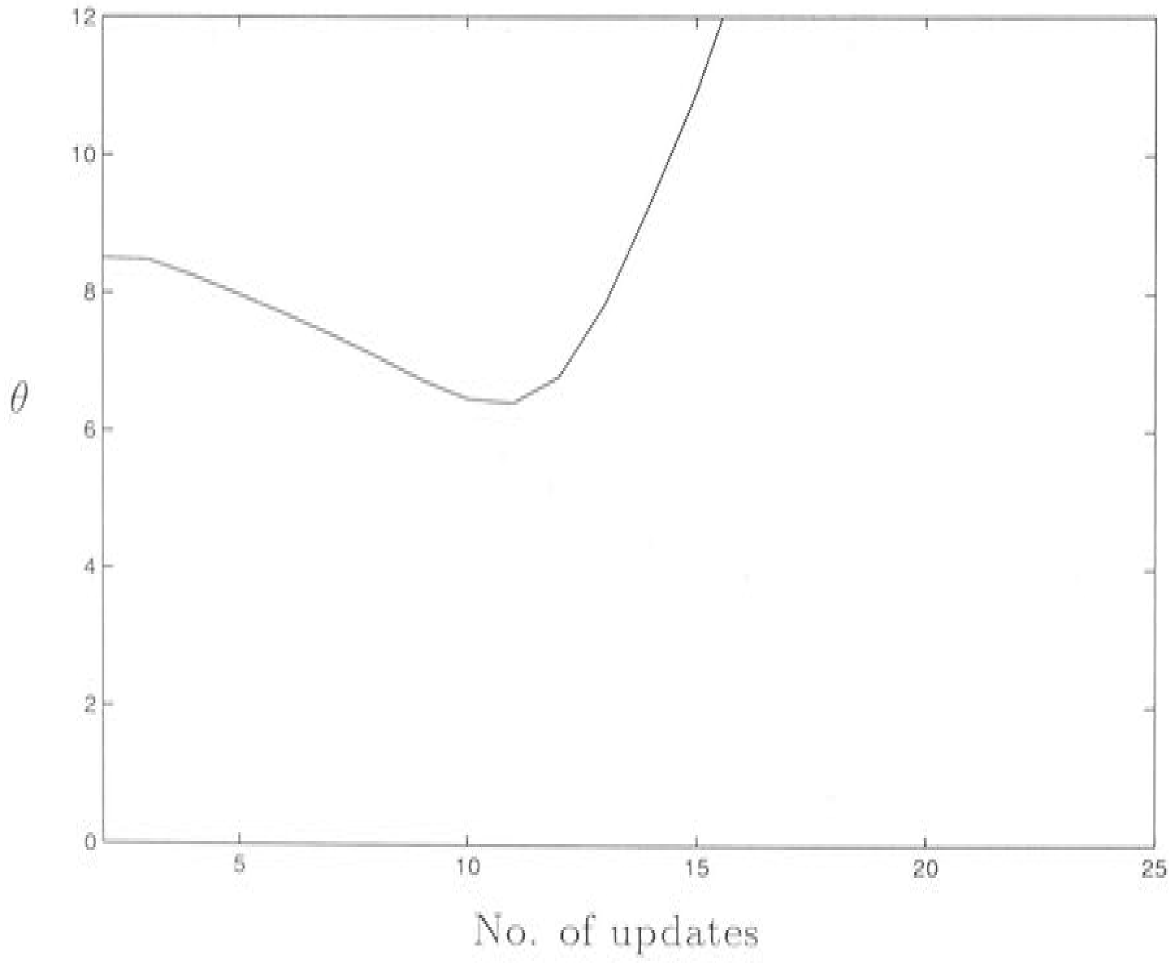


Figure 10.13: Successive smoothing parameter updates on grid 3 generated by the MINGCV algorithm, for the data set sine.dat.

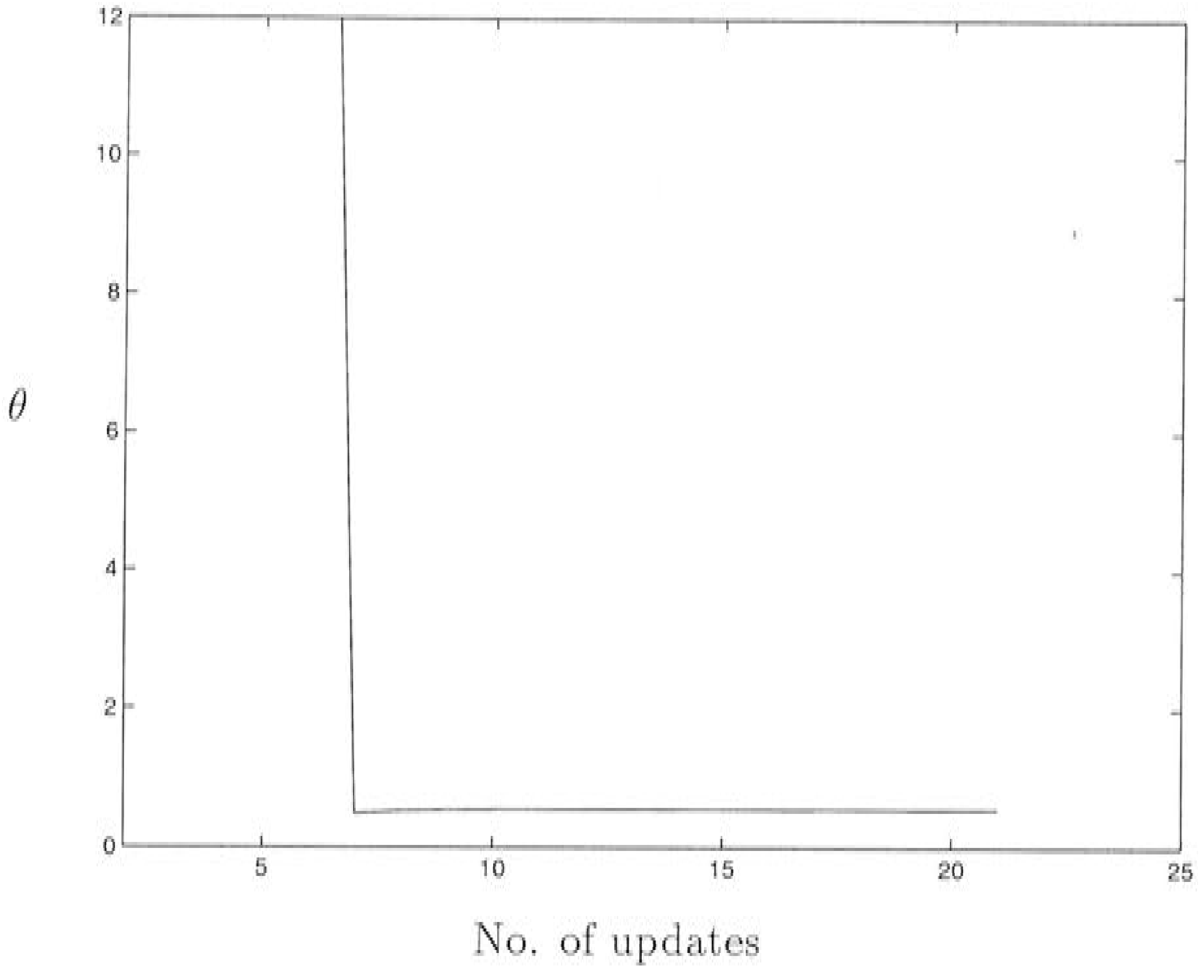


Figure 10.14: Successive smoothing parameter updates on grid 2 generated by the MINGCV algorithm, for the data set sine.dat.

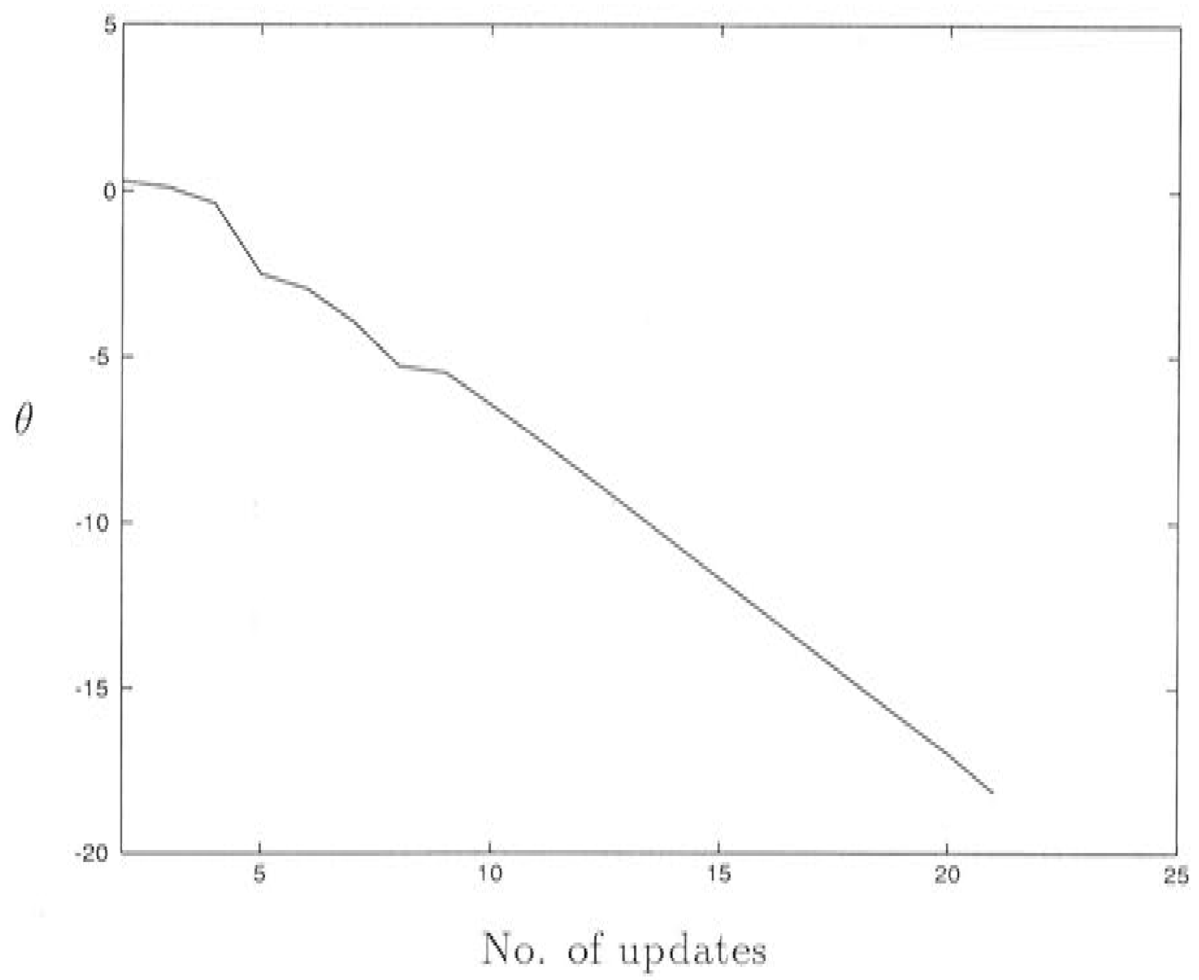


Figure 10.15: Successive smoothing parameter updates on grid 1 generated by the MINGCV algorithm, for the data set sine.dat.

λ	GCV		
	grid 6	grid 5	grid 4
1000	.0396	0.0329	0.0320
5000	.0393	**0.0318**	**0.0308**
10000	.0391	0.0319	0.0309
25500	**.0389**	0.0322	0.0313
30000	**.0389**	**0.0318**	0.0329
40000	.0390	0.0325	0.0316
50000	.0409	**0.0317**	**0.0305**
70000	**.0394**	0.0329	0.0323
100000	.0399	0.0335	0.0330

Table 10.6: GCV values for the piecewise constant approximation to the smoothing spline on each grid, for different prescribed smoothing parameters, where local minima are emphasised.

algorithm appears to oscillate between the two. Initially the numbers tend towards the smaller optimum, although there is some oscillation, which would be expected as this grid is relatively fine. At 12 updates the numbers jump towards higher values close to that of the larger optimum. This is followed by a sharp dive towards lower values.

The behaviour of the θ estimate is clearly reliant on the underlying structure of the GCV estimate. The errant behaviour of the GCV estimate could be due to discretisation error introduced by the piecewise constant approximation or stochastic error in the estimate of Tr. This is discussed further in Chapter 11. It was also interesting to note the resilience of the GCV to changes in the smoothing parameter in the above analysis. Increasing λ by a factor of 10 changes the GCV only around 2%. This implies that a high degree of accuracy in the θ estimate is not necessary to obtain a solution that represents the data in an optimal way. Further analysis revealed a number of factors that explain more about the behaviour of the algorithm. These findings are discussed below.

10.2.1 Stochastic error in the trace estimate

The MINGCV algorithm has a fundamental dependence on the trace estimate Tr, due to its presence in the calculation of the GCV and all its derivatives. This factor distinguishes the MINGCV algorithm from the OPTRSS algorithm, which did not involve the trace estimate in the optimisation process. The data set sine.dat is very small, having only 101 observations. The stochastic error in the Tr estimate is therefore much higher for the analyses discussed in this chapter than it would be for a large data set, given that the relative standard deviation of the Tr estimate is bounded by $(2/n)^{1/2}$ as discussed in section 8.2.1. While the MINGCV algorithm is intended for application to large data sets, an investigation of the influence of stochastic error on this analysis was necessary to understand the behaviour of the algorithm. This involved looking at how different realisations of the random vector $\mathbf{t}$ affected the performance of the algorithm. For example, using a different random vector $\mathbf{t}$ to that used to produce the results in Table 10.5 gave the results in Table 10.7, with full results in section A.8.

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6	11	9.93	3.47	13.6	0.0459
5		non	convergence		
4		non	convergence		
3		non	convergence		
2		non	convergence		
1		non	convergence		

Table 10.7: Results generated by the MINGCV algorithm for the data set sine.dat, using a second random vector $\mathbf{t}$.

λ	GCV					
	grid 1	grid 2	grid 3	grid 4	grid 5	grid 6
1000	0.0349	0.0351	0.0363	0.0335	0.0353	0.0463
5000	0.0355	0.0360	0.0371	0.0348	0.0359	0.0461
10000	0.0358	0.0362	0.0374	0.0353	0.0362	0.0460
25500	0.0361	0.0366	0.0378	0.0359	0.0364	**0.0459**
40000	0.0364	0.0368	0.0380	0.0362	0.0365	0.0460
60000	0.0367	0.0371	0.0383	0.0365	0.0367	0.0462

Table 10.8: GCV values for the piecewise constant approximation to the smoothing spine on each grid, for different prescribed smoothing parameters, for a second random vector $\mathbf{t}$. Local minima are emphasised.

Convergence is direct on grid number 6, but there are clearly problems on grid number 5. In order to calculate where the minimum GCV value should be, the piecewise constant system was again solved for different smoothing parameter values and GCV values were calculated, as shown in Table 10.8. These trends are different to those that would be expected for this problem, in that there is no minimum GCV value on grid number 5. Instead, the GCV continues to decrease as θ gets smaller. This implies that the optimal solution is too fine in structure to be represented on a grid of 24 points, and θ is forced to zero in an attempt to represent this solution. If this was the case, the optimal signal must be greater than 24. This is considerably larger than the analytic value of 8.2. Thus this particular random vector $\mathbf{t}$ is clearly associated with particularly high stochastic error in the trace estimate.

The results for a third random vector are shown in Tables 10.9 and section A.9. The estimates are again stable on the coarse grid but behave erratically on all other grids. The results in Table 10.10 indicate that the GCV structure is analytic in its behaviour, with a single minimum on all grids. However, it can be seen from section A.9 that, following an update to a large θ value on grid number 5, $d^2GCV/d\theta^2$ becomes negative. This is a sign that the θ estimate is in the extreme range of the GCV curve in Figure 8.1. The optimisation procedure would be expected to perform poorly in this region. A procedure involving dampening the update to θ was found to improve this situation. This is discussed further in the next section.

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6	8	11.29	3.61	10.7	0.0448
5		non	convergence		
4		non	convergence		
3		non	convergence		
2		non	convergence		
1		non	convergence		

Table 10.9: Results generated by the MINGCV algorithm for the data set sine.dat, for a third random vector $\mathbf{t}$.

The difference in the results in Tables 10.5, 10.7 and 10.9 show that the trace estimate is clearly an important issue in this algorithm, particularly when it is used on small data sets such as sine.dat. One possible solution is to use several random vectors, and take the average of the result. For example, to

λ	GCV					
	grid 1	grid 2	grid 3	grid 4	grid 5	grid 6
1000	0.0386	0.0395	0.0407	0.0375	0.0391	0.0492
5000	0.0379	0.0386	0.0398	0.0371	0.0379	0.0482
10000	0.0373	0.0380	0.0391	0.0368	0.0375	0.0474
25500	0.0367	0.0373	0.0384	0.0365	0.0370	0.0459
40000	**0.0366**	**0.0371**	**0.0383**	**0.0363**	**0.0369**	0.0453
60000	**0.0366**	**0.0371**	**0.0383**	0.0364	**0.0369**	**0.0449**
80000	**0.0366**	**0.0371**	**0.0383**	0.0364	**0.0369**	**0.0449**
100000	0.0373	0.0377	0.0389	0.0369	0.0373	**0.0449**

Table 10.10: GCV values for the piecewise constant approximation to the smoothing spline on each grid, for different prescribed smoothing parameters, for a third random vector. Local minima are emphasised.

reduce the variance of the stochastic error by a factor of 10 would require 10 random vectors. This option is explored in section 11.1.

One common result of the preceding analyses is that convergence on the coarse grid is always fast and direct, regardless of the instability of the performance on other grids. This shows that the algorithm obviously has potential for high speed performance for systems that are well conditioned.

10.2.2 Dampening the θ updates

One way to address the synchronisation issues observed in the above analysis is to damp the correction to the θ estimate with each update. This can be used to stabilise the algorithm in situations where the θ updates have high error. Also, basic iteration cannot efficiently respond to large θ updates, particularly on fine grids. A dampening factor of 1/2 was tried for the random vector used in Table 10.9, meaning that the amount by which the θ is updated, calculated by (8.14), is multiplied by 1/2. The results are shown in Table 10.11, with full results in section A.10.

The effect of the damping is to improve convergence on finer grids, but slow it down on the coarse grid. On the coarsest grid, convergence to 4 decimal places was delayed by a few updates. On grid number 5, however, the numbers are much more stable than those in Table 10.10, and the smoothing parameter does not jump into the region of negative $d^2GCV/d\theta^2$. This indicates that the issue of synchronising the various iterative processes in the algorithm is

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6	8	11.29	3.61	10.7	0.0448
5		non	convergence		
4		non	convergence		
3		non	convergence		
2		non	convergence		
1		non	convergence		

Table 10.11: Results generated by the MINGCV algorithm for a dampening factor of 1/2, for the data set sine.dat.

fundamental to its success. If the required change is smaller, basic iteration is powerful enough to make the algorithm stable. On the coarse grid, dampening is clearly wasteful, as SOR can keep up with the full update. However, coarse grid updates are not computationally expensive. As soon as the grids get finer, SOR is only capable of making very small changes to the solution with each iteration in response to changing θ. Clearly the requirement of a different level of smoothness resulting from updating θ introduces error components with a frequency too low to be reduced on fine grids.

10.3 Conclusion

The analysis so far has characterised, at a preliminary level, the behaviour of the MINGCV algorithm, or the algorithm for estimating minimum GCV smoothing splines, developed in Chapter 8. At this point, the algorithm has only been tested on the piecewise constant discretised system in (3.5). It appears that the MINGCV algorithm will converge efficiently on grids of an appropriate resolution, not too coarse to represent the complexity in the process, and not too fine to render the system poorly conditioned. Thus for an extremely smooth process such as a single sine curve, the algorithm will only converge on the coarser grids. Divergence on finer grids occurs due to poor conditioning, which leads to poor synchronisation of the two iterative process updating both the smoothing parameter estimate and the corresponding solution estimate. SOR iteration cannot respond efficiently to larger changes in θ for a poorly conditioned system, so the updates of θ diverge away from the optimal value. For a finer scale process, the algorithm will converge efficiently on finer grids, because the conditioning improves as the smoothing parameter is decreased. This was the case for the fine scale process corresponding to the smoothing spline fit to the data set sine.dat with the smoothing parameter fixed at $\lambda = 5$.

The above results have identified two potential difficulties associated with the MINGCV algorithm. Firstly, due to the small size of the test data sets, stochastic error in the Tr estimate was found to have a considerable influence on the resulting smoothing spline solution, and the convergence behaviour of the algorithm. Using different random vectors $\mathbf{t}$, it is possible to get convergence to an accurate solution estimate, convergence to a less accurate estimate, divergence, or a GCV structure which does not have a local minimum. This can be overcome by averaging results from multiple simulations of the random vector $\mathbf{t}$, as discussed in the next chapter. Secondly, it was found that the GCV structure corresponding to the discretised smoothing spline solution may have multiple minima, or maxima. This deviation from analytic behaviour could be due to either stochastic error in the estimate of Tr, and the corresponding GCV estimate, or the effect of the piecewise constant discretisation. These issues will be addressed in following chapters.

Chapter 11

Minimising GCV for the univariate quadratic B-spline smoothing spline system

The MINGCV algorithm defined in section 8.2 can deliver an accurate solution to the piecewise constant discretised system in equation (3.5) with a good estimate of the optimal smoothing parameter. However, it clearly has problems adjusting to the required corrections on finer grids. It therefore has a tendency to be slow and unstable. The quadratic B-spline approximation discussed in section 3.2 has clear potential to overcome the difficulties of reducing errors on finer grids by allowing a more accurate estimate of the analytic solution on coarse grids, and a smoother transition between the grids. The MINGCV algorithm was therefore applied to the quadratic B-spline discretised system in equation (3.29), still maintaining the initial settings used in the previous chapter, and beginning with the data set sine.dat. The results from this chapter have been published in Hancock and Hutchinson [48].

As in Chapter 10, the convergence criteria limiting updates of the smoothing parameter on a given grid and further refinements of the grid resolution were not emplaced for the investigations presented in this chapter. This allowed a clearer understanding of the convergence process. The results for the quadratic B-spline discretised system were a significant improvement on those obtained for the piecewise constant system. Summary results are shown in Table 11.1, but the effect is best seen by looking at more detailed results in section B.1. Convergence to 4 decimal places is achieved on grids 6 and 5. Section B.1 shows that divergence occurs on grids 4 and finer. Figures 11.1, 11.2 and 11.3 shows rapid convergence on the two coarsest grids, but a pattern of oscillatory divergence sets in on grid 4. However, the divergence on fine grids is much slower than that observed for the piecewise constant system, and $dGCV/d\theta$ is very small, implying that the estimates are not far from the minimum.

The performance of the algorithm for the quadratic B-spline system verifies the previous argument that divergence occurs on fine grids because basic iteration cannot respond to large changes in the smoothing parameter, particularly when the smoothing parameter is very high. This is a result of the poor conditioning of fine grid discretisations of smoothing spline systems corresponding to smooth processes. When a quadratic B-spline discretisation was used, the solution was very accurately represented on the coarse grid so little change was required on finer grids. Figure 11.4 clearly shows the superior representation given by B-spline representation on the coarse grid.

An important benefit of using the quadratic B-spline approach is that the underlying GCV structure appears to be more stable. Values of the GCV for different smoothing parameter values were again calculated, by solving the quadratic B-spline discretised system for fixed smoothing parameters. The results in Table 11.2 show that the GCV structure is consistent with analytic behaviour on all grids. The fact that a change of discretisation has led to

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6	6	7.25	2.78	6.1	0.0312
5	4	8.33	2.68	7.3	0.0309
4		non	convergence		
3		non	convergence		
2		non	convergence		
1		non	convergence		

Table 11.1: Results generated by the MINGCV algorithm, using quadratic B-spline discretisation.

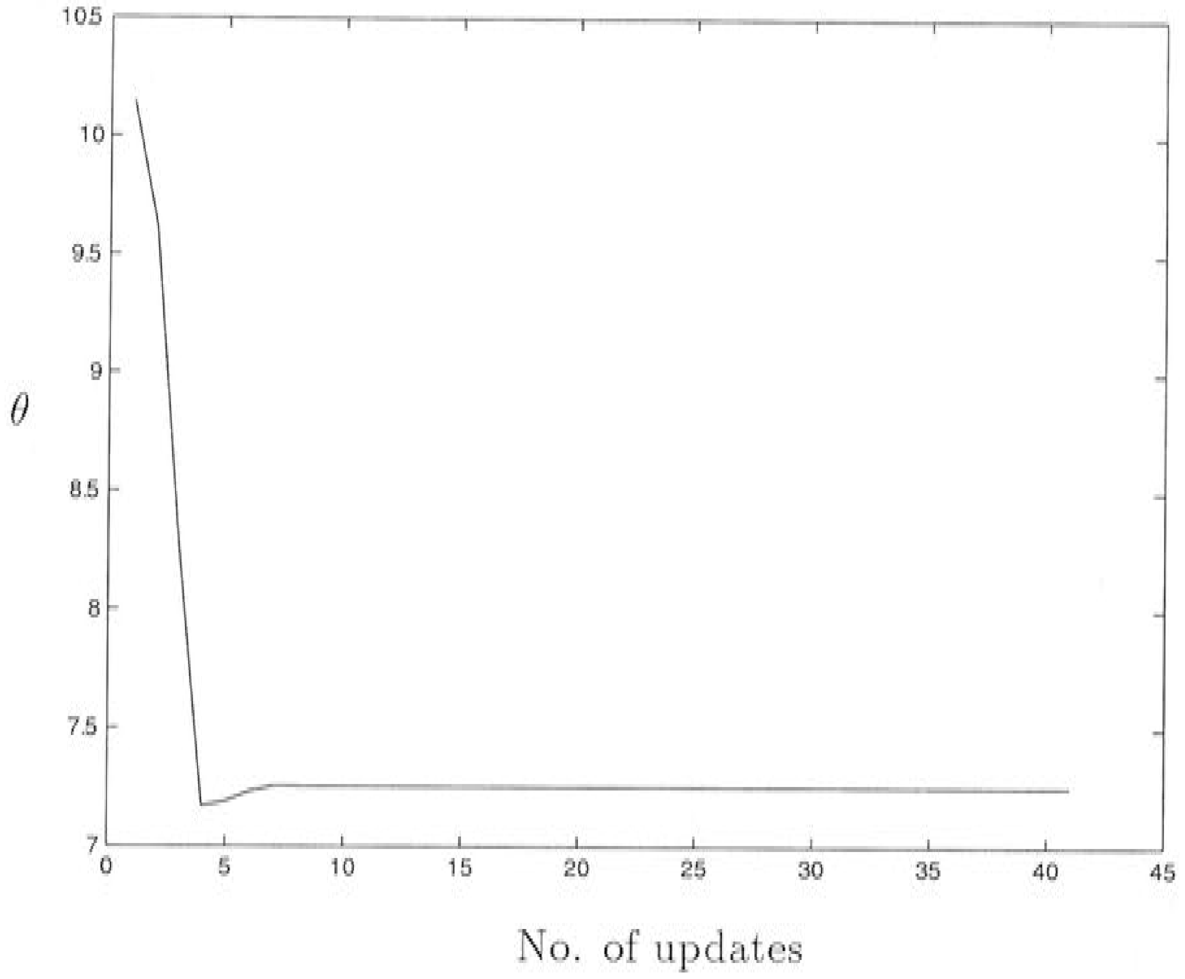


Figure 11.1: Successive smoothing paramater updates on grid no. 6, generated by the MINGCV algorithm, for the data set sine.dat.

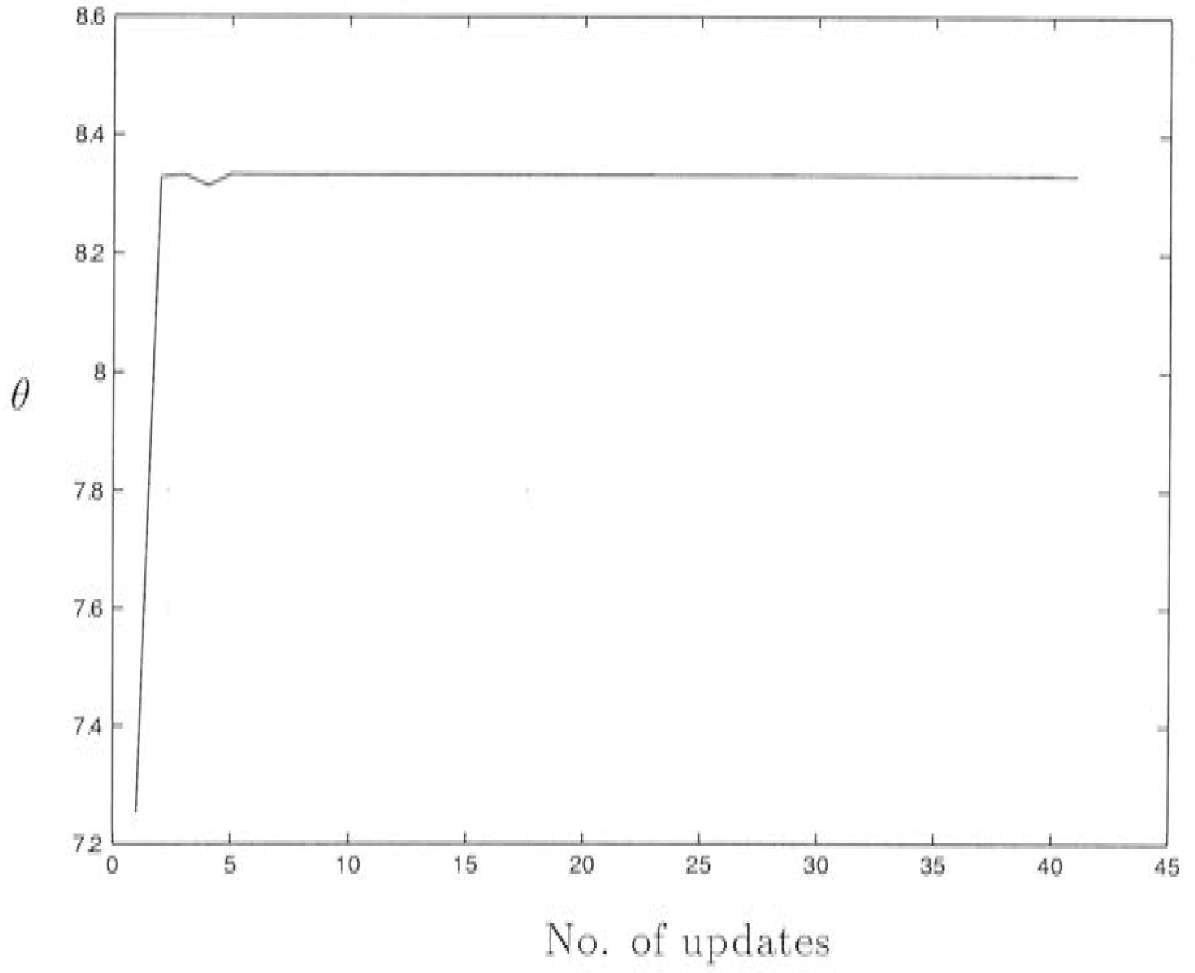


Figure 11.2: Successive smoothing paramater updates on grid no. 5, generated by the MINGCV algorithm, for the data set sine.dat.

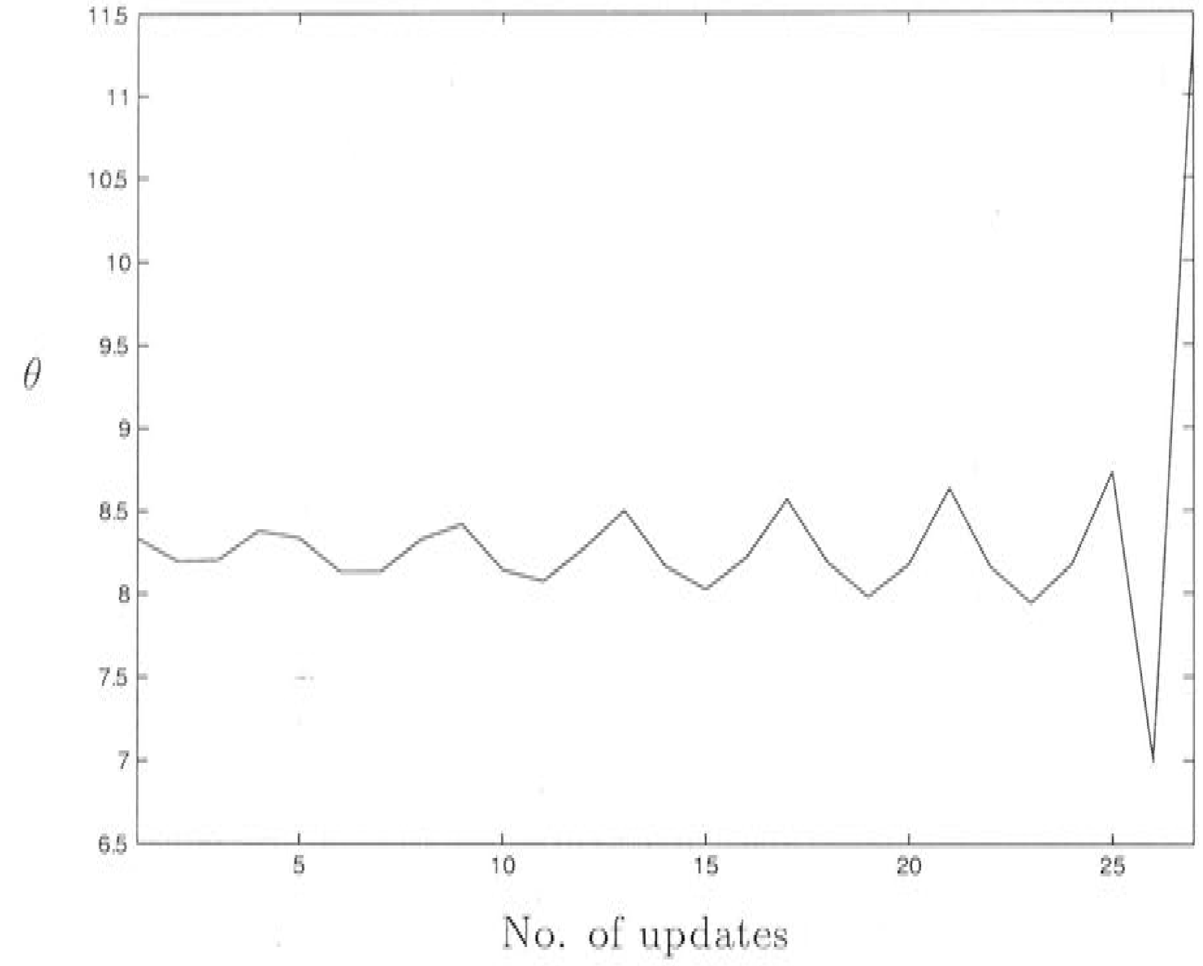


Figure 11.3: Successive smoothing paramater updates on grid no. 4, generated by the MINGCV algorithm, for the data set sine.dat.

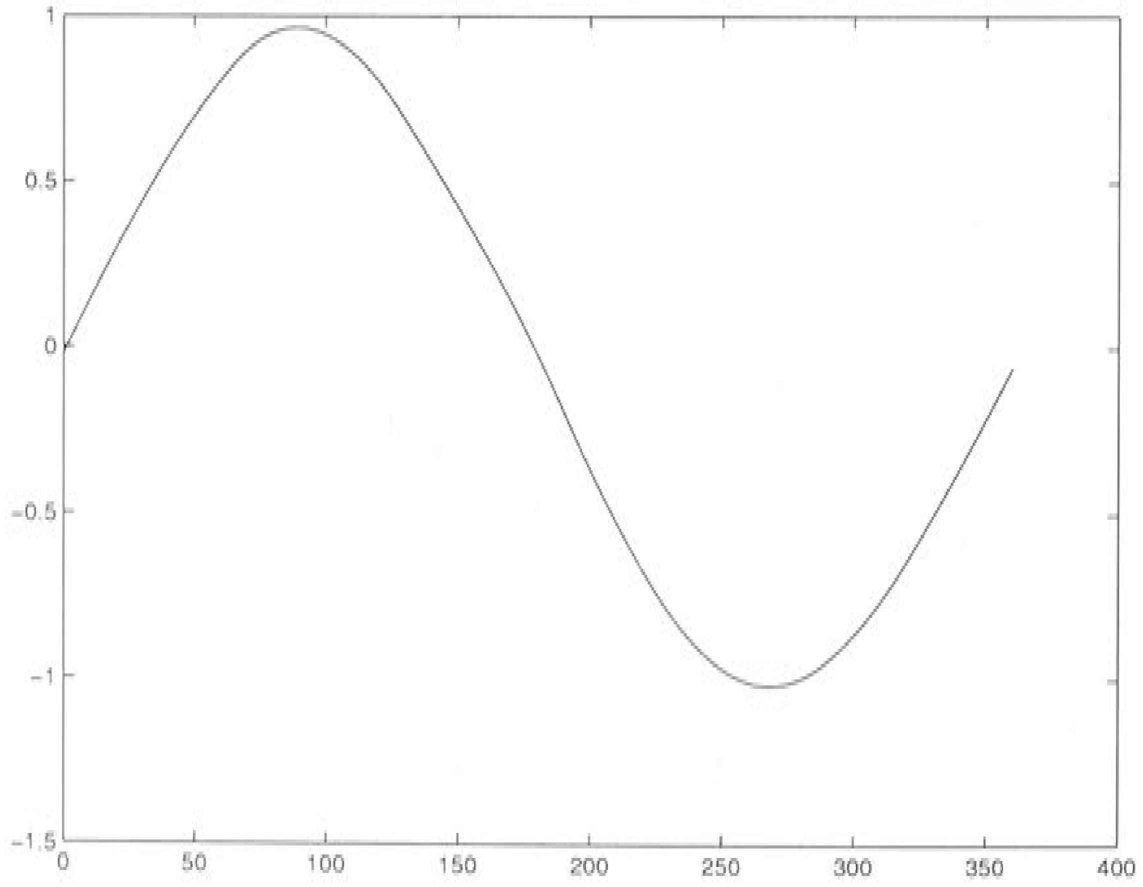


Figure 11.4: Quadratic B-spline approximation to the smoothing spline solution on grid 6, generated by the MINGCV algorithm for the data set sine.dat.

this improvement suggests that stochastic error in the trace estimate was not responsible for the erroneous GCV structure observed for the MINGCV algorithm with piecewise constant discretisation, discussed in section 10.2.

To test this further, ANUSPLIN was used to calculate $Tr(A) = \mathbf{t}\hat{\mathbf{t}}$ analytically for a sample of 9 random vectors and 7 prescribed smoothing parameters, where $\hat{\mathbf{t}}$ is the vector of fitted values corresponding to the analytic spline fit to the data vector $\mathbf{t}$. This allowed the calculation of the stochastic GCV estimate for each random vector in the absence of discretisation error or error due to the numerical solution not having converged exactly. These results are shown in Table 11.3. Although there are considerable differences in the estimates of the GCV for each random vector, the GCV structure seems consistent across the random vectors. The GCV always increases monotonically either side of a single minimum, even if the minimum corresponds to a smoothing parameter much smaller than the analytic value. It therefore seems that, while stochastic error in the Tr estimate does shift the position of the GCV curve in Figure 8.1 considerably, it does not cause the structure to behave in the anomalous way observed in Table 10.6, 10.8 and 10.10, which feature multiple minima. Instead, it is concluded that discretisation error was largely responsible for the errors observed for the piecewise constant discretisation. The higher continuity of quadratic elements gives accuracy to the solution and the corresponding statistical characteristics that piecewise constant discretisation does not allow.

λ	GCV					
	l= 6	l= 5	l= 4	l=3	l=2	l=1
1000	0.0314	0.0314	0.0314	0.0313	0.0316	0.0312
2000	**.0312**	.0311	.0309	.0309	.0309	.0309
4000	.0313	**.0309**	**.0308**	**.0308**	**.0308**	**.0308**
5000	.0313	**.0309**	**.0308**	**.0308**	**.0308**	**.0308**
8000	.0315	.0310	.0309	.0309	.0309	.0309
25500	.0318	.0315	.0314	.0314	.0314	.0314
40000	.0321	.0318	.0317	.0317	.0317	.0317

Table 11.2: GCV values for the quadratic B-spline approximation to the smoothing spline on each grid, for different smoothing parameters. Local minima are emphasised.

λ	GCV								
	$\mathbf{t}_1$	$\mathbf{t}_2$	$\mathbf{t}_3$	$\mathbf{t}_4$	$\mathbf{t}_5$	$\mathbf{t}_6$	$\mathbf{t}_7$	$\mathbf{t}_8$	$\mathbf{t}_9$
5000	0.0358	0.0344	0.0389	0.0319	0.0323	0.0361	0.0320	0.0336	0.0320
10000	0.0352	0.0342	0.0376	**0.0318**	0.0322	0.0352	0.0320	0.0337	0.0320
20000	0.0349	0.0339	0.0365	**0.0318**	**0.0321**	0.0343	0.0322	0.0338	0.0322
25500	0.0349	**0.0338**	0.0362	0.0319	**0.0321**	0.0342	0.0323	0.0339	0.0323
40000	**0.0348**	**0.0338**	0.0357	0.0319	**0.0321**	**0.0339**	0.0325	0.0339	0.0325
60000	0.0351	0.0339	**0.0354**	0.0323	0.0324	0.0340	0.0329	0.0341	0.0329
100000	0.0356	0.0344	0.0356	0.0329	0.0331	0.0345	0.0330	0.0347	0.0337

Table 11.3: Stochastic GCV estimates calculated using analytic thin plate splines, for 9 different random vectors **t** and for different smoothing parameters.

11.1 Stochastic error in the trace estimate

While the results in Tables 11.1 and section B.1 are more stable than those in Chapter 10, they are still susceptible to disruption due to stochastic error in the Tr estimate. The two different random vectors **t** used in Chapter 11 were again used to investigate the effect of stochastic error on the performance of the MINGCV algorithm. Table 11.4 and section B.2 shows that a different random vector causes the θ estimate to be considerably different. These estimates are much closer to the analytic optimum of 10.1464 than the estimates in Table 11.1, indicating that this particular random vector is likely to be giving a more accurate trace estimate. The results are consistent with the GCV structure shown in Table 11.5. The presence of a single minimum is evident on all grids, at approximately $\lambda = 60000$. This agrees well with the value reached by the minimum GCV algorithm.

Table 11.6 and section B.3 shows the results for yet another random vector. The θ estimate is very stable on the coarse grid, and close to the optimum. The estimate becomes more unstable as the grids get finer, eventually becoming

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6	5	10.95	2.98	10.6	0.0369
5	13	10.92	2.95	11.1	0.0368
4		non	convergence		
3		non	convergence		
2		non	convergence		
1		non	convergence		

Table 11.4: Results generated by the MINGCV algorithm, for a second random vector **t**.

λ	GCV					
	l= 6	l= 5	l= 4	l=3	l=2	l=1
1000	0.0411	0.0392	0.0392	0.0392	0.0393	0.0407
5000	0.0392	0.0383	0.0383	0.0383	0.0384	0.0388
10000	0.0383	0.0378	0.0378	0.0378	0.0378	0.0380
25500	0.0373	0.0371	0.0371	0.0371	0.0371	0.0372
40000	0.0370	**0.0368**	**0.0368**	**0.0368**	0.0369	0.0369
60000	**0.0369**	**0.0368**	**0.0368**	**0.0368**	**0.0368**	**0.0368**
80000	0.0370	0.0369	0.0369	0.0369	0.0369	0.0370
100000	0.0373	0.0372	0.0372	0.0372	0.0372	0.0372

Table 11.5: GCV values for the quadratic B-spline approximation to the smoothing spline on each grid, for different prescribed smoothing parameters, for a second random vector **t**. Local minima are emphasised.

too small and causing $d^2GCV/d\theta^2$ to become negative. This could possibly be corrected using the dampening procedure described in section 10.2.2. However, the GCV structure in Table 11.7 shows that the problem is likely to be similar to that identified in section 10.2.1. There is no local minima on grids 5-2. This analysis demonstrates that the stochastic error in the trace estimate has an enormous impact on the performance of the MINGCV algorithm for small data sets. In this case the stochastic error has distorted the solution to the point of exact interpolation.

In order to resolve this fundamental difficulty, the algorithm was modified to calculate 10 different random vectors **t** and work out the vectors **b** corresponding to each random vector. Estimates of Tr, $dTr/d\theta$ and $d^2Tr/d\theta$ were then calculated for each **t**. The averages of these estimates across all random vectors were used in the calculation of the GCV and its derivatives. As discussed in

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6	3	10.14	2.92	11.5	0.0368
5	13	8.94	2.74	13.6	0.0363
4		non	convergence		
3		non	convergence		
2		non	convergence		
1		non	convergence		

Table 11.6: Results generated by the MINGCV algorithm for a third random vector **t**.

λ	GCV					
	l= 6	l= 5	l= 4	l=3	l=2	l=1
1000	0.0378	0.0353	0.0353	0.0353	0.0355	0.0369
5000	0.0371	0.0358	0.0358	0.0359	0.0359	**0.0363**
10000	0.0369	0.0360	0.0360	0.0360	0.0361	0.0363
25500	**0.0368**	0.0363	0.0363	0.0363	0.0363	0.0364
40000	**0.0368**	0.0365	0.0365	0.0365	0.0365	0.0366
60000	0.0370	0.0368	0.0368	0.0368	0.0368	0.0368
80000	0.0374	0.0371	0.0371	0.0371	0.0371	0.0372
100000	0.0378	0.0376	0.0376	0.0376	0.0376	0.0376

Table 11.7: GCV values for the quadratic B-spline approximation to the smoothing spline on each grid, for different prescribed smoothing parameters, for a third random vector **t**. Local minima are emphasised.

section 10.2.1, this reduces the upper bound on the variance in the Tr estimate by a factor of 10. The results are shown in Table 11.8 and section B.4.

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6	4	10.38	2.93	5.8	0.0327
5	5	10.11	2.86	6.7	0.0325
4	10	9.88	2.86	7.3	0.0324
3	8	9.87	2.80	7.5	0.0324
2	8	9.86	2.80	7.5	0.0324
1	1	9.86	2.80	7.5	0.0324

Table 11.8: Results generated by the MINGCV algorithm, using the average of 10 different random vectors to calculate stochastic estimates of the signal and the GCV.

Convergence occurs on all grids, and the numbers are close to the analytic optimum. The improvement in stability achieved by reducing the stochastic error in this way indicates that stochastic error in the Tr estimate does considerably hamper the performance of the MINGCV algorithm. While the above procedure clearly reduces the computation efficiency of the algorithm considerably, the minimum GCV algorithm was originally designed for large data sets, which reduces the standard error in Tr to at most $(2/n)^{1/2}$. It was therefore anticipated that measures such as the above would not be necessary for very large data sets.

11.2 A first order correction

The oscillatory behaviour of the θ estimate suggested a form of correction to help the solution estimate keep up with the θ updates. The first derivative estimate, $d\mathbf{u}/d\theta$, can be used to correct the solution after the calculation of a θ update, using

$$\mathbf{u}(\theta_{q+1}) = \mathbf{u}(\theta_q) + \frac{d\mathbf{u}}{d\theta}(\theta_q)(\theta_{q+1} - \theta_q) \tag{11.1}$$

The results of adding this correction after each update, using only the original random vector $\mathbf{t}$, are shown in Table 11.9 and section B.5. The performance of the algorithm has improved considerably compared to that demonstrated in Table 11.1.

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6	5	7.25	2.78	5.1	0.0312
5	4	8.33	2.68	7.3	0.0309
4	7	8.24	2.62	8.3	0.0308
3	13	8.30	2.61	8.3	0.0308
2	1	8.30	2.61	8.4	0.0308
1	1	8.30	2.61	8.4	0.0308

Table 11.9: Results generated by the MINGCV algorithm, applying a first order correction to the solution after each θ update.

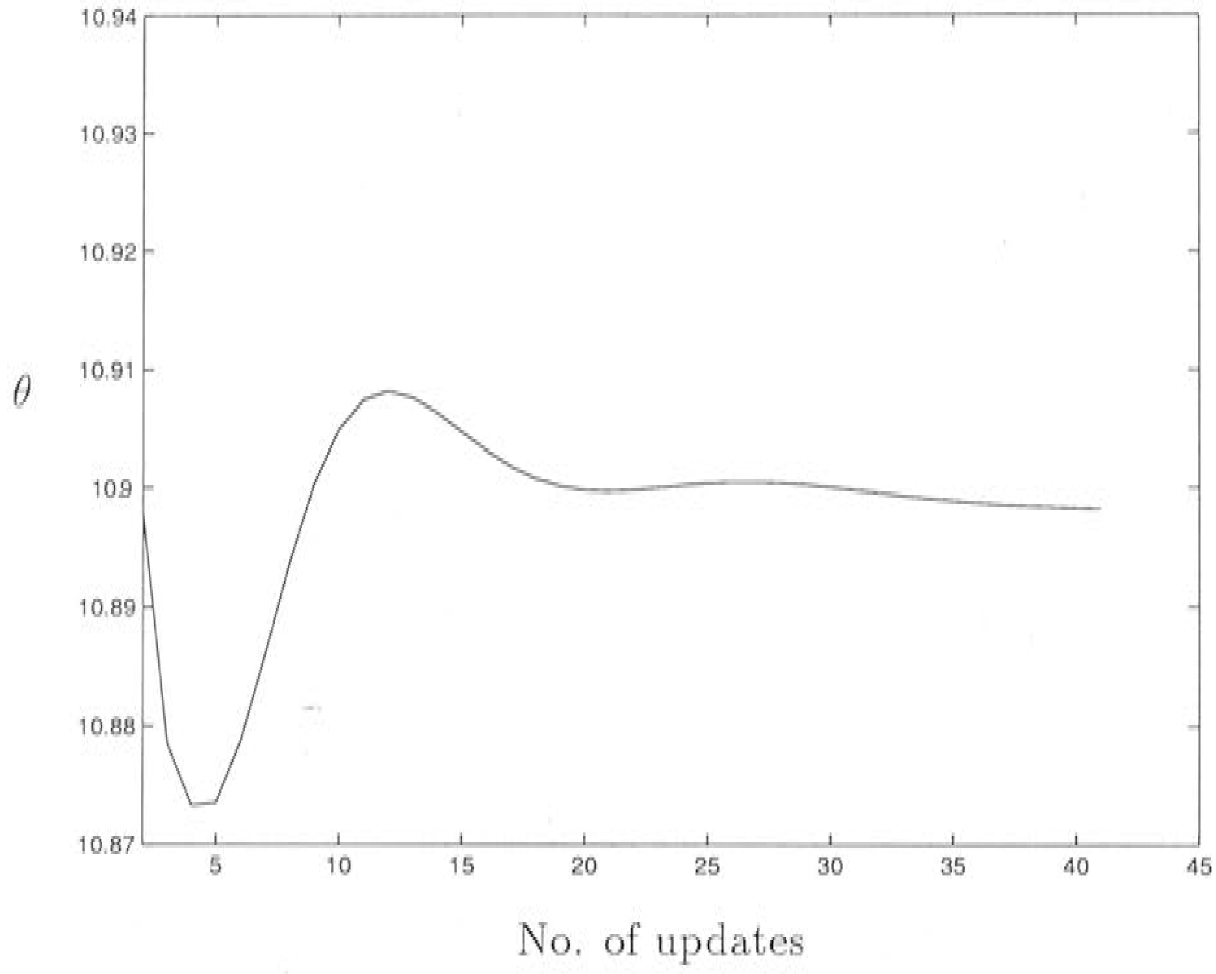


Figure 11.5: Successive smoothing parameter updates on grid no. 3, using a first order correction to the solution estimate following each θ update, for a second random vector.

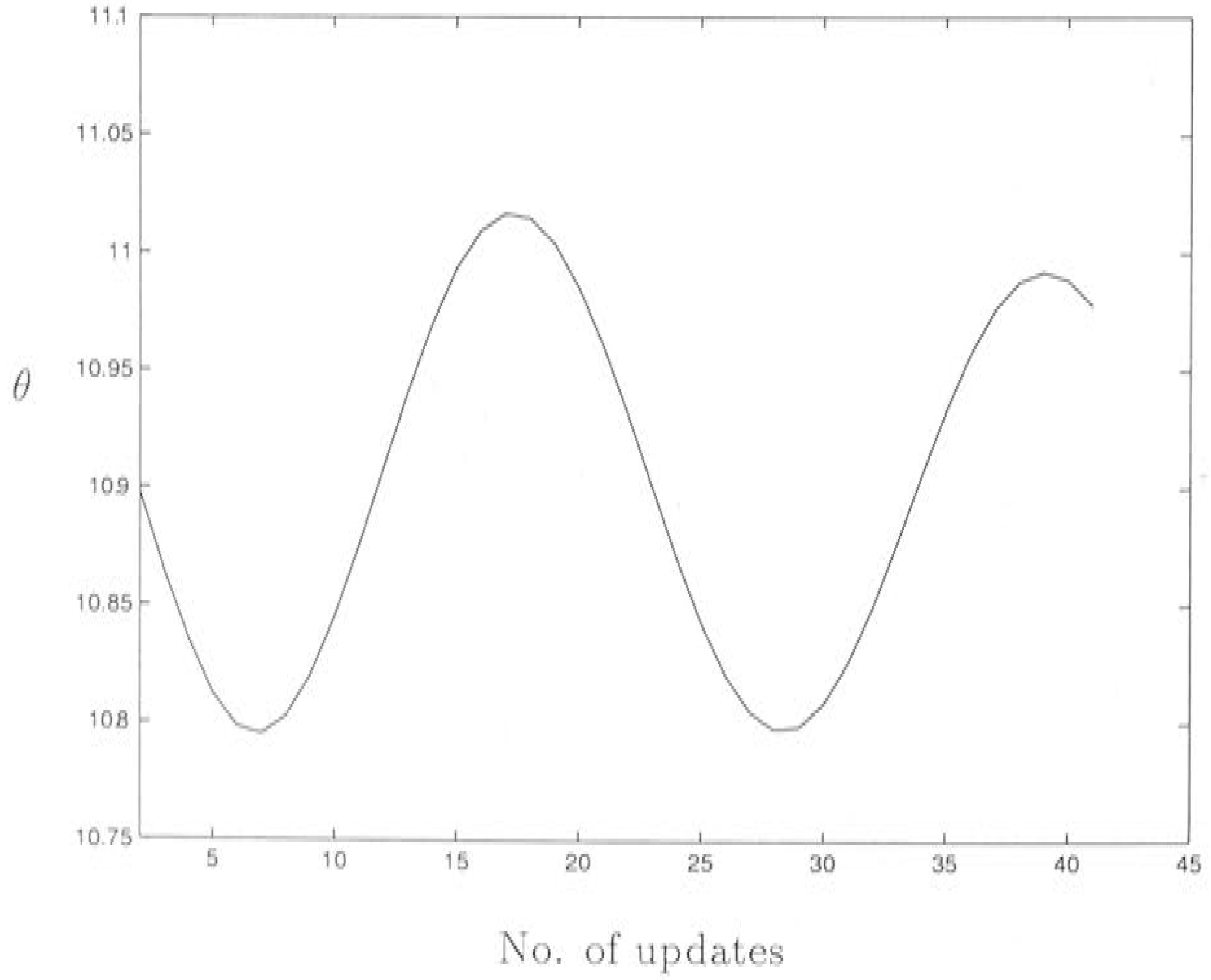


Figure 11.6: Successive smoothing parameter updates on grid no. 3, without using the first order correction, for the second random vector.

Convergence to 4 decimal places occurs on grids 6, 5 and 4, and convergence to 2 decimal places occurs on all grids. Thus the correction has allowed convergence on 4 additional grids in comparison to the results before applying the correction, from Table 11.1.

The correction was also tried using the other 2 random vectors $\mathbf{t}$, shown in section B.6 and B.7. In section B.6, the improvement was also significant, with convergence on all grids. Figures 11.5 and 11.6 show that the correction has removed oscillatory behaviour on grid number 3. In section B.7, $d^2GCV/d\theta^2$ still became negative on the grid number 4, a problem associated with stochastic error not poor convergence. However, convergence on the grid number 5 is much faster, and gets to four decimal places accuracy in 7 updates. This was not achieved in 20 updates without the correction.

The correction has been very helpful because it has helped overcome one of the main problems in the design of the MINGCV algorithm and the OPTRSS algorithm. It reduces the extent to which the solution must be re-estimated when a new smoothing parameter is specified. If large changes to the solution are required in response to the updates, the implementation of the multigrid principle is disrupted because the updates introduce error into the solution estimate that may not have a frequency suitable to the current grid.

11.3 Different data sets

The MINGCV algorithm was run on a data set with a lower smoothing parameter, to see to what extent the above characteristics were particular to the data set sine.dat. The data set bumpy.dat, described in section 9.3, was used. The results are shown in Table 11.10 and section B.8. The value of $d^2GCV/d\theta^2$ quickly became negative on the coarse grid, because there were not enough coefficients to make the solution sufficiently complex. This problem is resolved on the grid number 5, and convergence to four decimal places is achieved quickly, on both grid numbers 4 and 5. On grid number 3 the estimate oscillates, but Figure 11.7 shows that the behaviour is convergent. On the two finest grids the estimate is again very slow to change, and appears to be converging on grid 2.

The pattern in the above results is similar to that observed for sine.dat, though

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6		non	convergence		
5	5	5.94	15.19	12.7	0.197
4	8	6.66	15.09	13.3	0.198
3		non	convergence		
2		non	convergence		
1		non	convergence		

Table 11.10: Results generated by the MINGCV algorithm, for the data set bumpy.dat.

it appears to have shifted to finer grids, as would be expected. Convergence is faster on the middle grids, but the coarsest grid is too coarse to allow optimisation. The finest grids are still finer than is necessary to explain the true variation in the data.

Applying the correction in equation (11.1) to this analysis gave a marked improvement, as shown in Table 11.11 and section B.9. Convergence is faster on the coarser grids. On grid 3, Figure 11.8 shows that the correction has removed the oscillatory behaviour from the θ estimate, speeding up convergence considerably. On the two finest grids, the estimates do not change significantly. The success of the correction indicates that the estimate of $d\mathbf{u}/d\theta$ is itself quite accurate, in that it can provide a helpful correction to the current solution estimate, particularly on the fine resolution grids.

Grid no.	No. of updates	Converged θ value	Converged R value	Converged signal value	Converged GCV value
6		non	convergence		
5	4	5.94	15.19	12.8	0.197
4	6	6.66	15.09	13.3	0.198
3	10	6.92	15.32	13.9	0.200
2	15	6.94	15.32	13	0.200
1	1	6.94	15.30	13.1	0.200

Table 11.11: Results generated by the MINGCV algorithm, for the data set bumpy.dat, applying the first order correction to the solution after each θ update.

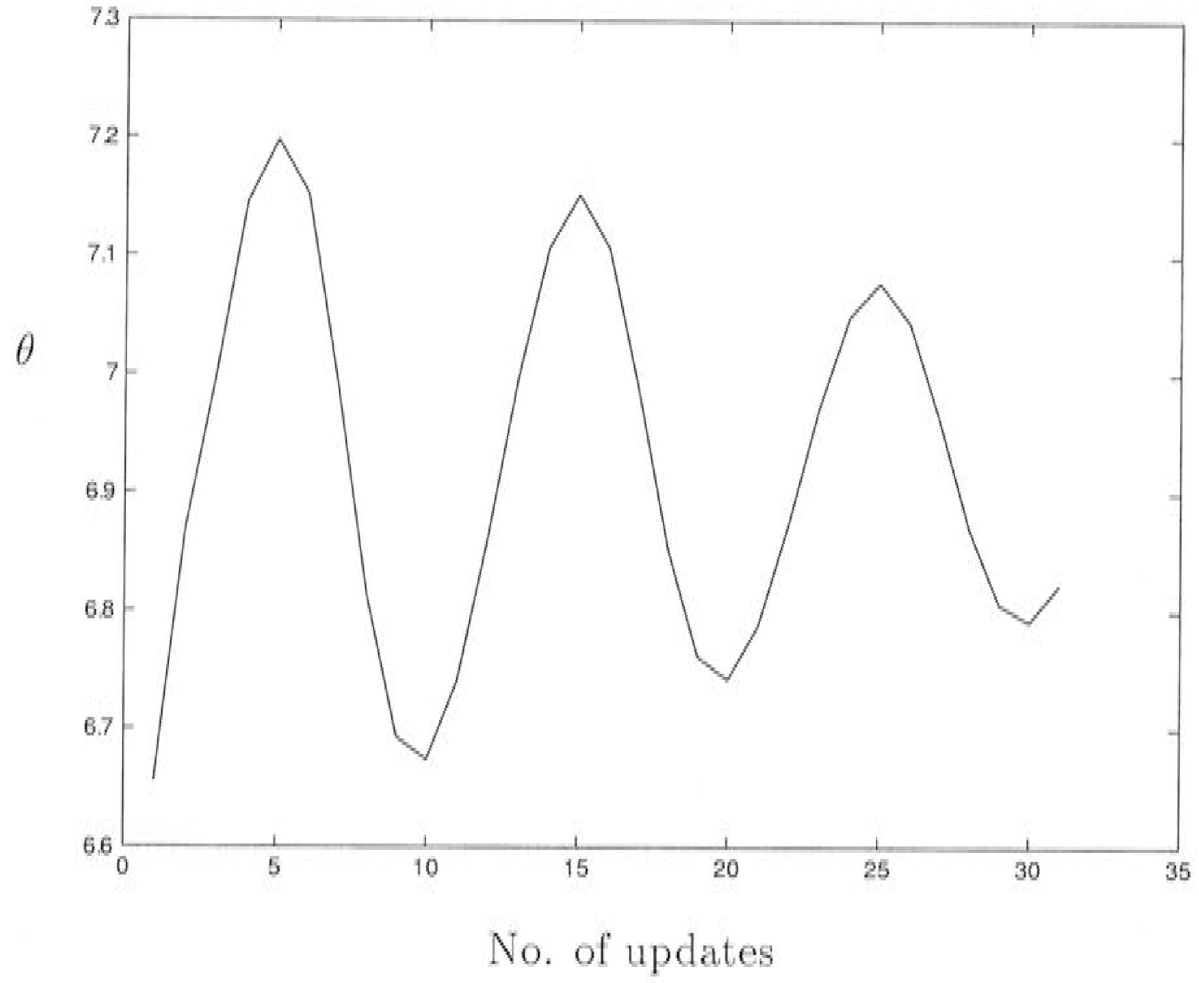


Figure 11.7: Successive smoothing parameter updates for grid 3, for the data set bumpy.dat, without using the first order correction.

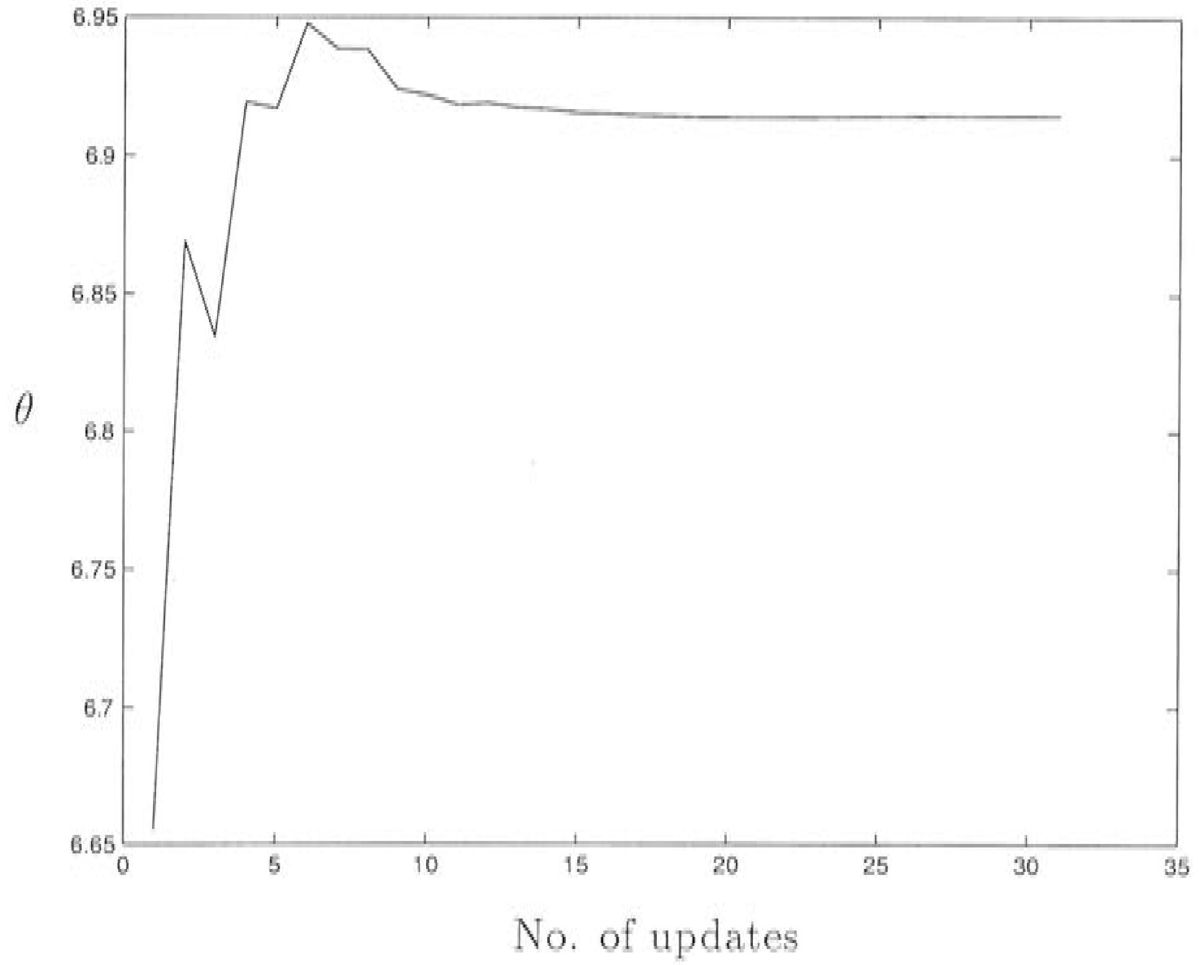


Figure 11.8: Successive smoothing parameter updates for grid 3, for the data set bumpy.dat, using the first order correction.

11.4 The convergence criteria

The restriction procedure described in section 5.3 was applied to the above data sets to produce a measure of how the solution changes between the grids. This technique was used to determine when the resolution was fine enough to represent accurately represent the spline solution. The value of $D = \|g_l\|/\|f_l\|$, measuring orthogonal distance between the fine grid solution and the restricted coarse grid solution, is shown for consecutive grids in Table 11.12. Clearly the differences between solutions before and after restriction are much smaller for the smooth data set, sine.dat, than for the more complex data set, bumpy.dat. The tolerance on D, denoted in the MINGCV algorithm stated in section 8.2 as $tol1$, was calculated by choosing the value below which no visible change could be seen in the solution estimate before and after restriction to a coarse resolution. For D values below this tolerance, it was considered inappropriate to continue refinement. For the data set sine.dat, no change could be seen between any of the grids, setting the cutoff on D to at least 0.5×10^{-4}. For bumpy.dat, changes in the solution estimate could be seen until the transition between grid 4 and grid 3, which corresponds to a D value of approximately 0.7×10^{-4}. This is of the same order of magnitude as 0.5×10^{-4}, implying that the coarsest grid is not too fine for the data set sine.dat. This is consistent with the fact that a grid coarser than approximately 8 grid points is too coarse to deliver the optimal signal value of 8.2. Based on these observations the value of $tol1$ was set conservatively to 0.5×10^{-4}.

data set	$D \times 10^4$				
	l= 5	l= 4	l= 3	l=2	l=1
sine.dat	0.455	0.113	0.000656	0.000000	0.000000
bumpy.dat	222	6.73	0.725	0.00902	0.000000

Table 11.12: D values for different grids, for the data sets sine.dat and bumpy.dat.

11.5 Improving the efficiency

The above analysis served the purposes of verifying that the MINGCV algorithm was working correctly, understanding the procedures involved, and determining the best approach to carrying out those procedures. The algorithm was then re-examined with the aim of further refinement to increase its efficiency. Some helpful modifications are discussed below.

11.5.1 Differentiation of Tr with respect to λ

It is demonstrated in Chapter 8 that the MINGCV algorithm is more efficient if equations (8.36) and (8.37) are used to differentiate Tr instead of equations (8.26) and (8.28). Once this option was realised, it was incorporated into the algorithm. Comparison of results in the tables of ?? shows that identical results were achieved for both systems. This is a validation that the procedures are correct.

11.5.2 Finite difference estimation of $\frac{d^2GCV}{d\theta^2}$

If the expensive calculation of $d^2GCV/d\theta^2$ could be replaced with a finite difference estimate, the efficiency of the MINGCV algorithm would almost double. Given that calculation of $d^2R/d\theta^2$ and $d^2Tr/d\theta^2$ would not be required, it would only be necessary to iteratively solve 3 systems instead of 5, assuming the derivatives of Tr were calculated using the method refered to in section 8.2.3. The calculation of $d^2GCV/d\theta^2$ using the formula in equation (8.17) was therefore replaced by the finite difference calculation

$$\frac{dGCV(\theta_{q+1})/d\theta - dGCV(\theta_q)/d\theta}{\theta_{q+1} - \theta_q} \tag{11.2}$$

The results for both the data sets sine.dat and bumpy.dat are shown in sections B.11 and B.12. To avoid numerical errors, a new estimate of $d^2GCV/d\theta^2$ was only calculated if the change in θ was greater than 0.1. As discussed in Chapter 8, the convergence criteria that prevent further updates of θ on a given grid and prevent further refinement to finer grids are now emplaced. According to the analysis in section 11.4, $tol1$ was set to 0.5×10^{-4}. The value of $tol2$, the tolerance on Q defined in section 8.1, was set to 0.01.

The finite difference estimates show good agreement with the estimates calculated using equation (8.17). For sine.dat, $tol1$ ensured that refinement did not proceed past grid 5, as would be expected from the results described in section 11.4. The MINGCV algorithm was therefore very fast for this data set, because it is extremely smooth. More iterations were required on the coarsest grid because no initial information was available. The data set bumpy.dat required an extra grid to represent the additional complexity, refining to grid number 4. Convergence was slightly slower on this grid, but was still much more efficient than the convergence patterns for the poorly conditioned systems observed in previous analyses. Overall the sets of results in sections B.11 and B.12 represent an efficient and direct path to an optimal solution on a resolution suitable to the complexity of the data set.

11.6 A modified, improved algorithm

As a result of the above analysis, it is recommended that the MINGCV algorithm, stated in section 8.2, be implemented with the following specifications.

- A quadratic B-spline is used as the discrete approximation to the natural cubic spline minimiser $f_\lambda(x)$ of equation (2.9).

- The basis element prolongation operator described in Chapter 5 is used to transfer the solution estimate to finer grids.

- Tr is differented by the method described in section 8.2.3, which reduces the number of systems that require iterative solution.

- The value of $d^2GCV/d\theta^2$ is estimated by finite differences, according to the process recommended by the results in section 11.5.2, rather than by estimation using equation (8.17).

- A first derivative correction to the solution estimate $\mathbf{u}$, described in section 11.2, is added to the solution estimate after each θ update.

The algorithm in section 8.2 can now be rewritten as shown on the next page. Note that we now require only one derivative of $\mathbf{u}_l$ with respect to the smoothing parameter, and no derivatives of $\mathbf{b}_l$.

```
while $D > tol1$
    $q = 0$
    while $Q > tol2$
        for $m = 0$ to 1
            $\mathbf{u}_l^{(m)}(\theta_q) = S_l{}^{v_1}(\mathbf{u}_l^{(m)}(\theta_q), \mathbf{v}_l^m)$
        end
        $\mathbf{b}_l(\theta_q) = S_l{}^{v_1}(\mathbf{b}_l(\theta_q), \mathbf{w}_l^0)$
        $\theta_{q+1} = -\frac{b}{2c}$
        $\mathbf{u}(\theta_{q+1}) = \mathbf{u}(\theta_q) + \mathbf{u}'(\theta_q)(\theta_{q+1} - \theta_q)$
        $q = q + 1$
    end
    for $m = 0$ to 1
        $\mathbf{u}_{l-1}^{(m)} = T_l\mathbf{u}_l^{(m)}$
    end
    $\mathbf{b}_{l-1} = T_l\mathbf{b}_l$
    end
    $l = l - 1$
end
$q = 0$
while $Q > tol2$
    for $m = 0$ to 1
        $\mathbf{u}_l^{(m)}(\theta_q) = S_1{}^{v_l}(\mathbf{u}_l^{(m)}(\theta_q), \mathbf{v}_l^m)$
    end
    $\mathbf{b}_l(\theta_q) = S_l{}^{v_1}(\mathbf{b}_l(\theta_q), \mathbf{w}_l^0)$
    $\theta_{q+1} = -\frac{b}{2c}$
    $\mathbf{u}(\theta_{q+1}) = \mathbf{u}(\theta_q) + \mathbf{u}'(\theta_q)(\theta_{q+1} - \theta_q)$
    $q = q + 1$
end
```

11.7 Conclusion

Chapters 10 and 11 have documented the process of developing an algorithm for calculating minimum GCV finite element univariate smoothing splines, denoted the MINGCV algorithm. In the initial phases of construction of the algorithm, the smoothing spline solution was approximated using a piecewise constant discretisation. Using this discretisation, the MINGCV algorithm converged to an accurate estimate of the analytic minimum GCV solution on grid resolutions where the smoothing spline system was well conditioned. On those resolutions where the system was poorly conditioned, the algorithm sometimes diverged. However, these grid resolutions were usually finer than necessary to represent both broadscale and finescale trends in the solution. For the piecewise constant smoothing spline system there were further difficulties associated with multiple minima in the GCV as a function of the smoothing parameter. This erroneous feature has been determined to be due to discretisation error.

Due to these problems with piecewise constant discretisation, a quadratic B-spline discretisation was used in the later stages of algorithm development. This stabilised the behaviour of the algorithm considerably on all grids. Stochastic GCV structures associated with this discretisation also featured single minima. The quadratic B-spline discretisation is clearly better suited to approximating smoothing splines, due to the higher order continuity allowing accurate approximation of smooth functions on coarse grids. Further improvements were achieved by applying a first order correction to the solution estimate after each smoothing parameter update. This allowed the solution estimate to respond more quickly to updates to the smoothing parameter. Using this correction, MINGCV algorithm converged on all grids for both test data sets, even grids which were unnecessarily fine and therefore associated with poorly conditioned systems. The average of multiple Tr estimates was taken to reduce stochastic error, and improve the accuracy of the solution estimate for small data sets.

In the final stages of algorithm development, the efficiency was further improved by replacing the expensive estimation of $d^2GCV/d\theta^2$ with a finite difference estimate. The convergence criteria for ceasing the updating process on a given grid, and for preventing further refinement of the grid resolution, were also emplaced. The resulting algorithm is a robust, efficient method for calculating finite element approximations to minimum GCV smoothing splines at a resolution appropriate for the scale of the data set.

Chapter 12

Minimising GCV for the bivariate quadratic B-spline thin plate smoothing spline system

Using the guidelines developed in Chapter 11, the iterative procedure for calculating minimum GCV bivariate finite element thin plate smoothing splines was constructed, from the methods described in Part 1. The univariate analysis gave an understanding of the behaviour of the individual processes that make up the MINGCV algorithm, and the key factors that influence its performance. This led to the development of design specifications that resulted in fast convergence to an accurate representation of the analytic solution for a range of simulated data sets of varying complexity.

Construction of the univariate MINGCV algorithm hinged upon maintaining a reasonably well-conditioned system of equations. The univariate analysis developed a series of techniques to avoid poor synchronisation of the double iteration, a direct result of a deterioration in conditioning. The bivariate analysis has involved more extensive testing of the algorithm on a larger and more diverse range of data sets, including sparse data sets with irregularly positioned data points. This has led to more refined modifications of the MINGCV procedure. This chapter demonstrates that the resulting algorithm gives an accurate representation of the analytic solution for all test data sets. Certain data set characteristics that may cause poor conditioning are also identified.

This chapter starts by outlining the specifications of the final bivariate MINGCV algorithm. The algorithm is the same as that defined in section 11.6, with the following conditions incorporated.

1. The initial value of the smoothing parameter λ was set so that

$$\frac{\lambda}{h^2} = \frac{1}{10} \tag{12.1}$$

This setting weights the two terms in equation (6.80) approximately equally, given that the diagonal elements of the roughness penalty matrix R have magnitude of around 10. The even weighting prevents the rank deficient term R from dominating the system, avoiding the problem of ill-conditioning, whilst still maintaining a smooth setting so as not to require fine scale structure in the initial phases of the iteration.

2. When iteration begins on a new grid, twice the number of smoothing iterations ($2v_1$) are performed before the smoothing parameter is updated. This is helpful because the right hand side of the equation for $d\mathbf{u}/d\theta$ (equation (8.7)) is dependent on the solution estimate $\mathbf{u}$. The algorithm therefore performs v_1 smoothing iterations to get an estimate of $\mathbf{u}$, and then performs v_1 iterations to update the estimate of $d\mathbf{u}/d\theta$. Without this procedure, the first smoothing parameter update would be performed with no information regarding the solution derivatives.

3. The finite difference estimate of $d^2GCV/d\theta^2$, introduced in section 11.5.2, is only updated if the $\Delta\theta$ used to calculate this derivative is greater than 0.1. This prevents numerical inaccuracies from being exacerbated by division by a very small number. This condition introduces a self correction behaviour into the algorithm, in that if the $d^2GCV/d\theta^2$ estimate is small, θ updates are likely to be large, which can result in a significant update of the $d^2GCV/d\theta^2$ estimate. Thus when the smoothing parameter estimate is not accurate, second derivative estimates will be improved. But when the solution process is close to convergence and θ updates are small, the $d^2GCV/d\theta^2$ estimate will not be changed, and the stable behaviour of the algorithm will not be disrupted.

4. The magnitude of the θ updates is not allowed to exceed 0.5. It was found to be important to avoid large smoothing parameter updates as they often

occur in the early stages of iteration, when the estimates of the solution and its derivatives still contain significant error. A large change in the smoothing parameter can disrupt the synchronisation of the process, and the algorithm will not converge, as discussed in Chapter 11.

5. It is important that the θ estimate does not become too small or too large, because the relationship between the smoothing parameter and the GCV deteriorates in the extremal regions of the GCV curve, as discussed in Chapter 8. To prevent the θ estimate from entering these regions, the following approximate bounds were imposed.

$$\lambda_{max} = 50h^2 \tag{12.2}$$

$$\lambda_{min} = 0.002h^2 \tag{12.3}$$

These bounds have been determined experimentally to be appropriate. They allow λ to differ from the 'even' weighting in equation (12.1) by no more than a factor of 500.

6. If θ reaches the minimum limit, no more iterations are performed and the solution estimate is transferred to a finer grid. Minimal values of θ are a sign that the current basis elements are not flexible enough to represent the optimal solution, and further iteration is likely to be detrimental to the performance of the algorithm. Maximal values indicate that that the θ estimates are diverging away from the optimal solution, at which point the algorithm is terminated.

7. It was considered sensible to emplace a minimum number of θ updates on each grid, to ensure that the solution estimate has a chance to stabilise. The minimum number of updates was set to 5. A maximum number of 16 updates, corresponding to 80 SOR iterations, was also emplaced. This is necessary if convergence on a particular grid is slow and oscillatory. This can sometimes occur for extremely smooth data sets, where the λ value makes little difference to the solution.

8. The restriction procedure, determining whether further refinement should occur, is not performed if the maximum iteration limit is reached or if the θ estimate hits its maximum or minimum limit. In these cases, the solution estimate has not converged on the current grid, so it cannot be assumed to reflect the structure of the optimal solution.

9. If the estimate of $d^2GCV/d\theta^2$ is negative then the previous $d^2GCV/d\theta^2$ estimate is used to update θ. This allows the $dGCV/d\theta$ estimate to continue to improve the θ estimate. If this condition is not imposed, the algorithm is unable to sensibly update θ. Negative estimates of $d^2GCV/d\theta^2$ are usually avoided by the limits in condition 5, but they can still erroneously occur in the early stages of iteration when the estimates of the solution characteristics are still inaccurate.

10. If the $d^2GCV/d\theta^2$ estimate is negative, the magnitude of the θ updates is limited to 0.25. The process must clearly be poorly synchronised if the $d^2GCV/d\theta^2$ estimates are negative, so it is likely to be unable to respond correctly to large θ updates.

11. If there are more than 3 consecutive negative $d^2GCV/d\theta^2$ updates, iteration terminates on the current grid and the resolution is refined. This condition and the two conditions above are necessary to control the difficult situation of negative $d^2GCV/d\theta^2$. When this occurs, the algorithm is unable to update accurately, and the constraints on its behaviour must be made more severe until the process stabilises. This situation usually occurs if the solution is approaching exact interpolation, as discussed in the following analysis. It can also occur if the system is poorly conditioned, which cannot be avoided for some data sets even though the algorithm is designed to maintain good conditioning. This is discussed in Chapter 13.

12. In some cases it was found to be necessary to update the estimate of $d^2GCV/d\theta^2$ even if $\Delta\theta$ was less than 0.1, contrary to condition 3. When the $d^2GCV/d\theta^2$ estimate is quite inaccurate, and also quite large, it can be difficult to update, because a large value of $d^2GCV/d\theta^2$ causes a small θ update. An additional condition was therefore incorporated, specifying that $d^2GCV/d\theta^2$ will be updated if θ has changed by more than 0.2 since the last $d^2GCV/d\theta^2$ update. This is still unlikely to be overly affected by numerical error, as $\Delta\theta$ must still be not significantly smaller than 0.1.

12.1 Results for different test data sets

The bivariate MINGCV algorithm described above was tested on a wide range of data sets, in order to vary the noise and complexity, as well as the distribution of the data point positions. The results below show that the MINGCV algorithm produced good estimates of the analytic spline solution for all data sets. Note that, for the following analyses, estimation of Tr was performed by averaging 10 different random vectors, as was done in section 11.1. This was done to reduce the stochastic error in the Tr estimate, which has been shown in previous chapters to have a major influence on the accuracy of the MINGCV solution estimate, as well as convergence behaviour, for small data sets.

12.1.1 121.dat

The data set 121.dat consists of 121 points on the bivariate function

$$x^2 + y^2 = z \tag{12.4}$$

The positions of the data points were obtained from a uniform random distribution on the region $[-3, 3] \times [-3, 3]$. The data values were perturbed by normally distributed random noise with standard deviation of 0.2, introducing small errors into the data. The data set, along with the analytic thin plate smoothing spline fit obtained from ANUSPLIN, is shown in Figure 12.1. Clearly the process is very simple, smooth and broadscale.

The spline solution is an accurate recreation of the parabolic function, thanks to the good cover of data points and the low noise level. The statistics for the analytic solution are shown in Table 12.1. The signal is less than half the number of data points, indicating a reliable fit, but it is not much less due to the low noise in the data. For the analyses in this chapter, the RMS (root mean square) residual is reported in the summary tables, instead of the residual sum of squares R, in order to show more clearly how closely the data are fitted. The minimum GCV estimate of the standard deviation of the noise discussed in section 2.2.2, denoted $\hat{\sigma}$, is also shown.

Summary results of the MINGCV algorithm for 121.dat are shown in Ta-

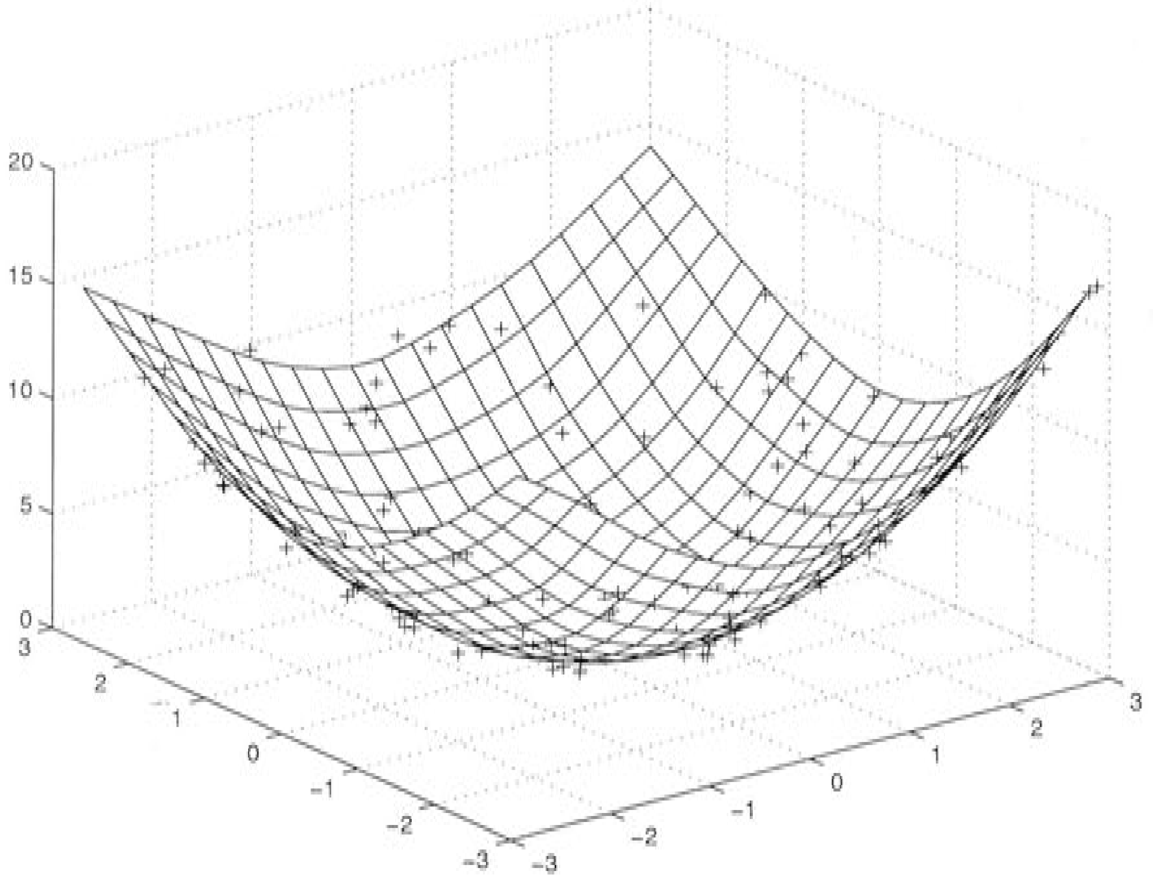


Figure 12.1: Analytic thin plate smoothing spline fit to the data set 121.dat.

λ	Signal	RMS residual	GCV	$\hat{\sigma}$
0.0134	56.6	0.151	0.0805	0.207

Table 12.1: Summary statistics for the analytic thin plate spline fit to 121.dat.

ble 12.2. The algorithm terminates at a relatively coarse grid, giving a solution estimate that is slightly smoother than the analytic solution. However, the estimate of $\hat{\sigma}$ is not greatly different for the two solutions. It therefore appears that both solutions have correctly identified the underlying data generation process. The difference in the signal values between the fine grid MINGCV fit and the analytic spline in this case is likely to be a consequence of choosing a very smooth underlying process with very low noise. In this situation, smoothness can be maintained with a relatively low RMS residual, so the signal value can be higher for a given smoothness.

The results in Table 12.2 do show how the discretisation constraints affect the solution estimate. The smoothing parameter consistently increases as the grid resolution is refined, and the RMS residual decreases. This is a sign that the coarse discretisation forced excessive smoothing, and the solution estimate benefited from the additional flexibility of the finer elements.

Grid no.	h	No. of updates	λ	Signal	RMS residual	GCV	$\hat{\sigma}$
6	1.20	13	0.0109	33.7	0.193	0.0717	0.227
5	0.600	5	0.0278	39.0	0.183	0.0726	0.222
4	0.300	5	0.0281	43.4	0.173	0.0727	0.216

Table 12.2: Results generated by the bivariate MINGCV algorithm for the data set 121.dat.

No. of iterations performed	Sobolev norm
1	40.6
2	20.9
3	25.6
4	22.8
5	18.5
6	16.8
7	15.1
8	14.8

Table 12.3: Sobolev norm values after each iteration for 121.dat, before the first θ update is performed.

The MINGCV solution is shown in Figure 12.2. It appears to be an accurate representation of the analytic solution, and Figures 12.3 and 12.4 show that there is little difference, except in the edge regions. Differences between the analytic solution and the finite element approximation at data sparse regions such as the edge regions can related to the calculation of the roughness penalty for the discretised bivariate smoothing spline equations. The analytic bivariate smoothing spline equations minimise roughness over the infinite plane (see equations (2.11) and (2.12)). This is a source of error in the finite element approximation of the bivariate roughness penalty, which minimises roughness only over the grid surrounding the data points and ignores the surrounding infinite region. This was not an issue for the univariate case because the second derivative of the natural cubic spline reduces to zero beyond the data points. For bivariate splines the second derivative does not reduce to zero beyond the data points. It could be argued in principle that minimisation over all $\mathcal{R}^2$ is not necessary, given that the aim is to best approximate the region in the vicinity of the data points. Nonetheless, we can expect less agreement between the MINGCV solution and the analytic solution for bivariate splines

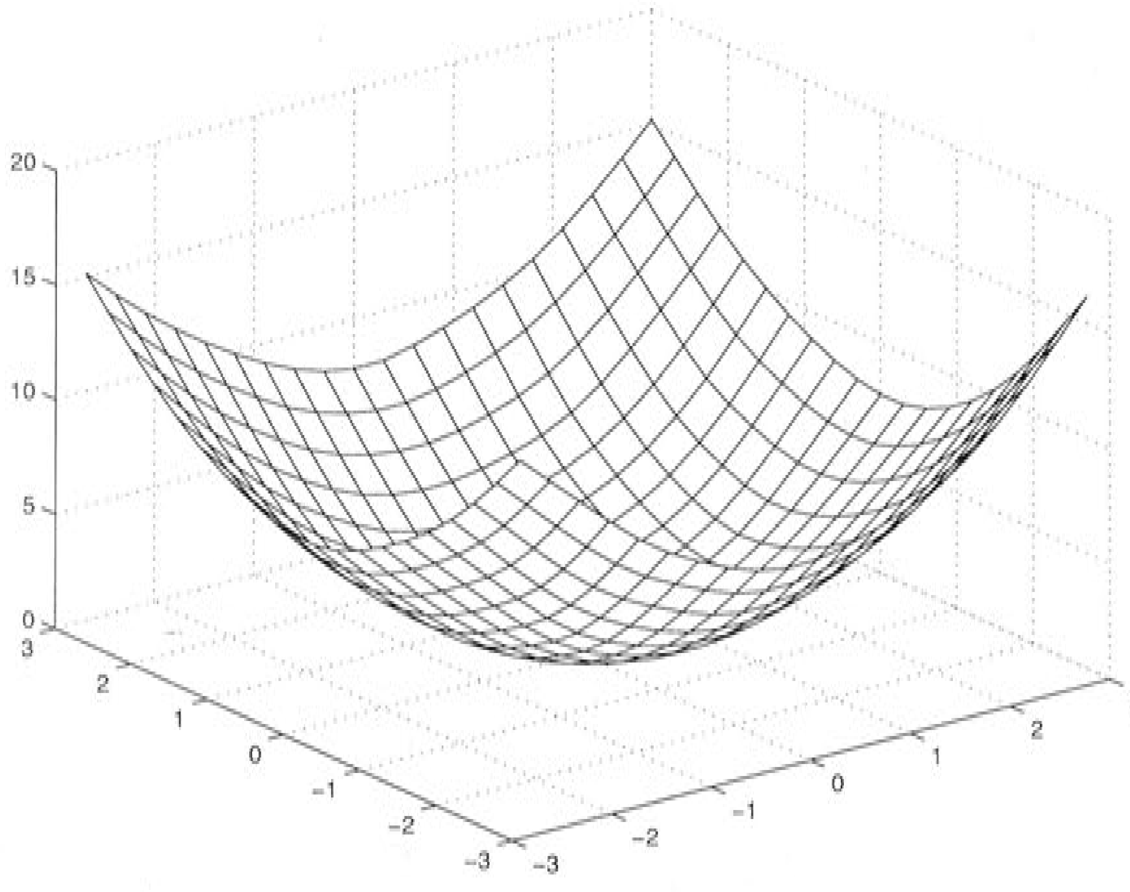


Figure 12.2: Biquadratic B-spline fit to the data set 121.dat.

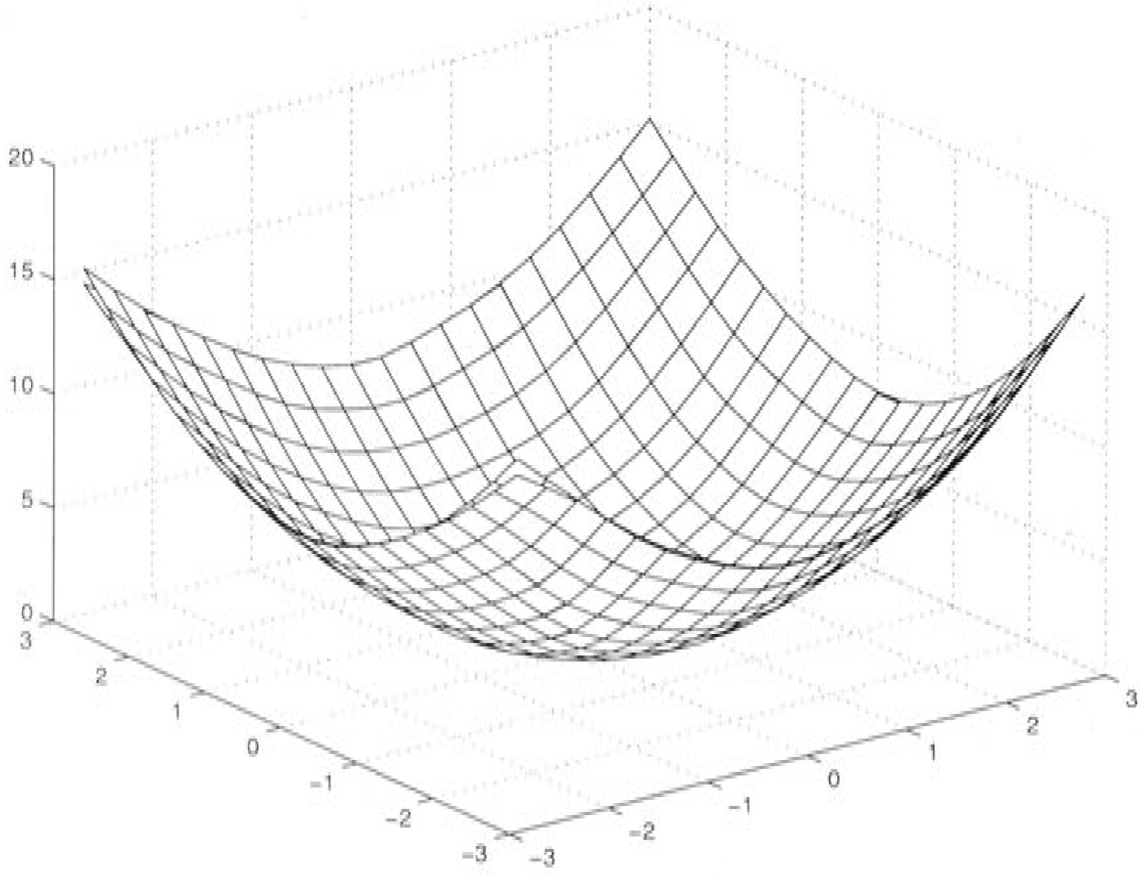


Figure 12.3: Overlay of the biquadratic B-spline solution and the analytic solution for the data set 121.dat.

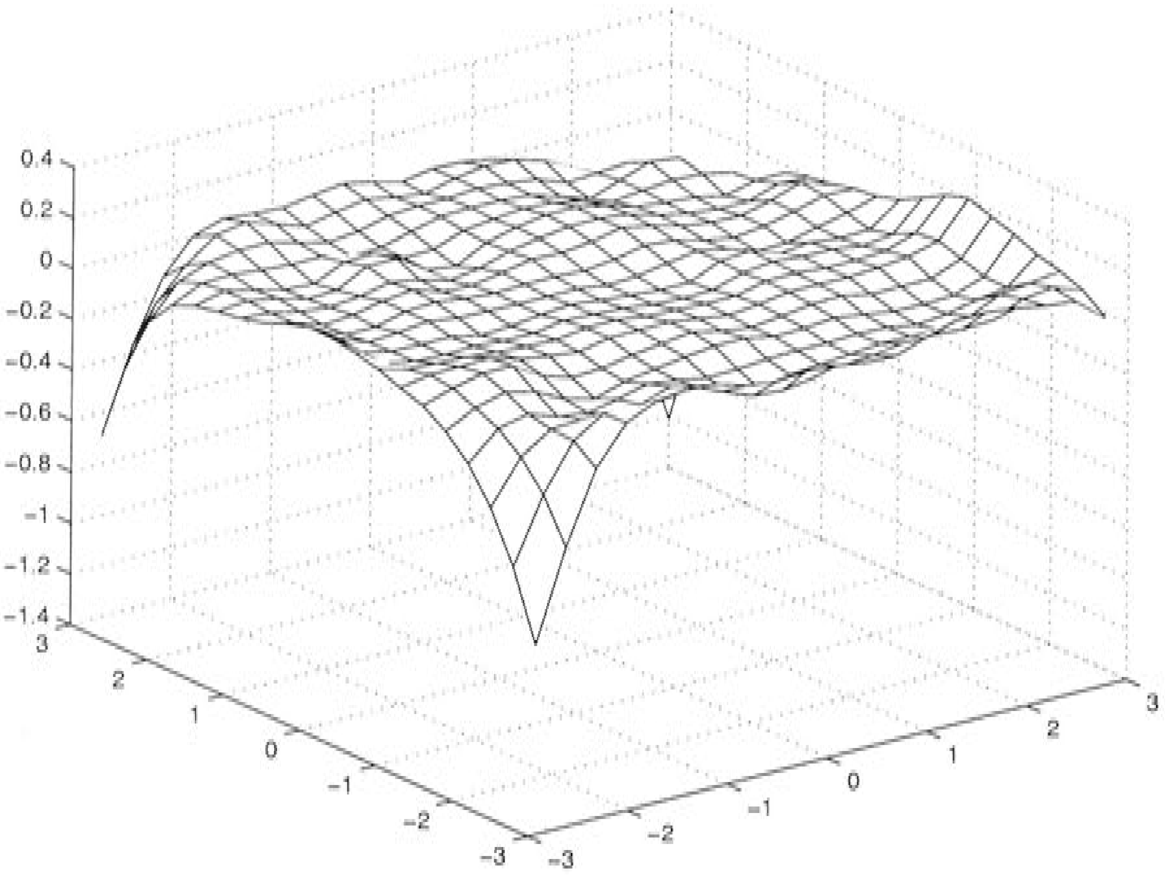


Figure 12.4: Difference between the biquadratic B-spline solution and the analytic solution for the data set 121.dat.

than univariate splines.

Furthermore, the localised minimisation of roughness for the bivariate MINGCV algorithm contributes to 'local poor conditioning' in data sparse regions and edge regions. If there is a relatively large patch with no data points, the roughness penalty term will dominate the system, and the MINGCV algorithm must effectively solve the biharmonic equation on that patch, with boundary information from the rest of the surface surrounding the patch. This was discussed in section 6.3. The biharmonic system is poorly conditioned on fine grids, and cannot be solved effectively by basic iteration, but the MINGCV algorithm is designed so that smooth components are mostly solved before fine grids are reached. However, on the edges the discretised minimum GCV solution is more likely to change with refinement because there is no condition on how the surface extends to infinity. The trends that are extrapolated to the edge regions depend on the solution values in the interior, which may change from grid to grid, particularly if there is a lot of fine scale structure in the solution. The MINGCV algorithm cannot solve for these changes to the edge regions on fine grids. The edge regions are therefore likely to contain higher numerical error, especially if there are no data point near the edge regions. This phenomenon will be further demonstrated in Chapter 13.

Further details of the performance of the MINGCV algorithm are shown in Table 12.3 and section C.1. Table 12.3 shows the benefit of extra iteration before updating the smoothing parameter estimate at the start of the first resolution. A measure of the curvature of the solution estimate, given by the Sobolev norm $\sqrt{\boldsymbol{\alpha}^T Z \boldsymbol{\alpha}}$, has been included in the bivariate analysis. This is the square root of the bivariate roughness penalty, as was shown in equation (6.78). The Sobolev norm has proven to be the most stable of the solution descriptors measured in this analysis. It takes around 5 iterations to settle down to a stable value, after which there are no significant further deviations on any grid. It therefore appears that a reasonable solution estimate has been reached at about this point, and further iteration makes only fine scale alterations.

It is beneficial to the performance of the algorithm to have some stabilisation of the solution estimate before updating the smoothing parameter, as the derivative estimates will be more likely to produce an accurate smoothing parameter update. This avoids smoothing parameter estimates that are in the end regions of the GCV curve in Figure 8.1, where the updating procedure is less effective.

On the coarsest grid, Table A.1.1 shows convergence is quite fast, with the $dGCV/d\theta$ estimate being reduced by 4 orders of magnitude in 13 updates. Fast convergence on the coarse grid would be expected given that the system is well conditioned. Some instability is evident at the fifth update, which is a point where the $d^2GCV/d\theta^2$ estimate is updated. The initial $d^2GCV/d\theta^2$ estimate was clearly quite inaccurate, given the final value of 0.004 corresponding to the converged solution estimate. This demonstrates the importance of condition 12. The initial value of $d^2GCV/d\theta^2$ was so large that the change in the θ estimate was unlikely to be greater than 0.1, so the $d^2GCV/d\theta^2$ estimate would have taken a long time to update without a condition considering cumulative changes in θ.

However, the sudden, large change in $d^2GCV/d\theta^2$ caused an unstable situation, where the $d^2GCV/d\theta^2$ estimate was relatively small in comparison to the $dGCV/d\theta$ estimate. This is a case where the criterion 4, preventing θ updates from exceeding 0.5, becomes important. Table A.1.1 shows that there are two updates of 0.5 before the $dGCV/d\theta$ estimates became smaller, the θ updates became smaller, and the algorithm stabilised, eventually converging on the coarse grid. This situation demonstrates the potential instability in the MINGCV algorithm, and the need to incorporate the conditions outlined at the start of this chapter to prevent erroneous deviations from destabilising the algorithm.

On the final two grids, Tables A.1.2 and A.1.3 show that the algorithm converged very quickly, requiring only the minimum possible number of updates. The GCV and the smoothing parameter estimate change very little between grid number 5 and grid number 4, although the RMS residual and the signal do change. This shows that, even when the predictive capacity is the same, the solution characteristics depend on the discretisation.

12.1.2 Franke's principal test function

Four data sets, of varying noise levels, were created from Franke's principal test function [35]. This function is a combination of negative exponential expressions, and is given by

$$0.75e^{-t_1} + 0.75e^{-t_2} + 0.2e^{-t_3} + 0.5e^{-t_4} \tag{12.5}$$

where

$$\begin{align}
t_1 &= ((9x-2)^2 + (9y-2)^2))/4 \tag{12.6}\\
t_2 &= \left(\frac{9x+1}{7}\right)^2 + (9y+1)/10 \tag{12.7}\\
t_3 &= (9x-4)^2 + (9y-7)^2 \tag{12.8}\\
t_4 &= ((9x-7)^2 + (9y-3)^2)/4 \tag{12.9}\\
& \tag{12.10}
\end{align}$$

The function decays from 4 centroids, 3 located in the unit square, as shown in Figure 12.5. Franke's principal test function was also used in the analysis finite element smoothing spline analysis by Hutchinson [67]. One hundred data point positions were selected from a uniform random distribution on the unit square. Four noisy data sets were created by adding Gaussian noise to the sample values, with standard deviations of 1/2, 1/16, 1/128 and 0 respectively.

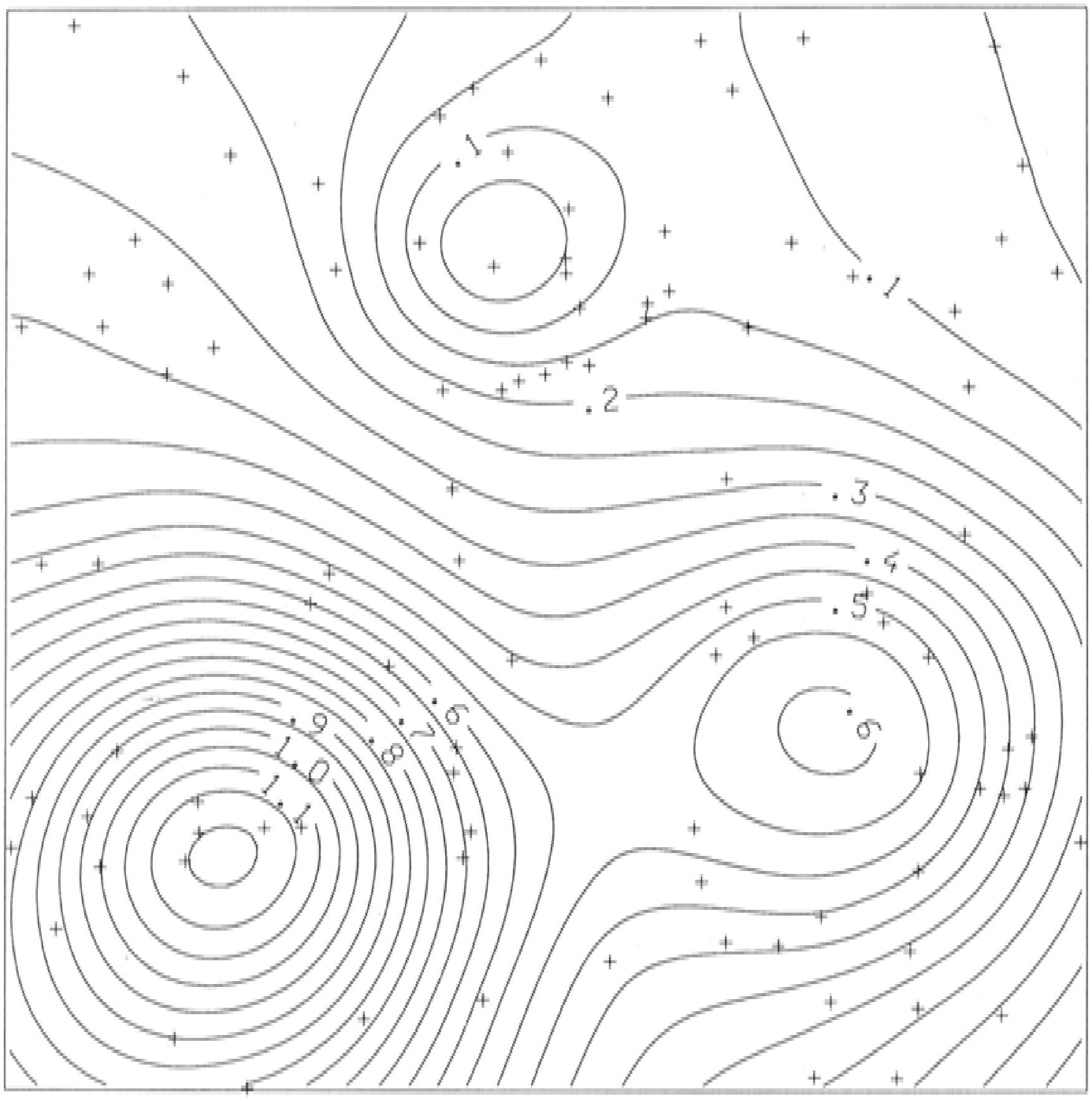


Figure 12.5: Contours of Franke's principal test function, on the unit square.

Franke1.dat

Franke1.dat is the noisiest data set, having a standard deviation of 1/2. The data set, along with the analytic thin plate spline fit, are shown in Figure 12.6. The solution is so smooth as to be almost a flat plane. Summary statistics for the analytic spline are shown in Table 12.4. The standard derivation estimate shows that the fitted function accurately isolates the noise in the data.

λ	Signal	RMS residual	GCV	$\hat{\sigma}$
0.0616	7.2	0.413	0.198	0.428

Table 12.4: Summary results for the analytic thin plate spline fit to franke1.dat.

Results of the MINGCV algorithm for franke1.dat are shown in Table 12.5

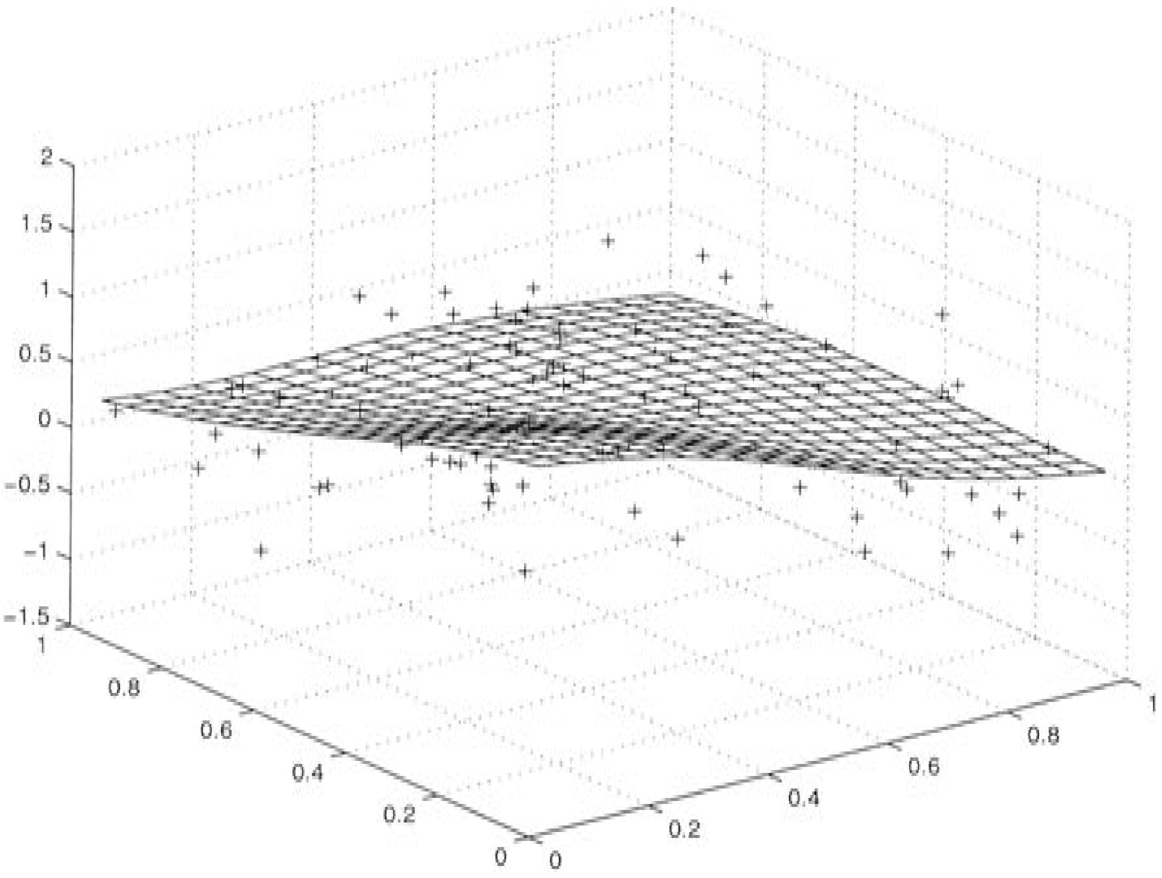


Figure 12.6: Analytic thin plate spline fit to the data set franke1.dat.

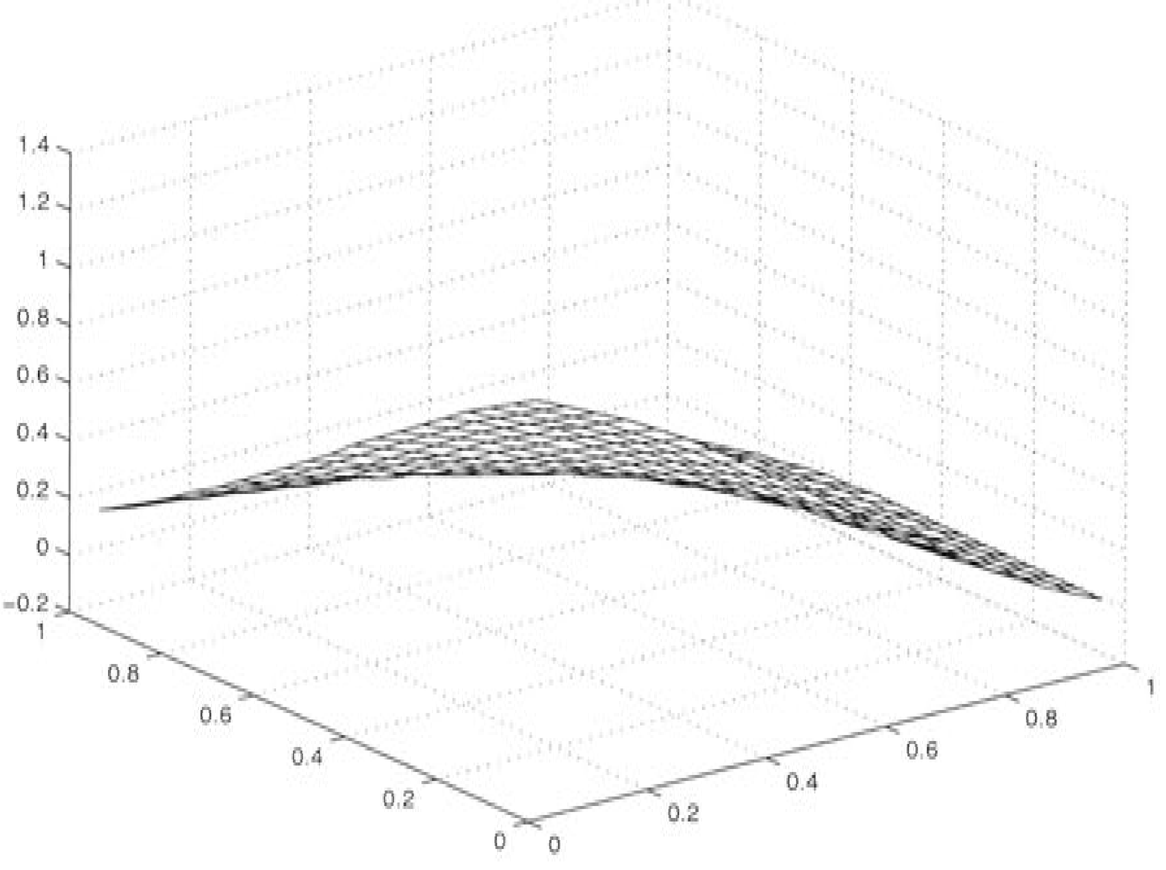


Figure 12.7: Biquadratic B-spline fit to the data set franke1.dat.

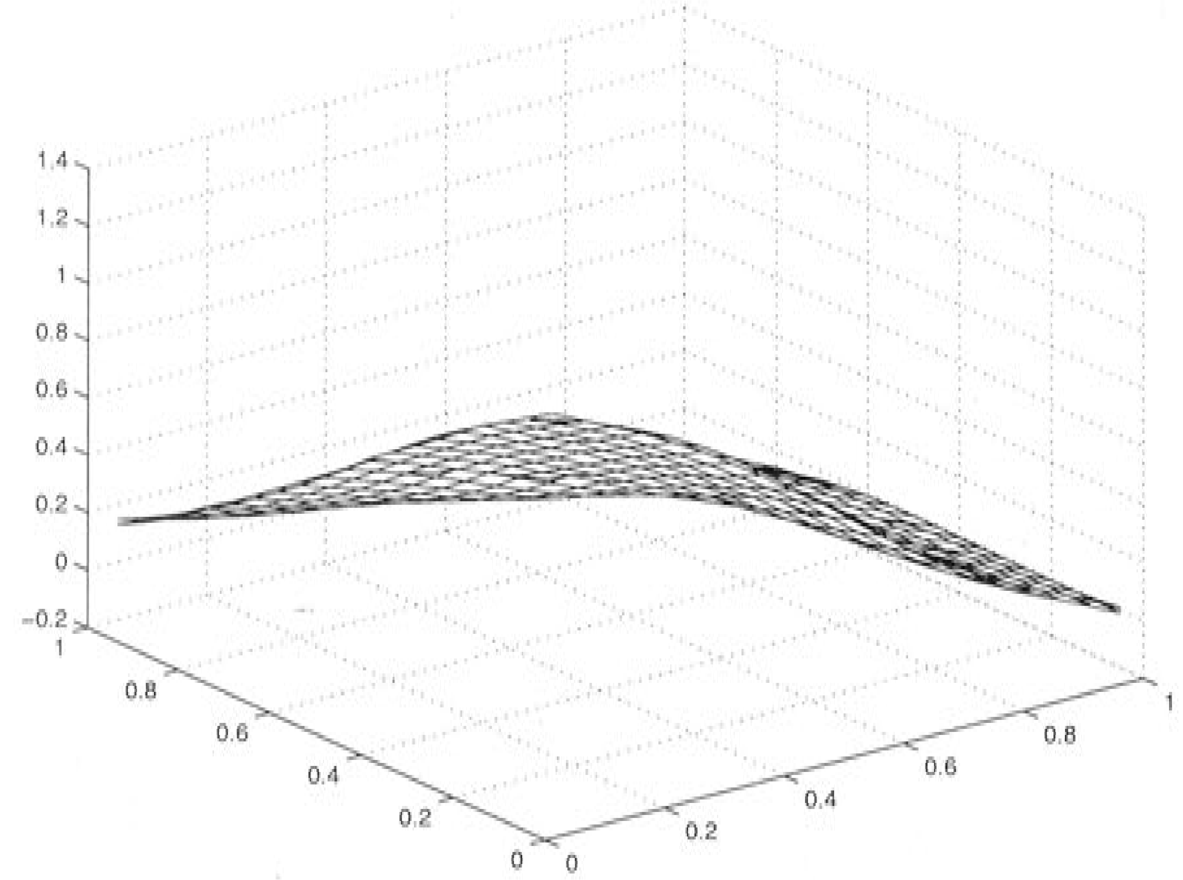


Figure 12.8: Overlay of the biquadratic B-spline solution and the analytic solution for the data set franke1.dat.

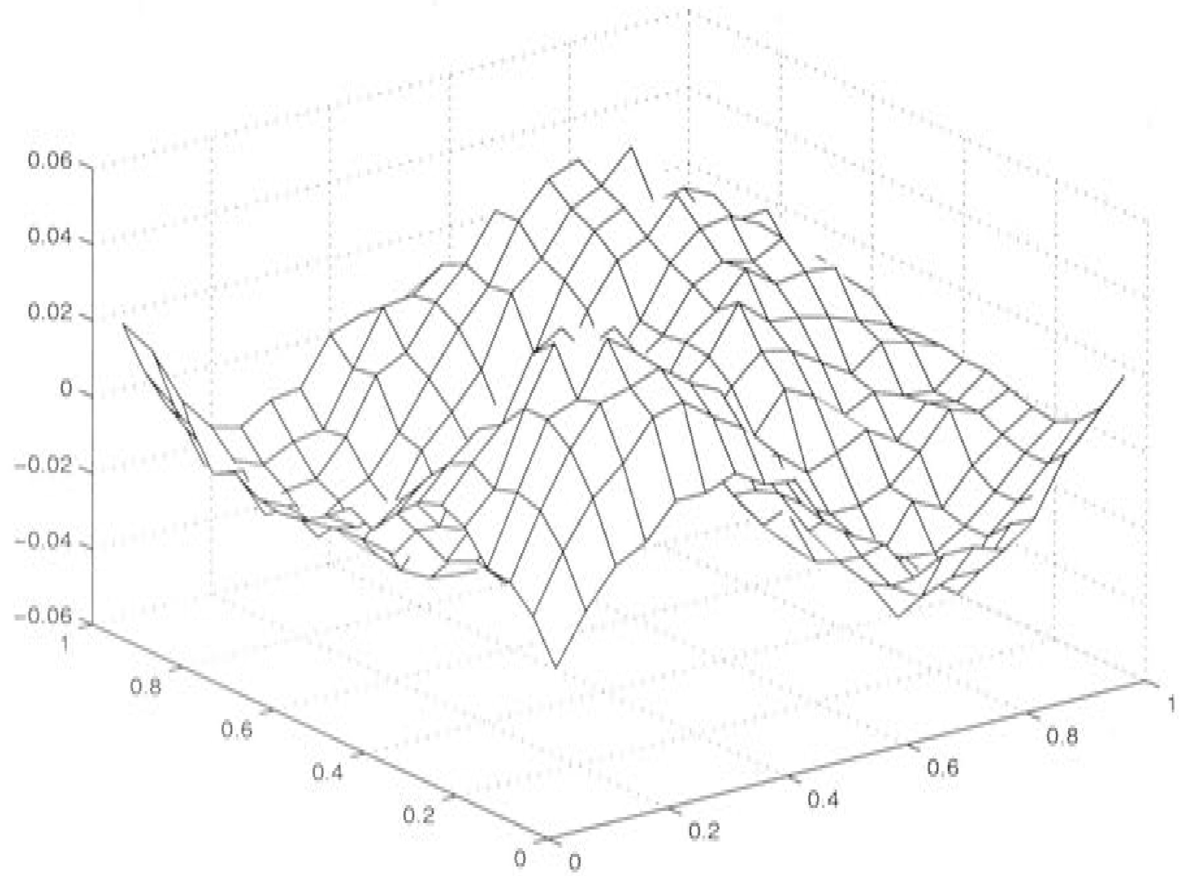


Figure 12.9: Difference between the biquadratic B-spline solution and the analytic solution for the data set franke1.dat.

and Figures 12.7, 12.8 and 12.9. The MINGCV surface differs very little from the analytic function. The algorithm terminates on a very coarse grid, clearly indicating the lack of fine scale structure in the data set. There is therefore good agreement between the analytic and finite element surfaces at the edge regions, as there is no need to solve edge regions at a fine resolution. There is little difference between the statistics associated with the analytic spline and the finite element representation, with the exception of the λ value.

Grid no.	h	No. of updates	λ	Signal	RMS residual	GCV	$\hat{\sigma}$
6	0.200	11	0.236	5.20	0.421	0.197	0.433
5	0.100	5	0.238	5.30	0.421	0.197	0.432

Table 12.5: Results generated by the bivariate MINGCV algorithm for the data set franke1.dat.

This is likely to be an artifact of the coarse discretisation at which the solution is estimated. In the case of this data set, an appropriate smoothness can be achieved at a coarse discretisation, but λ does not need to be precisely optimised because a certain amount of smoothness is enforced by the coarse basis elements.

Detailed results are shown in section C.2. The algorithm converges quickly on both grids. The fact that θ increases so much from the starting value shows that the roughness penalty term clearly dominates this system. We would not expect the algorithm to work for this data set if it had started on a fine grid. The solution estimate again changes very little between the two grids, as can be seen from the Sobolev norm and the estimate of Tr. This shows that that the solution estimate was accurate on the coarse grid, even though the $dGCV/d\theta$ estimate at the end of grid number 5 is quite different to that at the end of grid number 6. It has been generally the case for the analyses in this study that derivative estimates are far more sensitive to iteration than fundamental solutions characteristics like the norm, Tr, R and the GCV.

Franke2.dat

Franke2.dat, the data set with a noise level of 1/16, and the analytic spline fit are shown in Figure 12.10. The signal value, given in Table 12.6, is in the ideal range of values slightly less than half the number of data points. This is likely to give a well conditioned system, associated with a reliable solution with a well-defined optimum. The MINGCV summary results for franke2.dat, shown in Table 12.7, show that all analytic solution characteristics are well estimated by the MINGCV algorithm. Good agreement is also shown in Figures 12.11, 12.12 and 12.13.

$\lambda \times 10^3$	Signal	RMS residual	GCV $\times 10^2$	$\hat{\sigma}$
0.956	36.3	0.0427	0.450	0.0535

Table 12.6: Summary statistics for the analytic thin plate smoothing spline fit to the data set franke2.dat.

The trend of increasing signal and decreasing RMS residual as the grids are refined is evident, indicating the gradual development of fine scale structure. The full results in section C.3 show that convergence is very fast on all grids. On the coarsest grid $dGCV/d\theta$ is reduced by 4 orders of magnitude in 7 updates, on grid number 5 there is a reduction of 4 orders of magnitude in 5 updates, and convergence is similar on the final grid. This is an example of an optimal performance of the MINGCV algorithm, with fast convergence to an accurate estimate of the analytic solution.

Grid no.	h	No. of updates	$\lambda \times 10^3$	Signal	RMS residual	GCV $\times 10^2$	$\hat{\sigma}$
6	0.200	7	0.508	29.0	0.0474	0.445	0.0562
5	0.100	5	0.858	32.9	0.0447	0.445	0.0546
4	0.0500	5	0.964	35.3	0.0435	0.451	0.0540

Table 12.7: Results generated by the bivariate MINGCV algorithm for the data set franke2.dat.

It is interesting to note that, according to the GCV, the predictive capacity doesn't necessarily improve with refinement. The coarser grid solutions have a lower GCV than the fine grid solution, although the fine grid solution is a more accurate representation of the analytic solution. It certainly possible for a function to have a lower GCV than the analytic spline fit. It seems that in this case the coarse grid finite element approximation is a slightly better predictor than the analytic solution.

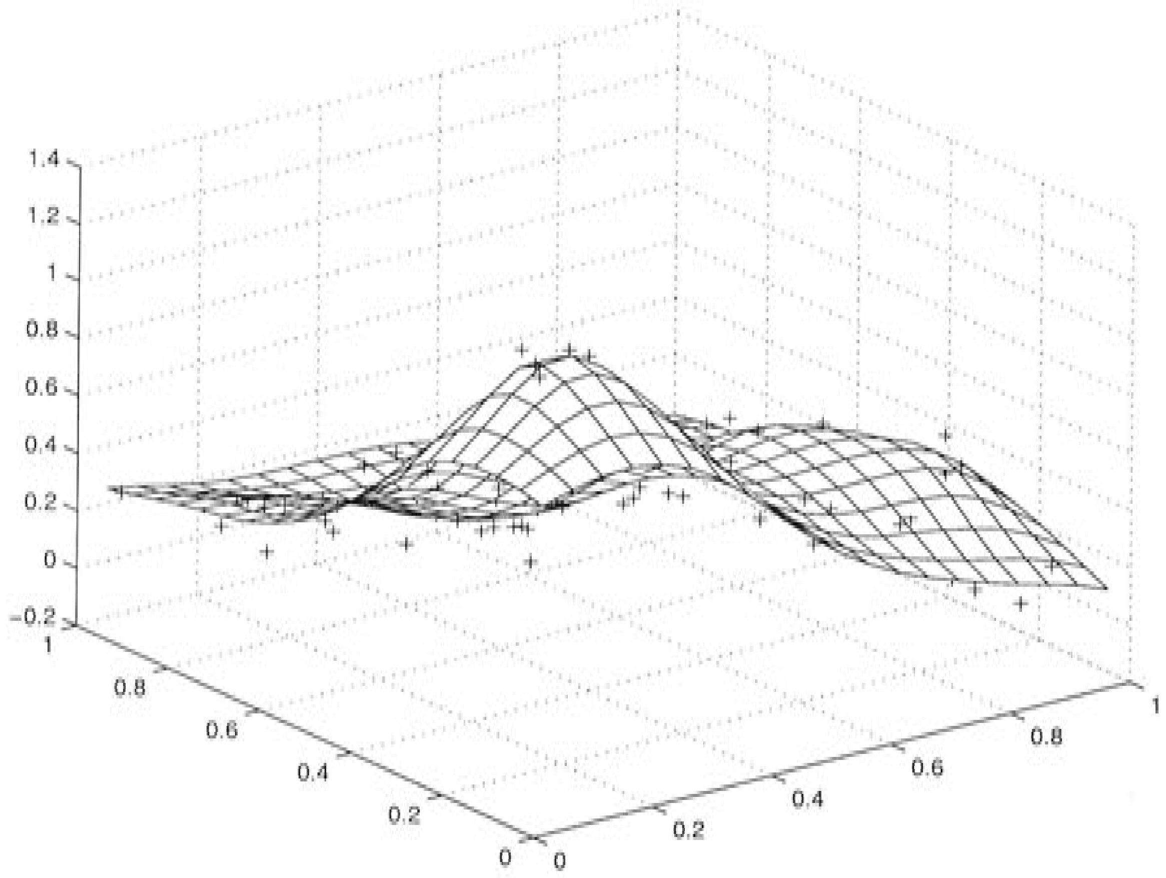


Figure 12.10: Analytic thin plate spline fit to the data set franke2.dat.

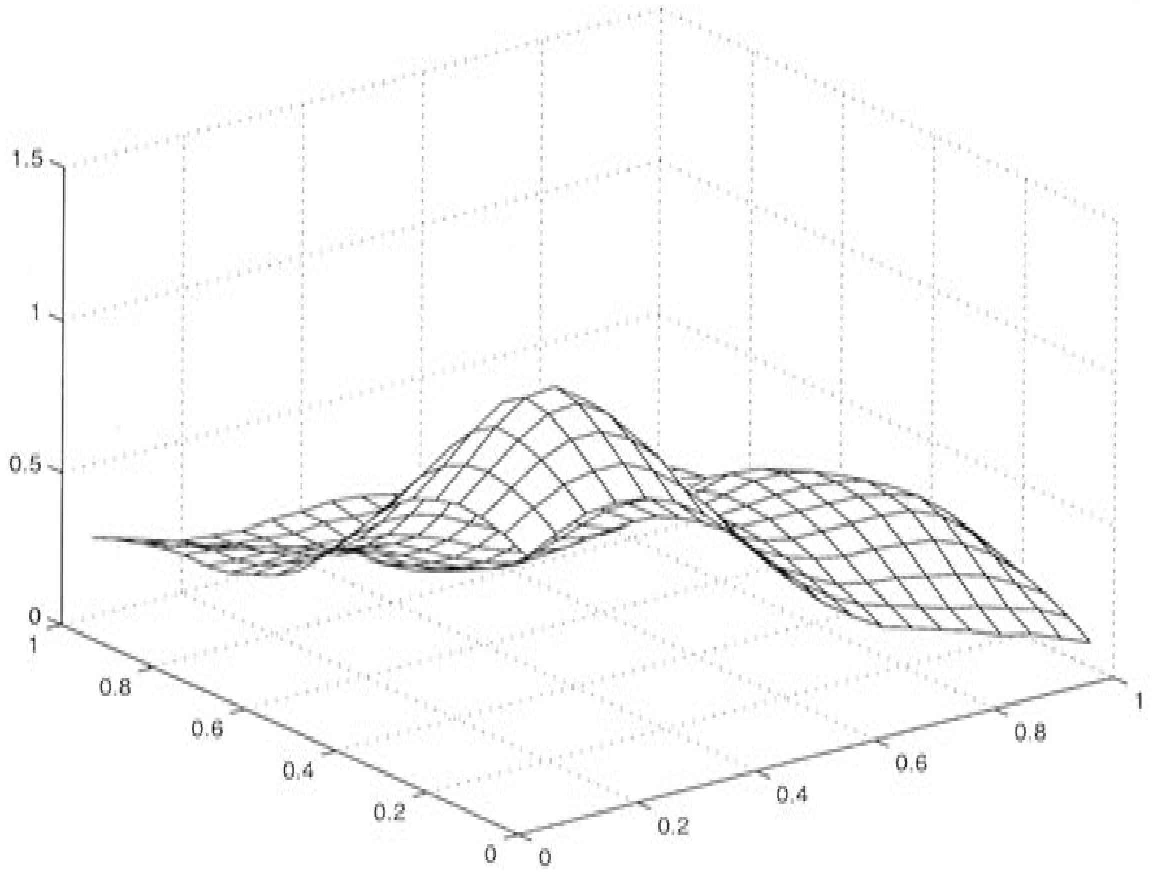


Figure 12.11: Biquadratic B-spline fit to the data set franke2.dat.

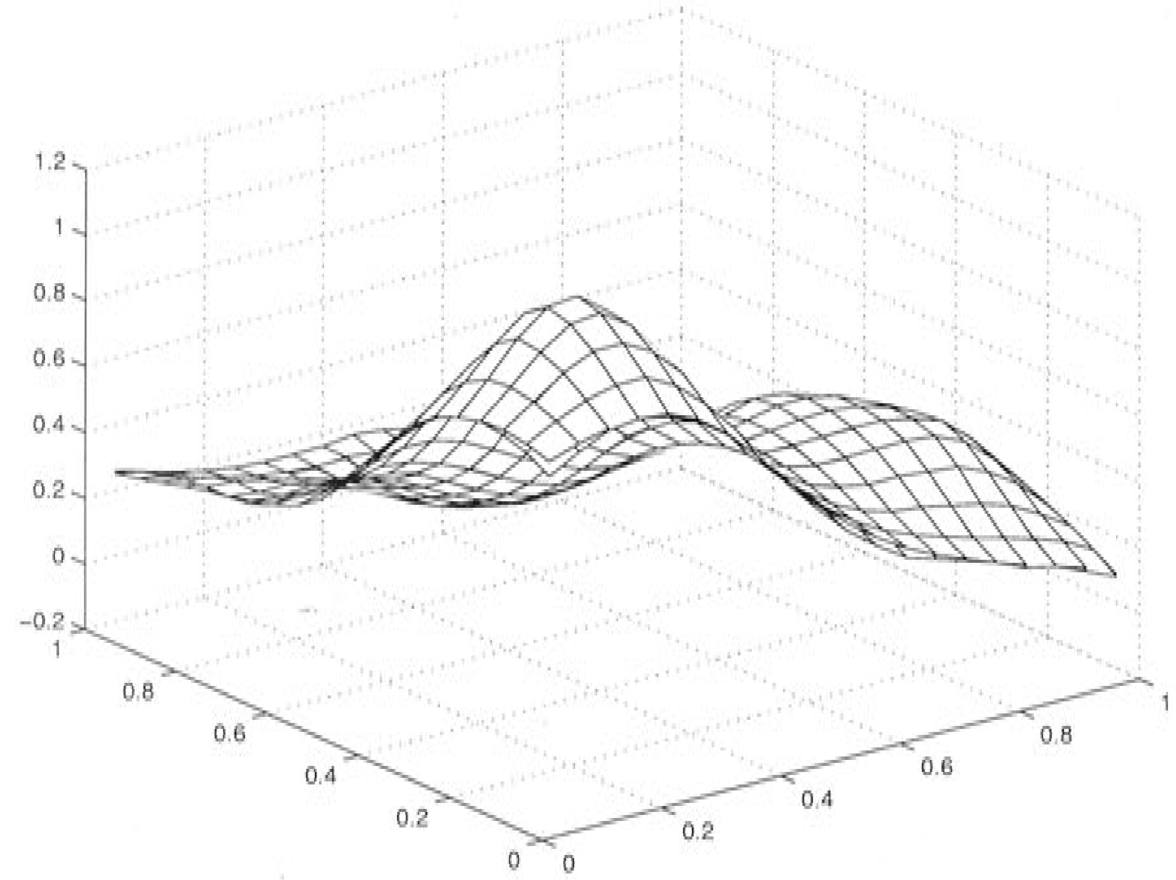


Figure 12.12: Overlay of the biquadratic B-spline solution and the analytic solution for the data set franke2.dat.

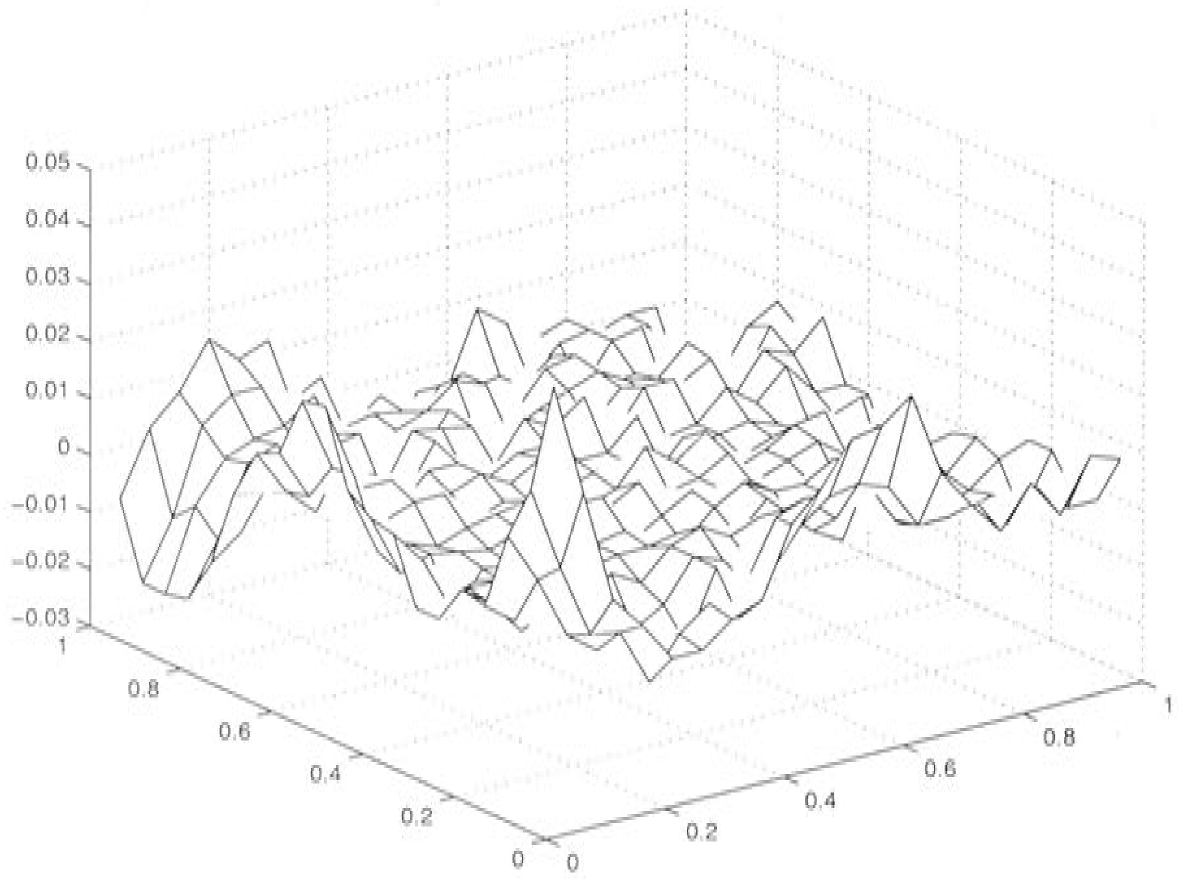


Figure 12.13: Difference between the biquadratic B-spline solution and the analytic solution for the data set franke3.dat.

Franke3.dat

Reducing the noise to 1/128 gives franke3.dat, shown in Figure 12.14, along with the analytic spline fit. The results of the MINGCV algorithm, in Table 12.9, are in good agreement with the analytic results in Table 12.8, although they

are not as close as for the previous data set. This could be due to the higher curvature of this function at end regions, as shown in Figure 12.14. Both franke2.dat and franke3.dat only required two refinements to represent the final solution. Convergence results are also similar for these two data sets. This is understandable given that the underlying function is the same and the signal values are in middle regions rather than the extremal regions of exact interpolation or flat plane regression.

$\lambda \times 10^3$	signal	RMS residual $\times 10^2$	GCV $\times 10^3$	$\hat{\sigma} \times 10^2$
0.0702	73.8	0.359	0.188	0.702

Table 12.8: Summary statistics for the analytic thin plate spline fit to the data set franke3.dat.

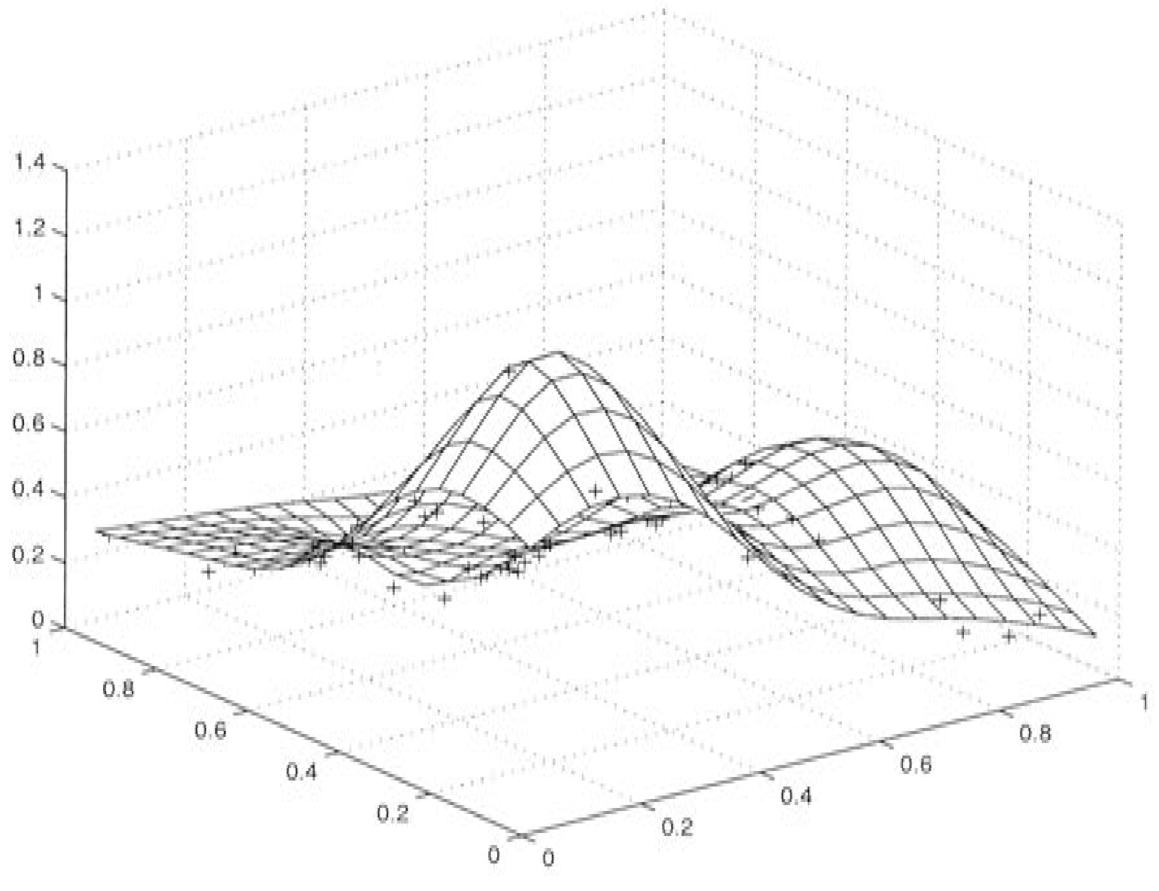


Figure 12.14: Analytic thin plate spline fit to the data set franke3.dat.

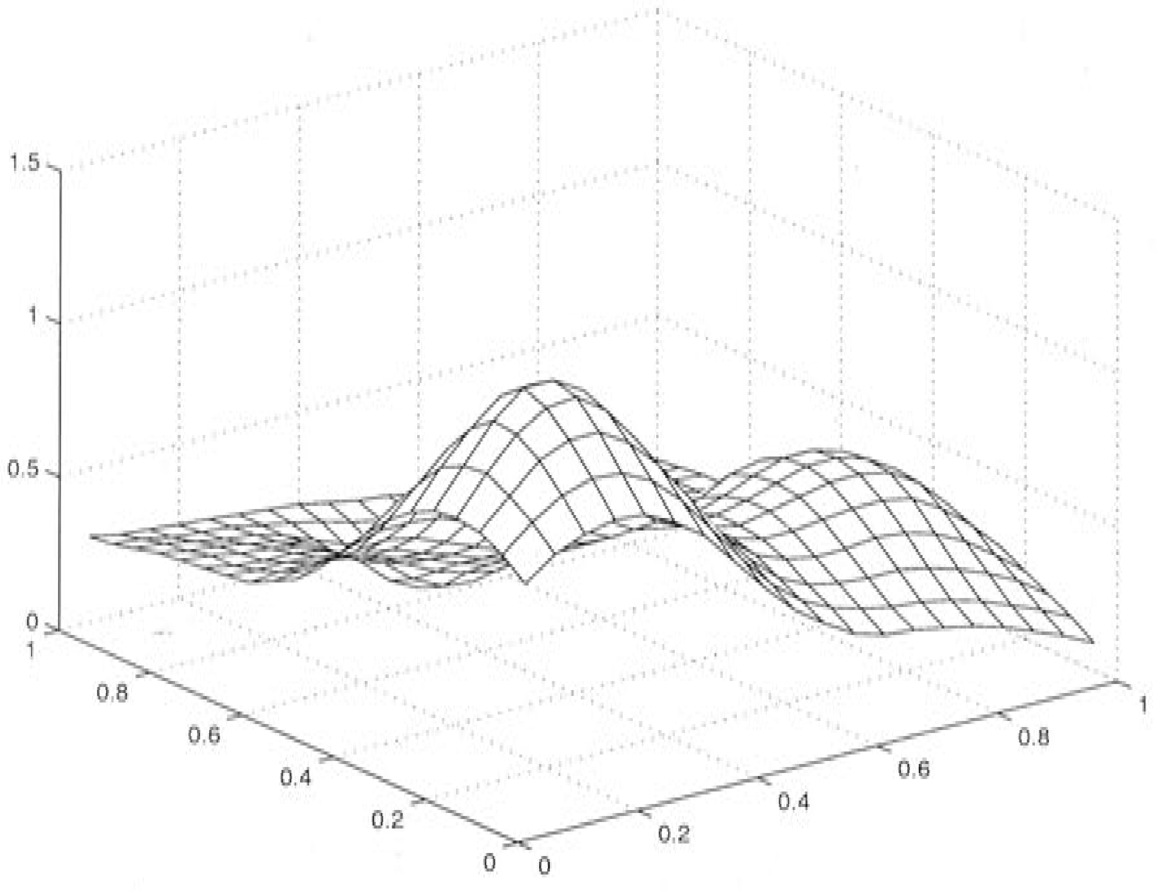


Figure 12.15: Biquadratic B-spline fit to the data set franke3.dat.

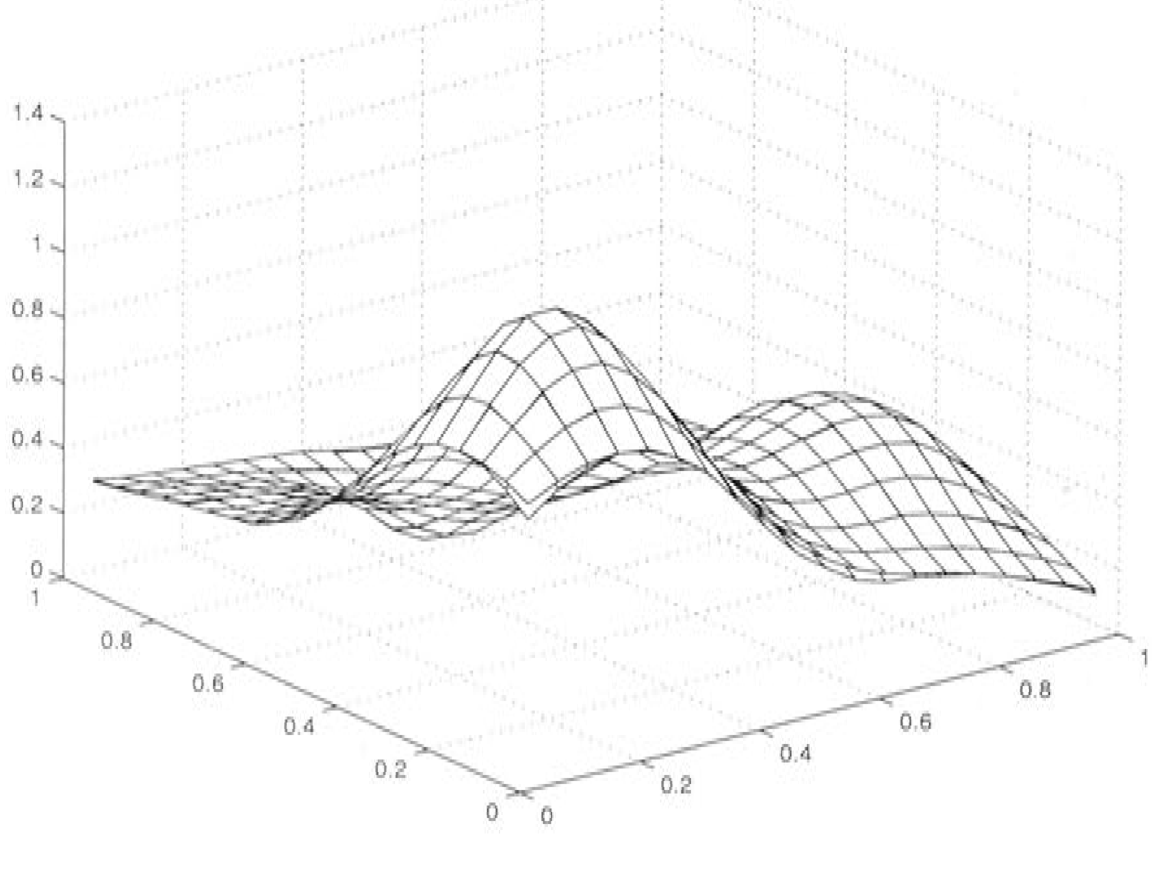


Figure 12.16: Overlay of the biquadratic B-spline solution and the analytic solution for the data set franke3.dat.

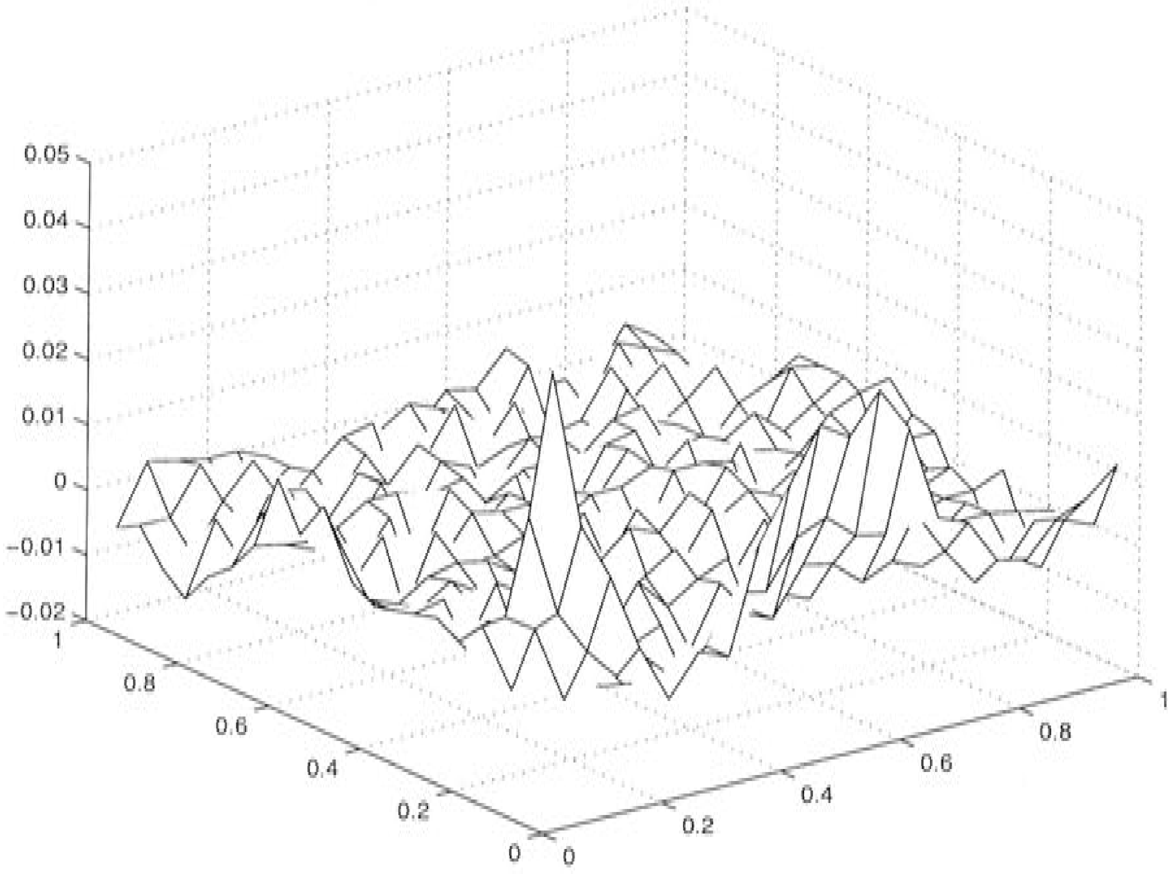


Figure 12.17: Difference between the biquadratic B-spline solution and the analytic solution for the data set franke3.dat.

Grid no.	h	No. of updates	$\lambda \times 10^3$	Signal	RMS residual	GCV $\times 10^3$	$\hat{\sigma} \times 10$
6	0.2	10	0.102	36	0.0169	0.726	0.213
5	0.1	5	0.0530	61.3	0.00476	0.152	0.0765
4	0.05	5	0.0813	67.1	0.00431	0.171	0.0750

Table 12.9: Results generated by the bivariate MINGCV algorithm for the data set franke3.dat.

Franke4.dat

The case of zero noise is a good test for the MINGCV algorithm. The θ estimate will be forced to its minimum on several grids, so we can determine whether the updates are able to respond correctly to constant visits to the extreme region of the GCV curve. However, poor conditioning should not be a problem as the rank deficient component will be virtually zero. The summary statistics for the analytic solution in Table 12.10, and the graph in Figure 12.18, show that the analytic spline exactly interpolates the data. The MINGCV results in Table 12.11 show five refinements of the grid, at which point the signal estimate shows that exact interpolation has essentially occurred. The difference between the analytic and finite element solution is small, as shown in Figures 12.19, 12.20 and 12.21, even though the approximate λ and RMS residual values do not get as small as the analytic values.

Looking at section C.5 for further detail, we see that the θ estimate has hit the minimum possible value on all grids except the final grid. Some instability is evident at the third update of grid number 3, in that there is a decrease in consecutive θ updates combined with an increase in consecutive positive $dGCV/d\theta$ updates. This contradicts the analytic relationship between the GCV and the smoothing parameter λ, shown in Figure 8.1, which is an indication that the process is poorly synchronised at this point, and the algorithm does not recognise the point of minimum GCV.

$\lambda \times 10^{12}$	Signal	RMS residual $\times 10^{10}$	GCV $\times 10^{4}$	$\hat{\sigma} \times 10^{6}$
0.577	100	0.209	0.148	0.284

Table 12.10: Summary statistics for the analytic thin plate spline fit to the data set franke4.dat.

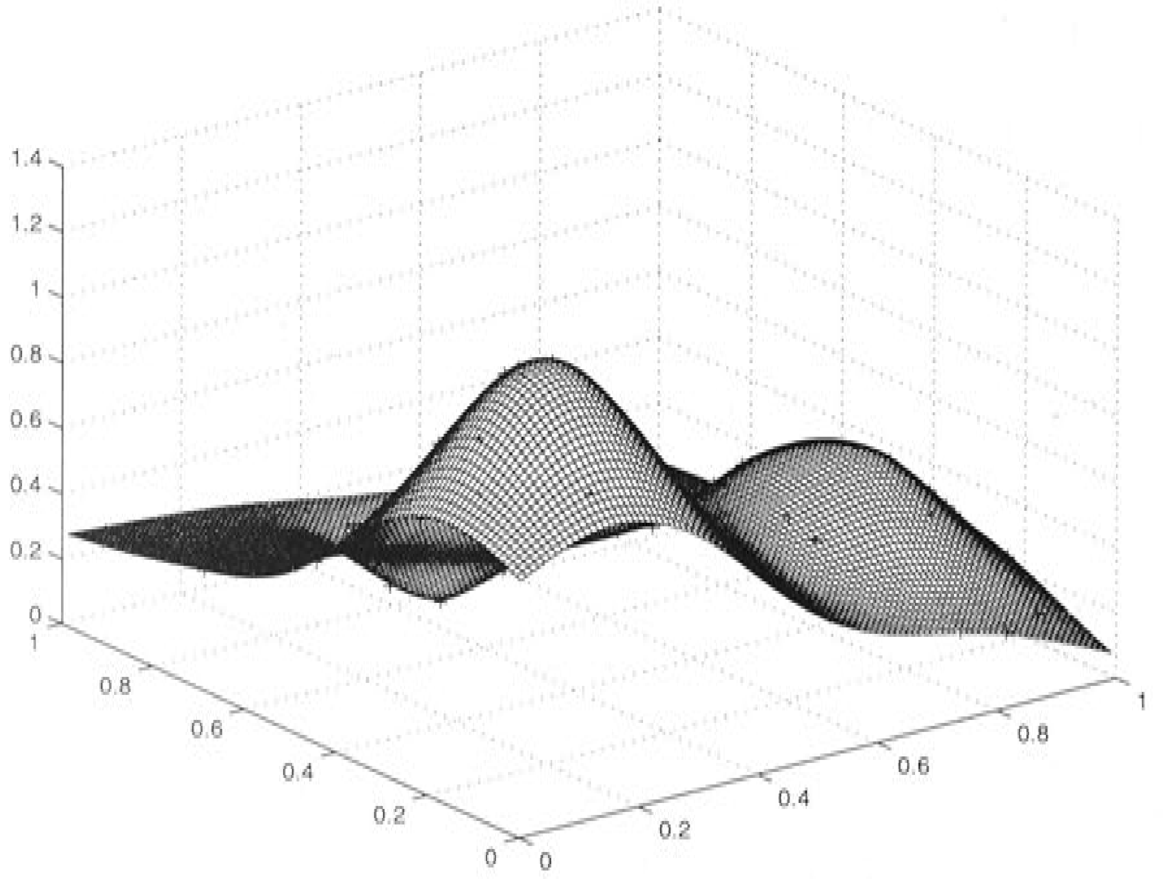


Figure 12.18: Analytic thin plate spline fit to the data set franke4.dat.

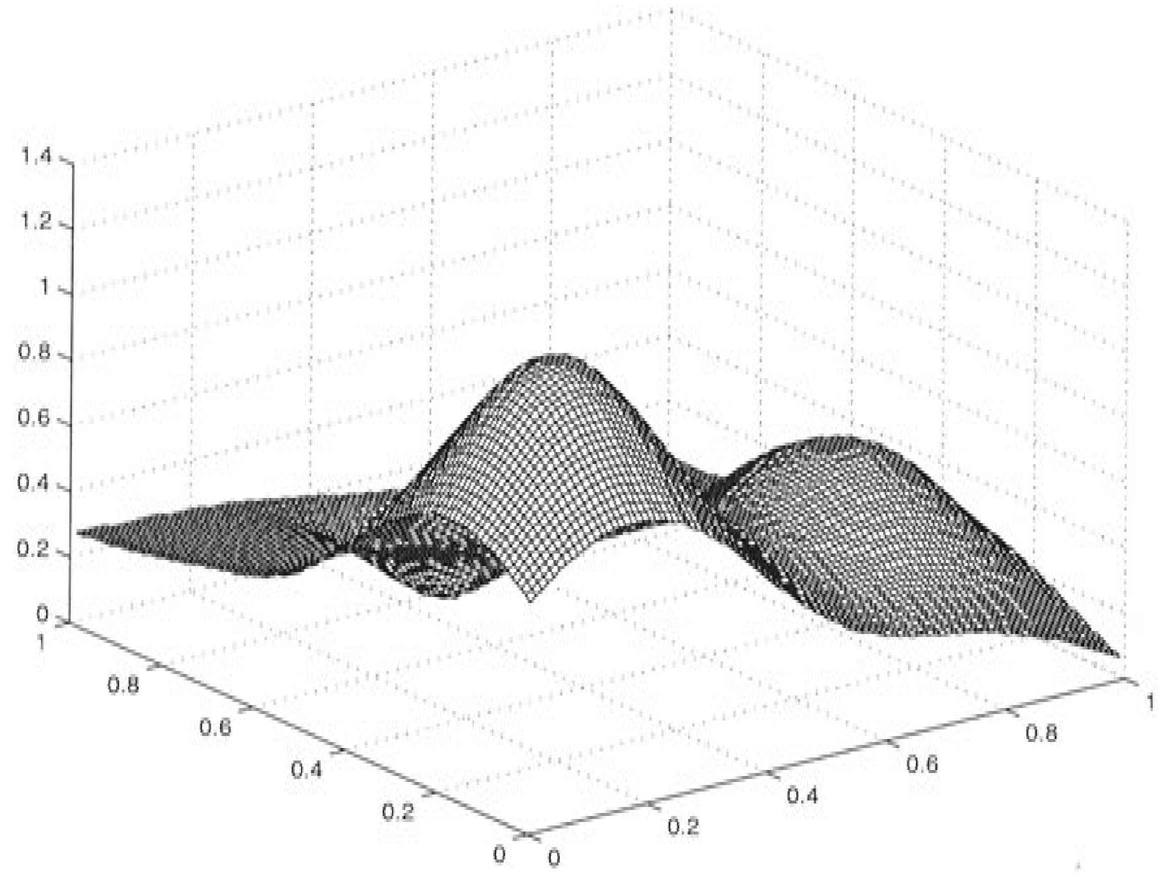


Figure 12.19: Biquadratic B-spline fit to the data set franke4.dat.

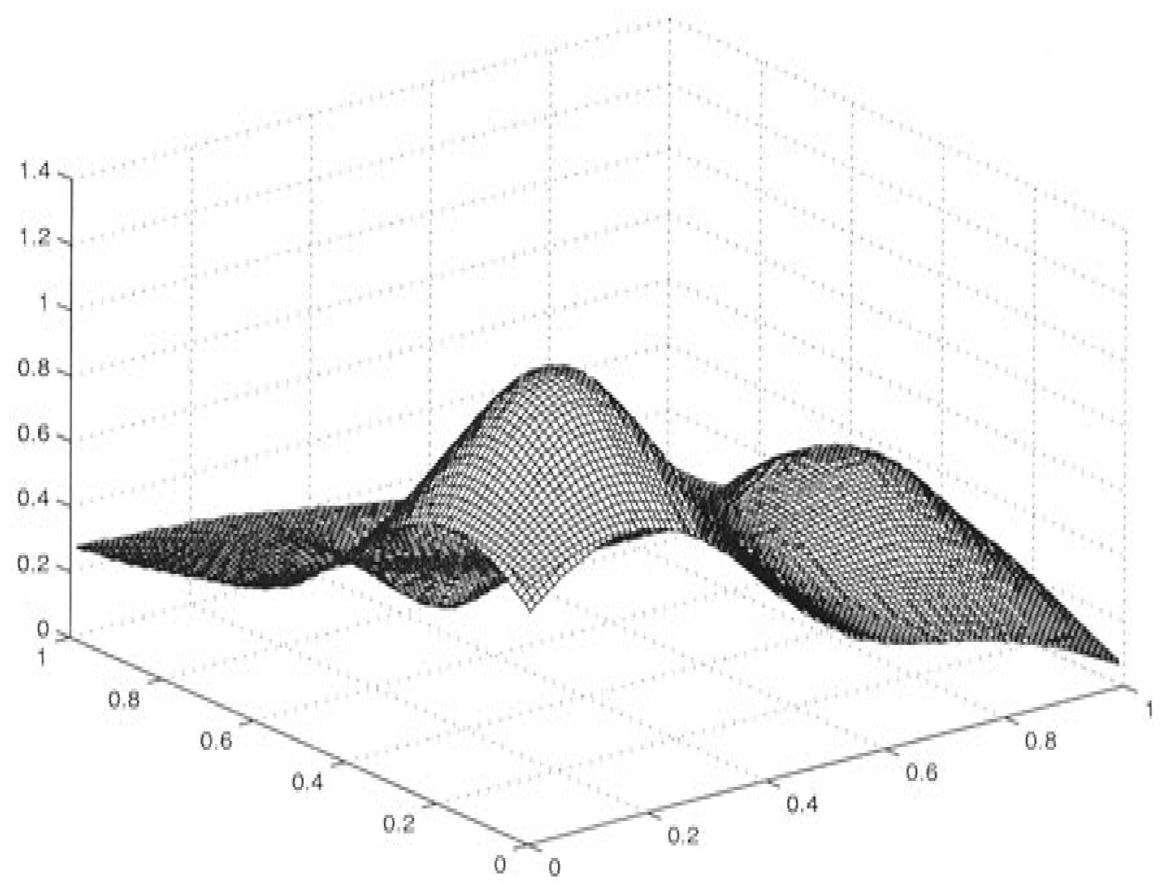


Figure 12.20: Overlay of the biquadratic B-spline solution and the analytic solution for the data set franke4.dat.

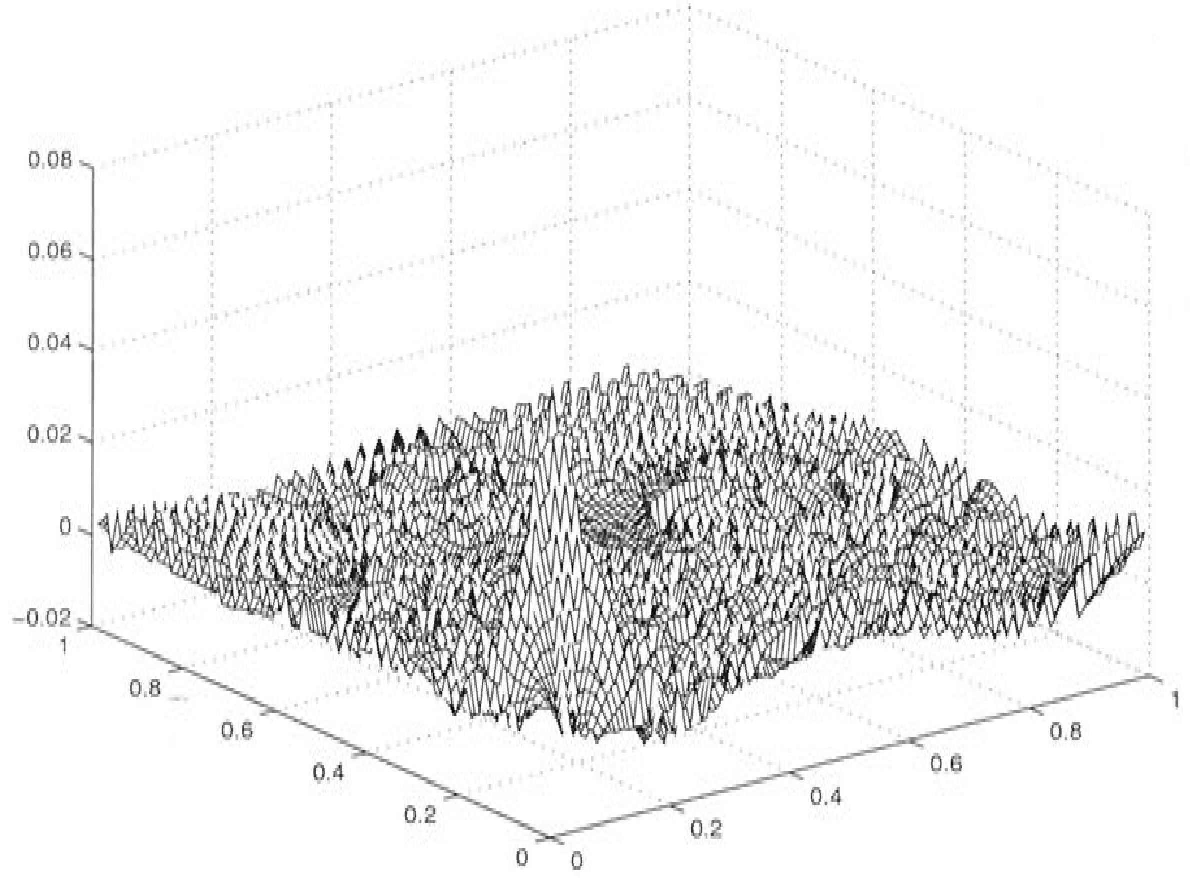


Figure 12.21: Difference between the biquadratic B-spline solution and the analytic solution for the data set franke4.dat.

Grid no.	h	No. of updates	$\lambda \times 10^3$	Signal	RMS residual $\times 10^3$	GCV $\times 10^2$	$\hat{\sigma} \times 10$
6	0.256	14	0.131	28.3	0.306	0.177	0.352
5	0.128	3	0.0328	57.5	0.0479	0.0130	0.0735
4	0.0640	2	0.00819	83.5	0.00408	0.00106	0.0100
3	0.0320	5	0.00205	96.2	0.000647	0.000571	0.00331
2	0.0160	6	0.000512	99.4	0.000503	0.000334	0.00631
1	0.00800	15	0.000815	99.2	0.000445	0.000623	0.00488

Table 12.11: Results generated by the bivariate MINGCV algorithm for the data set franke4.dat.

This would have resulted in a negative estimate of $d^2GCV/d\theta^2$, so condition 9 is activated. This situation also occurs on grid number 2 at the third and fifth updates. The negative $d^2GCV/d\theta^2$ estimates are a sign that the minimum possible θ value has been set too low in this case, because the θ estimate is already at the point where its relationship with the GCV has broken down. The final grid delivers the required flexibility, and the smoothing parameter estimate increases. This recovery is a sign that the synchronisation of the updates was not destroyed on previous grids, indicating that the conditions preventing iteration from lingering at extremal values of θ are working effectively.

On the final grid, convergence is quite slow, and the Tr, R and GCV estimates do not settle down even though the θ estimate barely changes. This is understandable considering that the GCV curve for exact interpolation will not have a local minimum. Also, the plots of R and Tr versus λ in Chapter 2 show that, for extremal values of λ, both R and Tr are relatively insensitive to changes in λ. It is therefore not surprising that, for the case of zero noise, the MINGCV algorithm cannot stabilise. It does, however, give an estimate that is not significantly different from the optimal result.

12.1.3 The peaks function

Further testing of the MINGCV algorithm was aimed at creating data sets that more closely resembled the environmental data sets for which this algorithm was intended. This involved generating a data set with areas of clumped data points, which might correspond to more heavily populated areas in a spatial region such as the Australian continent, and also large areas where data measurements are very sparse. This could cause the conditioning of the finite element equations to deteriorate, even at coarse resolutions, because there are large areas of the surface where smoothing is the only priority, as discussed in section 12.1.1.

The generating function was chosen to be the peaks function obtained from [51]. The peaks function is obtained by scaling and translating Gaussian distributions, and is given by

$$f(x,y) = 3(1-x)^2 e^{-(y+1)^2-x^2} - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{-x^2-y^2} - \frac{1}{3} e^{-(x+1)^2-y^2} \tag{12.11}$$

This function is shown in Figure 12.22. Although environmental processes are unlikely to behave in such a regular manner, this function was deemed to be a good test function because of its Gaussian nature and its spatial anisotropy.

Peaks.dat

The data set peaks.dat was created by sampling 190 points from the peaks function and adding Gaussian random noise with standard deviation of 0.2.

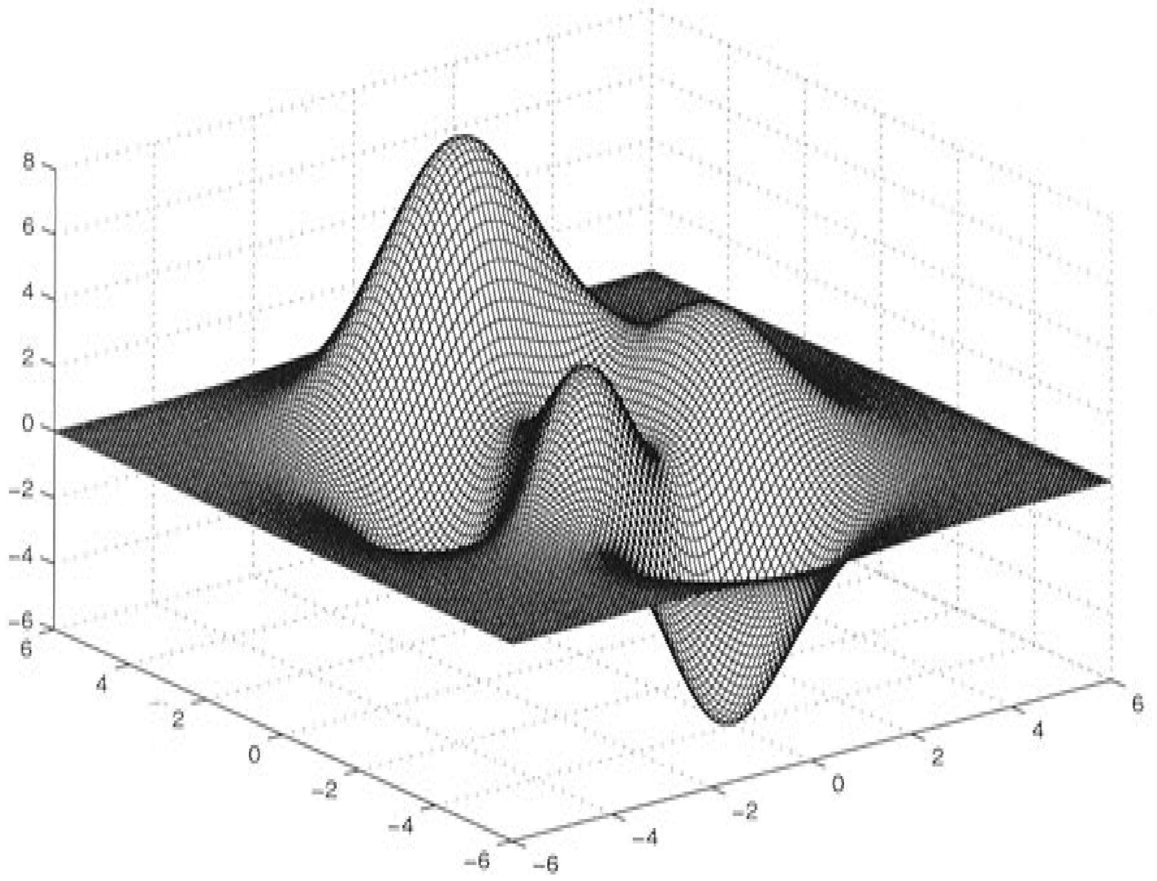


Figure 12.22: The peaks function.

This is a very small amount of noise compared to the magnitude of the data values. The data point positions are shown in Figure 12.23. They were chosen by eye to be clumped, with large areas of sparsity. The data set peaks.dat, along with the analytic spline fit, is shown in Figure 12.24. The analytic spline does not capture all the peaks in Figure 12.22, because they are not represented by data. The MINGCV results depicted in Figures 12.25, 12.26 and 12.27 show larger disagreement with the analytic solution than previous data sets. It can be seen that the regions where deviations are large in comparison to the rest of the surface correspond to areas where there are little or no data points, particularly at edge regions. This is a result of the roughness penalty issues discussed in section 12.1.1. However, these deviations do not necessarily indicate high error, considering that the analytic spline is not accurate in recreating the generating function in data sparse regions.

Table 12.12 shows that the analytic spline fit has a signal slightly less than half the number of data points. The MINGCV algorithm required four refinements

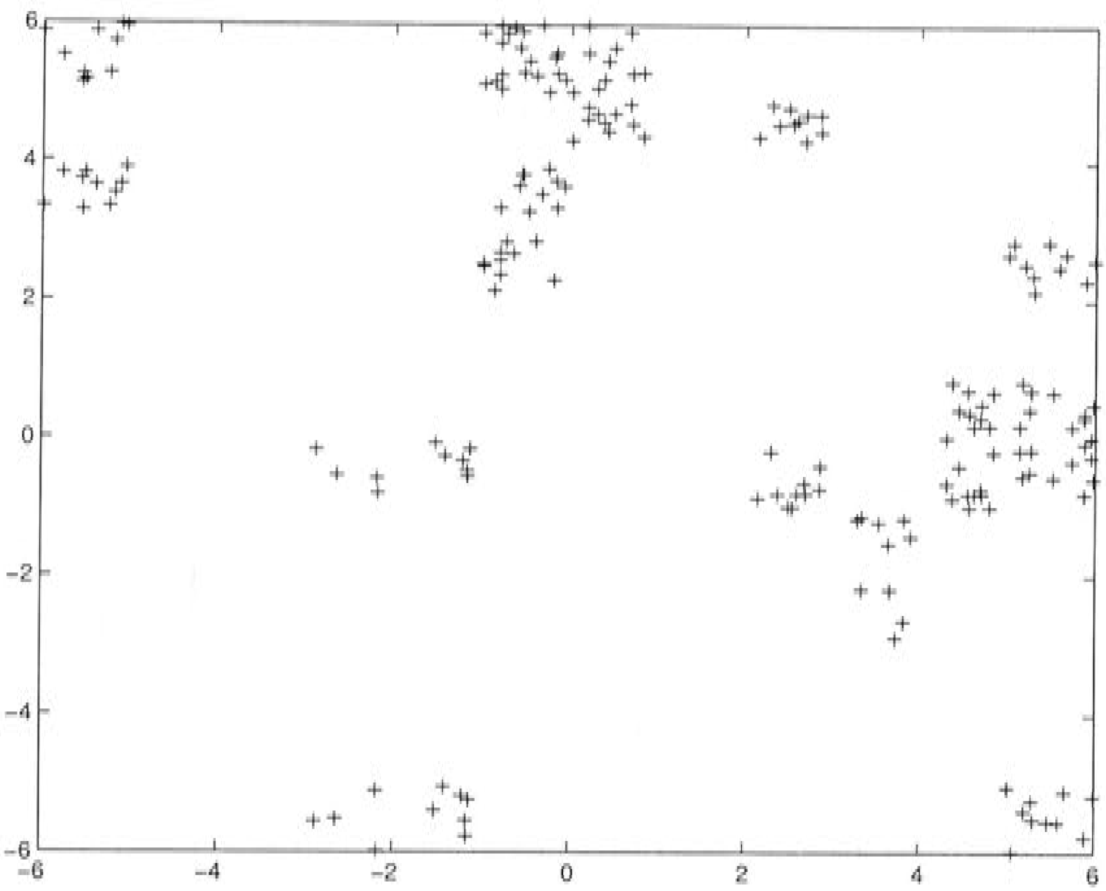


Figure 12.23: Data point positions for the data set peaks.dat.

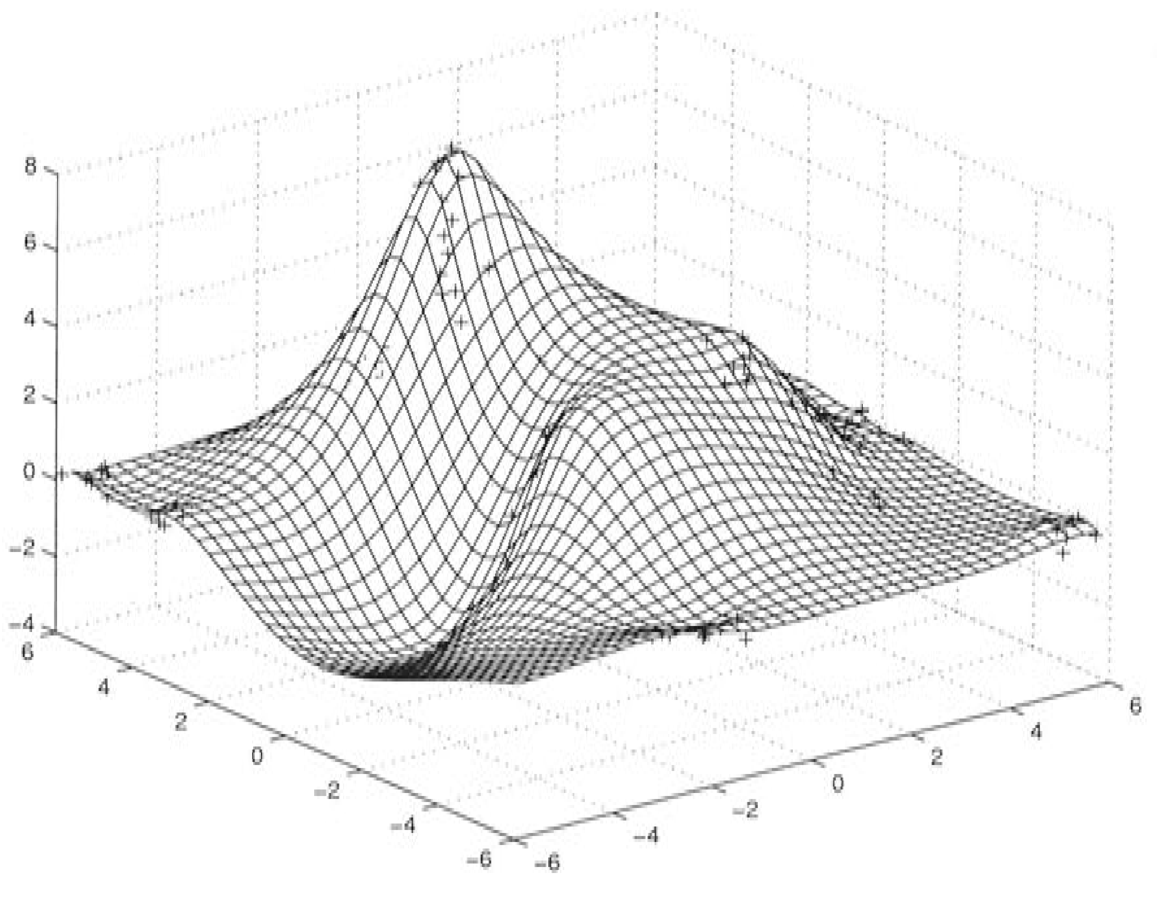


Figure 12.24: Analytic smoothing spline fit to the data set peaks.dat.

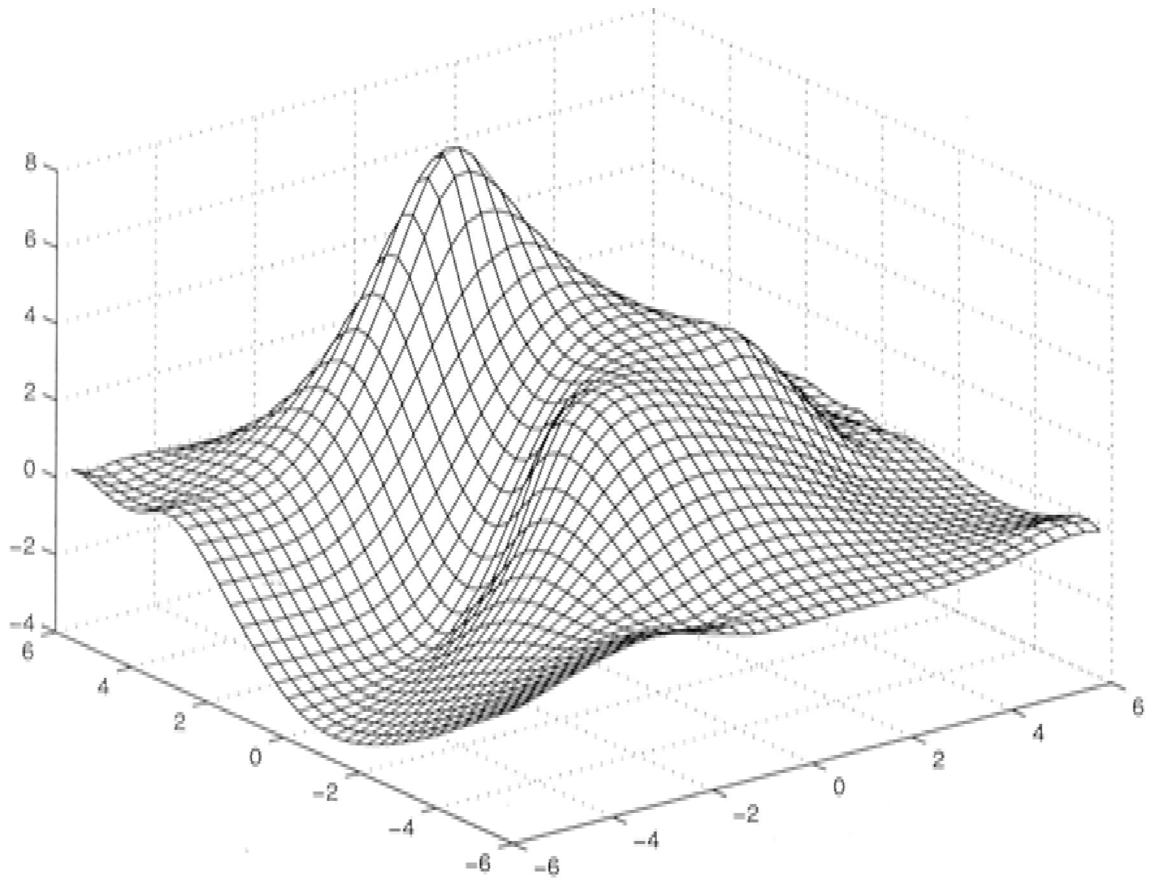


Figure 12.25: Biquadratic B-spline fit to the data set peaks.dat.

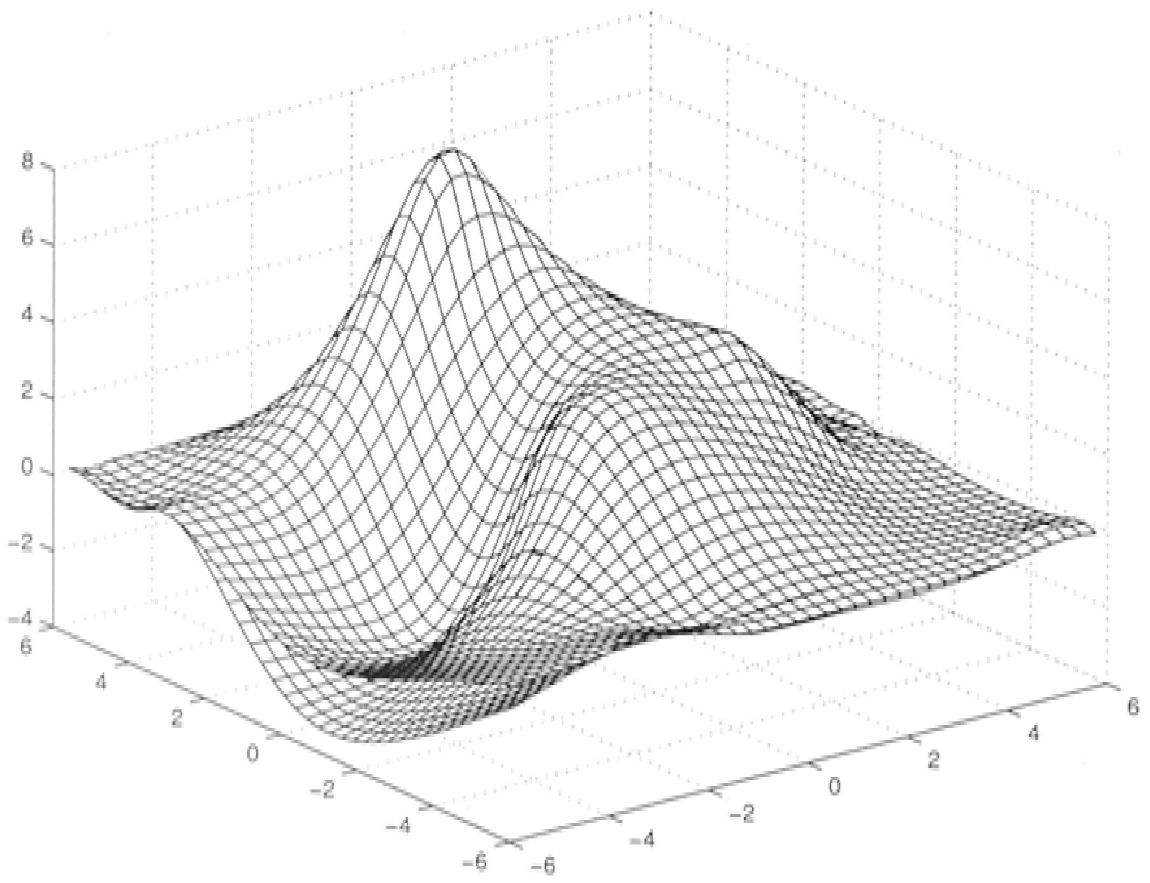


Figure 12.26: Overlay of the biquadratic B-spline solution and the analytic solution for the data set peaks.dat.

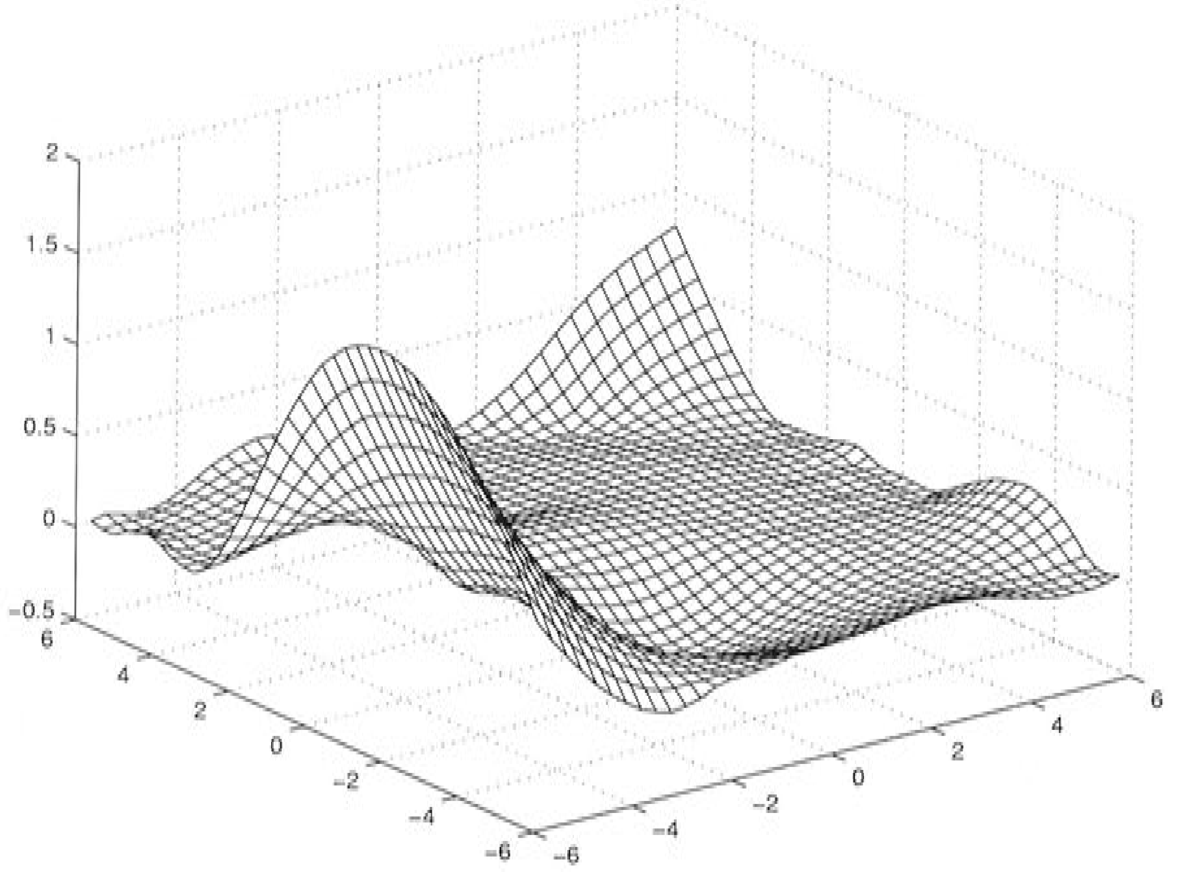


Figure 12.27: Difference between the biquadratic B-spline solution and the analytic solution for the data set peaks.dat.

$\lambda \times 10^2$	Signal	RMS residual	GCV	$\hat{\sigma}$
0.754	87.2	0.113	0.0438	0.154

Table 12.12: Summary statistics for the analytic smoothing spline fit to the data set peaks.dat.

Grid no.	h	No. of updates	$\lambda \times 10^2$	Signal	RMS residual	GCV	$\hat{\sigma}$
6	2.40	11	1.15	38.0	0.412	0.264	0.451
5	1.20	5	0.288	65.0	0.151	0.0540	0.186
4	0.600	8	0.780	76.0	0.138	0.0538	0.179
3	0.300	5	0.847	87.0	0.125	0.0529	0.170
2	0.150	5	0.843	92.4	0.118	0.0528	0.165

Table 12.13: Results generated by the bivariate MINGCV algorithm for the data set peaks.dat.

to represent this solution, as shown in Table 12.13. This can be attributed to the clumped distribution of the data set, as fine basis elements would be required to isolate the trends in the data clumps.

The full results in section C.6 show that the θ estimate hits the minimum possible value on the coarse grid, implying that this grid is too coarse to represent the complexity required for the optimal solution. Instability is evident on the grid number 5 at the third and fourth updates, where $dGCV/d\theta$ starts to become more positive as θ continues to decrease. This again indicates that the minimum limit on θ may be too low. The minimum limit is soon reached, however, and estimates are transferred to a finer grid. On the three remaining grids the behaviour is well conditioned. The reason for this is apparent from the behaviour on grid number 4. The smoothing parameter has to increase before the optimal solution is reached, which is a sign that the roughness penalty is having some influence on the solution, instead of the smoothness imposed by the coarse initial grids.

Peaks15.dat

The peaks.dat data set is not realistic due to the low noise. The noise level was therefore increased so that the standard deviation was 1.5, producing the data set peaks15.dat. This data set, along with the analytic spline fit, are shown in Figure 12.28. Tables 12.14 and 12.15, and Figures 12.29, 12.30 and 12.31, show good agreement between the analytic solution and the MINGCV estimate away from the edge regions. A relatively large number of refinements was again required to obtain this solution, but the final level is twice as coarse as the final level for peaks.dat. The full results in section C.7 shows slow convergence on the first grid, which may indicate that the optimum is again not well-defined on this grid. This is quite possible given that there is not much flexibility in the representation of a clumped data set at a very coarse discretisation.

The solution changes significantly between the first two resolutions, which further confirms that the coarsest grid was unable to represent the trends in the data.

λ	signal	RMS residual	GCV	$\hat{\sigma}$
0.129	33.3	0.916	1.23	1.01

Table 12.14: Summary statistics for the analytic thin plate smoothing spline fit to the data set peaks15.dat.

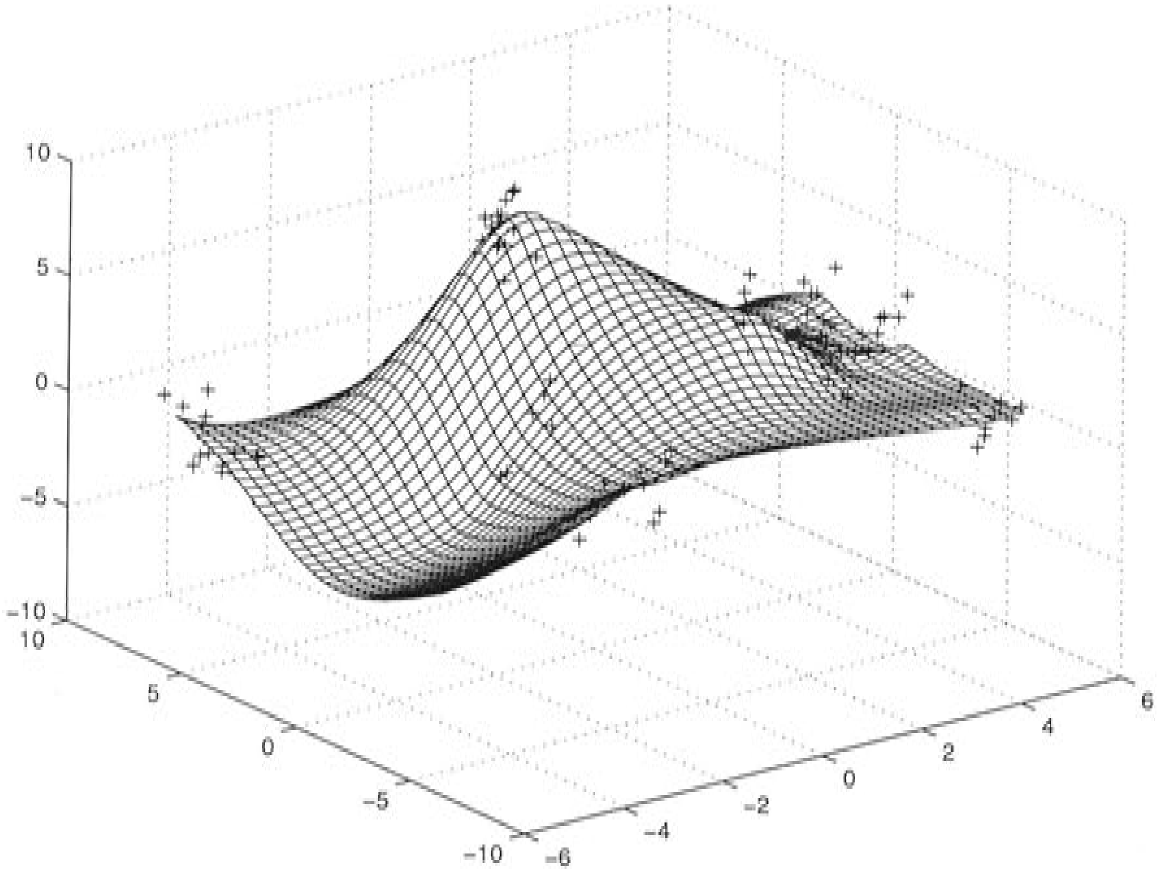


Figure 12.28: Analytic thin plate spline fit to the data set peaks15.dat.

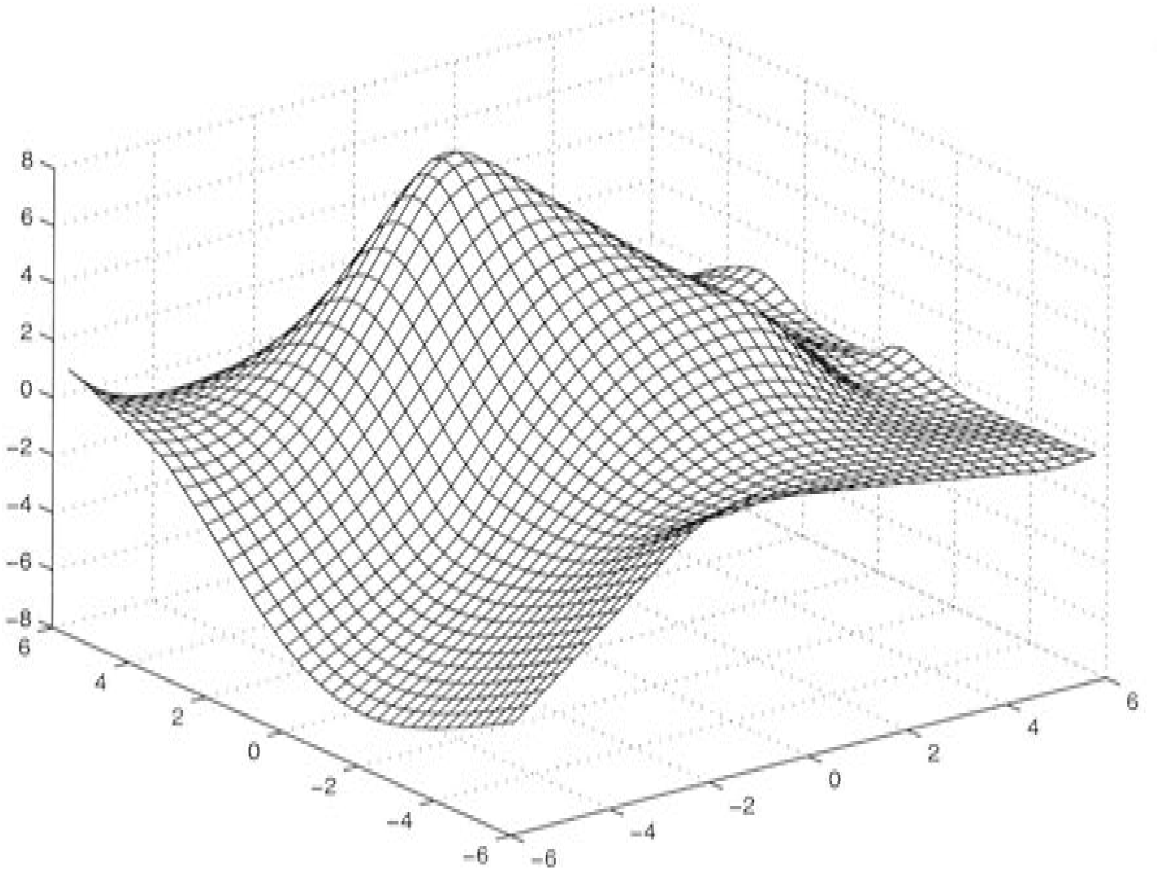


Figure 12.29: Biquadratic B-spline fit to the data set peaks15.dat.

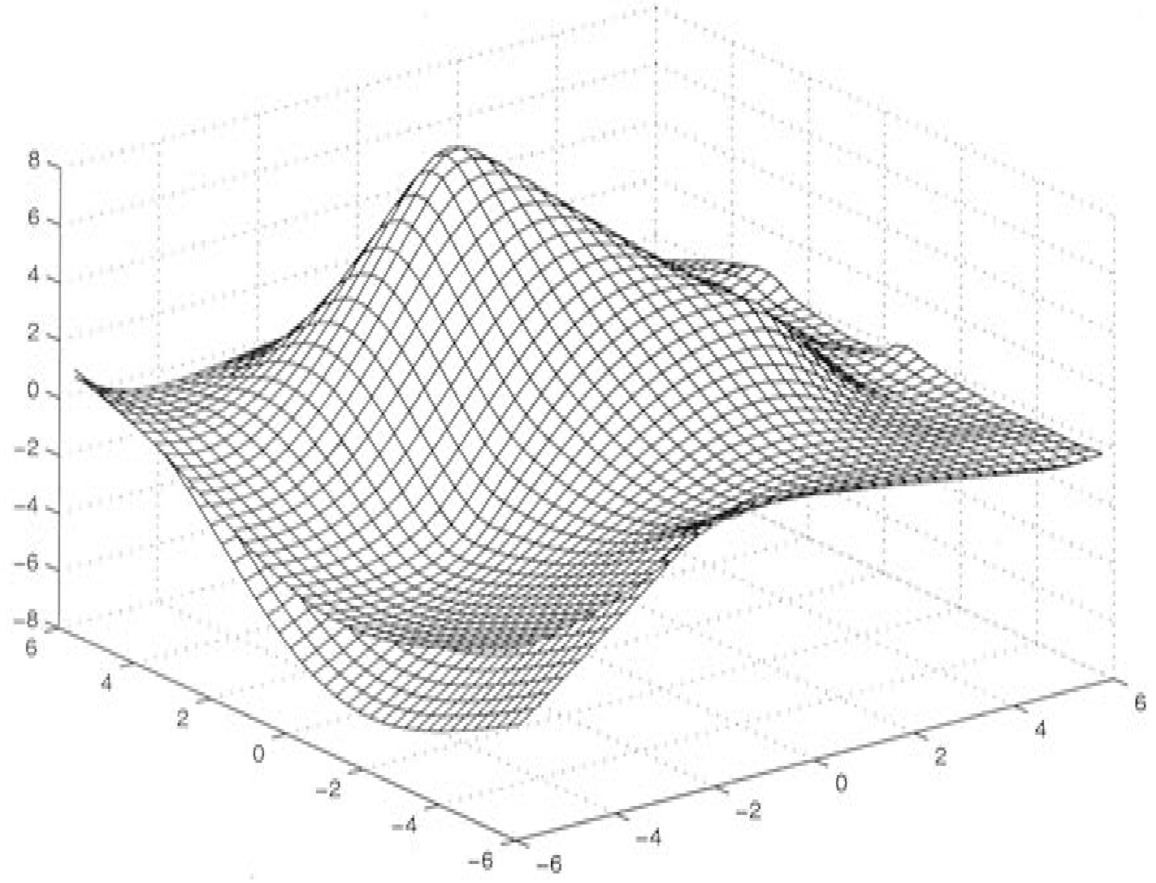


Figure 12.30: Overlay of the biquadratic B-spline solution and the analytic solution for the data set peaks15.dat.

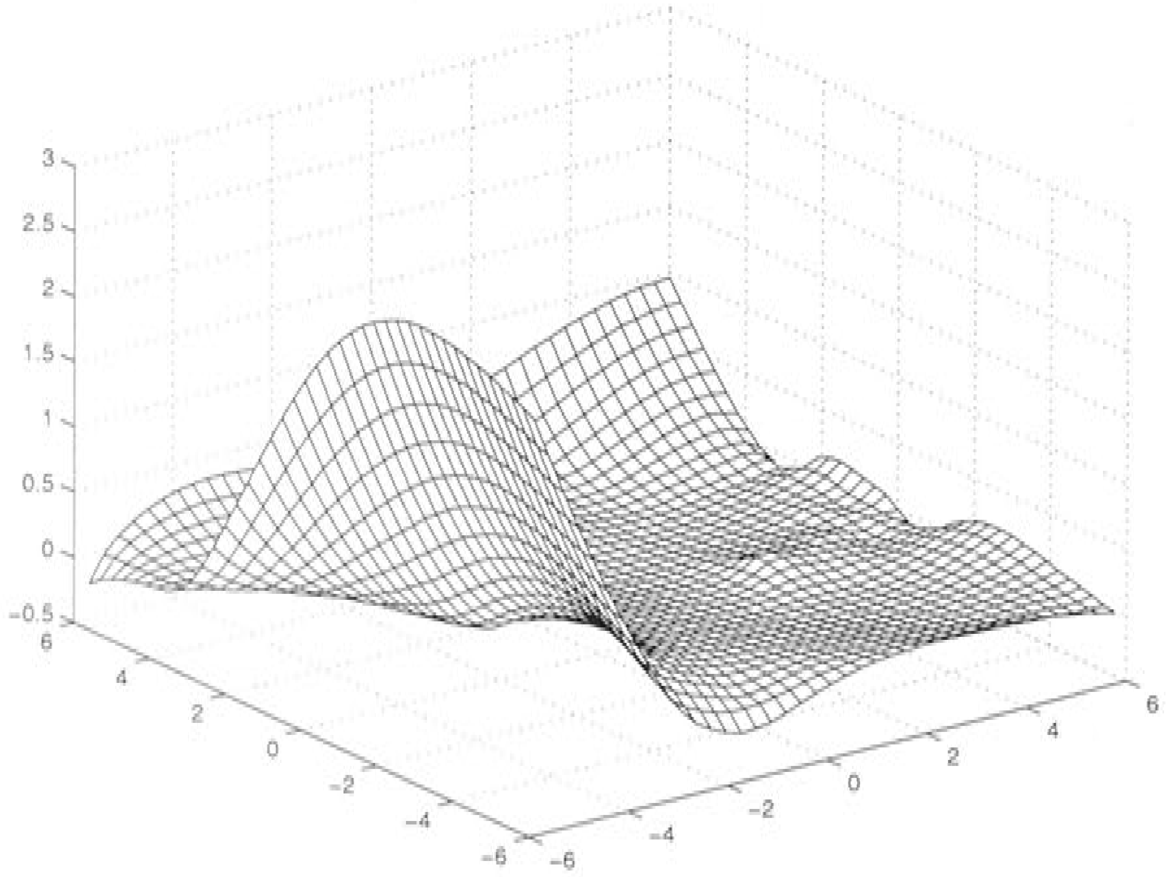


Figure 12.31: Difference between the biquadratic B-spline solution and the analytic solution for the data set peaks15.dat.

Grid no.	h	No. of updates	λ	Signal	RMS residual	GCV	$\hat{\sigma}$
6	2.40	11	0.155	28.	1.04	1.49	1.13
5	1.20	5	0.125	37.	0.956	1.40	1.06
4	0.600	5	0.159	38.	0.936	1.38	1.05
3	0.300	5	0.169	40.	0.928	1.38	1.04

Table 12.15: Results generated by the bivariate MINGCV algorithm for the data set peaks15.dat.

Convergence is quite fast and stable on all remaining grids. The RMS residual shows that the data are fitted more closely as the grids are refined. According to previously observed trends we would expect the smoothing parameter estimate to increase with refinement, but this is not the case for the grid numbers 6 and 5. This is further evidence that the coarse grid solution was not an accurate representation.
The results for both peaks.dat and peaks15.dat indicate that, for data sets with a clumped distribution more refinement will be necessary to represent the trends in the clumps, even though the underlying process might be quite smooth.

Peaks0.dat

As the name suggests, peaks0.dat is the data set peaks.dat with no noise. Although this is unrealistic, it was chosen to test the algorithm by combining very fine scale trends with large areas of data sparsity. This could force the algorithm to have to solve poorly conditioned equations. The data set and analytic spline fit are shown in Figure 12.32.
The results of running the MINGCV algorithm in Tables 12.16 and section C.8 show similar behaviour to that observed for franke4.dat, the other data set featuring zero noise. The θ estimate hits the minimum possible value on all grids except the last two grids, on which it fails to optimise. Although the algorithm does not converge, there are no signs that poor conditioning on fine grids is affecting the performance. Tables 12.17 and 12.16 and Figures 12.33, 12.34 and 12.35 show that the MINGCV estimate of the analytic solution is accurate away from the end regions.

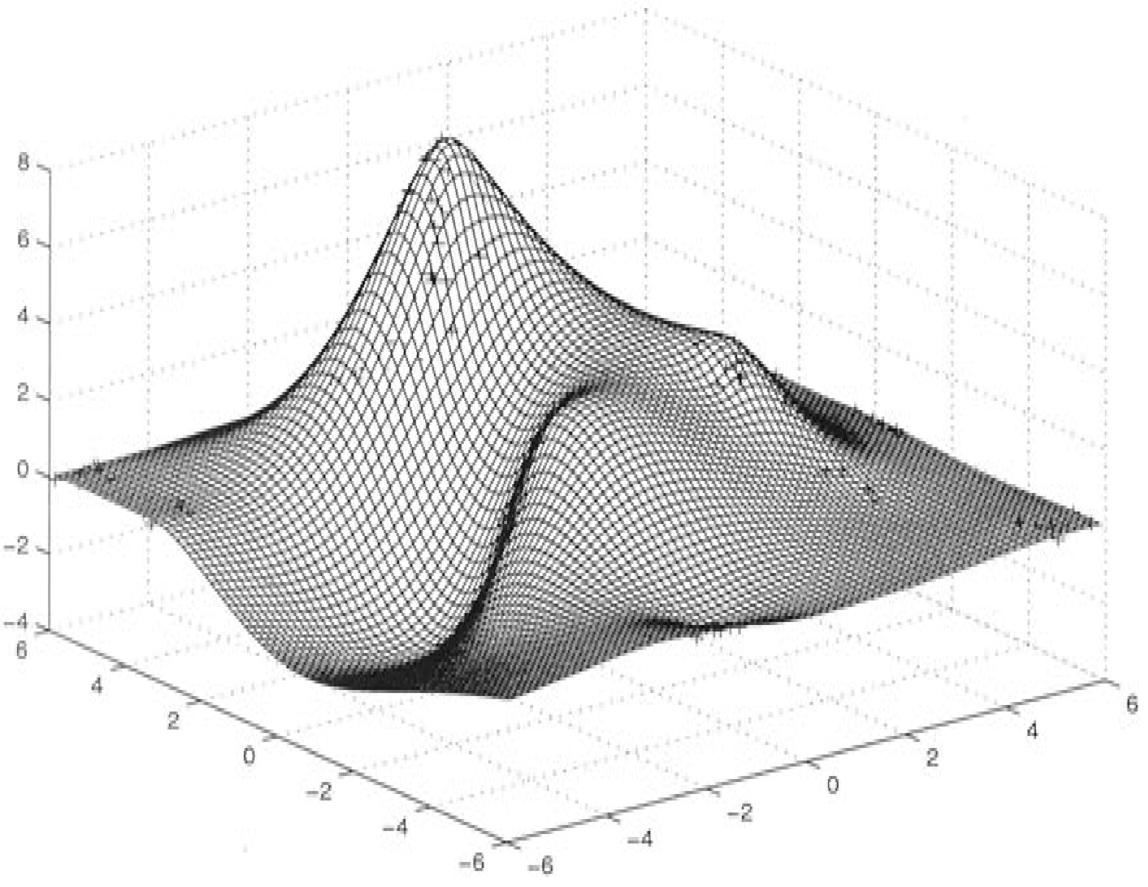


Figure 12.32: Analytic thin plate smoothing spline fit to the data set peaks0.dat.

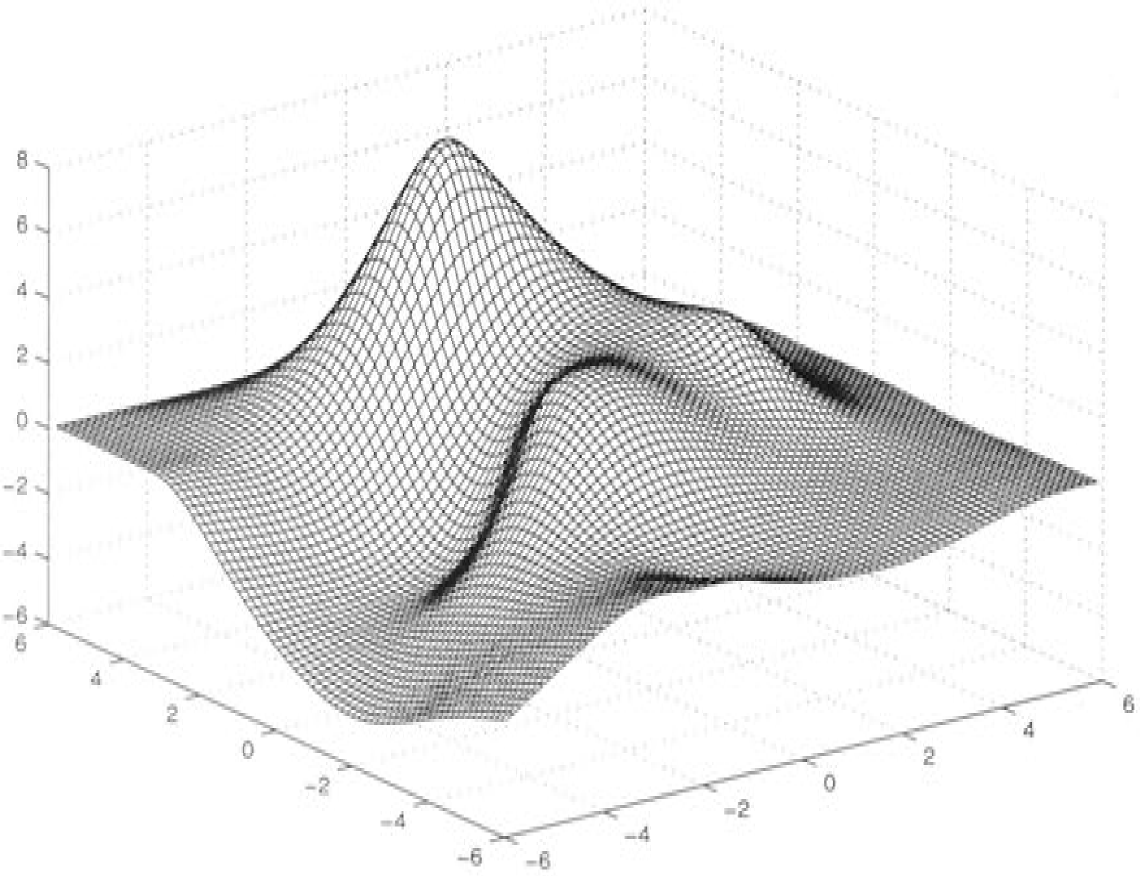


Figure 12.33: Biquadratic B-spline fit to the data set peaks0.dat.

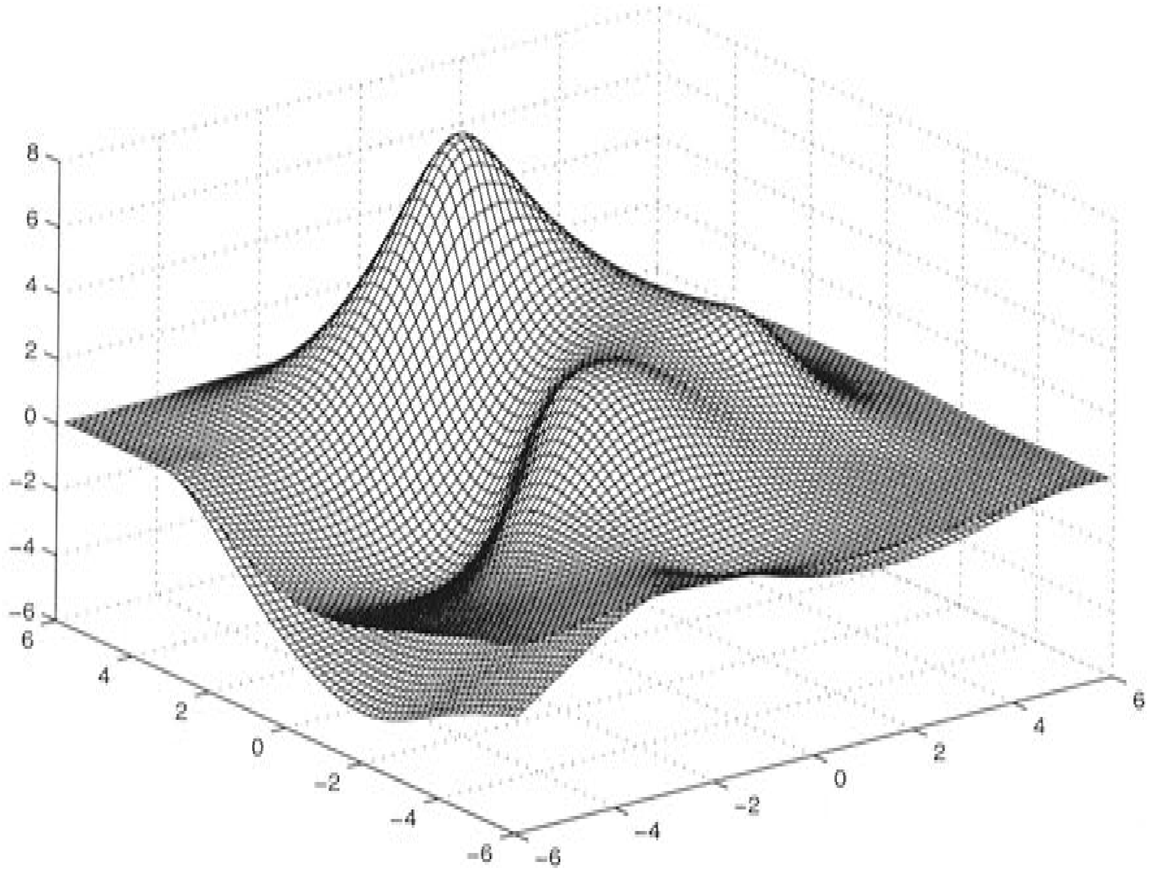


Figure 12.34: Overlay of the biquadratic B-spline solution and the analytic solution for the data set peaks0.dat.

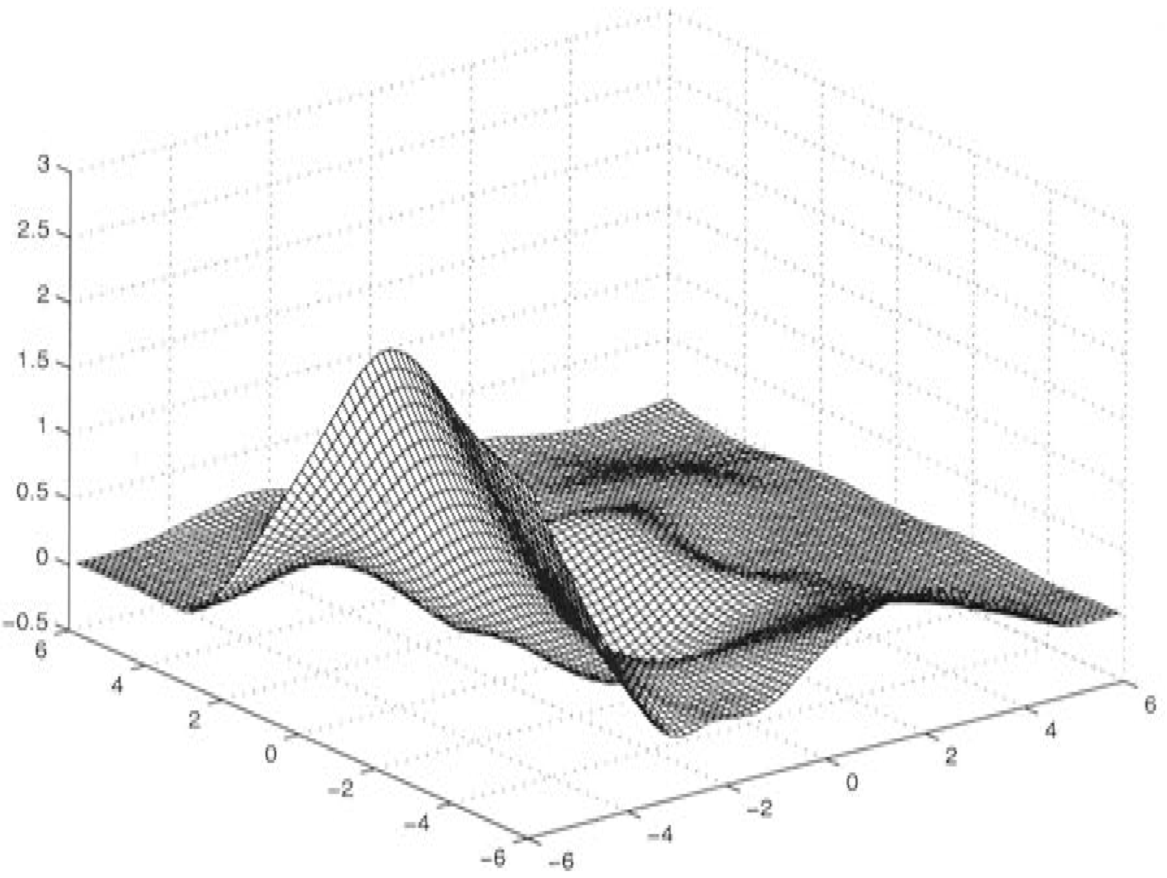


Figure 12.35: Difference between the biquadratic B-spline solution and the analytic solution for the data set peaks0.dat.

Grid no.	h	No. of updates	$\lambda \times 10^4$	Signal	RMS residual	GCV $\times 10^4$	$\hat{\sigma} \times 10$
7	2.40	11	115	38	0.382	2390	0.428
6	1.20	3	28.8	66	0.0458	51.8	0.0567
5	0.600	3	7.20	110	0.00965	5.553	0.0149
4	0.300	2	1.80	150	0.00272	2.71	0.00594
3	0.150	5	0.450	181	0.000650	1.79	0.00303
2	0.075	2	0.213	187	0.000275	0.203	0.00207
1	0.0375	17	0.100	189	0.000148	0.749	0.00193
0	0.0187	5	0.122	189	0.000193	0.213	0.00244

Table 12.16: Results generated by the bivariate MINGCV algorithm for the data set peaks0.dat.

$\lambda \times 10^{11}$	Signal	RMS residual $\times 10^{10}$	GCV $\times 10^3$	$\hat{\sigma} \times 10^6$
0.492	190	0.476	0.195	0.816

Table 12.17: Summary statistics for the analytic thin plate spline fit to the data set peaks0.dat.

12.2 Conclusions

The bivariate MINGCV algorithm has been tested on a number of data sets, with varying complexity, spatial distribution and noise level. As a result of this testing, a number of controls on the behaviour of the algorithm were emplaced, to maintain stability in the face of difficult situations presented by some types of data sets. The resulting MINGCV algorithm gave an accurate representation of the analytic solution for all test data sets at a resolution appropriate to the scale of the data generation process. Convergence was efficient for all data sets, although for the situation of exact interpolation the minimum GCV is poorly defined and the algorithm cannot fully settle. The solution estimates generated in this case are still accurate representations of the analytic solution.

The typical behaviour of the bivariate MINGCV algorithm for smooth processes is to converge quickly on the coarse grid and not refine the resolution. For fine scale processes, the optimum solution cannot be obtained on the coarse grid, as smoothness is enforced by the coarse basis elements. On finer grids, the discretisation has less control on smoothness, and the influence of the roughness penalty term increases. This is often reflected by an increase in the smoothing parameter on finer grids, in order to maintain some of the smoothness that was passed down from the coarse discretisation.

Issues did arise with regions of data sparsity, especially at the edges of grids. In these regions, there was less agreement between the MINGCV approximation and the analytic solution. The MINGCV solution is not as accurate in these regions, because, although the smoothing parameter estimate converges, the solution may not have converged to the analytic solution in these particular areas. This is because the roughness penalty term is dominant for the subsystem corresponding to these areas of the grid, and the subsystem is therefore poorly conditioned on fine grids. The optimal solution at the edge regions may well change considerably from grid to grid, because there are no conditions on the smoothness of the solution beyond the grid boundary. The algorithm therefore has to solve for smooth changes in the solution on fine grids, a situation which is known from previous analysis to cause instability. This phenomenon will be further discussed in the following chapter.

Chapter 13

Performance of the MINGCV algorithm for large temperature data sets

Previous chapters have documented the development and refinement of the bivariate MINGCV algorithm. The final algorithm was shown to yield an accurate approximation to the analytic solution for a wide range of simulated data sets with varying noise, spatial distribution and spatial complexity. However, simulated data sets are limited in their ability to recreate the features of a 'real' environmental data set consisting of noisy observations of an environmental process. The simulated data sets used in Chapter 12 were all generated from a relatively simple functions, so that only a limited amount of data are required to capture all the variability. Environmental processes often feature complex interactions both at the regional scale and the microscale. A denser data set can therefore give rise to quite a different interpolated surface than a sparser data set. The implications of this are demonstrated in the examples below, which consider temperature data for both the African and Australian continents. Temperature was chosen because, once an accurate estimate of the elevation lapse rate is known, past research shows that annual temperature can be accurately spatially interpolated with a bivariate thin plate spline with independent variables latitude and longitude [84, 60].

Maximum and minimum temperatures at various scales are known to be influenced by a variety of factors. Jarvis and Stuart [75] describe both surface and topographic controls on temperature. Features of surface type include the effects of absorbed solar radiation, surface roughness, internal boundary layers and urbanisation. The latter two tend to be microscale effects. Effects of internal boundary layers are difficult to incorporate into interpolated surfaces because the digital land cover and soils data that affect such processes are often not available at fine scales. Urban heating effects are also localised in that the influence on temperature declines steeply at the edge of suburban areas [75].

These effects interact with topographic controls. At a broader scale, temperature declines roughly linearly with increasing elevation, at a typical 'standard' lapse rate of -6.5^oC per kilometre, though this does vary with season and location. However, cold air drainage on slopes operates on cold nights to produce local lapse rate inversions [60]. Other influences include the evaporation rate on slopes, given that damp soils create a cooling effect. This is one factor that lowers the temperature at valley bottoms. It has been incorporated into daily temperature interpolation models by Jarvis and Stuart [75] by representing the distance to nearest rivers. Changes in slope also affect the depth of the surface-cooled atmostpheric layer, which is greater on concave rather than convex surfaces, leading to lower minima on concave slopes. The height of a point relative to the valley floor has also been considered as a measure of susceptibility to frost [107]. Further effects include adiabatic warming as a result of the descent of air from mountains and plateaus by fohn winds, as well as coastal shape and situation [75, 76].

Temperature is therefore driven by many different processes that operate to varying extents on different spatial scales. From a spatial interpolation perspective, the ability of the fitted surface to detect these different levels of variation depends on the density of the data network, and the time scale at which the interpolation is required. Averaging the data to represent long term monthly mean or annual mean temperature removes significant variability from the system. Studies by Hutchinson [60] and Price et al. [96] fitted thin plate smoothing splines to monthly mean maximum and minimum temperature using latitude, longitude and elevation as explanatory variables. Both reported standard errors of around 0.5^oC for minimum temperature, and smaller errors for maximum temperature. Hutchinson [60] attributed the higher errors in minimum temperature interpolation to cold air drainage in winter months. This could not be predicted by the partial spline model used in the study, which assumes a spatially constant elevation lapse rate (see equation (1.6)).

Jarvis and Stuart [75, 76] is the only study to report partial spline fitting of daily temperature. Given the high complexity of temperature influences at a daily time scale, a partial spline with 10 guiding covariates was used. The study found that maximum temperature was consistently better explained by fewer guiding covariates than minimum temperature. Maximum temperature was also determined to be more strongly influenced by terrain.

Long term mean values of monthly maximum temperature for the both the African continent and the Australian continent were used to test the MINGCV algorithm. As discussed above, a partial spline model with latitude and longitude as independent variables, and a linear dependence on elevation, is appropriate for interpolating average temperature, particularly average maximum temperature. The partial spline model is given in equation (1.6). Given that the MINGCV algorithm only allows 2 independent variables, ANUSPLIN was used to estimate the elevation lapse rate by fitting a partial spline. The linear elevation trend was then removed from the data to correct the measurements to sea level. This is a common practice in temperature modeling (e.g. Lennon and Turner [84]). A bivariate spline can then be fitted to the corrected data.

13.1 Spatial interpolation of temperature data for the African continent

There were a total of 1498 points in the African temperature data set. The distribution of the data values is shown in Figure 13.1. There is relatively even coverage throughout most of the continent. However, the shape of the continent is such that a significant proportion of a rectangular grid fitted over this continent will be uncovered with data, particularly in the lower left corner. This is therefore a challenging situation for the MINGCV algorithm, given that large sparse areas at grid boundaries lead to poor conditioning, as discussed in Chapter 12.

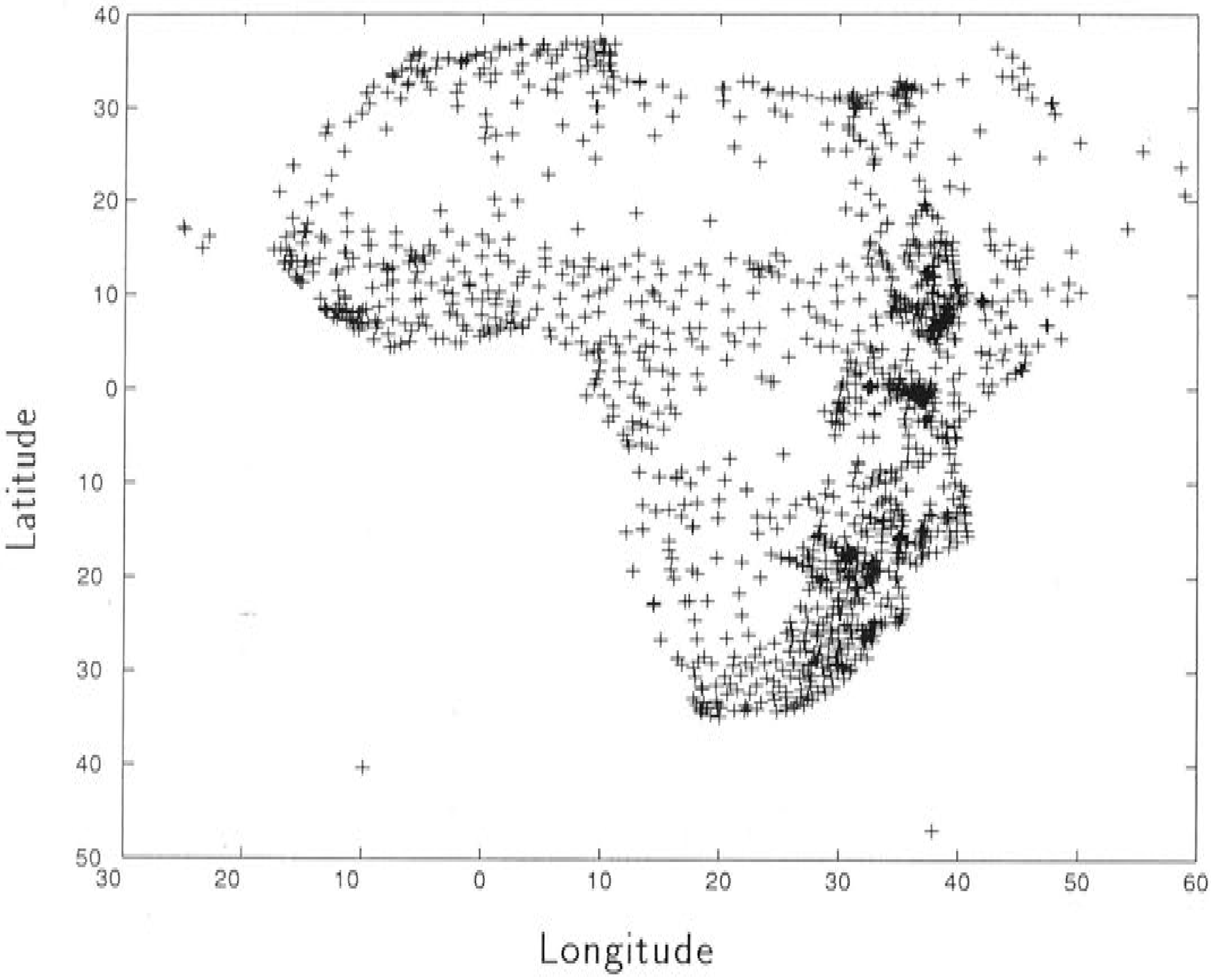


Figure 13.1: Data point locations for the African temperature data set.

1498 points is too many for efficient computation of the analytic spline, so the SPLINB routine within ANUSPLIN was used to calculate an approximation using a subset of 1000 knots [68]. The SPLINB results are shown in Table 13.1. The elevation lapse rate was estimated to be -6.22^o per kilometre, close to the standard rate.

λ	Signal	RMS residual	GCV	$\hat{\sigma^2}$
0.0925	469.1	0.829	1.46	1.00

Table 13.1: Summary statistics generated by SPLINB for the African temperature data set.

The results for the MINGCV algorithm are shown in Table 13.2, with full results in section D.1. The results are very close to those obtained for the partial spline fitted by SPLINB using three independent variables, latitude, longitude and elevation.

The final grid resolution of 1^o is quite fine, implying local scale spatial trends

Grid no.	h	No. of updates	λ	Signal	RMS residual	GCV	$\hat{\sigma}$
6	16.0	12	0.512	35.4	1.72	3.13	1.75
5	8.00	5	0.128	83.0	1.39	2.17	1.43
4	4.00	5	0.0320	212.	1.10	1.66	1.19
3	2.00	5	0.0643	381.	0.906	1.48	1.05
2	1.00	5	0.0732	494.	0.796	1.42	0.973

Table 13.2: Results generated by the bivariate MINGCV algorithm for the African temperature data set.

have been detected in long term annual mean temperature for Africa. Section D.1 shows that the θ estimate decreased to the minimum possible limit on the first 3 grids, and then converged on the final two grids. The signal is slightly greater that a quarter of the number of data points, which indicates that the smoothing procedure has reliably isolated the noise in the data. The refinements show a steady increase in fine scale structure, as reflected by the increasing signal values and decreasing RMS residual. The GCV improves with each refinement, though it has clearly stabilised on grid numbers 3 and 2.

The surface produced by the above run is shown in Figure 13.2. Bearing in mind that the surface represents sea level temperatures, the map shows a general trend of increasing temperatures with decreasing distance from the equator and increasing distance from the coast, as expected.

These positive results are encouraging, and appear to indicate that the MINGCV algorithm is robust in dealing with real environmental data sets with a sparse distribution of data points. However, slight modifications of the algorithm revealed that it is in fact unstable for this data set. The above analysis had an initial resolution of 16^o. The resolution of the coarsest grid depends on the desired resolution of the output grid, which is specified by the user. The coarse resolution is determined by successive doubling of the output grid resolution until further doubling would result in a coarse grid with less than 3 rows or 3 columns, which could not support a single basis element. When the output resolution was modified to give an initial coarse resolution of 25.6^o, it failed to converge. The results are shown in Table 13.3 and section D.2.

The θ estimate hits the minimum possible value on the first 4 grids. On grid number 5, the θ estimate begins to increase, and continues to become much larger than the value corresponding to the minimum GCV. As a result

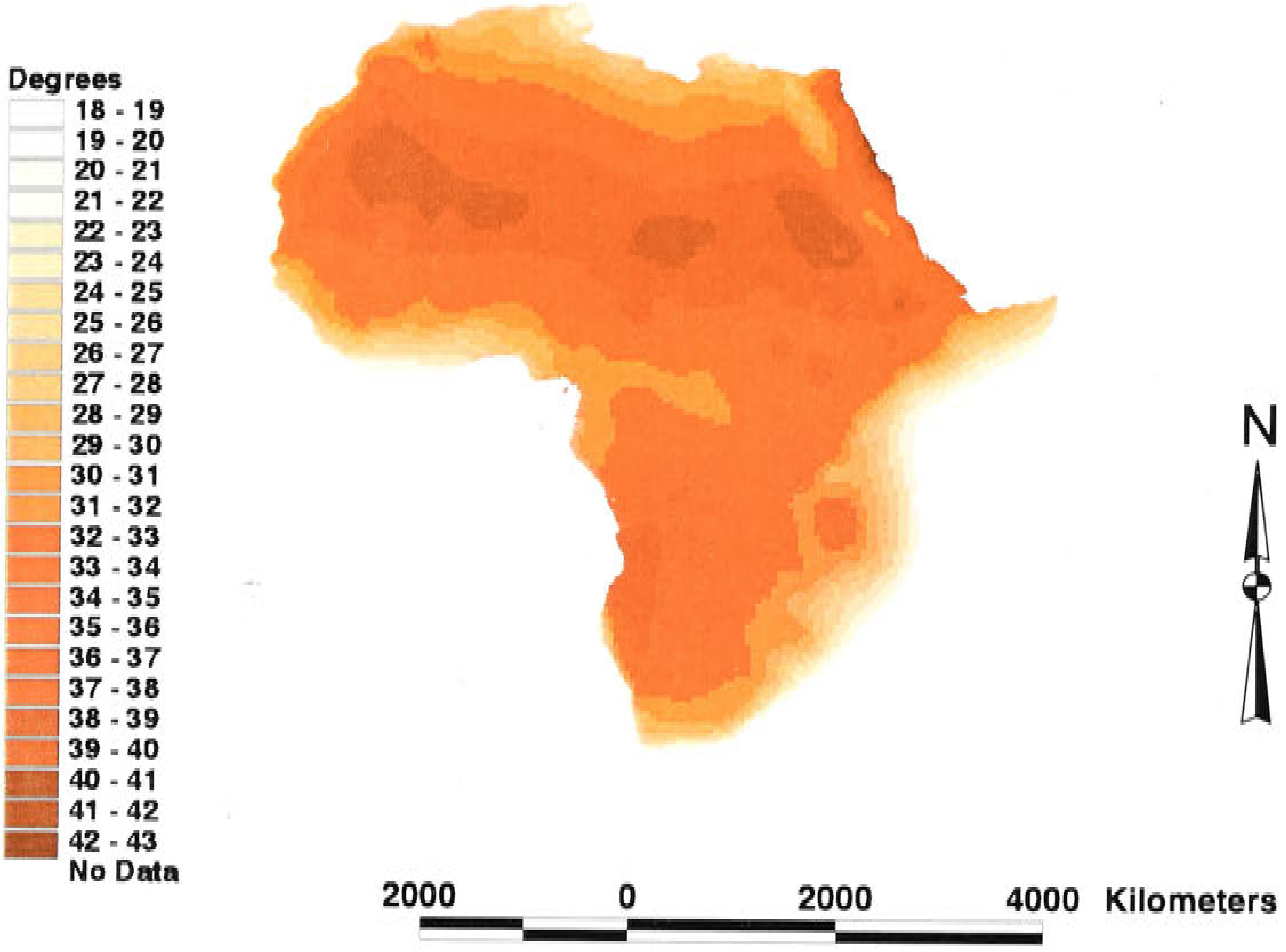


Figure 13.2: Minimum GCV biquadratic B-spline surface representing annual mean maximum temperature, standardised to sea-level, for the African continent.

Grid no.	h	No. of updates	λ	Signal	RMS residual	GCV	$\hat{\sigma}$
8	25.6	7	1.31	19.2	2.00	4.12	2.02
7	12.8	4	0.328	44.9	1.60	2.73	1.63
6	6.40	5	0.0819	109.	1.34	2.11	1.40
5	3.20	5	0.0205	290.	1.02	1.62	1.14
4	1.60	6	0.249	329.	1.01	1.54	1.14
3	0.800	2	0.528	299.	1.18	1.83	1.32
2	0.400	2	1.12	271.	1.81	3.45	2.01
1	0.200	2	2.00	264.	2.99	11.4	3.30
0	0.100	1	0.500	264.	1.15	13.2	1.27

Table 13.3: Results generated by the bivariate MINGCV algorithm for the African temperature data set, with an initial grid resolution of 25.6^o.

the GCV begins to increase markedly and so does R. This indicates that the derivative estimates were unable to detect a minimum GCV when θ was in the vicinity of the optimal value. Once again this is typical behaviour of the MINGCV algorithm on fine grids where the system is poorly conditioned, and was observed many times during the unidimensional analysis described in Chapter 11. Poor conditioning causes slow convergence of the SOR iterations, which results in poor synchronisation of the θ updates and the updates of the solution and its derivatives.

The errant solution estimate is shown in Figure 13.3. This figure clearly shows the cause of the problem. The large area of ocean in the lower left corner of the grid has produced a poorly conditioned system. The problem of poor convergence in data sparse areas near boundaries was discussed in Chapter 12. The algorithm essentially has to solve the biharmonic equation at a fine resolution, with little help from coarse resolutions due to lack of fixed specification of the solution values beyond the grid boundaries. The solution estimate has therefore diverged in this region of the grid, which then introduced error into the calculation of the solution estimate and corresponding derivatives on the rest of the grid. The reason why this occurred for the initial resolution of 25.6^o and not the initial resolution of 16^o has not been attributed to any particular process. It is assumed that, for this data set, chances of divergence are significant, so small changes in the resolution of each grid may trigger divergence.

A practical solution to this problem is to reduce the areas of data sparsity by dividing the continent up into 2 sections and solving for splines on the two grids separately. Two sections which avoid empty space are shown in Figure 13.4. With this subdivision we have eliminated virtually all the empty space that was problematic in the above analysis. The disadvantage of this approach is clearly that less data and therefore less information is available for the interpolation on each section. Summary results for the two halves are shown in Tables 13.5 and 13.6.

Grid no.	h	No. of updates	λ	Signal	RMS residual	GCV	$\hat{\sigma}$
6	8.00	10	0.128	54.6	1.46	2.42	1.51
5	4.00	4	0.0320	151.	1.17	2.04	1.29
4	2.00	8	0.135	216.	1.00	1.78	1.16
3	1.00	7	0.164	254.	0.942	1.76	1.12

Table 13.4: Results generated by the MINGCV algorithm for African temperature data set, for the top section of the African continent.

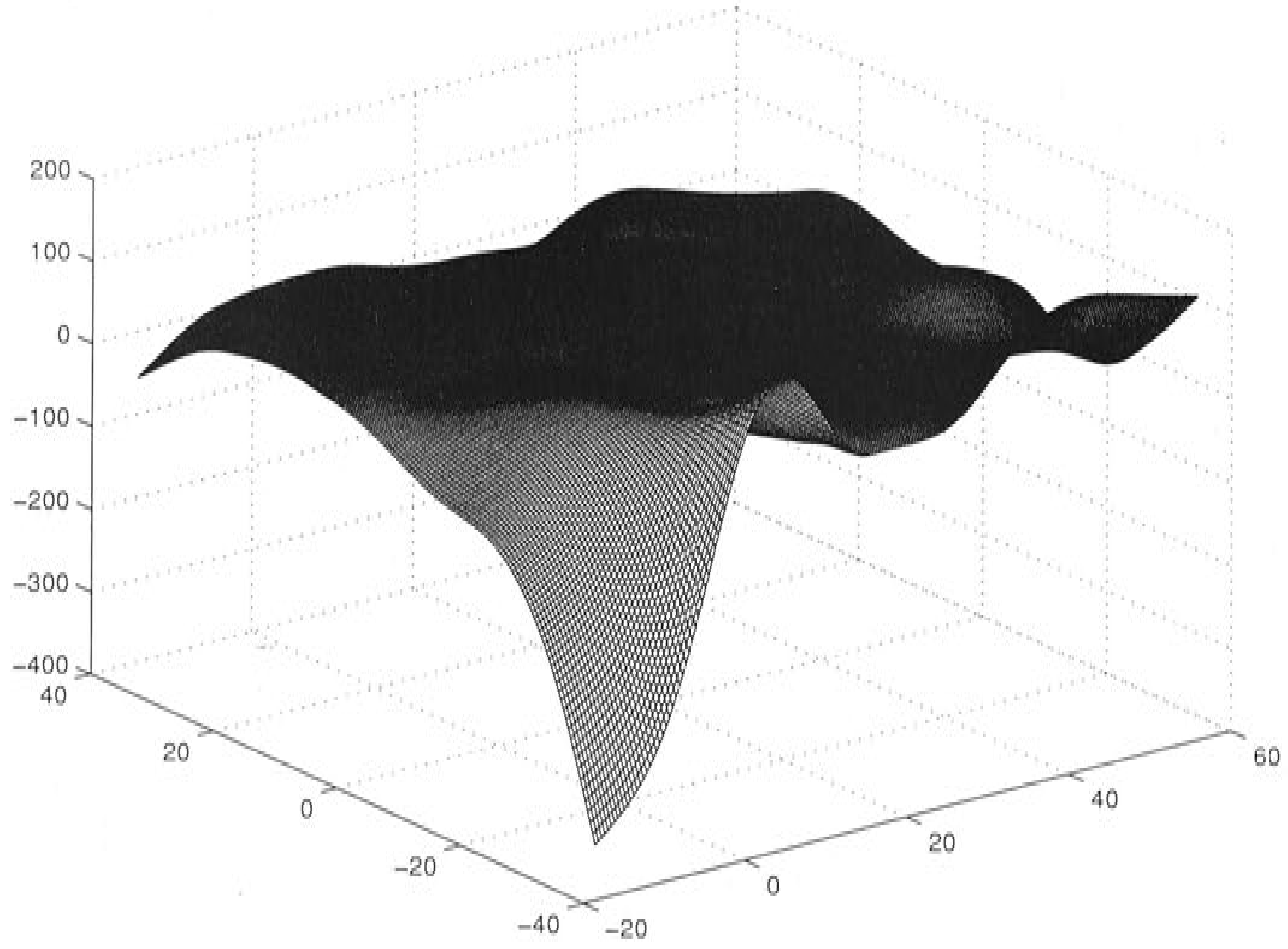


Figure 13.3: Surface produced by the bivariate MINGCV algorithm for the African temperature data set, with initial grid resolution 25.6° instead of 16°.

Comparing these Tables with Table 13.2, it can be seen that both halves converged to solutions with similar GCV, R and resolution to the MINGCV solution for the full data set. The surface produced by combining these two halves is shown in Figure 13.5. The patterns and trends are similar to those in Figure 13.2, with some fine scale differences. In general the surface generated by combining two sections predicts slightly cooler temperatures in the top half

Grid no.	h	No. of updates	λ	Signal	RMS residual	GCV	$\hat{\sigma}$
6	16.0	9	0.512	17.3	1.69	3.03	1.71
5	8.00	6	0.128	39.1	1.42	2.23	1.45
4	4.00	4	0.0320	97.1	1.10	1.58	1.17
3	2.00	7	0.0310	206.	0.859	1.29	0.987
2	1.00	7	0.0577	267.	0.770	1.26	0.930

Table 13.5: Results generated by the MINGCV algorithm for the African temperature data set, for the bottom half of the African continent.

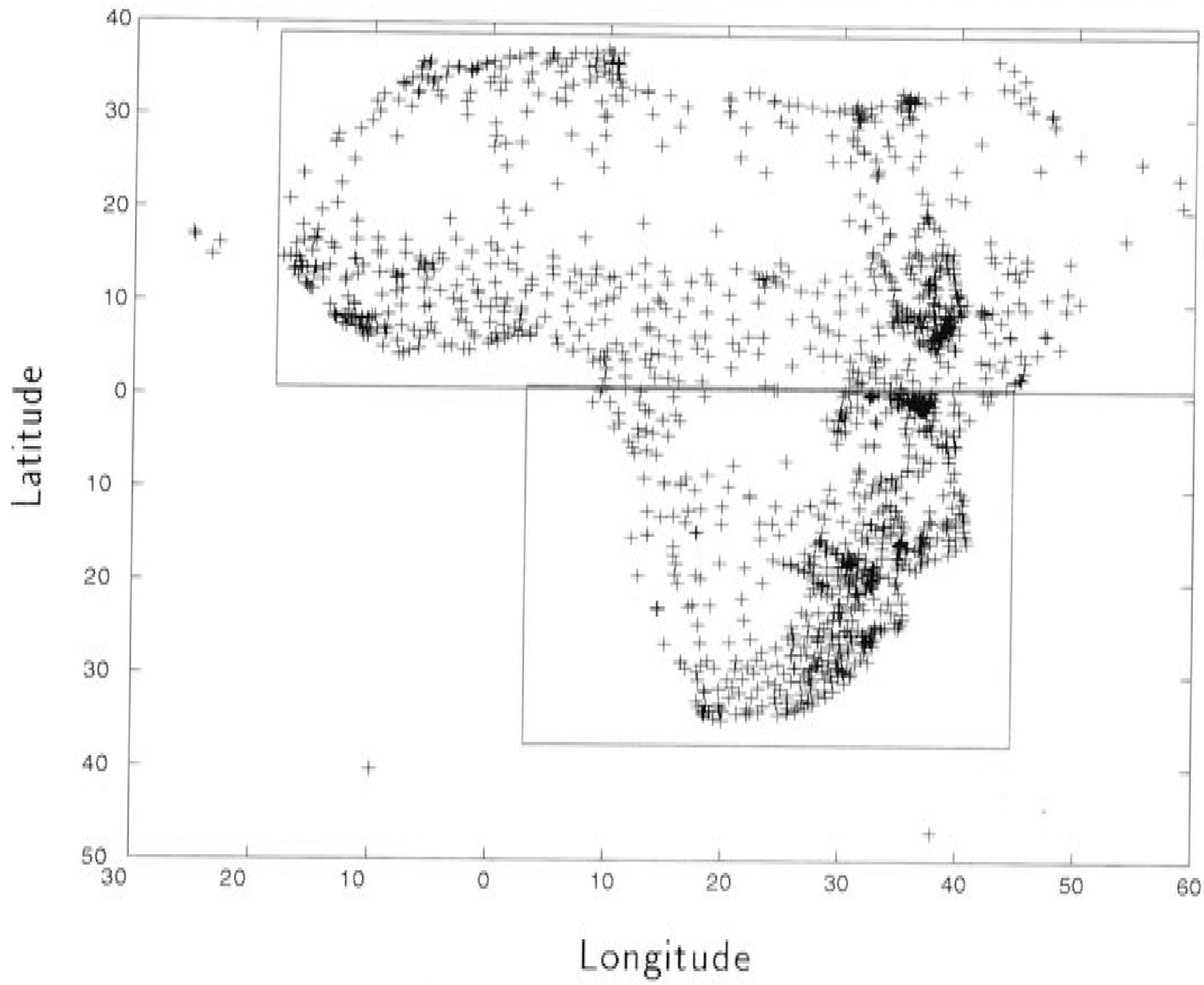


Figure 13.4: The two grids used to cover the African continent and avoid large areas of ocean.

of the continent. Given that the differences are relatively minor, this option is still an efficient alternative to analytic methods for a large data set.

13.2 Spatial interpolation of temperature data for the Australian continent

As a second test, the MINGCV algorithm was run on annual mean maximum temperature data, standardised to sea-level, for the Australian continent. The data set, consisting of 1134 data observations, is mapped in Figure 13.6. The distribution is clumped and uneven, with a strong bias towards coastal regions. However, the shape of the continent is more rectangular than that of the African continent, so there is less of a problem with large data sparse holes at the grid boundary. According to previous analysis, particularly that with the

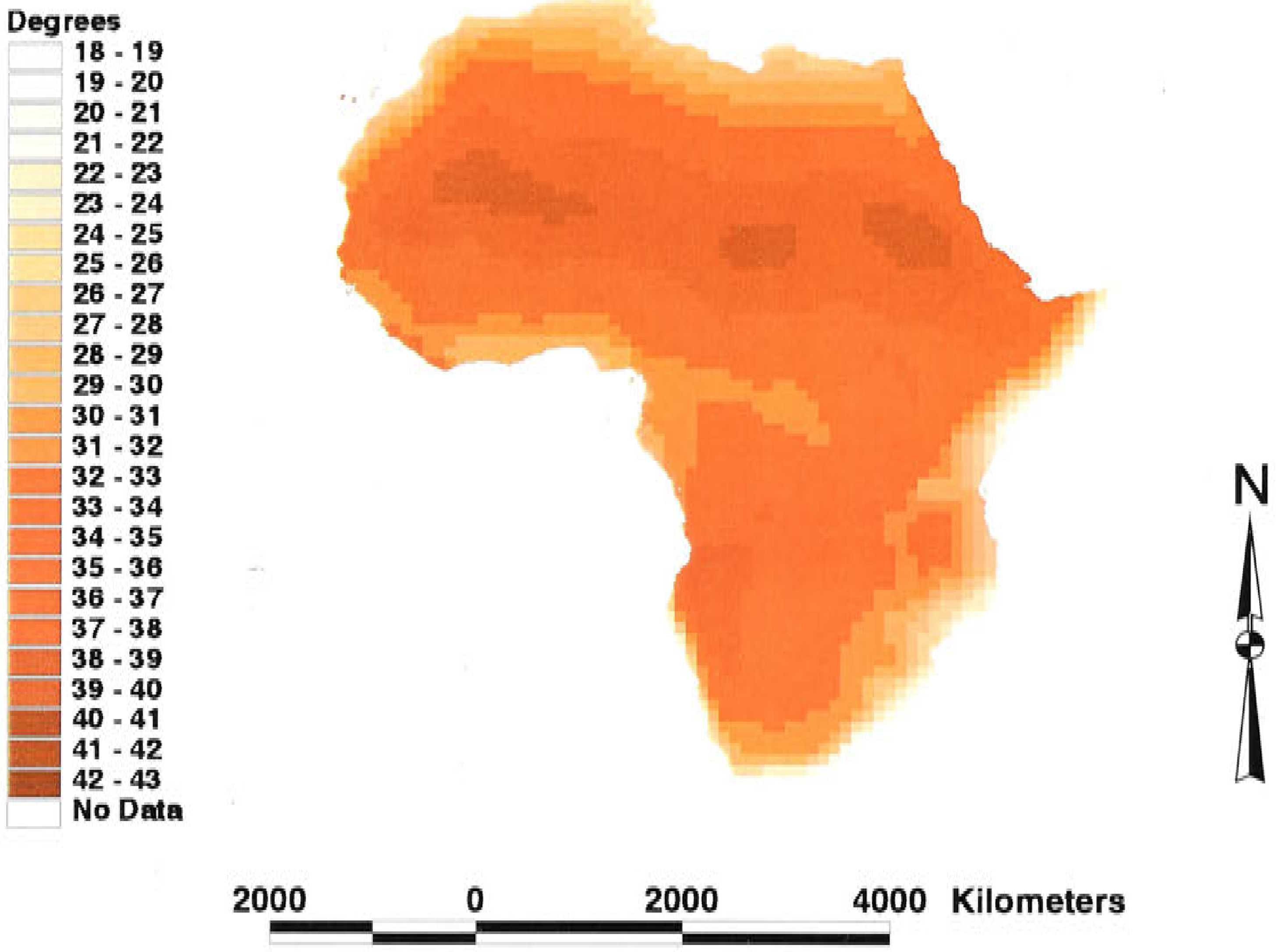


Figure 13.5: Combination of temperature surfaces produced by the bivariate MINGCV algorithm for the top and bottom segments of the African continent.

peaks.dat data set in Chapter 12, we would expect the MINGCV algorithm to converge for this data set.

This time, SPLINA was used to calculate the analytic partial spline solution, instead of using the SPLINB knot based approximation as was done for the African data set. The data set was regarded as just small enough for a single SPLINA run. The results are shown in Table 13.6.

λ	Signal	RMS residual	GCV	$\hat{\sigma}^2$
0.0335	358.5	0.497	0.527	0.600

Table 13.6: Summary statistics generated by SPLINA for the Australian temperature data set.

The partial spline estimate of the lapse rate was -7.67^o per kilometer, higher than that for Africa. After using this lapse rate to standardise the data to

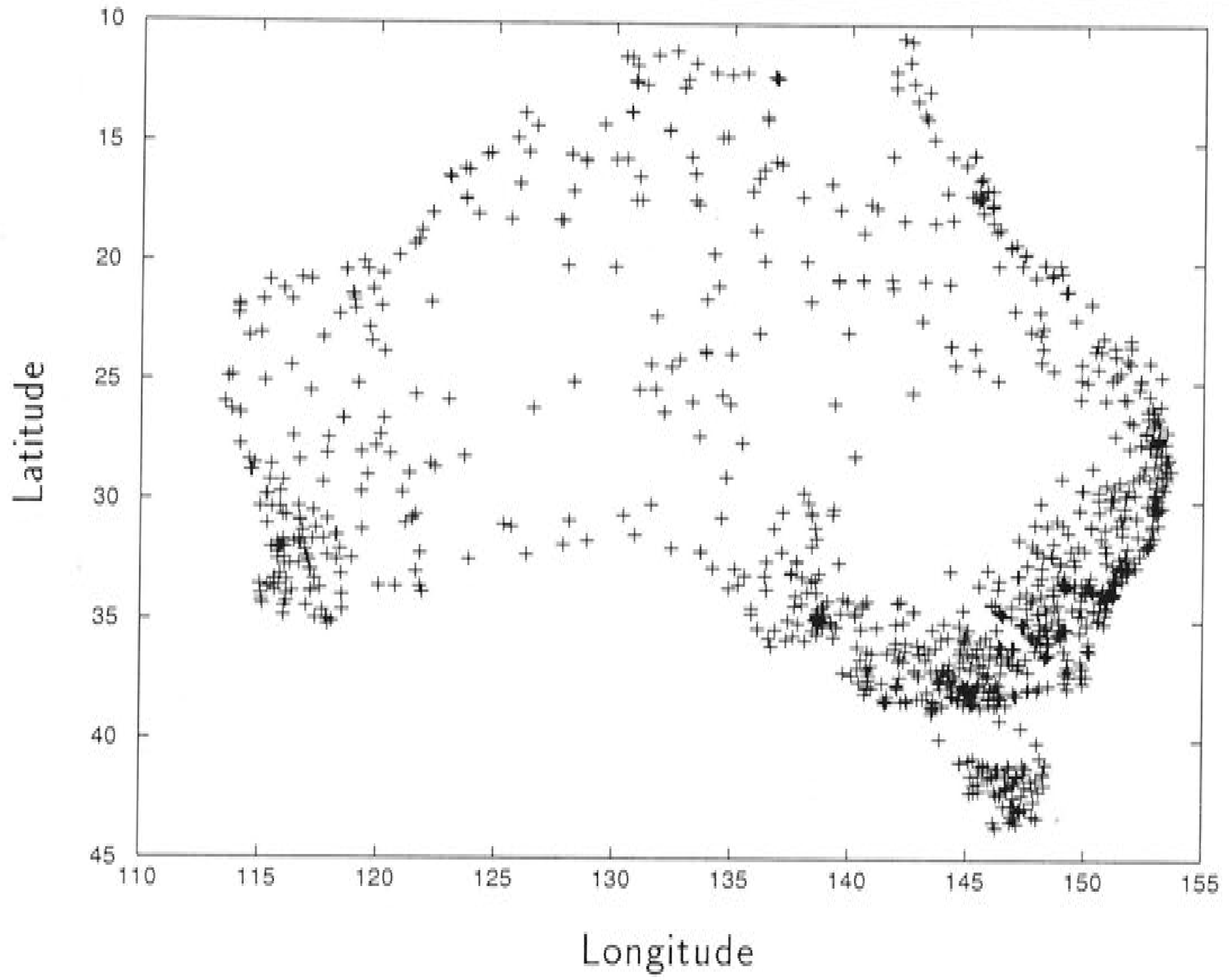


Figure 13.6: Data point locations for the Australian temperature data set.

sea level, the bivariate MINGCV algorithm was run to produce the results in Table 13.7 and section D.3. The statistical characteristics of the MINGCV solution agree well with those of the analytic solution. Estimates of the GCV and $\hat{\sigma}$ are almost identical. The signal is well below half the number of data points, which indicates the processes are broadscale. This is demonstrated in Figure 13.8, which shows a gradual inland gradient.

The results are similar to those for the Africa data set in Table 13.2. If anything, the statistics indicate less noise for the Australian data than for the African data, given that the final resolution is finer, and the $\hat{\sigma}$ estimate is lower, for the Australian data set. This could be expected, as the Australian data is of higher quality than the African data set, with fewer errors in the recorded station locations and data values.

The more detailed results for the MINGCV algorithm in section D.3 indicate that the θ estimate hit the minimum possible value on the 3 coarsest grids. On the two final grids, convergence was reasonably fast. The θ estimate does not

Grid no.	h	No. of updates	λ	Signal	RMS residual	GCV	$\hat{\sigma}$
6	8.00	8	0.128	34.8	0.947	0.959	0.962
5	4.00	3	0.0320	81.8	0.809	0.766	0.840
4	2.00	7	0.0131	188.	0.644	0.597	0.705
3	1.00	5	0.0272	270.	0.559	0.538	0.640
2	0.500	5	0.0271	336.	0.512	0.529	0.610

Table 13.7: Results generated by the bivariate MINGCV algorithm, for the Australian temperature data set.

increase markedly on these finer grids, but the signal increases significantly. This shows the inability of the coarser grid to represent the trends in this data set.

The solution produced by the MINGCV algorithm, shown in Figure 13.7, shows no signs of instability at the grid boundaries. Testing of the algorithm using different initial grid resolutions was performed to see whether the process suffered from similar instability to that observed for the African temperature data. As anticipated, the algorithm converged for all the tests. Given the rectangular shape of the Australian continent, and observing the stable behaviour at the boundaries in Figure 13.7, we would expect the MINGCV algorithm to converge efficiently for this data set.

13.3 The computational saving

To give a practical perspective on the computational efficiency of the MINGCV method, we compare the number operations required by this method to that required for a non-automated procedure for approximating finite element minimum GCV thin plate smoothing splines. The automised minimisation of the GCV has two main sources of computational workload; the solution of the equations for $d\mathbf{u}/d\theta$ and the extra iteration required to adjust to periodic changing of the smoothing parameter. The first source doubles the workload in comparison to that required to solve the system for a fixed smoothing parameter, and we estimate that the second source triples the workload. Most of the work in finding the structure of the optimal solution is done on the coarse grids, and iteration on fine grids serves as a fine tuning process. So the cost of updating the smoothing parameter is kept to a minimum. In total this estimate means that the automated minimisation of the GCV makes the solution process 6 times as slow as the process for solving for a fixed smoothing parameter. To manually minimise the GCV, we estimate that the finite element system would have to be solved around 30 times in order to converge to an accurate estimate of the minimum GCV smoothing parameter. This is 5 times more work than the automated procedure.

Another advantage of the MINGCV algorithm is that it determines an appropriate grid resolution for representation of the optimal solution. By simultaneously solving for the optimal smoothing parameter whilst solving the spline system, the MINGCV algorithm has a reasonably accurate estimate of the optimal solution by the time the finer resolutions are reached, so a suitable resolution for the final grid can be accurately determined. To solve the system for several fixed smoothing parameters, one would have to determine an appropriate final resolution each time. As many of the trial parameters will not be close to the optimal value, the solution process may involve visiting grids that whose resolution is unnecessarily fine.

13.4 Conclusion

Accurate approximation of analytic bivariate thin plate smoothing spline solutions were obtained for real environmental data sets using the MINGCV algorithm. The algorithm converged for sea-level temperature data for both the African and the Australian continent. The problems associated with lower accuracy at grid boundaries noted in Chapter 12 were more extensive for the African temperature data set. For some initial grid resolutions, the MINGCV algorithm failed to converge to a solution for this data set, due to divergence at the lower left grid boundary. This phenomenon was discussed in previous chapters, although African temperature presented the only data set so far for which the bivariate MINGCV algorithm did not converge to a solution. The unstable behaviour occurs when the algorithm has to represent a process with fine scale variability with data empty regions surrounding grid boundaries. The subsystem of equations corresponding to the data empty regions is ill-conditioned, resulting in poor synchronisation of the double iteration. One option in this situation is to subdivide the region and solve for the spline solution on each section, avoiding data empty 'holes'. This was demonstrated in the above analysis to be an efficient alternative for the African temperature data set, although it is clearly not as accurate in approximating the analytic solution.

Temperature data from the more rectangular shaped continent of Australia presented no problem for the MINGCV algorithm. The algorithm was found to be stable, efficient and accurate in approximating the analytic solution for this data set. This result emphasises the robustness of the MINGCV algorithm, considering that the coverage of the data throughout the Australian continent is clumped and sparse. The above analysis demonstrates that thin plate smoothing spline interpolation of environmental data can be performed efficiently using the MINGCV algorithm. The algorithm is therefore useful in environmental modelling and management applications.

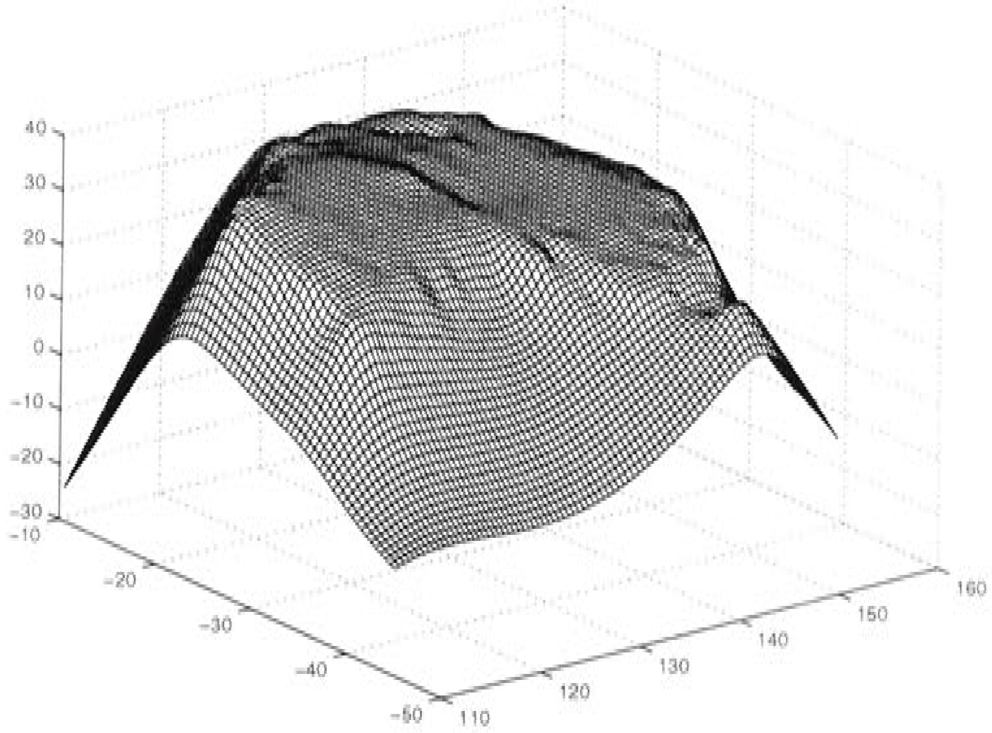


Figure 13.7: Minimum GCV biquadratic B-spline surface for mean annual temperature, standardised to sea level, for the Australian continent.

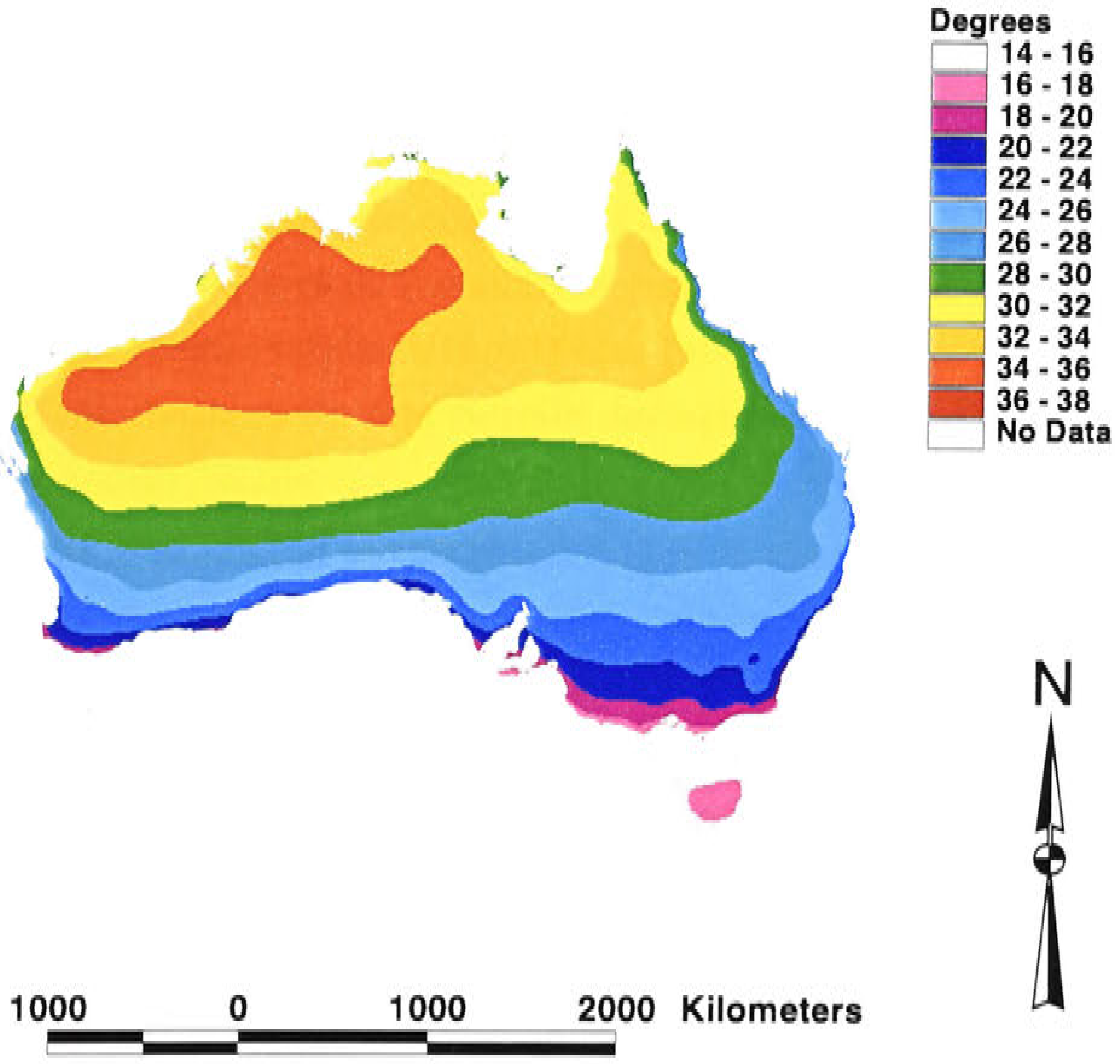


Figure 13.8: Map showing the surface in 13.7.

Chapter 14

Conclusion

This project developed an efficient, accurate, robust algorithm for generating finite element approximations to minimum GCV bivariate thin plate smoothing splines for large data sets. A diverse range of simulated data sets was used to test the algorithm, representing processes of varying spatial scales, from smooth, broadscale processes to those with more fine scale trends. The distribution of the data points was varied, so that the data sets included large data sparse areas and dense clumps of data points. The algorithm generated accurate approximations to the analytic solution for all test data sets. Testing on continental temperature data sets also produced accurate surfaces, demonstrating that the algorithm is suitable for application to large environmental data sets.

The algorithm presented in this thesis builds on past research of techniques for fast computation of thin plate smoothing splines for large data sets using finite element discretisations, including studies by Terzopoulos [108], O'Sullivan [94], Szeliski [106] and Hutchinson [67]. This study adopted a simple finite element discretisation using biquadratic splines, and presented an efficient iterative scheme for optimising the smoothing parameter in the thin plate smoothing spline equations to minimise the GCV. Optimising smoothness by minimising GCV is well known to be a suitable way of minimising the prediction error of the fitted surface, and is therefore important in many applications of thin plate smoothing splines. In this case, the primary motivation was the spatial interpolation of surface climate data, which often require fitting minimum

GCV thin plate smoothing splines to thousands of data points corresponding to a continent wide network of weather stations [70].

During the course of this project, a number of approaches to discretising the thin plate smoothing spline system and numerically solving for the optimal solution were investigated. Initial testing of different multigrid approaches for numerically solving the univariate discretised system emphasised the advantages of choosing a simple nested grid algorithm to solve for discretised smoothing splines. This testing showed that the smoothing spline equations are poorly conditioned on fine grid resolutions. When fitted to noisy data, smoothing spline solutions inherently contain smooth broad scale components, which means that the underlying structure cannot be efficiently obtained by iterating on fine grid resolutions. The v-cycle multigrid algorithm, which begins the solution process at the finest resolution, therefore performed poorly for the smoothing spline system. The nested grid algorithm, which starts at the coarsest resolution, is ideal for efficiently obtaining the underlying broadscale trends in the spline solution. Finer scale structure can then be developed by refining the grid resolution.

The efficiency and robustness of the algorithm developed by this project is largely due the implementation of the quadratic B-spline framework. Comparison of the results of discretising the system using piecewise constants with those generated using quadratic B-splines demonstrated many advantages of the quadratic B-spline basis for approximating thin plate smoothing spline solutions. The first order continuity of the quadratic B-spline approximation significantly improved the accuracy of the discretised solution on coarse grids, in comparison to the piecewise constant approximation. This meant that the smooth components of the solution were accurately estimated before transfer to finer grids. The lower discretisation error of the quadratic B-spline solution had further advantages when optimising the GCV, because the approximation was better able to reflect the properties of the analytic solution, especially with regard to the variation of the GCV with the smoothing parameter.

The techniques developed during this study for transferring quadratic B-spline functions between grids of varying coarseness allowed for efficient, accurate intergrid transfer. Preliminary testing of the algorithm showed that this was important in maintaining a well conditioned system, in that accurate prolongation of a smooth coarse grid estimate avoided the introduction of error components that could not be efficiently reduced on fine resolutions. Quadratic B-spline restriction techniques were also used to develop an effective criterion for determining the final grid resolution. Least squares projection of the fine grid solution onto a coarser grid allowed the change in fine scale variability between the two discretisations to be quantified. The use of this process for deciding when to stop refinement prevented the grids from becoming unnecessarily fine, and thus allowed maintenance of a well conditioned system throughout the solution process. The resolution of the final grid, as determined by the least squares restriction, also provided insight into the scale of the data generation process.

The construction of a method for numerically minimising the GCV required extending the methods in Hutchinson [67]. This involved deriving formulae for calculating the derivatives of the GCV, the residual sum of squares and the trace of the influence matrix with respect to the smoothing parameter. Relationships identified between these derivatives, the solution estimate and derivatives of the solution estimate with respect to λ, led to the derivation of expressions that could be evaluated in a computationally efficient manner. A second order Taylor series expression for the GCV as a function of the smoothing parameter could then be evaluated in order to estimate the point of minimum GCV. This procedure was central to the functioning of the algorithm, and it was found to converge efficiently and accurately.

To estimate GCV values corresponding to different smoothing parameters for the finite element approximation to the thin plate spline solution, it was necessary to use the stochastic estimate of the trace of the influence matrix developed by Hutchinson [59]. Testing of the trace approximation showed that its observed standard error agreed with the bounds published in [59]. The suitability of this estimate for application to large data sets was also confirmed.

Empirical analysis of the behaviour of the algorithm has established the behaviour of the process of double iteration. In the case of this study, double iteration was performed to update two interdependent quantities, the solution estimate and the estimate of the optimal smoothing parameter. It was found that, for poorly conditioned systems, basic iteration was not fast enough to allow the solution estimate to respond to the smoothing parameter updates. It was also observed that, as the difference between current λ estimates and the minimum GCV smoothing parameter increased, the estimates of the derivatives of the GCV were less effective at producing accurate estimates of the minimum GCV smoothing parameter.

The results of this testing led to the identification of appropriate controls on the initial value of λ, the maximum amount by which λ can be updated, and lower and upper bounds on the λ estimate. Convergence was improved by making first order correction to the solution estimate after each λ update. This allowed the solution estimate to respond more quickly to the changes in the smoothing parameter, avoiding inefficient oscillatory patterns in the double iteration. The result was a significant improvement in convergence rates on finer grids.

The production of the bivariate algorithm for application to large data sets involved vector FORTRAN 90 coding. The roughness penalty calculations were efficiently incorporated, taking advantage of the sparseness, bandedness and repetition in the bivariate thin plate smoothing spline system. The computational efficiency of tensor product systems was also exploited. These factors optimised computational speed and storage.

When the final algorithm was tested on 'real' environmental data sets, including temperature data from the Australian and African continents, accurate approximations to the analytic solution were achieved. However, some difficulty was presented by the geometry of the African continent. A large area of ocean in the lower left corner of the grid produced a gap in data coverage extending inland from the lower grid boundary. This part of the solution corresponded to particularly poorly conditioned components in the spline system, and led to instability on fine grids. Further developments of this algorithm could be aimed at addressing this difficulty. To overcome this problem here, the African region was subdivided into two parts to avoid fitting over areas of ocean. The procedure then converged efficiently. For the Australian continent, convergence to an accurate approximation was highly efficient.

In conclusion, this project has addressed thin plate smoothing spline modelling from a number of different perspectives, including numerical discretisation and solution, statistical modelling using thin plate smoothing splines, and practical aspects of spatially interpolating environmental data. This study also demonstrates the process of implementing a numerical algorithm. Each technical component of the algorithm was tested on a wide range of trial data sets. Optimal techniques were selected and combined into procedures for solving the discretised thin plate smoothing spline system. The technical complexity of the procedures was gradually increased by varying the smoothing requirements, to eventually achieve the aim of minimum GCV smoothing. As a result of this process, the behaviour of these procedures, when applied to the thin plate smoothing spline system, is now well understood. The resulting algorithm was designed for spatially interpolating noisy environmental data sets, and has been found to be suitable for this purpose. The relevance of this procedure to environmental modelling is in the efficient optimisation of the finite element approximations to thin plate smoothing splines by minimising GCV. This is the key contribution of this study.

Bibliography

[1] H. Akima. A method for bivariate interpolation and smooth surface fitting for irreguarly distributed data points. *ACM Transactions of Mathematical Software*, 4:148–159, 1978.

[2] I. Altas, J. Dym, M.M. Gupta, and R.P. Manohar. Multigrid solution of automatically generated high-order discretizations for the biharmonic equation. *SIAM J. Sci. Comput.*, 19(5):1575–1585, 1998.

[3] G.T. Anthony and M.G. Cox. The fitting of extremely large data sets by bivariate splines. In J.C. Mason and M.G. Cox, editors, *Algorithms for approximation*. Oxford University Press, New York, 1987.

[4] Z. Bai, M. Fahey, and Golub G. Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, 74:71–89, 1996.

[5] N.S. Bakhvalov. Convergence of a relaxation method with natural constraints on an elliptic operator. *USSR Computational Math. and Math. Phys.*, 6(5):101–135, 1966.

[6] R.E. Bank and C.D. Craig. Sharp estimates for multigrid rates of convergence with general smoothing and acceleration. *SIAM Journal of Numerical Analysis*, 22:617–633, 1985.

[7] R.K. Beatson, G. Goodsell, and M.F.D Powell. On multigrid techniques for thin plate spline interpolation in two dimensions. In *The Mathematics of Numerical Analysis (Park City, UT, 1995) (Providence RI)*, volume 32 of *Lectures in Appl. Math.*, pages 77–97, Providence RI, 1996. Amer. Math. Soc. - SIAM.

[8] R.K. Beatson and G.N. Newman. Fast evaluation of radial basis functions. I. *Comput. Math. Appl.*, 24:7–19, 1992.

[9] R.K. Beatson and M.J.D. Powell. An iterative method for thin plate spline interpolation that employs approximations to Lagrange functions. *Pitman Res. Notes Math. Ser.*, 303:17–39, 1994.

[10] M. Berman. Automated smoothing of image and other regularly spaced data. *IEEE transactions on pattern analysis and machine intelligence*, 16(5):460–468, May 1994.

[11] V. Bhaskaran and K. Konstantinides. *Image and Video Compression Standards: Algorithms and Architectures.* Kulwer Academic Publishers, Boston, MA, 1996.

[12] D. Braess. On the combination of the multigrid method and conjugate gradients. In *Multigrid Methods II*, volume 1228 of *Lecture Notes in Mathematics, Proceedings, Cologne*, pages 52–64. Springer-Verlag, Berlin, 1985.

[13] D. Braess. *Finite elements: theory, fast solvers, and applications in solid mechanics.* Cambridge University Press, Cambridge, United Kingdom, 1997.

[14] J.H. Bramble. *Multigrid methods.* Longman Scientific and Technical, Essex, UK, 1993.

[15] A. Brandt. Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. In *Proceedings of the third international conference on Numerical Methods in Fluid Mechanics*, pages 82–89, New York, 1973. Springer-verlag.

[16] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31:330–390, 1977.

[17] A. Brandt. Guide to multigrid development. In W. Hackbusch and W. Trottenberg, editors, *Multigrid Methods*, pages 220–312. Springer-Verlag, Berlin, 1981.

[18] I.C. Briggs. Machine contouring using minimum curvature. *Geophysics*, 39(1):39–48, 1974.

[19] W. Briggs and S. McCormick. Introduction. In *Multigrid methods*, chapter 1. Society for Industrial and Applied Mathematics, Pennsylvania, 1987.

[20] W.L. Briggs. *A multigrid tutorial.* Society for Industrial and Applied Mathematics, Pennsylvania, 1987.

[21] Lin-An Chen, Wenyaw Chan, and Tzong-Shi Lee. Tensor product polynomial splines. *Comm. Statist. Theory Methods*, 26(9):2093–2111, 1997.

[22] P. Christen, M. Hegland, O.M.Nielsen, S. Roberts, P.E. Strazdins, and I. Altas. Scalable parallel algorithms for surface fitting and data mining. *Parallel Computing*, 27:941–961, 2001.

[23] P.G. Ciarlet and P.A. Raviart. A mixed finite element method for the biharmonic equation. In C. de Boor, editor, *Mathematical aspects of finite elements in partial differential equations*, pages 125–145. Academic Press, New York, 1974.

[24] P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–403, 1979.

[25] N. Cressie. *Statistics for Spatial Data.* John Wiley and Sons, Canada, 1991.

[26] G. Cressman. An operational objective analysis system. *Monthly Weather Review*, 87:367–374, 1959.

[27] C. de Boor. *A practical guide to splines.* Springer-Verlag, 1978.

[28] P.J. Diggle and M.F. Hutchinson. On spline smoothing with autocorrelated errors. *Australian Journal of Statistics*, 31:166–182, 1989.

[29] J. Duchon. Fonctions splines et vecteurs aleatiors. In *Tech. Report 213*, Seminaire d'Analyse Numerique. Universite Scientifique et Medicale, Grenoble, 1975.

[30] J. Duchon. Fonctions-spline et esperances conditionnelles de champs gaussiens. *Ann. Sci. Univ. Clermont Ferrand II Math.*, 14:19–27, 1976.

[31] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobelov spaces. In *Constructive Theory of Functions of Several Variables*, pages 85–100. Springer-Verlag, Berlin, 1977.

[32] R.P. Fedorenko. The speed of convergence of one iterative process. *USSR Computational Math. and Math. Phys.*, 4(3):227–235, 1964.

[33] D.R. Ferguson, P.D. Frank, and A.K. Jones. Surface shape control using constrained optimization on the B-spline representation. *Comput. Aided Geom. Design*, 5(2):87–103, 1988.

[34] D.R. Forsey and R.H. Bartels. Hierarchical B-spline refinement. *Computer Graphics*, 22(4):205–211, 1988.

[35] R. Franke and G. Nielson. Smooth interpolation of large sets of scattered data. *International Journal for Numerical Methods in Engineering*, 15:1691–1704.

[36] D.A. Girard. A fast 'Monte Carlo cross-validation' procedure for large least squares problems with noisy data. *Numerische Mathematik*, 56:1–23, 1989.

[37] G. Golub and Z. Bai. Bounds for the trace of the inverse and the determinant of symmetric positive definite matrices. *Annals of numerical mathematics*, 4:29–38, 1997.

[38] G. Golub and U. von Matt. Generalized cross validation for large-scale problems. *Journal of Computational and Graphical Statistics*, 6(1):1–34, March 1997.

[39] G.H. Golub and U.V. Matt. Generalized cross-validation for large scale problems: revised version. *Journal of Computational and Graphical Statistics*, 6:1–34, 1997.

[40] P.J. Green and B.W. Silverman. *Non-parametric regression and generalised linear models: A roughness penalty approach.* Chapman and Hall, London, 1994.

[41] A. Greenbaum. *Iterative methods for solving linear systems.* Frontiers in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[42] Eric Grosse. Tensor spline approximation. *Linear Algebra Appl.*, 34:29–41, 1980.

[43] L. Guenni and M.F. Hutchinson. Spatial interpolation of the parameters of a rainfall model from ground-based data. *Journal of Hydrology*, 212-213:335–347, 1998.

[44] J. Gwinner and A. Rott. Least squares solution of bivariate surface fitting problems using tensor product splines. *BIT*, 40(4):715–725, 2000.

[45] W. Hackbusch. A fast iterative method for solving Poisson's equation in a general region. In R. Bulirsch, R.D. Griegorieff, and J. Schroder, editors, *Proceedings, Oberwolfach, 1976*, volume 631 of *Lecture Notes in Mathematics*, pages 51–62. Springer, Berlin, 1978.

[46] W. Hackbusch. *Multigrid methods and applications.* Springer-Verlag, Berlin, Heidelberg, 1985.

[47] L.A. Hageman and D.M. Young. *Applied Iterative Methods.* Academic Press, London, 1981.

[48] P.A. Hancock and M.F. Hutchinson. An iterative procedure for calculating minimum generalised cross validation smoothing splines. *Australian and New Zealand Applied Mathematics Journal*, (in press).

[49] P.A. Hancock, M.F. Hutchinson, S.M. Turton, and A.L. Lewis. Thin plate smoothing spline interpolation of long term monthly mean rainfall for the Wet Tropics region of north-eastern Australia. *Proceedings of the International Congress on Modelling and Simulation (MODSIM 2001)*, 2:943–948, 2001.

[50] M.R. Hanisch. Multigrid preconditioning for the biharmonic Dirichlet problem. *SIAM J. Numer. Anal.*, 30(1):184–214, 1993.

[51] D. Hanselman and B. Littlefield. *The student edition of MATLAB: High performance numerical computation and visualisation software*. Prentice Hall, Englewood Cliffs, 1995.

[52] X. He and P. Shi. Bivariate tensor-product B-splines in a partly linear model. *Journal of Multivariate Analysis*, 58:162–181, 1996.

[53] M. Hegland, S. Roberts, and I. Altas. Finite element thin plate splines for data mining applications. In M. Daehlen, T. Lyche, and L.L. Schumaker, editors, *Mathematical models for curves and surfaces II*, pages 245–252. Vanderbilt University Press, Nashville, TN, 1998. Available as a Mathematical Research Report MRR057-97, School of Mathematical Sciences, The Australian National University.

[54] M. Hegland, S. Roberts, and I. Altas. Finite element thin plate splines for surface fitting. In *Computational Techniques and Applications: CTAC97*, pages 289–296, Singapore, 1998. World Scientific Press.

[55] P.W. Hemker. On the order of prolongations and restrictions in multigrid procedures. *Journal of Computational and Applied Mathematics*, 32:423–429, 1990.

[56] M.R. Hestenes and E.L. Stiefel. Methods of conjugate gradients for solving linear systems. *Nat. Bur. Std. J. Res.*, 49:409–436, 1952.

[57] M.F. Hutchinson. A new method for estimating the spatial distribution of mean seasonal and annual rainfall applied to the Hunter valley. *Australian Meteorology Magazine*, 31:179–184, 1983.

[58] M.F. Hutchinson. A new method for gridding elevation and streamline data with automatic removal of spurious pits. *Journal of Hydrology*, 106:211–232, 1989.

[59] M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18:1059–1076, 1989.

[60] M.F. Hutchinson. The application of thin plate smoothing splines to continent-wide data assimilation. In J.D. Jasper, editor, *Data Assimi-*

lation Systems: BMRC Research Report No. 27, pages 104–113. Bureau of Meteorology, Melbourne, 1991.

[61] M.F. Hutchinson. On thin plate splines and kriging. *Computing and Science in Statistics*, 25:55–62, 1993.

[62] M.F. Hutchinson. Splines - more than just a smooth interpolator. *Geoderma*, 62:45–67, 1994.

[63] M.F. Hutchinson. Interpolating mean rainfall using thin plate smoothing splines. *International Journal of Geographic Information Systems*, 9(4):385–403, 1995.

[64] M.F. Hutchinson. Stochastic space-time weather models from ground-based data. *Agricultural and Forest Meteorology*, 73:237–264, 1995.

[65] M.F. Hutchinson. Interpolation of rainfall data with thin plate smoothing splines: I. Two dimensional smoothing of data with short range correlation. *Journal of Geographic Information and Decision Analysis*, 2(2):153–167, 1998.

[66] M.F. Hutchinson. Interpolation of rainfall data with thin plate smoothing splines: II. Analysis of topographic dependence. *Journal of Geographic Information and Decision Analysis*, 2(2):168–185, 1998.

[67] M.F. Hutchinson. Optimising the degree of data smoothing for locally adaptive finite element bivariate smoothing splines. *ANZIAM Journal*, pages C774–C796, 2000. `http://anziamj.austms.org.au/v42/CTAC99/Hutch`.

[68] M.F. Hutchinson. *ANUSPLIN version 4.2 User Guide*, 2002. `http://cres.anu.edu.au/outputs/anusplin.html`.

[69] M.F. Hutchinson and F.R. de Hoog. Smoothing noisy data with spline functions. *Numerische Mathematik*, 47(99-106), 1985.

[70] M.F. Hutchinson and J.L. Kesteven. Monthly mean climate surfaces for Australia, 1998. `http://cres.anu.edu.au/outputs/climatesurfaces/index.html`.

[71] M.F. Hutchinson, H.A. Nix, and J.P. McMahon. Climate constraints on cropping systems. In C.J. Pearson, editor, *Field Crop Ecosystems of the World*, pages 37–58. Elsevier, 1992.

[72] M.F. Hutchinson and J. Sharples. Topographic dependent interpolation of alpine precipitation. *Journal of Applied Meteorology*, (under review).

[73] A. Ionescu, Y. Candau, E. Mayer, and I. Colda. Analytical determination and classification of pollutant concentration fields using air pollution monitoring network data- methodology and application in the Paris area, during episodes with peak nitrogen dioxide levels. *Environmental Modelling and Software*, Sp. Iss. SI 2000(15):565–573, 2000.

[74] E. Isaaks and R. Srivistava. *An Introduction to Applied Geostatistics.* Oxford University Press, Oxford, 1989.

[75] C.H. Jarvis and N. Stuart. A comparison among strategies for interpolating maximum and minimum daily air temperatures. Part I: The selection of "guiding" topographic and land cover. *Journal of Applied Meteorology*, 40(6):1060–1074, 2001.

[76] C.H. Jarvis and N. Stuart. A comparison among strategies for interpolating maximum and minimum daily air temperatures. Part II: The interaction between number of guiding variables and the type of interpolation method. *Journal of Applied Meteorology*, 40(6):1075–1084, 2001.

[77] S.J. Jeffrey, J.O. Carter, K.B. Moodie, and A.R Beswick. Using spatial interpolation to construct a comprehensive archive of Australian climate data. *Environmental Modelling and Software*, 16:309–330, 2001.

[78] D. Jespersen. Multigrid methods for partial differential equations. In *Studies in Numerical Analysis*, volume 24 of *Studies in Mathematics.* Mathematical Association of America, 1984.

[79] G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Appl.*, 33:82–95, 1971.

[80] M. Kohler. Universally consistent regression function estimation using hierarchial B-splines. *Journal of Multivariate Analysis*, 68:138–164, 1999.

[81] R. Kraft. Adaptive and linearly independent multilevel B-splines. In A. Le Mehaute, C. Rabut, and L.L. Schumaker, editors, *Surface Fitting and Multiresolution methods*, pages 209–216. Vanderbilt University Press, Nashville, TN, 1997.

[82] P. Lancaster. Moving weighted least-squares methods. In B. Sahney, editor, *Proceedings NATO Advanced Study Institute on Polynomial and Spline Approximation*, pages 103–119, Calgary, 1979.

[83] R.J.A. Lapeer and R.W. Prager. 3D shape recovery of a newborn skull using thin-plate splines. *Computerised Medical Imaging and Graphics*, 24(3):193–204, 2000.

[84] J.L. Lennon and J.R.G. Turner. Predicting the spatial distribution of climate: temperature in Great Britain. *Journal of Animal Ecology*, 64:370–392, 1995.

[85] H. Luhtkepohl. *Handbook of Matrices.* John Wiley and Sons, Chichester, 1996.

[86] B.G. Mackey. A spatial analysis of the environmental relations of rainforest structural types. *Journal of Biogeography*, 20:303–336, 1993.

[87] B.G. Mackey, H.A. Nix, M.F. Hutchinson, and J.P. Macmahon. Assessing representativeness of places for conservation reservation and heritage listing. *Environmental Management*, 12(4):501–514, 1988.

[88] B.G. Mackey, H.A. Nix, J.A. Stein, and S.E. Cork. Assessing the representativeness of the Wet Tropics Queensland World Heritage property. *Biological Conservation*, 50:279–303, 1989.

[89] M.J. Marsden. Quadratic B-spline interpolation. *Bulletin of the American Statistical Society*, 80(5):903–906, 1974.

[90] G. Matheron. The intrinsic random functions and their applications. *Adv. Appl. Probab.*, 5:439–468, 1973.

[91] D.H. McClain. Drawing contours from arbitrary data points. *The computer journal*, 17:318–324, 1974.

[92] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *Journal of Applied Mathematical Physics (ZAMP)*, 30:292–304, 1979.

[93] H.A. Nix. A biogeographic analysis of Australian elapid snakes. In R. Longmore, editor, *Atlas of elapid snakes of Australia*, number 7 in Australian Flora and Fauna Series, pages 4–15, Canberra, 1986. AGPS.

[94] F. O'Sullivan. An iterative approach to two-dimensional Laplacian smoothing with application to image restoration. *Journal of the American Statistical Association*, 85(409):213–219, 1990.

[95] M.C. Pease. *Methods of matrix algebra.* Academic, New York, 1965.

[96] D.T. Price, D.W. McKenney, I.A. Nalder, M.F. Hutchinson, and J.L. Kesteven. A comparison of two statistical methods for spatial interpolation of Canadian monthly mean climate data. *Agricultural and Forest Meteorology*, 101:81–94, 2000.

[97] T. Ramsay. Spline smoothing over difficult regions. *J.R. Statist. Soc. B*, 64 Part 2:307–319, 2002.

[98] J.A. Rice. *Mathematical Statistics and Data Analysis.* Wadsworth Publishing Company, Belmont, California, 2nd edition, 1995.

[99] I.J. Schoenberg. Spline functions and the problem of graduation. *Proceedings of the National Academy of Science*, 52, 1964.

[100] I.J. Schoenberg. On spline functions. In O. Shisha, editor, *Inequalities*, pages 255–291. Academic Press, New York, 1967.

[101] D. Shepard. A two dimensional interpolation function for irregularly spaced data. In *Proc. 23rd National Conference ACM*, pages 517–523, 1968.

[102] R. Sibson and G. Stone. Computation of thin-plate splines. *SIAM J. Sci. Stat. Comput.*, 12(6):1304–1313, 1991.

[103] N.K. Spencer. Parallel multigrid solution of boundary value problems. Master's thesis, The Australian National University, 1992.

[104] G. Strang. *Linear Algebra and its Applications.* Harcourt Brace Jovanovich, Inc., Orlando, Florida, 3rd edition, 1988.

[105] C.J. Swain. A FORTAN IV program for interpolating irregularly spaced data using the difference equations for minimum curvature. *Comput. Geosci.*, 1:231–240, 1976.

[106] R. Szeliski. Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on pattern analysis and machine intelligence*, 12(6):513–528, June 1990.

[107] R.C. Tabony. Relations between minimum temperature and topography in Great Britian. *Journal of Climatology*, 5:502–530, 1985.

[108] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics and Image Processing*, 24:52–96, 1983.

[109] A.H. Thiessen. Precipitation averages for large areas. *Monthly Weather Review*, 39:1082–1084, 1911.

[110] A.H. Vermeulen, R.H. Bartels, and G.R. Heppler. Integrating products of B-splines. *SIAM Journal of Scientific and Statistical Computing*, 13(4):1025–1038, 1992.

[111] H. Wackernagel. *Multivariate Geostatistics.* Springer-Verlag, Berlin, 1998.

[112] G. Wahba. Bayesian "confidence intervals" for the cross-validated smoothing spline. *Journal of the Royal Statistical Society Series B*, 45:281–299, 1983.

[113] G. Wahba. Spline models for observational data. In *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics, Philadelphia, 1990.

[114] G. Wahba, D.R. Johnson, F. Gao, and J. Gong. Adaptive tuning of numerical weather prediction models: Randomized GCV in three and four dimensional data assimilation. *Monthly Weather Review*, 123:3358–3369, 1995.

[115] P. Wesseling. *An Introduction to Multigrid Methods.* John Wiley and Sons, Chichester, 1992.

[116] S.N. Wood and J.W. Horwood. Spatial-distribution functions and abundances inferred from sparse noisy plankton data - an application of constrained thin-plate splines. *Journal of plankton research*, 17(6):1189–1208, June 1995.

[117] D.M. Young. *Iterative solution of large linear systems.* Academic, New York, 1971.

[118] X. Zheng and R. Basher. Thin plate smoothing spline modeling of spatial climate data and its application to mapping South Pacific rainfalls. *Monthly Weather Review*, 123:3086–3102, 1995.

[119] X. Zheng and R. Basher. Mapping rainfall fields and their ENSO variation in the data-sparse tropical south-west Pacific Ocean Region,. *International Journal of Climatology*, 18:137–251, 1998.

Appendix A

Results for Chapter 10

The tables in this appendix are discussed in Chapter 10. They report updates of various quantities relevant to the OPTRSS and MINGCV algorithms described in Chapter 8. The notation is explained as follows:

q: the number of updates performed on a given grid.

θ: the logarithm of the smoothing parameter.

θ_q: the q^{th} update of the logarithm of the smoothing parameter.

S: the prescribed residual on a given grid, for the OPTRSS algorithm.

R: the estimate of the residual sum of squares.

Tr: the estimate of $tr(I - A)$, where A is the influence matrix.

GCV: the estimate of the generalised cross validation.

A.1 Results generated by the OPTRSS algorithm for the data set sine.dat

q	θ_q	$dR/d\theta$	$S - R$
0	10.14	0.925	-0.454
1	9.67	0.054	0.029
2	10.18	0.072	-0.005
3	10.12	0.077	-0.001
4	10.10	0.061	0.005
5	10.18	0.070	-0.002
6	10.15	0.068	0.000
7	10.15	0.068	0.000
8	10.15	0.068	0.000
9	10.15	0.068	0.000

Grid 6

q	θ_q	$dR/d\theta$	$S - R$
0	10.15	0.156	0.042
1	10.42	0.133	-0.047
2	10.06	0.149	0.010
3	10.13	0.128	0.009
4	10.20	0.131	-0.011
5	10.12	0.142	0.006
6	10.16	0.138	-0.003
7	10.14	0.136	0.001
8	10.15	0.136	0.000
9	10.15	0.137	-0.001

Grid 5

q	θ_q	$dR/d\theta$	$S - R$
0	10.15	0.137	-0.019
1	10.00	0.152	-0.020
2	9.88	0.154	-0.003
3	9.86	0.162	0.014
4	9.95	0.161	0.019
5	10.06	0.155	0.017
6	10.17	0.148	0.009
7	10.23	0.141	0.002
8	10.25	0.137	-0.004
9	10.22	0.135	-0.007

Grid 4

q	θ_q	$dR/d\theta$	$S - R$
0	10.22	0.127	-0.014
1	10.06	0.128	-0.008
2	10.00	0.129	-0.009
3	9.93	0.131	-0.006
4	9.89	0.133	-0.003
5	9.86	0.135	-0.001
6	9.86	0.137	0.002
7	9.87	0.138	0.004
8	9.90	0.138	0.005
9	9.93	0.139	0.006

Grid 3

q	θ_q	$dR/d\theta$	$S-R$
0	9.93	0.133	0.006
1	10.01	0.132	0.005
2	10.05	0.132	0.005
3	10.09	0.132	0.005
4	10.12	0.132	0.005
5	10.16	0.132	0.005
6	10.19	0.132	0.004
7	10.23	0.132	0.004
8	10.26	0.131	0.004
9	10.29	0.131	0.004

Grid 2

q	θ_q	$dR/d\theta$	$S-R$
0	10.29	0.132	0.004
1	10.34	0.132	0.004
2	10.37	0.132	0.003
3	10.39	0.132	0.003
4	10.42	0.132	0.003
5	10.44	0.132	0.003
6	10.47	0.132	0.003
7	10.49	0.132	0.003
8	10.52	0.132	0.003
9	10.54	0.132	0.003

Grid 1

A.2 Results generated by the OPTRSS algorithm, with a starting λ value of $\lambda_0 = 500$, for the data set sine.dat

q	θ_q	$dR/d\theta$	$S-R$
0	5.73	5.249	2.10
1	5.81	0.483	-0.04
2	5.73	.032	0.06
3	7.49	0.006	0.06
4	17.10	0.150	-6.40
5	-25.54	0.741	-0.24
6	-25.87	0.069	0.05
7	-25.18	0.005	0.06
8	-12.54	0.000	0.06
9	192.88	0.000	-6.78

Grid 6

A.3 Results generated by the OPTRSS algorithm, with a starting λ value of $\lambda_0 = 5000000$, for the data set sine.dat

q	θ_q	$dR/d\theta$	$S - R$
0	13.12	-7.198	-39.98
1	10.14	1.228	-0.51
2	9.46	0.060	0.03
3	10.00	0.068	0.01
4	10.09	0.068	0.00
5	10.14	0.062	0.00
6	10.17	0.070	0.00
7	10.14	0.068	0.00
8	10.15	0.068	0.00
9	10.15	0.068	0.00

Grid 6

q	θ_q	$dR/d\theta$	$S - R$
0	10.15	0.156	0.042
1	10.42	0.133	-0.047
2	10.06	0.149	0.011
3	10.13	0.128	0.009
4	10.20	0.131	-0.011
5	10.12	0.142	0.006
6	10.16	0.138	-0.003
7	10.14	0.136	0.001
8	10.15	0.136	0.000
9	10.15	0.137	-0.001

Grid 5

q	θ_q	$dR/d\theta$	$S - R$
0	10.15	0.137	-0.019
1	10.00	0.152	-0.020
2	9.88	0.154	-0.003
3	9.86	0.162	0.014
4	9.95	0.161	0.019
5	10.06	0.155	0.017
6	10.17	0.148	0.009
7	10.23	0.141	0.002
8	10.25	0.137	-0.004
9	10.22	0.135	-0.007

Grid 4

q	θ_q	$dR/d\theta$	$S - R$
0	10.22	0.127	-0.014
1	10.06	0.128	-0.008
2	10.00	0.129	-0.009
3	9.93	0.131	-0.006
4	9.89	0.133	-0.003
5	9.86	0.135	-0.001
6	9.86	0.137	0.002
7	9.87	0.138	0.004
8	9.90	0.138	0.005
9	9.93	0.139	0.006

Grid 3

q	θ_q	$dR/d\theta$	$S-R$
0	9.93	0.139	0.006
1	10.01	0.133	0.006
2	10.05	0.132	0.005
3	10.09	0.132	0.005
4	10.12	0.132	0.005
5	10.16	0.132	0.005
6	10.19	0.132	0.005
7	10.23	0.132	0.004
8	10.26	0.132	0.004
9	10.29	0.131	0.004

Grid 2

q	θ_q	$dR/d\theta$	$S-R$
0	10.29	0.131	0.004
1	10.34	0.132	0.004
2	10.37	0.132	0.004
3	10.39	0.132	0.003
4	10.42	0.132	0.003
5	10.44	0.132	0.003
6	10.47	0.132	0.003
7	10.49	0.132	0.003
8	10.52	0.132	0.003
9	10.54	0.132	0.003

Grid 1

A.4 Results generated by the OPTRSS algorithm, with prescribed S values of 2.8208 on each grid, for the data set sine.dat

q	θ_q	$dR/d\theta$	$S-R$
1	10.14	7.20	-40.6
2	8.89	1.38	-1.2
3	0.90	7.78E-02	-0.622
4	-96.32	6.18E-03	-0.600
5	-1569.10	4.07E-04	-0.599
6	NaN	NaN	-0.599
7	NaN	NaN	NaN
8	NaN	NaN	NaN
9	NaN	NaN	NaN
10	NaN	NaN	NaN

Grid 6

A.5 Results generated by the OPTRSS algorithm, with a lower threshold on λ updates of λ/h^3, for the data set sine.dat

q	θ_q	$dR/d\theta$	$S-R$
0	10.14	0.055	-0.634
1	9.70	0.050	-0.637
2	9.70	0.050	-0.637
3	9.70	-0.042	-0.635
4	9.70	0.047	-0.636
5	9.70	0.046	-0.636
6	9.70	0.046	-0.636
7	9.70	0.046	-0.636
8	9.70	0.046	-0.636
9	9.70	0.046	-0.636

Grid 6

q	θ_q	$dR/d\theta$	$S-R$
0	9.70	0.160	0.097
1	10.31	0.122	-0.064
2	9.79	0.155	0.039
3	10.04	0.125	-0.002
4	10.02	0.144	-0.004
5	9.99	0.143	0.008
6	10.05	0.137	-0.005
7	10.01	0.139	0.004
8	10.03	0.138	-0.002
9	10.02	0.140	0.000

Grid 5

q	θ_q	$dR/d\theta$	$S-R$
0	10.02	0.137	0.037
1	10.29	0.138	0.015
2	10.40	0.129	0.008
3	10.46	0.131	0.005
4	10.50	0.130	-0.003
5	10.48	0.129	-0.008
6	10.42	0.131	-0.010
7	10.34	0.134	-0.008
8	10.28	0.139	-0.002
9	10.27	0.143	0.004

Grid 4

q	θ_q	$dR/d\theta$	$S-R$
0	10.27	0.150	-0.144
1	9.07	0.188	-0.092
2	8.11	0.241	-0.009
3	7.62	0.276	0.061
4	7.58	0.285	0.098
5	7.80	0.272	0.104
6	8.15	0.247	0.090
7	8.53	0.221	0.067
8	8.89	0.197	0.043
9	9.20	0.137	0.037

Grid 3

q	θ_q	$dR/d\theta$	$S-R$
0	9.20	0.182	0.119
1	10.07	0.179	0.114
2	10.71	0.176	0.112
3	11.34	0.174	0.108
4	11.96	0.171	0.105
5	12.58	0.169	0.102
6	13.18	0.167	0.099
7	13.77	0.165	0.096
8	14.35	0.163	0.093
9	14.93	0.161	0.091

Grid 2

q	θ_q	$dR/d\theta$	$S-R$
0	14.93	0.161	0.128
1	16.28	0.162	0.128
2	17.07	0.162	0.128
3	17.86	0.162	0.128
4	18.65	0.162	0.127
5	19.44	0.161	0.127
6	20.23	0.161	0.127
7	21.01	0.161	0.127
8	21.80	0.161	0.127
9	22.59	0.161	0.127

Grid 1

A.6 Results generated by the OPTRSS algorithm, with the smoothing parameter fixed at $\lambda = 5$, for the data set sine.dat

q	θ_q	$dR/d\theta$	$S - R$
0	5.10	5.448	-2.240
1	5.10	0.504	-0.104
2	5.10	0.035	-0.005
3	5.10	-0.002	-0.000
4	5.10	0.000	0.000
5	5.10	0.000	0.000
6	5.10	0.000	0.000
7	5.10	0.000	0.000
8	5.10	0.000	0.000
9	5.10	0.000	0.000
10	5.10	0.000	0.000
11	5.10	0.000	0.000
12	5.10	0.000	0.000
13	5.10	0.000	0.000
14	5.10	0.000	0.000
15	5.10	0.000	0.000
16	5.10	0.000	0.000
17	5.10	0.000	0.000
18	5.10	0.000	0.000
19	5.10	0.000	0.000

Grid 6

q	θ_q	$dR/d\theta$	$S - R$
0	5.10	0.029	-0.013
1	4.66	0.015	-0.005
2	4.31	0.001	-0.001
3	3.39	0.000	-0.000
4	3.02	0.000	0.000
5	3.02	0.000	0.000
6	3.02	0.000	0.000
7	3.02	0.000	0.000
8	3.02	0.000	0.000
9	3.02	0.000	0.000
10	3.02	0.000	0.000
11	3.02	0.000	0.000
12	3.02	0.000	0.000
13	3.02	0.000	0.000
14	3.02	0.000	0.000
15	3.02	0.000	0.000
16	3.02	0.000	0.000
17	3.02	0.000	0.000
18	3.02	0.000	0.000
19	3.02	0.000	0.000

Grid 5

q	θ_q	$dR/d\theta$	$S-R$
0	3.02	0.025	-0.010
1	2.61	0.031	-0.013
2	2.19	0.006	-0.002
3	1.84	0.004	-0.001
4	1.69	0.003	-0.000
5	1.64	0.003	-0.000
6	1.63	0.003	-0.000
7	1.62	0.003	0.000
8	1.62	0.003	0.000
9	1.62	0.003	0.000
10	1.62	0.003	0.000
11	1.62	0.003	0.000
12	1.62	0.003	0.000
13	1.62	0.003	0.000
14	1.62	0.003	0.000
15	1.62	0.003	0.000
16	1.62	0.003	0.000
17	1.62	0.003	0.000
18	1.62	0.003	0.000
19	1.62	0.003	0.000

Grid 4

q	θ_q	$dR/d\theta$	$S-R$
0	1.62	0.352	0.007
1	1.64	0.318	0.012
2	1.63	0.340	-0.010
3	1.60	0.330	0.004
4	1.61	0.025	-0.010
5	1.61	0.333	-0.002
6	1.61	0.332	0.001
7	1.61	0.333	-0.001
8	1.61	0.332	0.000
9	1.61	0.333	0.000
10	1.61	0.333	0.000
11	1.61	0.333	0.000
12	1.61	0.333	0.000
13	1.61	0.333	0.000
14	1.61	0.333	0.000
15	1.61	0.333	0.000
16	1.61	0.333	0.000
17	1.61	0.333	0.000
18	1.61	0.333	0.000
19	1.61	0.333	0.000

Grid 3

q	θ_q	$dR/d\theta$	$S-R$
0	1.61	0.479	-0.054
1	1.50	0.451	0.077
2	1.52	0.467	0.075
3	1.63	0.414	-0.049
4	1.60	0.442	-0.012
5	1.60	0.436	0.002
6	1.61	0.439	-0.004
7	1.60	0.436	0.005
8	1.61	0.436	0.004
9	1.61	0.438	-0.002
10	1.61	0.437	0.000
11	1.61	0.437	0.000
12	1.61	0.436	0.000
13	1.61	0.437	0.000
14	1.61	0.437	0.000
15	1.61	0.437	0.000
16	1.61	0.437	0.000
17	1.61	0.437	0.000
18	1.61	0.437	0.000
19	1.61	0.437	0.000

Grid 2

q	θ_q	$dR/d\theta$	$S-R$
0	1.61	0.427	-0.012
1	1.58	0.438	-0.005
2	1.57	0.445	0.004
3	1.58	0.449	0.009
4	1.60	0.448	0.009
5	1.62	0.446	0.005
6	1.63	0.444	0.001
7	1.63	0.442	-0.002
8	1.63	0.441	-0.004
9	1.62	0.441	-0.003
10	1.61	0.442	-0.002
11	1.61	0.443	-0.001
12	1.60	0.444	0.000
13	1.60	0.444	0.001
14	1.61	0.445	0.001
15	1.61	0.445	0.001
16	1.61	0.445	0.000
17	1.61	0.445	0.000
18	1.61	0.445	0.000
19	1.61	0.445	0.000

Grid 1

A.7 Results generated by the MINGCV algorithm,for the data set sine.dat

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.14	0.0097	0.0187	0.9391	1.7095	0.9203	0.0473	3.9351	95.03	0.044
1	9.6302	-0.0002	0.0004	0.0423	0.047	0.8423	0.2033	3.4519	94.58	0.039
2	10.2243	0.0001	0.0007	0.0744	0.0595	0.9004	-0.0174	3.4894	95.10	0.039
3	10.0849	0.0000	0.0006	0.0671	0.054	0.8984	0.042	3.4806	94.97	0.039
4	10.0606	-0.0000	0.0006	0.0622	0.0544	0.8972	0.0497	3.4743	94.95	0.0389
5	10.1293	0.0000	0.0006	0.0675	0.0571	0.8996	0.0231	3.4806	95.01	0.0389
6	10.0999	-0.0000	0.0006	0.0656	0.0555	0.8988	0.0351	3.4784	94.99	0.0389
7	10.1049	0.0000	0.0006	0.0659	0.0561	0.899	0.0329	3.4787	94.99	0.0389
8	10.1032	0.0000	0.0006	0.0659	0.056	0.8989	0.0336	3.4786	94.99	0.0389
9	10.1026	-0.0000	0.0006	0.0658	0.0559	0.8989	0.0338	3.4786	94.99	0.0389
10	10.1031	0.0000	0.0006	0.0658	0.0559	0.8989	0.0336	3.4786	94.99	0.0389
11	10.103	0.0000	0.0006	0.0658	0.0559	0.8989	0.0337	3.4786	94.99	0.0389
12	10.103	0.0000	0.0006	0.0658	0.0559	0.8989	0.0336	3.4786	94.99	0.0389
13	10.103	0.0000	0.0006	0.0658	0.0559	0.8989	0.0337	3.4786	94.99	0.0389
14	10.103	0.0000	0.0006	0.0658	0.0559	0.8989	0.0337	3.4786	94.99	0.0389
15	10.103	0.0000	0.0006	0.0658	0.0559	0.8989	0.0337	3.4786	94.99	0.0389
16	10.103	0.0000	0.0006	0.0658	0.0559	0.8989	0.0337	3.4786	94.99	0.0389
17	10.103	0.0000	0.0006	0.0658	0.0559	0.8989	0.0337	3.4786	94.99	0.0389
18	10.103	0.0000	0.0006	0.0658	0.0559	0.8989	0.0337	3.4786	94.99	0.0389
19	10.103	0.0000	0.0006	0.0658	0.0559	0.8989	0.0337	3.4786	94.99	0.0389

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.103	0.0006	0.0005	0.1465	0.0092	1.5694	-0.6597	2.7886	94.27	0.0317
1	8.8597	0.0000	0.0004	0.1495	-0.0148	2.5278	-0.8794	2.6395	91.69	0.0317
2	8.7341	0.0001	0.0005	0.1638	-0.0006	2.6507	-0.8539	2.6285	91.36	0.0318
3	8.4695	-0.0000	0.0007	0.1597	0.0111	2.8751	-0.8566	2.5793	90.63	0.0317
4	8.5414	-0.0000	0.0006	0.1603	0.0071	2.8175	-0.8511	2.5928	90.84	0.0317
5	8.553	0.0000	0.0006	0.1618	0.0079	2.8067	-0.8539	2.5952	90.87	0.0317
6	8.5249	-0.0000	0.0006	0.1602	0.0077	2.8304	-0.854	2.5899	90.79	0.0317
7	8.5491	0.0000	0.0006	0.1611	0.0071	2.8102	-0.8533	2.5946	90.86	0.0317
8	8.538	-0.0000	0.0006	0.1608	0.0075	2.8193	-0.8541	2.5924	90.83	0.0317
9	8.5406	-0.0000	0.0006	0.1607	0.0073	2.8173	-0.8537	2.5929	90.83	0.0317
10	8.5424	0.0000	0.0006	0.1609	0.0073	2.8157	-0.8538	2.5933	90.84	0.0317
11	8.5399	-0.0000	0.0006	0.1608	0.0074	2.8178	-0.8538	2.5928	90.83	0.0317
12	8.5415	0.0000	0.0006	0.1608	0.0073	2.8165	-0.8538	2.5931	90.84	0.0317
13	8.5409	0.0000	0.0006	0.1608	0.0074	2.817	-0.8538	2.593	90.84	0.0317
14	8.5409	-0.0000	0.0006	0.1608	0.0073	2.817	-0.8538	2.593	90.84	0.0317
15	8.5411	0.0000	0.0006	0.1608	0.0073	2.8168	-0.8538	2.593	90.84	0.0317
16	8.5409	-0.0000	0.0006	0.1608	0.0073	2.817	-0.8538	2.593	90.84	0.0317
17	8.541	-0.0000	0.0006	0.1608	0.0073	2.8169	-0.8538	2.593	90.84	0.0317
18	8.541	-0.0000	0.0006	0.1608	0.0073	2.8169	-0.8538	2.593	90.84	0.0317
19	8.541	-0.0000	0.0006	0.1608	0.0073	2.8169	-0.8538	2.593	90.84	0.0317

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.541	0.0001	0.0011	0.2045	0.025	3.5524	-1.379	2.5653	91.27	0.0311
1	8.4897	0.0001	0.0011	0.1825	0.0057	3.1658	-1.7138	2.5389	91.13	0.0309
2	8.4245	-0.0002	0.0008	0.1734	-0.016	3.3875	-1.5801	2.5354	90.92	0.031
3	8.6555	-0.0002	0.0006	0.1451	-0.0337	2.9463	-1.5556	2.5451	91.65	0.0306
4	9.0265	-0.0001	0.0004	0.1266	-0.0362	2.414	-1.3096	2.5905	92.67	0.0305
5	9.2708	0.0001	0.0003	0.1278	-0.0347	2.1681	-1.1316	2.6329	93.25	0.0306
6	9.0632	0.0003	0.0005	0.1667	-0.0213	2.5172	-1.1704	2.636	92.76	0.0309
7	8.4544	0.0004	0.0013	0.2241	0.04	3.4361	-1.4575	2.5605	91.01	0.0312
8	8.1751	-0.0002	0.0011	0.1977	-0.0053	3.8102	-1.8104	2.4927	90.02	0.0311
9	8.3261	-0.0005	0.0007	0.1562	-0.0407	3.5037	-1.7472	2.4978	90.58	0.0307
10	9.0137	-0.0003	0.0004	0.1057	-0.0423	2.3453	-1.3878	2.5754	92.64	0.0303
11	9.7391	-0.0001	0.0002	0.081	-0.0361	1.6437	-0.9077	2.6454	94.21	0.0301
12	10.5455	0.0000	0.0001	0.061	-0.0267	1.1115	-0.5864	2.723	95.38	0.0302
13	10.9408	0.0001	0.0000	0.0585	-0.0252	0.9069	-0.5151	2.7869	95.79	0.0307
14	9.5177	0.0008	0.0002	0.1899	-0.0374	2.0936	-1.1417	2.7619	93.71	0.0318
15	6.3829	-0.0028	0.0014	0.201	-0.0479	7.4782	-2.1294	2.0234	79.90	0.032
16	8.3358	0.0008	0.0033	0.2723	0.1736	3.5047	-1.9274	2.7302	90.64	0.0336
17	8.111	0.0001	0.0016	0.2259	0.0437	3.901	-1.7642	2.4778	89.78	0.031
18	8.0293	0.0000	0.0011	0.2293	-0.0048	4.0684	-1.878	2.5169	89.45	0.0318
19	8.0255	0.0000	0.0011	0.2293	-0.0048	4.0672	-1.8892	2.4715	89.44	0.0312

Grid 4

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.0255	-0.0001	0.0007	0.1339	0.0037	2.4913	-0.9962	2.7023	92.28	0.032
1	8.5097	0.0000	0.0005	0.1294	-0.0049	2.1966	-0.8552	2.7079	92.80	0.0318
2	8.4796	0.0001	0.0004	0.1365	-0.0139	2.1753	-0.9539	2.7209	92.84	0.0319
3	8.2439	0.0001	0.0005	0.1563	-0.0266	2.455	-1.2782	2.7191	92.35	0.0322
4	7.9745	0.0002	0.0006	0.1831	-0.0366	2.8559	-1.6224	2.7031	91.58	0.0326
5	7.6971	0.0002	0.0007	0.2136	-0.0406	3.3224	-1.8759	2.6723	90.64	0.0329
6	7.4038	0.0003	0.0008	0.2476	-0.0376	3.8629	-1.9891	2.6267	89.55	0.0331
7	7.0838	0.0003	0.0009	0.2844	-0.0249	4.4839	-1.9483	2.5649	88.23	0.0333
8	6.7467	0.0003	0.0011	0.3168	-0.0036	5.1484	-1.8201	2.485	86.67	0.0334
9	6.4701	0.0001	0.0012	0.3232	0.0059	5.6623	-1.7993	2.3966	85.25	0.0333
10	6.4132	-0.0004	0.0011	0.2824	-0.0213	5.6599	-2.0002	2.3381	84.93	0.0327
11	6.7985	-0.0008	0.0008	0.1931	-0.0507	4.7098	-2.0324	2.3581	86.99	0.0315
12	7.8523	-0.0008	0.0005	0.0842	-0.02	2.7718	-1.1237	2.4535	90.91	0.03
13	9.3352	-0.0006	0.0004	0.0274	0.0152	1.4655	-0.2921	2.5656	93.40	0.0297
14	10.9293	-0.0005	0.0003	0.0126	0.0227	1.0034	0.0178	2.6609	94.58	0.03
15	12.8368	-0.0004	0.0002	0.0084	0.0166	0.8444	0.0466	2.7302	95.18	0.0304
16	15.5323	-0.0004	0.0001	0.0043	0.0103	0.7207	0.0198	2.7743	95.50	0.0307
17	19.3192	-0.0004	0.0001	0.0009	0.0062	0.5921	0.0129	2.7972	95.71	0.0308
18	24.8494	-0.0003	0.0000	-0.0012	0.0034	0.479	0.0276	2.8053	95.87	0.0308
19	38.0293	-0.0003	-0.0000	-0.0024	0.001	0.3976	0.047	2.8058	96.01	0.0307

Grid 3

A.8 Results generated by the MINGCV algorithm for the data set sine.dat, using a second random vector t

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.14	0.0114	0.0215	0.9247	1.6537	0.6553	-0.038	3.9351	87.54	0.0519
1	9.6155	0.0000	0.0011	0.0535	0.1141	0.6382	0.4019	3.4513	87.14	0.0459
2	9.5792	0.0000	0.0005	0.0452	0.0585	0.6017	0.2742	3.4524	87.12	0.0459
3	9.6476	0.0000	0.0004	0.0468	0.0588	0.6284	0.3187	3.4548	87.16	0.0459
4	9.7373	-0.0001	0.0001	0.044	0.0311	0.6519	0.2912	3.4575	87.22	0.0459
5	10.7158	0.0005	0.0008	0.106	0.0874	0.8634	0.2561	3.5312	87.98	0.0461
6	10.1483	0.0001	0.0005	0.0711	0.0605	0.7939	0.2419	3.4829	87.52	0.0459
7	9.9521	0.0000	0.0003	0.0571	0.0514	0.7304	0.3294	3.4684	87.37	0.0459
8	9.9888	0.0000	0.0004	0.0603	0.0541	0.7256	0.2799	3.4717	87.40	0.0459
9	9.9048	-0.0000	0.0003	0.0553	0.0502	0.7091	0.3059	3.4664	87.34	0.0459
10	9.9437	0.0000	0.0004	0.0571	0.0505	0.7139	0.2934	3.4688	87.36	0.0459
11	9.9287	0.0000	0.0004	0.0566	0.0508	0.7128	0.2974	3.468	87.35	0.0459
12	9.9281	-0.0000	0.0004	0.0565	0.0506	0.7118	0.2976	3.4679	87.35	0.0459
13	9.9305	0.0000	0.0004	0.0567	0.0508	0.7124	0.2968	3.4681	87.35	0.0459
14	9.9269	-0.0000	0.0004	0.0565	0.0507	0.7117	0.2976	3.4678	87.35	0.0459
15	9.9287	0.0000	0.0004	0.0565	0.0507	0.7119	0.2971	3.468	87.35	0.0459
16	9.9279	-0.0000	0.0004	0.0565	0.0507	0.7119	0.2973	3.4679	87.35	0.0459
17	9.9281	-0.0000	0.0004	0.0565	0.0507	0.7118	0.2973	3.4679	87.35	0.0459
18	9.9282	0.0000	0.0004	0.0565	0.0507	0.7119	0.2973	3.4679	87.35	0.0459
19	9.928	0.0000	0.0004	0.0565	0.0507	0.7119	0.2973	3.4679	87.35	0.0459

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	9.928	-0.0002	0.0004	0.1576	-0.0081	2.2512	-0.7586	2.7603	88.50	0.0356
1	9.9301	0.0002	-0.0004	0.1494	-0.0024	2.1217	0.4124	2.7281	87.28	0.0362
2	4.9566	0.0005	0.0001	0.1486	-0.0581	1.6551	-1.1155	2.8371	88.23	0.0368
3	4.6357	0.0005	0.0014	0.0311	0.0866	0.0293	-0.0912	2.315	80.98	0.0357
4	4.6689	0.0000	0.0005	0.0151	0.0503	0.2813	0.3573	2.3089	80.92	0.0356
5	3.4919	-0.0002	-0.0002	0.0024	0.0015	0.2511	0.2053	2.3055	80.93	0.0355
6	4.5727	0.0000	0.0000	0.0001	0.0002	0.0453	-0.0371	2.3039	80.75	0.0357
7	3.1888	-0.0002	-0.0001	0.0055	0.0132	0.275	0.3613	2.3063	80.91	0.0356
8	3.3191	0.0000	0.0001	0.0000	0.0000	0.0144	-0.1171	2.3038	80.73	0.0357
9	2.6235	-0.0001	-0.0001	0.0006	0.0019	0.0834	0.1386	2.304	80.74	0.0357
10	4.971	0.0000	0.0000	0.0000	-0.0001	0.0252	-0.0123	2.3038	80.70	0.0357
11	-21.5137	-0.0002	0.0000	0.0119	0.028	0.3999	0.5008	2.3092	81.02	0.0355
12	-21.7763	0.0001	0.0004	0.0012	0.0056	-0.0908	-0.3245	2.3041	80.66	0.0358
13	-21.9459	0.0000	-0.0001	0.0001	0.0006	0.0196	0.1178	2.3038	80.67	0.0358
14	-22.0642	0.0000	0.0000	0.0000	0.0001	-0.0042	-0.0358	2.3038	80.67	0.0358
15	-22.1563	0.0000	0.0000	0.0000	0.0000	0.0009	0.01	2.3038	80.67	0.0358
16	-22.2315	0.0000	0.0000	0.0000	0.0000	-0.0002	-0.0026	2.3038	80.67	0.0358
17	-22.2951	0.0000	0.0000	0.0000	0.0000	0.0000	0.0007	2.3038	80.67	0.0358
18	-22.3502	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0002	2.3038	80.67	0.0358
19	-22.3988	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	2.3038	80.67	0.0358

Grid 5

A.9 Results generated by the MINGCV algorithm for the data set sine.dat, for a third random vector t

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.14	0.0099	0.021	0.9247	1.6537	1.9288	-0.1127	3.9351	87.53	0.0519
1	9.675	-0.0014	0.0007	0.0547	0.1083	2.0012	0.7696	3.4537	86.46	0.0467
2	11.665	0.0051	0.0134	0.6247	1.1255	2.3788	-0.2806	3.8594	91.13	0.0469
3	11.2802	0.0004	0.0032	0.2105	0.2083	2.2581	-0.6684	3.6286	90.24	0.045
4	11.1669	-0.0003	0.0029	0.1649	0.2093	2.3853	-0.3264	3.5877	89.96	0.0448
5	11.2757	-0.0002	0.0023	0.1773	0.1624	2.3785	-0.3521	3.6088	90.22	0.0448
6	11.3451	0.0002	0.0031	0.204	0.2301	2.3421	-0.37	3.6266	90.39	0.0448
7	11.2818	0.0000	0.0028	0.1854	0.1984	2.3443	-0.4006	3.6116	90.24	0.0448
8	11.2918	0.0000	0.0029	0.1883	0.2068	2.3577	-0.3657	3.6133	90.26	0.0448
9	11.2938	0.0000	0.0028	0.1879	0.2012	2.3541	-0.3746	3.6139	90.27	0.0448
10	11.2962	0.0000	0.0029	0.1891	0.2051	2.353	-0.3752	3.6145	90.27	0.0448
11	11.2931	0.0000	0.0028	0.1882	0.2032	2.3532	-0.376	3.6138	90.26	0.0448
12	11.2942	0.0000	0.0028	0.1885	0.2039	2.3537	-0.3743	3.614	90.27	0.0448
13	11.294	0.0000	0.0028	0.1884	0.2036	2.3535	-0.3752	3.614	90.27	0.0448
14	11.2941	0.0000	0.0028	0.1885	0.2038	2.3535	-0.375	3.614	90.27	0.0448
15	11.294	0.0000	0.0028	0.1884	0.2037	2.3535	-0.3751	3.614	90.27	0.0448
16	11.2941	0.0000	0.0028	0.1885	0.2037	2.3535	-0.375	3.614	90.27	0.0448
17	11.2941	0.0000	0.0028	0.1885	0.2037	2.3535	-0.375	3.614	90.27	0.0448
18	11.2941	0.0000	0.0028	0.1885	0.2037	2.3535	-0.375	3.614	90.27	0.0448
19	11.2941	0.0000	0.0028	0.1885	0.2037	2.3535	-0.375	3.614	90.27	0.0448

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	11.2941	0.0032	0.255	0.2348		1.9597	-0.5847	3.0128	90.27	0.0373
1	10.8166	0.0009	0.0029	0.2323	0.2097	2.3771	-0.4407	2.9665	89.45	0.0374
2	10.4876	0.0001	0.0014	0.1679	0.1002	2.497	-0.2711	2.8885	88.80	0.037
3	10.439	-0.0005	0.0003	0.1217	0.0019	2.4883	-0.3834	2.8671	88.71	0.0368
4	11.911	-0.0005	-0.0001	0.0431	-0.0221	1.2926	-0.2025	3.0192	91.97	0.036
5	6.4411	-0.0008	0.001	0.0618	0.1003	1.7747	0.7526	2.3388	77.50	0.0393
6	7.2415	-0.0005	0.0003	0.1224	0.0681	2.5543	0.7527	2.4218	79.21	0.039
7	8.8233	-0.0005	0.0007	0.2026	0.0892	3.7508	0.5883	2.6726	83.99	0.0383
8	9.5832	-0.0007	-0.0003	0.1321	-0.104	2.8409	-1.2389	2.7747	86.38	0.0376
9	7.4672	0.0005	0.0007	0.2026	0.0892	2.8911	0.512	2.4515	79.80	0.0389
10	8.0882	0.0007	-0.0003	0.1321	-0.104	3.1354	0.345	2.5373	81.66	0.0384
11	4.0659	-0.0004	-0.0007	0.151	0.077	-0.4149	-1.326	2.3098	75.29	0.0412
12	3.7561	-0.0006	-0.0001	0.1574	0.0139	0.3308	0.824	2.3054	75.27	0.0411
13	3.1951	-0.0008	0.0026	0.0197	0.0634	0.07	-0.1486	2.3038	75.18	0.0412
14	3.6891	-0.0003	-0.0005	0.0059	0.0248	0.2146	0.3117	2.3042	75.26	0.0411
15	2.9043	-0.0001	0.0002	-0.0001	-0.0003	0.0593	-0.0298	2.3038	75.15	0.0412
16	5.0858	-0.0002	-0.0003	0.0011	0.0037	0.8071	0.9771	2.3104	75.80	0.0406
17	3.6949	-0.0001	0.0000	0.0000	-0.0002	0.0509	-0.309	2.3038	75.25	0.0411
18	3.8651	-0.0006	-0.0004	0.0145	0.0336	0.2598	0.4084	2.3044	75.30	0.0411
19	3.1375	-0.0001	0.0003	-0.0001	0.0000	0.0768	-0.0357	2.3039	75.18	0.0412

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	3.1375	-0.0008	0.0021	0.2556	0.0426	5.513	-1.3273	1.8866	70.43	0.0384
1	5.9314	-0.0008	0.0014	0.2851	0.0575	5.9795	-0.5374	1.9882	72.67	0.038
2	6.5016	0.0000	0.0001	0.2707	-0.0701	4.6892	-1.5898	2.1754	75.55	0.0385
3	6.358	-0.0005	0.0003	0.2377	-0.0441	4.7626	-1.2697	2.0751	74.85	0.0374
4	7.9839	-0.001	-0.0007	0.1086	-0.129	2.7046	-1.25	2.5995	81.48	0.0395
5	6.6073	-0.0002	0.0012	0.2438	0.0436	4.6965	-0.7187	2.0847	76.04	0.0364
6	6.8021	-0.0002	-0.0007	0.2514	-0.1176	4.5042	-1.5307	2.2417	76.90	0.0383
7	6.5295	-0.0003	0.0007	0.2409	-0.018	4.6607	-1.1926	2.1144	75.68	0.0373
8	7.0512	0.0000	-0.0006	0.236	-0.0923	3.9935	-1.1517	2.3147	77.87	0.0386
9	7.0132	-0.0001	0.0003	0.2259	-0.031	4.0777	-1.0652	2.2378	77.82	0.0373
10	7.4476	-0.0004	-0.0004	0.1929	-0.0698	3.6069	-0.8539	2.3783	79.47	0.038
11	6.5042	-0.0005	0.0006	0.2316	-0.0228	4.7627	-1.1524	2.0763	75.54	0.0368
12	7.458	-0.0005	-0.0012	0.186	-0.148	3.5261	-1.2749	2.4587	79.43	0.0394
13	7.0147	0.0002	0.0012	0.2474	0.0361	4.1584	-0.8808	2.2149	77.90	0.0369
14	6.8605	-0.0003	-0.0002	0.2382	-0.0642	4.4324	-1.0776	2.215	77.20	0.0375
15	5.6466	-0.0008	0.0017	0.2651	0.0417	5.6719	-1.0408	1.9088	71.02	0.0382
16	6.1097	-0.0009	0.0006	0.2583	-0.0134	5.5414	-1.0129	2.0284	73.62	0.0378
17	7.5113	-0.0005	-0.0011	0.1662	-0.1588	3.0967	-1.437	2.5078	79.54	0.04
18	7.1033	0.0003	0.001	0.2516	0.0394	4.0833	-0.6793	2.2329	78.26	0.0368
19	6.8043	-0.0004	-0.0001	0.237	-0.0598	4.5285	-1.1006	2.1926	76.97	0.0374

Grid 4

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	6.8043	-0.0079	0.0066	0.3647	0.0148	12.4522	-2.3139	1.5138	59.13	0.0437
1	13.5552	-0.0042	-0.0005	0.1127	-0.2119	5.6709	-3.5578	1.915	69.55	0.04
2	13.4647	-0.0003	-0.0031	0.0177	-0.2165	0.5786	-0.3164	2.4553	79.98	0.0388
3	13.4391	0.0000	-0.0018	0.0016	-0.1622	0.0686	-0.6461	2.7444	81.44	0.0418
4	13.4613	0.0000	-0.0009	0.0152	-0.1116	0.1979	-0.7373	2.9026	82.48	0.0431
5	15.1637	0.0006	-0.0003	0.0251	-0.0696	-0.2007	-0.6432	2.9893	83.50	0.0433
6	15.8654	0.0003	-0.0004	0.0238	-0.0295	0.0603	-0.0253	3.0192	84.17	0.043
7	16.1419	0.0000	0.0001	0.0249	0.0039	0.3772	-0.0335	2.9899	84.68	0.0421
8	16.4685	-0.0001	0.0003	0.0234	0.0245	0.4417	0.0398	2.9383	85.18	0.0409
9	16.5761	0.0000	0.0004	0.0212	0.0312	0.3626	0.0155	2.8872	85.64	0.0398
10	16.6051	0.0000	0.0004	0.0185	0.028	0.2917	-0.0106	2.8484	86.06	0.0388
11	16.4926	0.0000	0.0003	0.0166	0.0205	0.2164	-0.0284	2.8224	86.45	0.0381
12	16.2137	0.0001	0.0002	0.0154	0.0131	0.1796	-0.0083	2.8102	86.81	0.0377
13	15.5102	0.0001	0.0001	0.0151	0.0069	0.1497	-0.0135	2.8095	87.14	0.0374
14	12.7135	0.0001	0.0000	0.0152	0.0017	0.1194	-0.0164	2.8178	87.44	0.0372
15	25.936	0.0001	0.0000	0.0155	-0.0015	0.1133	-0.0144	2.8301	87.71	0.0372
16	31.0394	0.0001	0.0000	0.015	-0.0018	0.1098	-0.0047	2.8433	87.98	0.0371
17	62.2224	0.0001	0.0000	0.0142	-0.0005	0.1123	-0.0056	2.855	88.22	0.0371
18	57.9038	0.0001	0.0000	0.0135	0.0012	0.1165	-0.0035	2.865	88.44	0.037
19	56.1666	0.0001	0.0000	0.0128	0.0027	0.1219	-0.0028	2.873	88.65	0.0369

Grid 3

A.10 Results generated by the MINGCV algorithm using a dampening factor of 1/2, for the data set sine.dat

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.14	0.0099	0.021	0.9247	1.6537	1.9288	-0.1127	3.9351	87.53	0.0519
1	9.675	-0.0014	0.0007	0.0619	0.0913	2.1374	0.5281	3.4655	86.96	0.0463
2	11.8576	0.0008	0.0014	0.1323	0.1028	2.4072	-0.0794	3.5588	89.28	0.0451
3	11.4243	0.0001	0.0034	0.2056	0.256	2.4453	-0.2605	3.6092	89.94	0.0451
4	11.1197	-0.0005	0.002	0.1556	0.1362	2.408	-0.2742	3.587	89.89	0.0448
5	11.3704	0.0001	0.0029	0.185	0.2056	2.3777	-0.3562	3.6084	90.17	0.0448
6	11.2767	-0.0001	0.0026	0.1826	0.188	2.3608	-0.371	3.6095	90.20	0.0448
7	11.295	0.0000	0.0028	0.1867	0.2029	2.3594	-0.3647	3.6119	90.23	0.0448
8	11.2897	0.0000	0.0028	0.1867	0.2003	2.3564	-0.3742	3.6125	90.25	0.0448
9	11.2935	0.0000	0.0028	0.1876	0.2024	2.3556	-0.3716	3.6132	90.26	0.0448
10	11.2935	-0.0000	0.0028	0.188	0.2031	2.3545	-0.3741	3.6136	90.26	0.0448
11	11.2936	-0.0000	0.0028	0.1882	0.2032	2.3541	-0.3742	3.6138	90.26	0.0448
12	11.2939	-0.0000	0.0028	0.1883	0.2035	2.3538	-0.3746	3.6139	90.26	0.0448
13	11.2939	0.0000	0.0028	0.1884	0.2035	2.3537	-0.3748	3.6139	90.27	0.0448
14	11.294	0.0000	0.0028	0.1884	0.2036	2.3536	-0.3749	3.614	90.27	0.0448
15	11.294	0.0000	0.0028	0.1884	0.2037	2.3536	-0.3749	3.614	90.27	0.0448
16	11.294	0.0000	0.0028	0.1884	0.2037	2.3535	-0.375	3.614	90.27	0.0448
17	11.294	-0.0000	0.0028	0.1884	0.2037	2.3535	-0.375	3.614	90.27	0.0448
18	11.2941	-0.0000	0.0028	0.1884	0.2037	2.3535	-0.375	3.614	90.27	0.0448
19	11.2941	-0.0000	0.0028	0.1885	0.2037	2.3535	-0.375	3.614	90.27	0.0448

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	11.2941	0.0015	0.0032	0.255	0.2348	1.9598	-0.5847	3.0128	90.27	0.0373
1	10.8166	0.0012	0.0032	0.2468	0.2402	2.2083	-0.4863	3.0085	89.99	0.0375
2	10.6732	0.0006	0.0025	0.1984	0.1844	2.264	-0.3257	2.9557	89.68	0.0371
3	10.615	0.0000	0.0014	0.15	0.0958	2.3468	-0.2421	2.9122	89.40	0.0368
4	10.7669	-0.0003	0.0006	0.1248	0.0279	2.2741	-0.3766	2.9121	89.47	0.0367
5	11.208	-0.0003	0.0004	0.1141	-0.0043	2.0886	-0.5431	2.9441	90.04	0.0367
6	11.7274	0.0000	0.0004	0.1228	0.0005	1.8235	-0.5848	2.9967	90.84	0.0367
7	11.2843	0.0005	0.001	0.1674	0.0377	1.903	-0.6985	3.0126	90.67	0.037
8	10.7942	0.0008	0.0018	0.207	0.1148	2.1442	-0.6003	2.9851	90.02	0.0372
9	10.6101	0.0006	0.0023	0.1977	0.1656	2.286	-0.3758	2.9431	89.57	0.037
10	10.5722	-0.0001	0.0019	0.1604	0.131	2.3309	-0.3496	2.9122	89.32	0.0369
11	10.6465	-0.0002	0.0011	0.1318	0.0536	2.2965	-0.5152	2.9042	89.28	0.0368
12	10.8819	-0.0003	0.0005	0.122	0.001	2.2068	-0.6625	2.9202	89.55	0.0368
13	11.2896	0.0001	0.0005	0.1258	-0.0004	2.0356	-0.6898	2.9569	90.16	0.0367
14	11.2159	0.0002	0.0009	0.1498	0.0351	2.0125	-0.6509	2.9764	90.29	0.0369
15	10.8924	0.0004	0.0015	0.1763	0.0904	2.1362	-0.5286	2.966	89.98	0.037
16	10.709	0.0004	0.0018	0.1784	0.127	2.2411	-0.3876	2.9423	89.65	0.037
17	10.6437	-0.0001	0.0017	0.1591	0.1122	2.2901	-0.3813	2.9206	89.43	0.0369
18	10.6793	-0.0001	0.0012	0.1392	0.0638	2.2796	-0.5044	2.9119	89.37	0.0368
19	10.815	0.0002	0.0008	0.13	0.0238	2.2271	-0.6241	2.9192	89.51	0.0368

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.815	-0.0000	0.0009	0.1368	0.0398	2.1261	-0.5474	2.8903	89.60	0.0364
1	10.8926	0.0000	0.0009	0.143	0.0346	2.1623	-0.5776	2.9074	89.63	0.0366
2	10.8505	-0.0000	0.0009	0.1402	0.0346	2.1801	-0.6091	2.9029	89.59	0.0365
3	10.8848	0.0000	0.0009	0.1453	0.0324	2.1953	-0.6413	2.9004	89.55	0.0365
4	10.8327	0.0001	0.0009	0.1482	0.031	2.1959	-0.6484	2.8991	89.49	0.0366
5	10.7688	0.0001	0.0009	0.1504	0.0307	2.2026	-0.6534	2.8963	89.42	0.0366
6	10.6992	0.0001	0.0009	0.1534	0.0305	2.233	-0.6711	2.8933	89.33	0.0366
7	10.6289	0.0001	0.0009	0.1566	0.0306	2.2767	-0.6904	2.8874	89.23	0.0366
8	10.5611	0.0001	0.0009	0.1603	0.031	2.3312	-0.7074	2.8801	89.10	0.0366
9	10.4928	0.0001	0.0009	0.164	0.0316	2.3932	-0.7184	2.8712	88.95	0.0367
10	10.4293	0.0001	0.0009	0.1672	0.0327	2.4574	-0.7191	2.8615	88.79	0.0367
11	10.3745	0.0001	0.001	0.1698	0.0344	2.5207	-0.7074	2.8511	88.64	0.0367
12	10.3315	0.0001	0.001	0.1714	0.0365	2.5786	-0.6813	2.8409	88.49	0.0366
13	10.3048	0.0000	0.001	0.1718	0.0391	2.6259	-0.6394	2.8314	88.36	0.0366
14	10.2967	-0.0000	0.001	0.1709	0.042	2.658	-0.5828	2.8233	88.27	0.0366
15	10.3085	-0.0000	0.0009	0.1687	0.0449	2.6721	-0.5152	2.8172	88.22	0.0366
16	10.3409	-0.0001	0.0009	0.1654	0.0478	2.6672	-0.4422	2.8133	88.23	0.0365
17	10.3937	-0.0001	0.0009	0.1611	0.0505	2.6434	-0.3698	2.812	88.28	0.0364
18	10.4657	-0.0001	0.0009	0.156	0.0528	2.6025	-0.3032	2.8133	88.38	0.0364
19	10.5547	-0.0002	0.0009	0.1504	0.0549	2.5468	-0.2462	2.817	88.52	0.0363

Grid 4

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.5547	-0.0002	0.001	0.146	0.0693	2.3821	-0.1922	2.9901	88.74	0.0384
1	10.7537	-0.0002	0.001	0.1451	0.0704	2.368	-0.1846	2.9859	88.78	0.0383
2	10.8389	-0.0002	0.001	0.1438	0.071	2.3459	-0.1645	2.9897	88.83	0.0383
3	10.9282	-0.0002	0.001	0.1425	0.072	2.3199	-0.1504	2.9909	88.88	0.0382
4	11.0104	-0.0002	0.001	0.1413	0.0725	2.2984	-0.1394	2.9928	88.94	0.0382
5	11.0933	-0.0002	0.001	0.1402	0.0732	2.2775	-0.1281	2.9947	89.00	0.0382
6	11.1734	-0.0002	0.001	0.1391	0.0738	2.2536	-0.1156	2.9965	89.06	0.0382
7	11.25	-0.0002	0.001	0.138	0.0745	2.2305	-0.1041	2.9985	89.12	0.0381
8	11.3243	-0.0001	0.001	0.1367	0.0752	2.2078	-0.0928	3.0005	89.19	0.0381
9	11.397	-0.0001	0.001	0.1355	0.0759	2.1848	-0.0817	3.0027	89.25	0.0381
10	11.4681	-0.0001	0.001	0.1342	0.0767	2.1615	-0.0704	3.0049	89.32	0.038
11	11.5373	-0.0001	0.001	0.1329	0.0775	2.138	-0.0592	3.0071	89.38	0.038
12	11.6045	-0.0001	0.001	0.1316	0.0782	2.1147	-0.0484	3.0094	89.45	0.038
13	11.6699	-0.0001	0.001	0.1304	0.079	2.0916	-0.0378	3.0117	89.51	0.038
14	11.7336	-0.0001	0.001	0.1291	0.0798	2.0687	-0.0275	3.0139	89.57	0.0379
15	11.7955	-0.0001	0.001	0.1278	0.0806	2.0461	-0.0175	3.0162	89.64	0.0379
16	11.8559	-0.0001	0.001	0.1266	0.0814	2.0237	-0.0077	3.0186	89.70	0.0379
17	11.9146	-0.0001	0.001	0.1253	0.0821	2.0013	0.002	3.0209	89.76	0.0379
18	11.9717	-0.0001	0.001	0.1241	0.0829	1.9791	0.0116	3.0232	89.82	0.0379
19	12.0271	-0.0001	0.001	0.1229	0.0837	1.957	0.0211	3.0255	89.87	0.0378

Grid 3

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	12.0271	-0.0000	0.0000	0.0121	0.0022	0.1439	0.0251	2.7782	88.70	0.0357
1	50.9255	-0.0000	0-.0000	0.012	0.0023	0.1452	0.0257	2.7792	88.71	0.0357
2	46.389	-0.0000	-0.0000	0.0119	0.0024	0.1456	0.025	2.779	88.72	0.0357
3	43.0275	-0.0000	-0.0000	0.0119	0.0025	0.1458	0.0247	2.7793	88.73	0.0357
4	40.0008	-0.0000	-0.0000	0.0118	0.0026	0.146	0.0244	2.7795	88.75	0.0356
5	37.3851	-0.0000	-0.0000	0.0117	0.0026	0.1459	0.024	2.7797	88.76	0.0356
6	35.0393	-0.0000	-0.0000	0.0117	0.0027	0.146	0.0236	2.7799	88.77	0.0356
7	32.8663	-0.0000	-0.0000	0.0116	0.0027	0.1459	0.0232	2.7801	88.78	0.0356
8	30.8545	-0.0000	-0.0000	0.0116	0.0028	0.1457	0.0229	2.7802	88.79	0.0356
9	28.9681	-0.0000	-0.0000	0.0115	0.0028	0.1455	0.0226	2.7804	88.80	0.0356
10	27.1621	-0.0000	-0.0000	0.0115	0.0028	0.1453	0.0222	2.7806	88.82	0.0356
11	25.4299	-0.0000	-0.0000	0.0114	0.0028	0.145	0.0219	2.7808	88.83	0.0356
12	23.7502	-0.0000	-0.0000	0.0114	0.0028	0.1447	0.0217	2.781	88.84	0.0356
13	22.1109	-0.0000	-0.0000	0.0113	0.0028	0.1443	0.0214	2.7812	88.85	0.0356
14	20.5001	-0.0000	-0.0000	0.0113	0.0028	0.144	0.0211	2.7814	88.86	0.0356
15	18.9084	-0.0000	-0.0000	0.0112	0.0028	0.1436	0.0209	2.7816	88.87	0.0356
16	17.3276	-0.0000	-0.0000	0.0112	0.0027	0.1432	0.0207	2.7819	88.88	0.0356
17	15.7505	-0.0000	-0.0000	0.0112	0.0027	0.1428	0.0204	2.7821	88.89	0.0356
18	14.1713	-0.0000	-0.0000	0.0111	0.0027	0.1428	0.0199	2.7823	88.90	0.0356
19	12.5889	-0.0000	-0.0000	0.0112	0.0025	0.144	0.018	2.7824	88.91	0.0356

Grid 2

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	13.1278	-0.0001	0.001	0.12	0.0883	1.9275	0.0513	2.8983	90.06	0.0361
1	13.1525	-0.0001	0.001	0.12	0.0883	1.9279	0.0514	2.8983	90.06	0.0361
2	13.1768	-0.0001	0.001	0.12	0.0883	1.9275	0.0515	2.8984	90.06	0.0361
3	13.2014	-0.0001	0.001	0.12	0.0883	1.9274	0.0516	2.8984	90.06	0.0361
4	13.2258	-0.0001	0.001	0.12	0.0883	1.9272	0.0516	2.8984	90.06	0.0361
5	13.2504	-0.0001	0.001	0.12	0.0883	1.9271	0.0518	2.8984	90.06	0.0361
6	13.2749	-0.0001	0.001	0.12	0.0883	1.9269	0.0518	2.8984	90.07	0.0361
7	13.2994	-0.0001	0.001	0.12	0.0884	1.9268	0.052	2.8984	90.07	0.0361
8	13.3239	-0.0001	0.001	0.1199	0.0884	1.9266	0.052	2.8985	90.07	0.0361
9	13.3484	-0.0001	0.001	0.1199	0.0884	1.9264	0.0521	2.8985	90.07	0.0361
10	13.3729	-0.0001	0.001	0.1199	0.0884	1.9263	0.0522	2.8985	90.07	0.0361
11	13.3974	-0.0001	0.001	0.1199	0.0884	1.9261	0.0523	2.8985	90.07	0.0361
12	13.4219	-0.0001	0.001	0.1199	0.0884	1.9259	0.0524	2.8985	90.07	0.0361
13	13.4465	-0.0001	0.001	0.1199	0.0884	1.9257	0.0526	2.8986	90.07	0.0361
14	13.471	-0.0001	0.001	0.1199	0.0884	1.9256	0.0527	2.8986	90.07	0.0361
15	13.4955	-0.0001	0.001	0.1199	0.0884	1.9254	0.0528	2.8986	90.07	0.0361
16	13.52	-0.0001	0.001	0.1199	0.0884	1.9252	0.0529	2.8986	90.07	0.0361
17	13.5446	-0.0001	0.001	0.1198	0.0884	1.925	0.053	2.8986	90.07	0.0361
18	13.5691	-0.0001	0.001	0.1198	0.0884	1.9249	0.0531	2.8986	90.07	0.0361
19	13.5936	-0.0001	0.001	0.1198	0.0885	1.9247	0.0532	2.8987	90.07	0.0361

Grid 1

Appendix B

Results for Chapter 11

The tables in this appendix are discussed in Chapter 11. They report updates of various quantities relevant to the OPTRSS and MINGCV algorithms described in Chapter 8. The notation is explained as follows:

q: the number of updates performed on a given grid.

θ: the logarithm of the smoothing parameter.

θ_q: the q^{th} update of the logarithm of the smoothing parameter.

R: the estimate of the residual sum of squares.

Tr: the estimate of $tr(I - A)$, where A is the influence matrix.

GCV: the estimate of the generalised cross validation.

B.1 Results generated by the MINGCV algorithm, using quadratic B-spline discretisation

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.1464	0.0034	0.0065	0.3418	0.5938	0.4699	-0.0546	3.0588	96.19	0.0334
1	9.6187	0.0004	0.0003	0.0632	0.0254	0.5013	0.0028	2.8841	95.93	0.0317
2	8.2639	0.0002	0.0002	0.0447	0.0226	0.4485	0.0935	2.813	95.28	0.0313
3	7.1707	-0.0000	0.0002	0.0193	0.0209	0.3331	0.1182	2.779	94.85	0.0312
4	7.1897	-0.0000	0.0002	0.0191	0.0211	0.3364	0.119	2.7789	94.85	0.0312
5	7.2321	-0.0000	0.0002	0.0198	0.0211	0.3439	0.1209	2.7797	94.87	0.0312
6	7.2541	-0.0000	0.0002	0.0203	0.0214	0.3469	0.1223	2.7801	94.87	0.0312
7	7.2542	-0.0000	0.0002	0.0203	0.0214	0.3468	0.1222	2.7801	94.87	0.0312
8	7.253	0.0000	0.0002	0.0203	0.0214	0.3466	0.122	2.7801	94.87	0.0312
9	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
10	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
11	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
12	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
13	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
14	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
15	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
16	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
17	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
18	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
19	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	7.2525	-0.0007	0.0007	0.0684	0.0156	2.238	-0.6422	2.6053	91.60	0.0314
1	8.3281	-0.0000	0.0005	0.0874	0.015	1.5285	-0.5079	2.6845	93.66	0.0309
2	8.3323	0.0000	0.0006	0.0918	0.0205	1.5829	-0.6241	2.686	93.67	0.0309
3	8.314	-0.0000	0.0006	0.0904	0.0196	1.5954	-0.5912	2.6829	93.64	0.0309
4	8.3341	0.0000	0.0006	0.0905	0.0189	1.5769	-0.595	2.6845	93.67	0.0309
5	8.3325	0.0000	0.0006	0.0905	0.0189	1.5792	-0.5976	2.6845	93.67	0.0309
6	8.3322	-0.0000	0.0006	0.0905	0.019	1.5794	-0.5969	2.6844	93.67	0.0309
7	8.3324	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
8	8.3323	0.0000	0.0006	0.0905	0.019	1.5793	-0.597	2.6844	93.67	0.0309
9	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
10	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
11	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
12	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
13	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
14	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
15	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
16	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
17	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
18	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
19	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.3323	0.0001	0.0006	0.1252	-0.0013	2.0787	-1.0172	2.6337	92.96	0.0308
1	8.1925	-0.0000	0.0006	0.1288	-0.0033	2.2976	-1.0531	2.6081	92.65	0.0307
2	8.2021	-0.0001	0.0005	0.1209	-0.0119	2.2756	-1.0054	2.6127	92.67	0.0307
3	8.3775	0.0000	0.0004	0.1174	-0.0177	2.0462	-0.9581	2.6355	93.05	0.0307
4	8.3331	0.0001	0.0005	0.1285	-0.0106	2.1019	-1.0578	2.635	92.96	0.0308
5	8.1305	0.0000	0.0008	0.1337	0.0051	2.3738	-1.1029	2.6062	92.51	0.0308
6	8.1313	-0.0001	0.0006	0.1226	-0.0053	2.3506	-1.0248	2.6023	92.51	0.0307
7	8.325	-0.0000	0.0004	0.1154	-0.0215	2.0904	-0.9735	2.6287	92.94	0.0307
8	8.4156	0.0001	0.0004	0.1233	-0.0191	2.0028	-1.0054	2.6449	93.13	0.0308
9	8.142	0.0001	0.0008	0.1374	0.0045	2.3561	-1.1331	2.6102	92.53	0.0308
10	8.0728	-0.0001	0.0007	0.1256	0.0012	2.4381	-1.039	2.5944	92.37	0.0307
11	8.271	-0.0001	0.0004	0.1142	-0.0217	2.1478	-0.9771	2.6208	92.82	0.0307
12	8.4987	0.0001	0.0003	0.1167	-0.0263	1.9061	-0.9462	2.6536	93.29	0.0308
13	8.1653	0.0001	0.0007	0.14	0.0007	2.3205	-1.1672	2.616	92.59	0.0308
14	8.0232	-0.0001	0.0008	0.1289	0.007	2.5168	-1.0508	2.5877	92.25	0.0307
15	8.2176	-0.0001	0.0004	0.1134	-0.0216	2.2077	-0.9795	2.6132	92.70	0.0307
16	8.5664	0.0001	0.0002	0.1104	-0.032	1.8309	-0.8916	2.6601	93.42	0.0308
17	8.1881	0.0001	0.0007	0.1416	-0.0041	2.2846	-1.1952	2.6214	92.64	0.0308
18	7.9792	-0.0002	0.0008	0.1316	0.0115	2.5864	-1.0612	2.5819	92.14	0.0307
19	8.1746	-0.0002	0.0004	0.1127	-0.0221	2.2565	-0.9802	2.607	92.61	0.0307

Grid 4

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.1746	-0.0001	0.0003	0.1087	-0.021	2.158	-0.9297	2.6163	93.12	0.0305
1	9.0554	-0.0001	0.0003	0.0963	-0.0137	1.8851	-0.7286	2.6364	93.76	0.0303
2	9.4306	-0.0001	0.0003	0.0827	-0.0061	1.6113	-0.5387	2.6588	94.27	0.0302
3	9.781	-0.0001	0.0002	0.0725	-0.0017	1.3793	-0.3892	2.6782	94.66	0.0302
4	10.0664	-0.0000	0.0002	0.0656	0	1.2138	-0.2975	2.6942	94.96	0.0302
5	10.271	-0.0000	0.0001	0.0614	-0.0005	1.1027	-0.2498	2.7071	95.17	0.0302
6	10.3755	-0.0000	0.0001	0.0587	-0.0021	1.0288	-0.2306	2.7183	95.33	0.0302
7	10.3688	0.0000	0.0001	0.0572	-0.0045	0.9799	-0.2298	2.728	95.42	0.0303
8	10.2272	0.0000	0.0001	0.0569	-0.0077	0.9571	-0.2487	2.7362	95.44	0.0303
9	9.888	0.0000	0.0000	0.0593	-0.013	0.9805	-0.3064	2.741	95.32	0.0305
10	9.1334	0.0000	0.0000	0.0716	-0.026	1.1684	-0.5128	2.7342	94.73	0.0308
11	7.5054	0.0001	0.0002	0.1496	-0.0808	2.4704	-1.7255	2.6438	90.66	0.0325
12	7.1449	0.0002	0.0005	0.219	-0.096	3.5955	-2.5451	2.5377	88.77	0.0325
13	6.8008	-0.0001	0.0012	0.2565	-0.0617	4.6778	-2.9552	2.4361	87.39	0.0322
14	6.8489	-0.0003	0.0015	0.2423	-0.0127	4.9029	-2.4895	2.3977	87.74	0.0315
15	7.0735	-0.0005	0.0014	0.2038	0.0158	4.4965	-1.8392	2.4159	88.76	0.031
16	7.4397	-0.0005	0.0012	0.1617	0.0265	3.7365	-1.3117	2.4639	90.15	0.0306
17	7.8798	-0.0005	0.001	0.1247	0.0322	2.9553	-0.947	2.5168	91.47	0.0304
18	8.3311	-0.0004	0.0009	0.0961	0.0387	2.3138	-0.6738	2.5613	92.53	0.0302
19	8.7529	-0.0003	0.0008	0.0758	0.0452	1.8664	-0.464	2.5942	93.33	0.0301

Grid 3

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.7529	-0.0003	0.0008	0.0744	0.0461	1.8285	-0.4409	2.5968	93.44	0.03
1	9.5311	-0.0003	0.0008	0.0722	0.0479	1.7755	-0.403	2.6002	93.60	0.03
2	9.9074	-0.0003	0.0008	0.0698	0.0497	1.7167	-0.3589	2.6038	93.76	0.0299
3	10.273	-0.0003	0.0008	0.0675	0.0516	1.6567	-0.3134	2.6075	93.92	0.0299
4	10.6278	-0.0003	0.0008	0.0651	0.0535	1.5985	-0.2692	2.6111	94.06	0.0298
5	10.9724	-0.0003	0.0008	0.0628	0.0554	1.5435	-0.2276	2.6147	94.19	0.0298
6	11.3082	-0.0003	0.0008	0.0606	0.0571	1.4921	-0.1889	2.6181	94.29	0.0297
7	11.6361	-0.0002	0.0008	0.0586	0.0588	1.4444	-0.1534	2.6214	94.39	0.0297
8	11.9572	-0.0002	0.0008	0.0566	0.0603	1.4004	-0.1211	2.6246	94.47	0.0297
9	12.2725	-0.0002	0.0008	0.0548	0.0617	1.3599	-0.0919	2.6277	94.54	0.0297
10	12.5827	-0.0002	0.0008	0.0532	0.063	1.3226	-0.0653	2.6306	94.61	0.0297
11	12.8883	-0.0002	0.0007	0.0516	0.0641	1.288	-0.0411	2.6334	94.66	0.0297
12	13.1896	-0.0002	0.0007	0.0502	0.0651	1.2558	-0.0189	2.636	94.72	0.0297
13	13.4868	-0.0002	0.0007	0.0489	0.066	1.2256	0.0015	2.6385	94.77	0.0297
14	13.7799	-0.0002	0.0007	0.0477	0.0668	1.1974	0.0203	2.6409	94.82	0.0297
15	14.0688	-0.0002	0.0007	0.0466	0.0675	1.171	0.0374	2.6432	94.86	0.0297
16	14.3537	-0.0002	0.0007	0.0456	0.0681	1.1463	0.0531	2.6454	94.90	0.0297
17	14.6345	-0.0002	0.0007	0.0447	0.0686	1.1233	0.0673	2.6475	94.94	0.0297
18	14.9113	-0.0002	0.0007	0.0438	0.069	1.1018	0.0801	2.6495	94.98	0.0297
19	15.184	-0.0002	0.0007	0.0431	0.0694	1.0817	0.0917	2.6514	95.02	0.0297

Grid 2

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	15.184	0.0000	0.0006	0.0907	0.0189	1.5471	0.0924	2.6515	95.02	0.0297
1	8.3323	-0.0002	0.0004	0.1127	-0.022	2.2495	0.093	2.6516	95.02	0.0297
2	8.6308	-0.0003	0.0008	0.0758	0.0452	1.8657	0.0937	2.6517	95.02	0.0297
3	9.1459	-0.0002	0.0007	0.0431	0.0694	1.0817	0.0943	2.6518	95.02	0.0297
4	15.4528	-0.0002	0.0007	0.043	0.0694	1.0804	0.095	2.6519	95.03	0.0297
5	15.7209	-0.0002	0.0007	0.043	0.0694	1.0792	0.0956	2.652	95.03	0.0297
6	15.9885	-0.0002	0.0007	0.043	0.0694	1.0779	0.0962	2.6522	95.04	0.0297
7	16.2555	-0.0002	0.0007	0.043	0.0694	1.0766	0.0969	2.6523	95.04	0.0297
8	16.522	-0.0002	0.0007	0.0429	0.0694	1.0754	0.0975	2.6524	95.04	0.0297
9	16.7879	-0.0002	0.0007	0.0429	0.0694	1.0742	0.0981	2.6525	95.04	0.0297
10	17.0533	-0.0002	0.0007	0.0429	0.0694	1.0729	0.0987	2.6526	95.04	0.0297
11	17.3181	-0.0002	0.0007	0.0429	0.0694	1.0717	0.0993	2.6527	95.05	0.0297
12	17.5824	-0.0002	0.0007	0.0428	0.0694	1.0705	0.0998	2.6528	95.05	0.0297
13	17.8462	-0.0002	0.0007	0.0428	0.0695	1.0693	0.1004	2.6529	95.05	0.0297
14	18.1095	-0.0002	0.0007	0.0428	0.0695	1.0681	0.101	2.653	95.05	0.0297
15	18.3722	-0.0002	0.0007	0.0428	0.0695	1.067	0.1015	2.6531	95.05	0.0297
16	18.6345	-0.0002	0.0007	0.0427	0.0695	1.0658	0.1021	2.6532	95.06	0.0297
17	18.8962	-0.0002	0.0007	0.0427	0.0695	1.0646	0.1026	2.6534	95.06	0.0297
18	19.1574	-0.0002	0.0007	0.0427	0.0695	1.0635	0.1032	2.6535	95.06	0.0297
19	19.4181	-0.0002	0.0007	0.0427	0.0695	1.0623	0.1037	2.6536	95.06	0.0297

Grid 1

B.2 Results generated by the MINGCV algorithm, for a second random vector t

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.1464	0.0026	0.0076	0.3418	0.5938	1.9649	-0.2636	3.0588	88.84	0.0391
1	9.7968	-0.001	0.0005	0.0618	0.0206	2.1308	-0.2374	2.894	88.11	0.0376
2	11.7322	0.0031	0.0069	0.3457	0.5555	1.2816	-0.4647	3.138	91.69	0.0377
3	11.2808	0.0007	0.0028	0.1703	0.2087	1.6325	-0.4272	3.0328	90.96	0.037
4	11.0147	0.0001	0.0021	0.1285	0.1519	1.7714	-0.3001	2.9938	90.50	0.0369
5	10.9474	-0.0000	0.0019	0.1185	0.1336	1.7893	-0.3001	2.9849	90.38	0.0369
6	10.9456	-0.0000	0.0019	0.1183	0.1334	1.7918	-0.3007	2.9845	90.37	0.0369
7	10.9459	-0.0000	0.0019	0.1183	0.1335	1.7921	-0.3008	2.9845	90.37	0.0369
8	10.9463	-0.0000	0.0019	0.1183	0.1335	1.7919	-0.3008	2.9845	90.37	0.0369
9	10.9465	-0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
10	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
11	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
12	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
13	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
14	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
15	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
16	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
17	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
18	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
19	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.9465	0.0000	0.0018	0.1404	0.119	2.1228	-0.4369	2.9412	89.94	0.0367
1	10.9355	-0.0000	0.0017	0.133	0.1119	2.0786	-0.3697	2.9482	89.92	0.0368
2	10.96	.0000	0.0017	0.133	0.1119	2.0292	-0.4136	2.95	89.97	0.0368
3	10.946	0.0000	0.0017	0.1349	0.1094	2.0526	-0.4172	2.95	89.94	0.0368
4	10.9149	0.0001	0.0017	0.139	0.1148	2.079	-0.4018	2.9456	89.87	0.0368
5	10.908	-0.0000	0.0017	0.1372	0.1166	2.0707	-0.4182	2.9439	89.86	0.0368
6	10.9267	-0.0000	0.0017	0.1332	0.1097	2.0492	-0.4335	2.9463	89.90	0.0368
7	10.9381	0.0000	0.0016	0.1328	0.1054	2.0456	-0.4313	2.9483	89.92	0.0368
8	10.9276	0.0000	0.0017	0.1355	0.1091	2.0605	-0.422	2.9471	89.90	0.0368
9	10.9156	-0.0000	0.0017	0.1368	0.1136	2.0687	-0.4182	2.9453	89.88	0.0368
10	10.9187	-0.0000	0.0017	0.1352	0.1125	2.061	-0.4221	2.9455	89.88	0.0368
11	10.9284	-0.0000	0.0017	0.1338	0.1089	2.0523	-0.4258	2.9468	89.90	0.0368
12	10.9295	0.0000	0.0017	0.1344	0.1084	2.0548	-0.4248	2.9472	89.90	0.0368
13	10.9229	0.0000	0.0017	0.1356	0.1108	2.0619	-0.4221	2.9463	89.89	0.0368
14	10.9198	-0.0000	0.0017	0.1356	0.112	2.0626	-0.422	2.9458	89.88	0.0368
15	10.9234	-0.0000	0.0017	0.1347	0.1107	2.0578	-0.4238	2.9462	89.89	0.0368
16	10.9269	0.0000	0.0017	0.1344	0.1094	2.0556	-0.4244	2.9467	89.90	0.0368
17	10.9256	0.0000	0.0017	0.1349	0.1099	2.0582	-0.4234	2.9466	89.90	0.0368
18	10.9226	0.0000	0.0017	0.1353	0.111	2.0608	-0.4226	2.9462	89.89	0.0368
19	10.9225	0.0000	0.0017	0.1351	0.111	2.0597	-0.423	2.9461	89.89	0.0368

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.9225	0.0000	0.0016	0.1401	0.1055	2.0915	-0.4509	2.9401	89.80	0.0368
1	10.9005	0.0001	0.0016	0.1424	0.1038	2.1114	-0.4616	2.937	89.74	0.0368
2	10.8687	0.0000	0.0016	0.1431	0.1033	2.1328	-0.4691	2.9343	89.67	0.0369
3	10.8413	0.0000	0.0016	0.144	0.1025	2.1567	-0.4753	2.9313	89.61	0.0369
4	10.819	0.0000	0.0016	0.1442	0.102	2.1782	-0.4763	2.9288	89.56	0.0369
5	10.8056	0.0000	0.0016	0.144	0.1015	2.195	-0.472	2.9266	89.53	0.0369
6	10.8021	0.0000	0.0016	0.1433	0.1011	2.2057	-0.463	2.9251	89.52	0.0369
7	10.8089	0.0000	0.0016	0.1422	0.1009	2.2083	-0.4499	2.9243	89.53	0.0368
8	10.825	-0.0000	0.0016	0.1409	0.1008	2.2029	-0.4345	2.9244	89.57	0.0368
9	10.8484	-0.0000	0.0016	0.1394	0.1007	2.1904	-0.4187	2.9252	89.62	0.0368
10	10.8768	-0.0000	0.0016	0.1379	0.1008	2.1721	-0.4041	2.9267	89.69	0.0368
11	10.9076	-0.0000	0.0015	0.1365	0.1009	2.1498	-0.3918	2.9288	89.75	0.0367
12	10.9381	-0.0000	0.0015	0.1353	0.101	2.1254	-0.3828	2.9311	89.82	0.0367
13	10.9656	-0.0000	0.0015	0.1344	0.1011	2.1011	-0.378	2.9336	89.88	0.0367
14	10.9877	-0.0000	0.0015	0.1339	0.1012	2.0787	-0.3776	2.9361	89.93	0.0367
15	11.0027	-0.0000	0.0015	0.1337	0.1012	2.0602	-0.3819	2.9382	89.96	0.0367
16	11.0095	0.0000	0.0015	0.134	0.101	2.0468	-0.3906	2.94	89.98	0.0367
17	11.0077	0.0000	0.0015	0.1346	0.1008	2.0398	-0.4033	2.9412	89.98	0.0367
18	10.9976	0.0000	0.0016	0.1356	0.1004	2.0395	-0.419	2.9418	89.96	0.0367
19	10.9801	0.0000	0.0016	0.1368	0.1	2.0459	-0.4367	2.9418	89.92	0.0367

Grid 4

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.9801	-0.0000	0.0016	0.1375	0.0994	2.0553	-0.446	2.9411	89.90	0.0368
1	10.9325	0.0000	0.0016	0.1382	0.0988	2.0614	-0.4517	2.9404	89.88	0.0368
2	10.9062	0.0000	0.0016	0.1387	0.0984	2.0674	-0.4569	2.9398	89.86	0.0368
3	10.8791	0.0000	0.0016	0.1391	0.0981	2.0741	-0.4624	2.9392	89.83	0.0368
4	10.8518	0.0000	0.0016	0.1396	0.0978	2.0821	-0.4687	2.9386	89.80	0.0368
5	10.8252	0.0000	0.0016	0.1401	0.0974	2.0911	-0.4756	2.9379	89.76	0.0368
6	10.7994	0.0000	0.0016	0.1407	0.0969	2.101	-0.4828	2.9371	89.73	0.0368
7	10.7746	0.0000	0.0016	0.1412	0.0965	2.1115	-0.4903	2.9362	89.68	0.0369
8	10.7508	0.0000	0.0016	0.1418	0.096	2.1227	-0.498	2.9352	89.64	0.0369
9	10.7281	0.0000	0.0016	0.1425	0.0955	2.1344	-0.5058	2.9342	89.60	0.0369
10	10.7068	0.0000	0.0016	0.1431	0.0949	2.1465	-0.5136	2.9331	89.55	0.0369
11	10.687	0.0000	0.0016	0.1437	0.0944	2.159	-0.5214	2.9319	89.51	0.037
12	10.6688	0.0000	0.0016	0.1443	0.0939	2.1717	-0.529	2.9308	89.46	0.037
13	10.6525	0.0000	0.0016	0.1449	0.0934	2.1845	-0.5363	2.9296	89.42	0.037
14	10.6382	0.0000	0.0016	0.1455	0.0929	2.1972	-0.5433	2.9284	89.38	0.037
15	10.6259	0.0000	0.0016	0.1461	0.0925	2.2097	-0.5498	2.9272	89.34	0.037
16	10.6158	0.0000	0.0016	0.1466	0.092	2.2219	-0.5558	2.926	89.30	0.0371
17	10.6079	0.0000	0.0016	0.1471	0.0916	2.2337	-0.5612	2.9249	89.27	0.0371
18	10.6022	0.0000	0.0016	0.1475	0.0913	2.2449	-0.5659	2.9237	89.23	0.0371
19	10.5988	0.0000	0.0016	0.1479	0.091	2.2554	-0.5698	2.9226	89.21	0.0371

Grid 3

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.5988	0.0000	0.0016	0.1481	0.0908	2.2579	-0.5719	2.9225	89.20	0.0371
1	10.5969	0.0000	0.0016	0.1482	0.0908	2.2592	-0.5726	2.9224	89.20	0.0371
2	10.5963	0.0000	0.0016	0.1482	0.0907	2.2603	-0.5732	2.9223	89.20	0.0371
3	10.5958	0.0000	0.0016	0.1483	0.0907	2.2613	-0.5736	2.9222	89.19	0.0371
4	10.5955	0.0000	0.0016	0.1483	0.0906	2.2622	-0.574	2.9221	89.19	0.0371
5	10.5952	0.0000	0.0016	0.1484	0.0906	2.2631	-0.5744	2.922	89.19	0.0371
6	10.595	0.0000	0.0016	0.1484	0.0906	2.264	-0.5747	2.9219	89.19	0.0371
7	10.5949	0.0000	0.0016	0.1484	0.0905	2.2648	-0.575	2.9218	89.19	0.0371
8	10.5948	0.0000	0.0016	0.1485	0.0905	2.2657	-0.5753	2.9217	89.19	0.0371
9	10.5948	-0.0000	0.0016	0.1485	0.0905	2.2665	-0.5756	2.9216	89.18	0.0371
10	10.595	-0.0000	0.0016	0.1486	0.0905	2.2673	-0.5759	2.9215	89.18	0.0371
11	10.5952	-0.0000	0.0016	0.1486	0.0904	2.2681	-0.5761	2.9214	89.18	0.0371
12	10.5955	-0.0000	0.0016	0.1486	0.0904	2.2689	-0.5764	2.9213	89.18	0.0371
13	10.596	-0.0000	0.0016	0.1487	0.0904	2.2696	-0.5766	2.9212	89.18	0.0371
14	10.5965	-0.0000	0.0016	0.1487	0.0904	2.2704	-0.5768	2.9211	89.18	0.0371
15	10.5972	-0.0000	0.0016	0.1487	0.0903	2.2711	-0.577	2.9211	89.17	0.0371
16	10.598	-0.0000	0.0016	0.1487	0.0903	2.2718	-0.5772	2.921	89.17	0.0371
17	10.5989	-0.0000	0.0016	0.1488	0.0903	2.2725	-0.5774	2.9209	89.17	0.0371
18	10.5999	-0.0000	0.0016	0.1488	0.0903	2.2732	-0.5775	2.9208	89.17	0.0371
19	10.6011	-0.0000	0.0016	0.1488	0.0903	2.2738	-0.5777	2.9207	89.17	0.0371

Grid 2

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.6011	0.0000	0.0016	0.148	0.0908	2.2579	-0.5719	2.9225	89.20	0.0371
1	10.5969	0.0000	0.0016	0.1481	0.0908	2.2592	-0.5726	2.9224	89.20	0.0371
2	10.5963	0.0000	0.0016	0.1482	0.0908	2.2603	-0.5732	2.9223	89.20	0.0371
3	10.5958	0.0000	0.0016	0.1482	0.0907	2.2613	-0.5736	2.9222	89.19	0.0371
4	10.5955	0.0000	0.0016	0.1483	0.0907	2.2622	-0.574	2.9221	89.19	0.0371
5	10.5952	0.0000	0.0016	0.1483	0.0906	2.2631	-0.5744	2.922	89.19	0.0371
6	10.595	0.0000	0.0016	0.1484	0.0906	2.264	-0.5747	2.9219	89.19	0.0371
7	10.5949	0.0000	0.0016	0.1484	0.0906	2.2648	-0.575	2.9218	89.19	0.0371
8	10.5948	0.0000	0.0016	0.1484	0.0905	2.2657	-0.5753	2.9217	89.19	0.0371
9	10.5948	-0.0000	0.0016	0.1485	0.0905	2.2665	-0.5756	2.9216	89.18	0.0371
10	10.595	-0.0000	0.0016	0.1485	0.0905	2.2673	-0.5759	2.9215	89.18	0.0371
11	10.5952	-0.0000	0.0016	0.1486	0.0905	2.2681	-0.5761	2.9214	89.18	0.0371
12	10.5955	-0.0000	0.0016	0.1486	0.0904	2.2689	-0.5764	2.9213	89.18	0.0371
13	10.596	-0.0000	0.0016	0.1486	0.0904	2.2696	-0.5766	2.9212	89.18	0.0371
14	10.5965	-0.0000	0.0016	0.1487	0.0904	2.2704	-0.5768	2.9211	89.18	0.0371
15	10.5972	-0.0000	0.0016	0.1487	0.0904	2.2711	-0.577	2.9211	89.17	0.0371
16	10.598	-0.0000	0.0016	0.1487	0.0903	2.2718	-0.5772	2.921	89.17	0.0371
17	10.5989	-0.0000	0.0016	0.1487	0.0903	2.2725	-0.5774	2.9209	89.17	0.0371
18	10.5999	-0.0000	0.0016	0.1488	0.0903	2.2732	-0.5775	2.9208	89.17	0.0371
19	10.6011	-0.0000	0.0016	0.1488	0.0903	2.2738	-0.5777	2.9207	89.17	0.0371

Grid 1

B.3 Results generated by the MINGCV algorithm, for a third random vector t

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.1464	0.0035	0.0074	0.3418	0.5938	0.9694	-0.1281	3.0588	89.49	0.0386
1	9.6793	-0.0001	0.0004	0.0623	0.0227	1.0958	-0.146	2.8874	89.00	0.0368
2	9.9558	-0.0001	0.0003	0.0627	0.0171	1.0397	-0.1687	2.9038	89.30	0.0368
3	10.1339	-0.0000	0.0004	0.0657	0.0238	1.0113	-0.165	2.9152	89.48	0.0368
4	10.1404	-0.0000	0.0004	0.0656	0.0235	1.0067	-0.1628	2.9162	89.49	0.0368
5	10.1407	0.0000	0.0004	0.0656	0.0235	1.0063	-0.1625	2.9162	89.49	0.0368
6	10.1406	-0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
7	10.1407	-0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
8	10.1407	-0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
9	10.1407	-0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
10	10.1407	-0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
11	10.1407	-0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
12	10.1407	-0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
13	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
14	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
15	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
16	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
17	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
18	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
19	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.1407	0.0001	0.0002	0.0963	0.0104	1.3645	-0.1334	2.8525	89.08	0.0363
1	9.6156	0.0001	0.0001	0.0962	-0.0046	1.4483	-0.2044	2.8041	88.39	0.0362
2	8.9663	0.0000	0.0001	0.1015	0.0016	1.5777	-0.1788	2.7516	87.43	0.0364
3	8.7668	0.0000	0.0002	0.0996	0.0085	1.6014	-0.1747	2.7286	87.11	0.0363
4	8.8082	-0.0000	0.0003	0.0976	0.0094	1.5902	-0.1867	2.7292	87.18	0.0363
5	8.9106	-0.0000	0.0003	0.0973	0.0075	1.5705	-0.2097	2.7379	87.34	0.0362
6	8.9723	0.0000	0.0002	0.098	0.0057	1.5595	-0.2236	2.7445	87.44	0.0363
7	8.9656	0.0000	0.0002	0.0986	0.0054	1.5623	-0.2198	2.7449	87.43	0.0363
8	8.9342	0.0000	0.0002	0.0987	0.006	1.5689	-0.2118	2.7422	87.38	0.0363
9	8.9198	-0.0000	0.0002	0.0985	0.0064	1.5713	-0.2087	2.7405	87.36	0.0363
10	8.9245	-0.0000	0.0002	0.0983	0.0065	1.57	-0.2101	2.7407	87.36	0.0363
11	8.934	-0.0000	0.0002	0.0983	0.0063	1.5681	-0.2124	2.7415	87.38	0.0363
12	8.938	0.0000	0.0002	0.0984	0.0061	1.5675	-0.2132	2.742	87.39	0.0363
13	8.9363	0.0000	0.0002	0.0984	0.0061	1.5679	-0.2126	2.7419	87.38	0.0363
14	8.9335	0.0000	0.0002	0.0984	0.0062	1.5685	-0.212	2.7416	87.38	0.0363
15	8.9325	0.0000	0.0002	0.0984	0.0062	1.5686	-0.2118	2.7415	87.38	0.0363
16	8.9331	-0.0000	0.0002	0.0984	0.0062	1.5685	-0.2119	2.7416	87.38	0.0363
17	8.934	-0.0000	0.0002	0.0984	0.0062	1.5683	-0.2121	2.7416	87.38	0.0363
18	8.9342	0.0000	0.0002	0.0984	0.0062	1.5683	-0.2122	2.7417	87.38	0.0363
19	8.934	0.0000	0.0002	0.0984	0.0062	1.5683	-0.2121	2.7417	87.38	0.0363

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.934	0.0002	0.0001	0.1231	-0.0093	1.6871	-0.3471	2.709	87.11	0.0361
1	6.8935	0.0001	-0.0006	0.1293	-0.0539	2.0358	-0.2634	2.4364	83.28	0.0355
2	6.8935	0.0007	-0.0004	0.1673	-0.0294	2.0934	-0.1923	2.4353	83.28	0.0355
3	6.8935	0.0006	-0.0005	0.1583	-0.0481	2.0472	-0.3693	2.4323	83.28	0.0354
4	6.8935	0.0006	-0.0005	0.1595	-0.0429	2.0814	-0.272	2.4338	83.28	0.0354
5	6.8935	0.0006	-0.0005	0.1594	-0.0446	2.0742	-0.3039	2.4341	83.28	0.0354
6	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0747	-0.2994	2.4339	83.28	0.0354
7	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0746	-0.2986	2.434	83.28	0.0354
8	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.3	2.434	83.28	0.0354
9	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2996	2.434	83.28	0.0354
10	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2998	2.434	83.28	0.0354
11	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2997	2.434	83.28	0.0354
12	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2997	2.434	83.28	0.0354
13	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2997	2.434	83.28	0.0354
14	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2997	2.434	83.28	0.0354
15	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2997	2.434	83.28	0.0354
16	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2997	2.434	83.28	0.0354
17	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2997	2.434	83.28	0.0354
18	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2997	2.434	83.28	0.0354
19	6.8935	0.0006	-0.0005	0.1593	-0.0443	2.0743	-0.2997	2.434	83.28	0.0354

Grid 4

B.4 Results generated by the MINGCV algorithm, using the average of 10 different random vectors t

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.1464	0.003	0.0067	0.3418	0.5938	1.178	-0.2295	3.0588	94.89	0.0343
1	9.6992	-0.0002	0.0002	0.0594	0.0127	1.308	-0.1319	2.8878	94.32	0.0328
2	10.693	0.0003	0.0011	0.0947	0.0848	1.1011	-0.2766	2.9608	95.53	0.0328
3	10.4285	0.0000	0.0007	0.0753	0.0465	1.1765	-0.2162	2.9364	95.23	0.0327
4	10.3827	0.0000	0.0006	0.0733	0.0415	1.1869	-0.2178	2.9329	95.18	0.0327
5	10.3808	0.0000	0.0006	0.0732	0.0413	1.1873	-0.2183	2.9328	95.17	0.0327
6	10.3805	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
7	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
8	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
9	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
10	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
11	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
12	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
13	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
14	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
15	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
16	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
17	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
18	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327
19	10.3806	0.0000	0.0006	0.0732	0.0412	1.1873	-0.2183	2.9328	95.17	0.0327

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.3806	0.0001	0.0005	0.1016	0.0277	1.559	-0.3529	2.8753	94.66	0.0324
1	10.2345	0.0000	0.0004	0.0978	0.0092	1.5545	-0.3927	2.8702	94.44	0.0325
2	10.1295	0.0000	0.0004	0.0986	0.0106	1.6097	-0.3723	2.8595	94.27	0.0325
3	10.1004	0.0000	0.0003	0.0975	0.0056	1.6168	-0.3814	2.8562	94.22	0.0325
4	10.121	0.0000	0.0003	0.0978	0.007	1.6093	-0.3922	2.8583	94.26	0.0325
5	10.1148	0.0000	0.0003	0.0978	0.0073	1.6121	-0.3839	2.8577	94.25	0.0325
6	10.1138	0.0000	0.0003	0.0977	0.0068	1.6124	-0.3857	2.8576	94.25	0.0325
7	10.115	0.0000	0.0003	0.0978	0.007	1.6119	-0.3865	2.8577	94.25	0.0325
8	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.3858	2.8577	94.25	0.0325
9	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325
10	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325
11	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325
12	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325
13	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325
14	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325
15	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325
16	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325
17	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325
18	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325
19	10.1147	0.0000	0.0003	0.0978	0.007	1.612	-0.386	2.8577	94.25	0.0325

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.1147	0.0000	0.0003	0.1084	-0.0014	1.7295	-0.4846	2.8453	94.06	0.0325
1	9.967	.0000	0.0003	0.1084	-0.0014	1.8129	-0.5152	2.8244	93.80	0.0324
2	9.8341	0.0000	0.0002	0.1119	-0.0079	1.879	-0.5291	2.8086	93.57	0.0324
3	9.8757	0.0000	0.0002	0.112	-0.0099	1.8484	-0.4831	2.8123	93.64	0.0324
4	9.873	-0.0000	0.0002	0.1111	-0.0078	1.8443	-0.4586	2.8123	93.64	0.0324
5	9.9041	-0.0000	0.0002	0.1102	-0.007	1.8272	-0.4393	2.8163	93.69	0.0324
6	9.9213	-0.0000	0.0002	0.1095	-0.0063	1.8178	-0.435	2.8187	93.73	0.0324
7	9.922	-0.0000	0.0002	0.1093	-0.0064	1.8166	-0.4419	2.8192	93.73	0.0324
8	9.9121	0.0000	0.0002	0.1095	-0.0072	1.8209	-0.4543	2.8183	93.71	0.0324
9	9.8976	0.0000	0.0002	0.1098	-0.0081	1.8279	-0.466	2.8167	93.68	0.0324
10	9.886	0.0000	0.0002	0.1101	-0.0089	1.8339	-0.4735	2.8154	93.66	0.0324
11	9.8793	0.0000	0.0002	0.1104	-0.0092	1.8376	-0.4764	2.8146	93.65	0.0324
12	9.8774	-0.0000	0.0002	0.1105	-0.0093	1.839	-0.4759	2.8144	93.65	0.0324
13	9.8784	-0.0000	0.0002	0.1105	-0.0091	1.839	-0.4739	2.8145	93.65	0.0324
14	9.8804	-0.0000	0.0002	0.1105	-0.0089	1.8383	-0.4718	2.8147	93.65	0.0324
15	9.8824	-0.0000	0.0002	0.1105	-0.0087	1.8376	-0.4703	2.8149	93.65	0.0324
16	9.8838	-0.0000	0.0002	0.1104	-0.0086	1.8372	-0.4694	2.8151	93.66	0.0324
17	9.8845	-0.0000	0.0002	0.1104	-0.0085	1.8369	-0.469	2.8152	93.66	0.0324
18	9.8847	0.0000	0.0002	0.1104	-0.0085	1.8369	-0.4689	2.8152	93.66	0.0324
19	9.8846	0.0000	0.0002	0.1104	-0.0085	1.837	-0.4689	2.8152	93.66	0.0324

Grid 4

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	9.8846	0.0000	0.0002	0.1119	-0.0099	1.8615	-0.4926	2.8136	93.61	0.0324
1	9.8837	-0.0000	0.0002	0.1129	-0.0107	1.8706	-0.5005	2.8124	93.60	0.0324
2	9.8555	0.0000	0.0002	0.1136	-0.0114	1.8899	-0.5077	2.8088	93.54	0.0324
3	9.8513	-0.0000	0.0002	0.1137	-0.0116	1.8957	-0.5127	2.8082	93.53	0.0324
4	9.858	-0.0000	0.0002	0.1136	-0.0115	1.8949	-0.5152	2.8089	93.54	0.0324
5	9.8691	-0.0000	0.0002	0.1135	-0.0113	1.891	-0.5154	2.8101	93.56	0.0324
6	9.8747	-0.0000	0.0002	0.1135	-0.0111	1.8891	-0.5146	2.8106	93.57	0.0324
7	9.8759	0.0000	0.0002	0.1135	-0.011	1.8894	-0.5138	2.8106	93.57	0.0324
8	9.8747	0.0000	0.0002	0.1136	-0.011	1.8907	-0.5133	2.8104	93.57	0.0324
9	9.8729	0.0000	0.0002	0.1137	-0.011	1.8922	-0.513	2.81	93.57	0.0324
10	9.8714	0.0000	0.0002	0.1137	-0.0111	1.8935	-0.5128	2.8098	93.57	0.0324
11	9.8704	0.0000	0.0002	0.1138	-0.0111	1.8944	-0.5127	2.8096	93.56	0.0324
12	9.8698	0.0000	0.0002	0.1138	-0.011	1.8951	-0.5126	2.8094	93.56	0.0324
13	9.8692	0.0000	0.0002	0.1138	-0.011	1.8956	-0.5124	2.8093	93.56	0.0324
14	9.8686	0.0000	0.0002	0.1139	-0.011	1.8961	-0.5122	2.8091	93.56	0.0324
15	9.868	-0.0000	0.0002	0.1139	-0.011	1.8966	-0.5119	2.809	93.56	0.0324
16	9.8674	0.0000	0.0002	0.1139	-0.0109	1.8969	-0.5116	2.8089	93.56	0.0324
17	9.8668	0.0000	0.0002	0.1139	-0.0109	1.8973	-0.5113	2.8088	93.56	0.0324
18	9.8662	0.0000	0.0002	0.1139	-0.0109	1.8976	-0.5111	2.8086	93.56	0.0324
19	9.8657	0.0000	0.0002	0.114	-0.0109	1.8979	-0.5108	2.8085	93.55	0.0324

Grid 3

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	9.8657	0.0000	0.0002	0.1142	-0.0111	1.9013	-0.514	2.8083	93.55	0.0324
1	9.8654	-0.0000	0.0002	0.1142	-0.0111	1.9025	-0.5152	2.8083	93.55	0.0324
2	9.8663	0.0000	0.0002	0.1142	-0.0112	1.9027	-0.5159	2.8083	93.55	0.0324
3	9.8662	-0.0000	0.0002	0.1143	-0.0112	1.9034	-0.5164	2.8083	93.54	0.0324
4	9.8664	0.0000	0.0002	0.1143	-0.0112	1.9039	-0.5169	2.8082	93.54	0.0324
5	9.8661	0.0000	0.0002	0.1144	-0.0112	1.9045	-0.5174	2.8082	93.54	0.0324
6	9.8655	0.0000	0.0002	0.1144	-0.0113	1.9052	-0.5177	2.8081	93.54	0.0324
7	9.865	0.0000	0.0002	0.1144	-0.0113	1.9058	-0.518	2.808	93.54	0.0324
8	9.8643	0.0000	0.0002	0.1145	-0.0113	1.9065	-0.5183	2.8079	93.54	0.0324
9	9.8638	0.0000	0.0002	0.1145	-0.0114	1.907	-0.5186	2.8078	93.54	0.0324
10	9.8632	0.0000	0.0002	0.1145	-0.0114	1.9075	-0.5188	2.8077	93.54	0.0324
11	9.8628	0.0000	0.0002	0.1146	-0.0114	1.908	-0.519	2.8076	93.53	0.0324
12	9.8623	0.0000	0.0002	0.1146	-0.0114	1.9084	-0.5193	2.8075	93.53	0.0324
13	9.8619	0.0000	0.0002	0.1146	-0.0114	1.9088	-0.5195	2.8075	93.53	0.0324
14	9.8616	0.0000	0.0002	0.1146	-0.0115	1.9092	-0.5196	2.8074	93.53	0.0324
15	9.8613	0.0000	0.0002	0.1146	-0.0115	1.9095	-0.5198	2.8073	93.53	0.0324
16	9.861	0.0000	0.0002	0.1146	-0.0115	1.9098	-0.52	2.8073	93.53	0.0324
17	9.8608	0.0000	0.0002	0.1147	-0.0115	1.9101	-0.5201	2.8073	93.53	0.0324
18	9.8607	0.0000	0.0002	0.1147	-0.0115	1.9103	-0.5203	2.8072	93.53	0.0324
19	9.8605	0.0000	0.0002	0.1147	-0.0115	1.9105	-0.5204	2.8072	93.53	0.0324

Grid 2

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	9.8605	0.0000	0.0002	0.1147	-0.0115	1.911	-0.5209	2.8071	93.53	0.0324
1	9.8604	0.0000	0.0002	0.1147	-0.0116	1.9112	-0.521	2.8071	93.53	0.0324
2	9.8603	0.0000	0.0002	0.1147	-0.0116	1.9113	-0.5211	2.8071	93.53	0.0324
3	9.8603	0.0000	0.0002	0.1147	-0.0116	1.9114	-0.5212	2.8071	93.53	0.0324
4	9.8603	-0.0000	0.0002	0.1147	-0.0116	1.9115	-0.5213	2.8071	93.53	0.0324
5	9.8604	-0.0000	0.0002	0.1147	-0.0116	1.9115	-0.5214	2.8071	93.53	0.0324
6	9.8604	-0.0000	0.0002	0.1147	-0.0116	1.9116	-0.5214	2.8071	93.53	0.0324
7	9.8604	-0.0000	0.0002	0.1147	-0.0116	1.9116	-0.5215	2.8071	93.53	0.0324
8	9.8605	-0.0000	0.0002	0.1148	-0.0116	1.9116	-0.5216	2.8071	93.53	0.0324
9	9.8605	-0.0000	0.0002	0.1148	-0.0116	1.9117	-0.5216	2.8071	93.53	0.0324
10	9.8605	-0.0000	0.0002	0.1148	-0.0116	1.9117	-0.5216	2.8071	93.53	0.0324
11	9.8605	-0.0000	0.0002	0.1148	-0.0116	1.9117	-0.5217	2.8071	93.53	0.0324
12	9.8605	-0.0000	0.0002	0.1148	-0.0116	1.9118	-0.5217	2.8071	93.53	0.0324
13	9.8605	-0.0000	0.0002	0.1148	-0.0116	1.9118	-0.5218	2.8071	93.53	0.0324
14	9.8605	-0.0000	0.0002	0.1148	-0.0116	1.9118	-0.5218	2.8071	93.53	0.0324
15	9.8606	-0.0000	0.0002	0.1148	-0.0116	1.9119	-0.5218	2.8071	93.53	0.0324
16	9.8606	-0.0000	0.0002	0.1148	-0.0116	1.9119	-0.5218	2.8071	93.53	0.0324
17	9.8606	-0.0000	0.0002	0.1148	-0.0116	1.9119	-0.5219	2.8071	93.53	0.0324
18	9.8606	-0.0000	0.0002	0.1148	-0.0116	1.9119	-0.5219	2.8071	93.53	0.0324
19	9.8606	-0.0000	0.0002	0.1148	-0.0116	1.912	-0.5219	2.8071	93.53	0.0324

Grid 1

B.5 Results generated by the MINGCV algorithm, applying a first order correction to the solution estimate

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.1464	0.0034	0.0065	0.3418	0.5938	0.4699	-0.0546	3.0588	96.19	0.0334
1	9.6187	0.0003	0.0001	0.059	0.0119	0.4953	-0.0065	2.8833	95.93	0.0316
2	7.0928	0.0000	0.0003	0.0218	0.0353	0.3503	0.1572	2.7784	94.82	0.0312
3	7.0426	0.0000	0.0001	0.0155	0.0172	0.3174	0.1219	2.7761	94.80	0.0312
4	7.3516	0.0000	0.0002	0.0226	0.0228	0.3578	0.1154	2.7822	94.91	0.0312
5	7.2503	-0.0000	0.0002	0.0203	0.0213	0.3462	0.1197	2.7801	94.87	0.0312
6	7.2527	0.0000	0.0002	0.0203	0.0214	0.3465	0.1223	2.7801	94.87	0.0312
7	7.2524	-0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
8	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
9	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
10	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
11	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
12	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
13	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
14	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
15	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
16	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
17	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
18	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312
19	7.2525	0.0000	0.0002	0.0203	0.0214	0.3465	0.1219	2.7801	94.87	0.0312

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	7.2525	-0.0007	0.0007	0.0684	0.0156	2.238	-0.6422	2.6053	91.60	0.0314
1	8.3281	-0.0000	0.0006	0.0887	0.019	1.6189	-0.549	2.6823	93.66	0.0309
2	8.4107	0.0000	0.0006	0.0918	0.0173	1.5285	-0.5868	2.6913	93.79	0.0309
3	8.3282	-0.0000	0.0006	0.0905	0.019	1.581	-0.6059	2.684	93.66	0.0309
4	8.3302	-0.0000	0.0006	0.0905	0.019	1.5806	-0.5969	2.6842	93.66	0.0309
5	8.3323	-0.0000	0.0006	0.0905	0.019	1.5792	-0.5968	2.6844	93.67	0.0309
6	8.3323	-0.0000	0.0006	0.0905	0.019	1.5793	-0.597	2.6844	93.67	0.0309
7	8.3324	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
8	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
9	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
10	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
11	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
12	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
13	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
14	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
15	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
16	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
17	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
18	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309
19	8.3323	0.0000	0.0006	0.0905	0.019	1.5792	-0.597	2.6844	93.67	0.0309

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.3323	0.0001	0.0006	0.1252	-0.0013	2.0787	-1.0172	2.6337	92.96	0.0308
1	8.1925	-0.0000	0.0006	0.1257	-0.0084	2.2755	-1.0471	2.6076	92.65	0.0307
2	8.2416	-0.0000	0.0005	0.1227	-0.0114	2.2216	-1.0074	2.6199	92.76	0.0308
3	8.3068	0.0000	0.0005	0.1247	-0.0092	2.1452	-1.016	2.628	92.90	0.0308
4	8.2321	0.0000	0.0006	0.1262	-0.0073	2.2225	-1.0615	2.6187	92.74	0.0308
5	8.2184	-0.0000	0.0006	0.1257	-0.0061	2.2383	-1.0461	2.6169	92.71	0.0308
6	8.2308	-0.0000	0.0006	0.1252	-0.0071	2.2239	-1.0377	2.6184	92.74	0.0308
7	8.2381	-0.0000	0.0006	0.1251	-0.0083	2.2168	-1.0397	2.6193	92.75	0.0308
8	8.24	0.0000	0.0006	0.1253	-0.0081	2.2149	-1.0394	2.6196	92.76	0.0308
9	8.2367	0.0000	0.0006	0.1254	-0.0076	2.2183	-1.0406	2.6192	92.75	0.0308
10	8.2351	-0.0000	0.0006	0.1253	-0.0074	2.2199	-1.0408	2.6189	92.74	0.0308
11	8.2356	-0.0000	0.0006	0.1253	-0.0076	2.2194	-1.0402	2.619	92.75	0.0308
12	8.2362	-0.0000	0.0006	0.1253	-0.0077	2.2187	-1.0402	2.6191	92.75	0.0308
13	8.2363	0.0000	0.0006	0.1253	-0.0077	2.2186	-1.0404	2.6191	92.75	0.0308
14	8.2361	0.0000	0.0006	0.1253	-0.0076	2.2188	-1.0404	2.6191	92.75	0.0308
15	8.2361	0.0000	0.0006	0.1253	-0.0076	2.2189	-1.0403	2.6191	92.75	0.0308
16	8.2361	0.0000	0.0006	0.1253	-0.0076	2.2189	-1.0403	2.6191	92.75	0.0308
17	8.2361	0.0000	0.0006	0.1253	-0.0076	2.2188	-1.0403	2.6191	92.75	0.0308
18	8.2361	0.0000	0.0006	0.1253	-0.0076	2.2188	-1.0403	2.6191	92.75	0.0308
19	8.2361	0.0000	0.0006	0.1253	-0.0076	2.2188	-1.0403	2.6191	92.75	0.0308

Grid 4

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.2361	-0.0000	0.0006	0.1317	-0.0131	2.363	-1.1763	2.6119	92.49	0.0308
1	8.2713	0.0000	0.0006	0.1348	-0.0147	2.3646	-1.1921	2.6122	92.56	0.0308
2	8.2441	-0.0000	0.0006	0.1358	-0.0145	2.4272	-1.2157	2.6072	92.49	0.0308
3	8.2648	-0.0000	0.0006	0.1351	-0.0136	2.4036	-1.202	2.6099	92.55	0.0308
4	8.2731	-0.0000	0.0006	0.1344	-0.0126	2.3893	-1.1889	2.6113	92.57	0.0308
5	8.2811	-0.0000	0.0006	0.134	-0.012	2.375	-1.1758	2.6126	92.59	0.0308
6	8.2826	-0.0000	0.0006	0.1338	-0.0118	2.3721	-1.1706	2.6127	92.59	0.0308
7	8.2842	-0.0000	0.0006	0.1337	-0.0117	2.3703	-1.1678	2.6128	92.59	0.0308
8	8.2851	-0.0000	0.0006	0.1337	-0.0117	2.3702	-1.1673	2.6127	92.60	0.0308
9	8.2867	-0.0000	0.0006	0.1336	-0.0117	2.3693	-1.1664	2.6128	92.60	0.0308
10	8.289	-0.0000	0.0006	0.1335	-0.0117	2.3676	-1.1648	2.613	92.61	0.0308
11	8.2918	-0.0000	0.0006	0.1333	-0.0116	2.365	-1.1621	2.6133	92.61	0.0308
12	8.2945	-0.0000	0.0006	0.1332	-0.0116	2.3621	-1.1592	2.6137	92.62	0.0308
13	8.2967	-0.0000	0.0006	0.1331	-0.0115	2.3595	-1.1566	2.6139	92.62	0.0308
14	8.2983	-0.0000	0.0006	0.133	-0.0116	2.3574	-1.1546	2.6141	92.63	0.0308
15	8.2994	-0.0000	0.0006	0.1329	-0.0116	2.3558	-1.1534	2.6143	92.63	0.0308
16	8.3003	-0.0000	0.0006	0.1329	-0.0117	2.3543	-1.1528	2.6144	92.63	0.0308
17	8.3011	-0.0000	0.0006	0.1328	-0.0118	2.3532	-1.1525	2.6145	92.63	0.0308
18	8.3017	-0.0000	0.0006	0.1328	-0.0119	2.3522	-1.1525	2.6145	92.64	0.0308
19	8.3022	-0.0000	0.0006	0.1328	-0.0119	2.3514	-1.1525	2.6146	92.64	0.0308

Grid 3

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.3022	0.0000	0.0006	0.1336	-0.0128	2.3644	-1.166	2.6138	92.61	0.0308
1	8.3008	-0.0000	0.0006	0.1338	-0.013	2.373	-1.1724	2.6133	92.59	0.0308
2	8.3031	-0.0000	0.0006	0.134	-0.0132	2.3748	-1.1765	2.6134	92.59	0.0308
3	8.3045	-0.0000	0.0006	0.1341	-0.0133	2.3768	-1.1796	2.6134	92.59	0.0308
4	8.3059	-0.0000	0.0006	0.1342	-0.0134	2.3782	-1.182	2.6134	92.59	0.0308
5	8.3062	-0.0000	0.0006	0.1344	-0.0135	2.3803	-1.184	2.6133	92.59	0.0308
6	8.3062	0.0000	0.0006	0.1345	-0.0136	2.3822	-1.1853	2.6132	92.59	0.0308
7	8.3057	0.0000	0.0006	0.1346	-0.0137	2.3843	-1.1863	2.613	92.59	0.0308
8	8.305	0.0000	0.0006	0.1347	-0.0137	2.3863	-1.1872	2.6128	92.58	0.0308
9	8.3044	0.0000	0.0006	0.1348	-0.0138	2.3881	-1.1881	2.6126	92.58	0.0308
10	8.3038	0.0000	0.0006	0.1349	-0.0139	2.3898	-1.189	2.6124	92.58	0.0308
11	8.3034	0.0000	0.0006	0.135	-0.0139	2.3913	-1.1898	2.6123	92.58	0.0308
12	8.3032	0.0000	0.0006	0.135	-0.014	2.3925	-1.1907	2.6122	92.58	0.0308
13	8.3031	-0.0000	0.0006	0.1351	-0.014	2.3936	-1.1915	2.6121	92.58	0.0308
14	8.3032	-0.0000	0.0006	0.1351	-0.014	2.3944	-1.1922	2.612	92.58	0.0308
15	8.3032	-0.0000	0.0006	0.1351	-0.014	2.3951	-1.1928	2.612	92.58	0.0308
16	8.3034	-0.0000	0.0006	0.1352	-0.0141	2.3957	-1.1934	2.6119	92.58	0.0308
17	8.3035	-0.0000	0.0006	0.1352	-0.0141	2.3962	-1.1938	2.6119	92.58	0.0308
18	8.3036	-0.0000	0.0006	0.1352	-0.0141	2.3966	-1.1942	2.6119	92.58	0.0308
19	8.3038	-0.0000	0.0006	0.1352	-0.0141	2.3969	-1.1944	2.6119	92.58	0.0308

Grid 2

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.3038	0.0000	0.0006	0.1354	-0.0142	2.3988	-1.1964	2.6118	92.57	0.0308
1	8.3038	0.0000	0.0006	0.1354	-0.0143	2.3998	-1.1972	2.6117	92.57	0.0308
2	8.3037	0.0000	0.0006	0.1354	-0.0143	2.4004	-1.1977	2.6117	92.57	0.0308
3	8.3037	0.0000	0.0006	0.1355	-0.0143	2.4007	-1.198	2.6116	92.57	0.0308
4	8.3036	0.0000	0.0006	0.1355	-0.0143	2.4011	-1.1982	2.6116	92.57	0.0308
5	8.3035	-0.0000	0.0006	0.1355	-0.0143	2.4015	-1.1985	2.6116	92.57	0.0308
6	8.3036	-0.0000	0.0006	0.1355	-0.0144	2.4017	-1.1987	2.6116	92.57	0.0308
7	8.3037	-0.0000	0.0006	0.1355	-0.0144	2.4018	-1.1989	2.6116	92.57	0.0308
8	8.3037	-0.0000	0.0006	0.1355	-0.0144	2.402	-1.1991	2.6116	92.57	0.0308
9	8.3038	-0.0000	0.0006	0.1355	-0.0144	2.4021	-1.1993	2.6116	92.57	0.0308
10	8.3038	-0.0000	0.0006	0.1355	-0.0144	2.4022	-1.1995	2.6116	92.57	0.0308
11	8.3039	-0.0000	0.0006	0.1355	-0.0144	2.4024	-1.1996	2.6116	92.57	0.0308
12	8.3039	-0.0000	0.0006	0.1356	-0.0144	2.4025	-1.1998	2.6116	92.57	0.0308
13	8.304	-0.0000	0.0006	0.1356	-0.0144	2.4026	-1.2	2.6116	92.57	0.0308
14	8.3041	-0.0000	0.0006	0.1356	-0.0144	2.4027	-1.2001	2.6116	92.57	0.0308
15	8.3041	-0.0000	0.0006	0.1356	-0.0144	2.4028	-1.2002	2.6116	92.57	0.0308
16	8.3042	-0.0000	0.0006	0.1356	-0.0144	2.4029	-1.2004	2.6116	92.57	0.0308
17	8.3042	-0.0000	0.0006	0.1356	-0.0144	2.403	-1.2005	2.6116	92.57	0.0308
18	8.3043	-0.0000	0.0006	0.1356	-0.0144	2.4031	-1.2006	2.6116	92.57	0.0308
19	8.3043	-0.0000	0.0006	0.1356	-0.0144	2.4031	-1.2007	2.6116	92.57	0.0308

Grid 1

B.6 Results of using the first order correction, for a second random vector t

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.1464	0.0026	0.0076	0.3418	0.5938	1.9649	-0.2636	3.0588	88.84	0.0391
1	9.7968	-0.001	0.0005	0.0603	0.0147	2.1305	-0.2456	2.8935	88.11	0.0376
2	12.0607	0.006	0.0107	0.6639	1.0602	2.1654	1.4992	3.3396	92.19	0.0397
3	11.4976	0.0016	0.0044	0.2438	0.3567	1.6834	-0.3015	3.0775	91.32	0.0373
4	11.1408	0.0004	0.0026	0.15	0.1933	1.7298	-0.298	3.0106	90.72	0.0369
5	10.9718	0.0001	0.0019	0.1218	0.1408	1.7812	-0.2894	2.9876	90.42	0.0369
6	10.9458	-0.0000	0.0019	0.1182	0.1335	1.7921	-0.298	2.9844	90.37	0.0369
7	10.9466	0.0000	0.0019	0.1184	0.1336	1.7918	-0.3009	2.9845	90.37	0.0369
8	10.9465	0.0000	0.0019	0.1184	0.1336	1.7918	-0.3009	2.9845	90.37	0.0369
9	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
10	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
11	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
12	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
13	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
14	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
15	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
16	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
17	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
18	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
19	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.9465	0.0026	0.0076	0.3418	0.5938	1.9649	-0.2636	3.0588	88.84	0.0391
1	9.7968	-0.001	0.0005	0.0603	0.0147	2.1305	-0.2456	2.8935	88.11	0.0376
2	12.0607	0.006	0.0107	0.6639	1.0602	2.1654	1.4992	3.3396	92.19	0.0397
3	11.4976	0.0016	0.0044	0.2438	0.3567	1.6834	-0.3015	3.0775	91.32	0.0373
4	11.1408	0.0004	0.0026	0.15	0.1933	1.7298	-0.298	3.0106	90.72	0.0369
5	10.9718	0.0001	0.0019	0.1218	0.1408	1.7812	-0.2894	2.9876	90.42	0.0369
6	10.9458	-0.0000	0.0019	0.1182	0.1335	1.7921	-0.298	2.9844	90.37	0.0369
7	10.9466	0.0000	0.0019	0.1184	0.1336	1.7918	-0.3009	2.9845	90.37	0.0369
8	10.9465	0.0000	0.0019	0.1184	0.1336	1.7918	-0.3009	2.9845	90.37	0.0369
9	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
10	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
11	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
12	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
13	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
14	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
15	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
16	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
17	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
18	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369
19	10.9465	0.0000	0.0019	0.1183	0.1335	1.7918	-0.3008	2.9845	90.37	0.0369

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.9465	-0.0000	0.0017	0.1404	0.1057	2.0912	-0.4515	2.9403	89.80	0.0368
1	10.898	00.0000	0.0016	0.1396	0.0994	2.1161	-0.4606	2.9342	89.72	0.0368
2	10.8882	-0.0000	0.0016	0.1389	0.0977	2.1276	-0.4635	2.9318	89.70	0.0368
3	10.8894	-0.0000	0.0016	0.1394	0.0978	2.1331	-0.4647	2.9311	89.71	0.0368
4	10.8897	0.0000	0.0016	0.1392	0.098	2.1363	-0.4632	2.931	89.71	0.0368
5	10.8925	-0.0000	0.0016	0.1393	0.0987	2.1366	-0.4599	2.9315	89.72	0.0368
6	10.8953	-0.0000	0.0016	0.1393	0.0994	2.1363	-0.4561	2.9321	89.73	0.0368
7	10.8979	-0.0000	0.0016	0.1393	0.0999	2.1352	-0.4519	2.9326	89.73	0.0368
8	10.8996	-0.0000	0.0016	0.1394	0.1002	2.134	-0.448	2.9329	89.74	0.0368
9	10.9004	-0.0000	0.0016	0.1394	0.1003	2.133	-0.4448	2.9331	89.74	0.0368
10	10.9008	-0.0000	0.0016	0.1394	0.1002	2.1322	-0.4424	2.9332	89.74	0.0368
11	10.9009	-0.0000	0.0016	0.1393	0.1001	2.1316	-0.4406	2.9332	89.74	0.0368
12	10.901	-0.0000	0.0016	0.1393	0.1	2.131	-0.4392	2.9332	89.74	0.0368
13	10.901	-0.0000	0.0016	0.1393	0.0999	2.1304	-0.4383	2.9331	89.74	0.0368
14	10.9011	-0.0000	0.0016	0.1392	0.0998	2.1297	-0.4377	2.9331	89.74	0.0368
15	10.9011	-0.0000	0.0016	0.1392	0.0997	2.129	-0.4373	2.9331	89.74	0.0368
16	10.9011	-0.0000	0.0016	0.1391	0.0996	2.1284	-0.4373	2.9331	89.74	0.0368
17	10.901	-0.0000	0.0016	0.1391	0.0995	2.1278	-0.4374	2.9331	89.74	0.0368
18	10.901	0.0000	0.0016	0.1391	0.0994	2.1272	-0.4377	2.9331	89.74	0.0368
19	10.9009	0.0000	0.0016	0.139	0.0993	2.1268	-0.4381	2.9331	89.74	0.0368

Grid 4

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.9009	0.0000	0.0016	0.1396	0.0987	2.1343	-0.4454	2.9325	89.73	0.0368
1	10.9	000	0.0016	0.1396	0.0987	2.1372	-0.4478	2.9319	89.72	0.0368
2	10.8982	0.0000	0.0016	0.1399	0.0978	2.1392	-0.4489	2.9315	89.71	0.0368
3	10.8974	0.0000	0.0016	0.1399	0.0976	2.1403	-0.4495	2.9313	89.71	0.0368
4	10.8973	-0.0000	0.0016	0.1399	0.0976	2.1412	-0.4503	2.9312	89.71	0.0368
5	10.8976	-0.0000	0.0016	0.1399	0.0976	2.1419	-0.451	2.9313	89.71	0.0368
6	10.8979	-0.0000	0.0016	0.14	0.0976	2.1426	-0.4516	2.9312	89.71	0.0368
7	10.898	0.0000	0.0016	0.1401	0.0976	2.1431	-0.4521	2.9312	89.70	0.0368
8	10.8979	0.0000	0.0016	0.1401	0.0975	2.1437	-0.4525	2.9311	89.70	0.0368
9	10.8978	0.0000	0.0016	0.1401	0.0975	2.1442	-0.4528	2.931	89.70	0.0368
10	10.8977	0.0000	0.0016	0.1402	0.0974	2.1447	-0.4531	2.931	89.70	0.0368
11	10.8976	0.0000	0.0016	0.1402	0.0974	2.1452	-0.4534	2.9309	89.70	0.0368
12	10.8975	0.0000	0.0016	0.1402	0.0974	2.1456	-0.4536	2.9309	89.70	0.0368
13	10.8975	0.0000	0.0016	0.1402	0.0973	2.146	-0.4539	2.9308	89.70	0.0368
14	10.8974	0.0000	0.0016	0.1403	0.0973	2.1464	-0.4541	2.9308	89.70	0.0368
15	10.8974	0.0000	0.0016	0.1403	0.0973	2.1467	-0.4543	2.9307	89.70	0.0368
16	10.8973	0.0000	0.0016	0.1403	0.0973	2.147	-0.4544	2.9307	89.70	0.0368
17	10.8973	0.0000	0.0016	0.1403	0.0972	2.1473	-0.4546	2.9306	89.70	0.0368
18	10.8973	0.0000	0.0016	0.1403	0.0972	2.1476	-0.4547	2.9306	89.70	0.0368
19	10.8972	0.0000	0.0016	0.1403	0.0972	2.1478	-0.4548	2.9306	89.70	0.0368

Grid 3

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.8932	0.0000	0.0016	0.1404	0.0971	2.1491	-0.4561	2.9305	89.70	0.0368
1	10.8973	0.0000	0.0016	0.1405	0.0971	2.1495	-0.4565	2.9305	89.69	0.0368
2	10.8973	0.0000	0.0016	0.1405	0.0971	2.1497	-0.4567	2.9305	89.69	0.0368
3	10.8973	0.0000	0.0016	0.1405	0.0971	2.1499	-0.4569	2.9305	89.69	0.0368
4	10.8973	0.0000	0.0016	0.1405	0.0971	2.15	-0.457	2.9304	89.69	0.0368
5	10.8972	0.0000	0.0016	0.1405	0.097	2.1502	-0.4571	2.9304	89.69	0.0368
6	10.8972	0.0000	0.0016	0.1405	0.097	2.1503	-0.4572	2.9304	89.69	0.0368
7	10.8971	0.0000	0.0016	0.1405	0.097	2.1504	-0.4573	2.9304	89.69	0.0368
8	10.897	0.0000	0.0016	0.1405	0.097	2.1506	-0.4573	2.9304	89.69	0.0368
9	10.897	0.0000	0.0016	0.1405	0.097	2.1507	-0.4574	2.9303	89.69	0.0368
10	10.8969	0.0000	0.0016	0.1405	0.097	2.1508	-0.4575	2.9303	89.69	0.0368
11	10.8969	0.0000	0.0016	0.1405	0.0969	2.1508	-0.4575	2.9303	89.69	0.0368
12	10.8969	0.0000	0.0015	0.1405	0.0969	2.1509	-0.4576	2.9303	89.69	0.0368
13	10.8968	0.0000	0.0015	0.1406	0.0969	2.151	-0.4576	2.9303	89.69	0.0368
14	10.8968	0.0000	0.0015	0.1406	0.0969	2.1511	-0.4577	2.9303	89.69	0.0368
15	10.8968	0.0000	0.0015	0.1406	0.0969	2.1512	-0.4577	2.9302	89.69	0.0368
16	10.8967	0.0000	0.0015	0.1406	0.0969	2.1512	-0.4578	2.9302	89.69	0.0368
17	10.8967	0.0000	0.0015	0.1406	0.0969	2.1513	-0.4578	2.9302	89.69	0.0368
18	10.8967	0.0000	0.0015	0.1406	0.0969	2.1514	-0.4578	2.9302	89.69	0.0368
19	10.8967	0.0000	0.0015	0.1406	0.0969	2.1514	-0.4579	2.9302	89.69	0.0368

Grid 2

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.8967	0.0000	0.0015	0.1406	0.0968	2.1516	-0.458	2.9302	89.69	0.0368
1	10.8966	0.0000	0.0015	0.1406	0.0968	2.1516	-0.4581	2.9302	89.69	0.0368
2	10.8966	0.0000	0.0015	0.1406	0.0968	2.1517	-0.4581	2.9302	89.69	0.0368
3	10.8966	0.0000	0.0015	0.1406	0.0968	2.1517	-0.4581	2.9302	89.69	0.0368
4	10.8966	0.0000	0.0015	0.1406	0.0968	2.1517	-0.4582	2.9302	89.69	0.0368
5	10.8966	0.0000	0.0015	0.1406	0.0968	2.1517	-0.4582	2.9302	89.69	0.0368
6	10.8966	0.0000	0.0015	0.1406	0.0968	2.1518	-0.4582	2.9302	89.69	0.0368
7	10.8966	0.0000	0.0015	0.1406	0.0968	2.1518	-0.4582	2.9302	89.69	0.0368
8	10.8966	0.0000	0.0015	0.1406	0.0968	2.1518	-0.4582	2.9302	89.69	0.0368
9	10.8966	0.0000	0.0015	0.1406	0.0968	2.1518	-0.4582	2.9302	89.69	0.0368
10	10.8966	0.0000	0.0015	0.1406	0.0968	2.1518	-0.4583	2.9302	89.69	0.0368
11	10.8966	0.0000	0.0015	0.1406	0.0968	2.1518	-0.4583	2.9302	89.69	0.0368
12	10.8966	0.0000	0.0015	0.1406	0.0968	2.1519	-0.4583	2.9302	89.69	0.0368
13	10.8966	0.0000	0.0015	0.1406	0.0968	2.1519	-0.4583	2.9302	89.69	0.0368
14	10.8966	0.0000	0.0015	0.1406	0.0968	2.1519	-0.4583	2.9302	89.69	0.0368
15	10.8966	0.0000	0.0015	0.1406	0.0968	2.1519	-0.4583	2.9302	89.69	0.0368
16	10.8966	0.0000	0.0015	0.1406	0.0968	2.1519	-0.4583	2.9302	89.69	0.0368
17	10.8966	0.0000	0.0015	0.1406	0.0968	2.1519	-0.4583	2.9302	89.69	0.0368
18	10.8966	0.0000	0.0015	0.1406	0.0968	2.1519	-0.4583	2.9302	89.69	0.0368
19	10.8966	0.0000	0.0015	0.1406	0.0968	2.1519	-0.4584	2.9302	89.69	0.0368

Grid 1

B.7 Results of using the first order correction, for a third random vector t.

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.1464	0.0035	0.0074	0.3418	0.5938	0.9694	-0.1281	3.0588	89.49	0.0386
1	9.6793	-0.0001	0.0003	0.0593	0.0124	1.0888	-0.1617	2.8867	89.00	0.0368
2	10.1793	0.0000	0.0004	0.0663	0.0248	0.9995	-0.1615	2.9188	89.52	0.0368
3	10.1481	0.0000	0.0004	0.0658	0.0239	1.0049	-0.1633	2.9167	89.49	0.0368
4	10.1403	0.0000	0.0004	0.0656	0.0235	1.0063	-0.1625	2.9162	89.49	0.0368
5	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
6	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
7	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
8	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
9	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
10	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
11	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
12	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
13	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
14	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
15	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
16	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
17	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
18	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368
19	10.1407	0.0000	0.0004	0.0656	0.0235	1.0062	-0.1626	2.9162	89.49	0.0368

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	10.1407	0.0001	0.0002	0.0963	0.0104	1.3645	-0.1334	2.8525	89.08	0.0363
1	9.6156	0.0001	0.0002	0.0984	-0.0008	1.367	-0.3481	2.8077	88.39	0.0363
2	9.0153	0.0000	0.0002	0.0998	0.0035	1.559	-0.2272	2.7506	87.51	0.0363
3	8.9008	-0.0000	0.0002	0.0984	0.0065	1.5725	-0.2074	2.7385	87.33	0.0363
4	8.9119	-0.0000	0.0002	0.0983	0.0068	1.5726	-0.2045	2.7395	87.34	0.0363
5	8.9331	-0.0000	0.0002	0.0984	0.0063	1.5685	-0.2103	2.7416	87.38	0.0363
6	8.9344	0.0000	0.0002	0.0984	0.0062	1.5682	-0.2125	2.7417	87.38	0.0363
7	8.9339	0.0000	0.0002	0.0984	0.0062	1.5683	-0.2122	2.7416	87.38	0.0363
8	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
9	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
10	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
11	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
12	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
13	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
14	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
15	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
16	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
17	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
18	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363
19	8.9338	0.0000	0.0002	0.0984	0.0062	1.5684	-0.2121	2.7416	87.38	0.0363

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	8.9338	0.0002	0.0001	0.1231	-0.0093	1.6871	-0.3471	2.709	87.11	0.0361
1	6.8944	0.0005	-0.0004	0.1454	-0.0454	1.8956	-0.4383	2.4396	83.28	0.0355
2	6.8944	0.0006	-0.0005	0.1615	-0.0386	2.1019	-0.182	2.4328	83.28	0.0354
3	6.8944	0.0006	-0.0005	0.1593	-0.0453	2.0604	-0.3396	2.4334	83.28	0.0354
4	6.8944	0.0005	-0.0005	0.1592	-0.0442	2.0783	-0.2803	2.4339	83.28	0.0354
5	6.8944	0.0006	-0.0005	0.1594	-0.0442	2.0738	-0.3048	2.4342	83.28	0.0354
6	6.8944	0.0006	-0.0005	0.1592	-0.0443	2.0744	-0.2984	2.4341	83.28	0.0354
7	6.8944	0.0006	-0.0005	0.1593	-0.0442	2.0742	-0.299	2.4341	83.28	0.0354
8	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2996	2.4341	83.28	0.0354
9	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2994	2.4341	83.28	0.0354
10	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2995	2.4341	83.28	0.0354
11	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2995	2.4341	83.28	0.0354
12	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2995	2.4341	83.28	0.0354
13	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2995	2.4341	83.28	0.0354
14	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2995	2.4341	83.28	0.0354
15	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2995	2.4341	83.28	0.0354
16	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2995	2.4341	83.28	0.0354
17	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2995	2.4341	83.28	0.0354
18	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2995	2.4341	83.28	0.0354
19	6.8944	0.0006	-0.0005	0.1593	-0.0443	2.074	-0.2995	2.4341	83.28	0.0354

Grid 4

B.8 Results generated by the MINGCV algorithm, for the data set bumpy.dat

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	6.6846	0.0471	0.0948	4.288	8.5061	0.249	0.1163	20.1473	94.66	0.2271
1	6.0539	0.0026	0.0096	0.3067	0.8877	0.1974	0.086	18.0753	94.55	0.2042
2	5.7828	0.0002	0.0004	0.0837	0.0716	0.1774	0.0832	17.9982	94.50	0.2036
3	5.3752	0.0001	0.0000	0.069	0.0166	0.1532	0.0488	17.9686	94.43	0.2035
4	5.3752	0.0001	-0.0001	0.066	0.0103	0.1515	0.0555	17.9684	94.43	0.2035
5	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0469	17.9682	94.43	0.2035
6	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
7	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
8	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
9	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
10	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
11	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
12	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
13	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
14	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
15	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
16	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
17	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
18	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035
19	5.3752	0.0001	-0.0001	0.0657	0.0094	0.1488	0.0463	17.9682	94.43	0.2035

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	5.3752	-0.003	0.0033	0.7793	0.3678	2.9599	0.3532	14.6721	86.64	0.1974
1	6.2884	0.001	0.0022	1.015	0.0923	2.6832	-0.3401	15.558	89.19	0.1975
2	5.8327	-0.0004	0.0037	0.9397	0.2545	2.8235	-0.1553	15.0828	87.93	0.197
3	5.9358	-0.0000	0.0033	0.9587	0.2126	2.794	-0.211	15.1817	88.22	0.197
4	5.9472	0.0000	0.0033	0.9631	0.2149	2.7943	-0.2105	15.1927	88.26	0.197
5	5.9429	0.0000	0.0034	0.9624	0.2167	2.7956	-0.2098	15.1885	88.24	0.197
6	5.9426	-0.0000	0.0033	0.9623	0.2165	2.7956	-0.2091	15.1882	88.24	0.197
7	5.9428	0.0000	0.0033	0.9623	0.2164	2.7955	-0.2094	15.1884	88.24	0.197
8	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
9	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
10	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
11	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
12	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
13	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
14	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
15	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
16	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
17	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
18	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
19	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	5.9428	-0.0035	0.0089	1.5324	0.0598	5.2973	-1.8202	14.1279	84.10	0.2017
1	6.3406	-0.0022	0.0074	1.52	0.0554	4.9781	-1.623	14.5482	86.20	0.1978
2	6.6358	-0.0002	0.0064	1.5514	-0.0014	4.5471	-1.6522	15.0593	87.58	0.1983
3	6.6612	0.0002	0.006	1.5191	-0.0643	4.3706	-1.7444	15.1236	87.69	0.1986
4	6.6362	-0.0002	0.0059	1.5033	-0.0639	4.4021	-1.7028	15.0696	87.58	0.1984
5	6.6621	-0.0000	0.0059	1.5104	-0.0475	4.3809	-1.667	15.1007	87.69	0.1983
6	6.6585	0.0000	0.0059	1.5095	-0.0514	4.3752	-1.6795	15.0979	87.68	0.1984
7	6.6525	-0.0000	0.0059	1.5079	-0.052	4.3848	-1.6782	15.0872	87.65	0.1983
8	6.656	-0.0000	0.0059	1.5091	-0.0498	4.3832	-1.674	15.092	87.67	0.1983
9	6.6561	0.0000	0.0059	1.5091	-0.0507	4.3816	-1.6757	15.0927	87.67	0.1983
10	6.6552	-0.0000	0.0059	1.5088	-0.0509	4.3827	-1.676	15.0913	87.66	0.1983
11	6.6556	-0.0000	0.0059	1.5089	-0.0506	4.3825	-1.6754	15.0918	87.66	0.1983
12	6.6557	0.0000	0.0059	1.5089	-0.0507	4.3823	-1.6756	15.092	87.66	0.1983
13	6.6555	-0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
14	6.6556	-0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
15	6.6556	-0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
16	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
17	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
18	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
19	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983

Grid 4

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	6.6556	-0.0017	0.008	1.5697	-0.0603	4.9147	-2.0978	14.9865	86.79	0.201
1	6.8693	-0.0009	0.007	1.5114	-0.0169	4.6086	-1.8104	15.0774	87.88	0.1972
2	6.9999	-0.0009	0.0064	1.4254	-0.0297	4.3351	-1.7021	15.2815	88.44	0.1973
3	7.1454	-0.0003	0.0053	1.3633	-0.0912	3.9741	-1.6172	15.5159	89.04	0.1977
4	7.1979	0.0002	0.0045	1.3548	-0.1734	3.8056	-1.6754	15.6965	89.24	0.1991
5	7.1529	0.0007	0.0042	1.4174	-0.2442	3.8562	-1.8226	15.7529	89.04	0.2007
6	6.9937	0.0009	0.0049	1.5485	-0.2644	4.1812	-2.0549	15.6328	88.35	0.2023
7	6.8127	0.0008	0.0065	1.6875	-0.1843	4.6359	-2.2133	15.3701	87.52	0.2027
8	6.6931	0.0002	0.0085	1.7497	-0.0129	5.0063	-2.1611	15.0918	86.97	0.2015
9	6.6739	-0.0006	0.0097	1.7074	0.1524	5.1078	-1.9291	14.9342	86.90	0.1997
10	6.7406	-0.0012	0.0096	1.6048	0.2196	4.94	-1.6983	14.9454	87.24	0.1983
11	6.8615	-0.0012	0.0086	1.4969	0.1781	4.6149	-1.5771	15.0911	87.82	0.1976
12	6.9974	-0.0008	0.0071	1.4154	0.0728	4.2656	-1.5606	15.3021	88.43	0.1977
13	7.1074	-0.0003	0.0058	1.3736	-0.0464	3.9945	-1.609	15.5084	88.88	0.1983
14	7.1522	0.0002	0.005	1.3791	-0.1484	3.8742	-1.7085	15.6496	89.05	0.1993
15	7.1073	0.0006	0.0047	1.4381	-0.2171	3.9493	-1.8638	15.6787	88.85	0.2006
16	6.9877	0.0007	0.0053	1.5405	-0.2334	4.2118	-2.0463	15.5757	88.33	0.2016
17	6.8524	0.0006	0.0064	1.6438	-0.1741	4.5619	-2.1597	15.3761	87.71	0.2019
18	6.7619	0.0002	0.0078	1.6932	-0.0514	4.8392	-2.1178	15.1709	87.30	0.201
19	6.7425	-0.0004	0.0087	1.67	0.0711	4.926	-1.9508	15.05	87.23	0.1998

Grid 3

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	6.7425	-0.0004	0.0087	1.6811	0.0605	4.96	-1.9773	15.0312	87.19	0.1997
1	6.8327	-0.0004	0.0087	1.6705	0.0689	4.9426	-1.9582	15.0369	87.30	0.1993
2	6.882	-0.0004	0.0086	1.6571	0.0774	4.9016	-1.9217	15.0486	87.45	0.1987
3	6.9278	-0.0003	0.0085	1.6434	0.0839	4.8476	-1.8782	15.0639	87.63	0.1981
4	6.9641	-0.0002	0.0083	1.6287	0.0896	4.7877	-1.8336	15.0824	87.79	0.1976
5	6.9898	-0.0001	0.0082	1.614	0.0939	4.7277	-1.7924	15.1025	87.94	0.1973
6	7.0057	-0.0001	0.008	1.6002	0.0965	4.6708	-1.7567	15.1229	88.05	0.197
7	7.0126	0.0000	0.0079	1.5873	0.0977	4.6205	-1.7289	15.1428	88.13	0.1969
8	7.0134	0.0000	0.0078	1.5756	0.0976	4.5784	-1.7099	15.162	88.18	0.1969
9	7.0114	0.0000	0.0078	1.5651	0.0963	4.5442	-1.6986	15.18	88.21	0.197
10	7.0096	0.0000	0.0078	1.5558	0.0939	4.5162	-1.6932	15.1969	88.22	0.1972
11	7.0097	0.0000	0.0077	1.5475	0.0907	4.4924	-1.6916	15.2129	88.23	0.1974
12	7.0124	0.0000	0.0077	1.5399	0.0869	4.4708	-1.6919	15.2284	88.25	0.1975
13	7.0176	-0.0001	0.0077	1.5329	0.0826	4.4498	-1.6927	15.2435	88.26	0.1976
14	7.0246	-0.0001	0.0076	1.5263	0.0779	4.4285	-1.6931	15.2584	88.29	0.1977
15	7.0321	-0.0001	0.0076	1.5201	0.0729	4.4067	-1.6932	15.2732	88.32	0.1978
16	7.0389	0.0000	0.0075	1.5145	0.0676	4.385	-1.6933	15.2876	88.35	0.1978
17	7.0443	0.0000	0.0074	1.5094	0.0619	4.3642	-1.694	15.3015	88.37	0.1979
18	7.0476	0.0000	0.0074	1.5049	0.0559	4.3452	-1.6961	15.3148	88.40	0.198
19	7.0488	0.0000	0.0073	1.5012	0.0496	4.3286	-1.7	15.3273	88.41	0.198

Grid 2

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	7.0488	0.0000	0.0073	1.503	0.0471	4.3343	-1.707	15.3261	88.40	0.1981
1	7.0479	0.0000	0.0073	1.5039	0.0455	4.3361	-1.7105	15.326	88.40	0.1981
2	7.0474	0.0000	0.0073	1.5043	0.0443	4.3367	-1.7128	15.3263	88.39	0.1981
3	7.0465	0.0000	0.0073	1.5047	0.0432	4.3367	-1.7145	15.3266	88.39	0.1981
4	7.0451	0.0000	0.0073	1.505	0.0423	4.3367	-1.7161	15.327	88.39	0.1981
5	7.0434	0.0000	0.0073	1.5052	0.0414	4.3369	-1.7178	15.3275	88.39	0.1982
6	7.0415	0.0000	0.0073	1.5054	0.0406	4.3371	-1.7196	15.328	88.39	0.1982
7	7.0396	0.0000	0.0073	1.5055	0.0398	4.3374	-1.7213	15.3285	88.38	0.1982
8	7.0379	0.0000	0.0073	1.5056	0.039	4.3376	-1.7231	15.329	88.38	0.1982
9	7.0362	0.0000	0.0073	1.5057	0.0382	4.3379	-1.7248	15.3295	88.38	0.1982
10	7.0346	0.0000	0.0073	1.5059	0.0375	4.3382	-1.7266	15.3301	88.37	0.1982
11	7.0332	0.0000	0.0073	1.506	0.0367	4.3386	-1.7283	15.3306	88.37	0.1983
12	7.032	0.0000	0.0073	1.5061	0.036	4.339	-1.7301	15.3311	88.37	0.1983
13	7.0311	0.0000	0.0073	1.5062	0.0353	4.3394	-1.7319	15.3316	88.37	0.1983
14	7.0304	0.0000	0.0073	1.5063	0.0346	4.3398	-1.7336	15.3321	88.36	0.1983
15	7.0299	0.0000	0.0073	1.5064	0.0339	4.3401	-1.7353	15.3326	88.36	0.1983
16	7.0296	0.0000	0.0073	1.5065	0.0332	4.3405	-1.737	15.3331	88.36	0.1984
17	7.0295	0.0000	0.0073	1.5065	0.0325	4.3408	-1.7386	15.3336	88.36	0.1984
18	7.0297	0.0000	0.0073	1.5066	0.0319	4.341	-1.7402	15.3341	88.35	0.1984
19	7.03	0.0000	0.0073	1.5067	0.0312	4.3413	-1.7417	15.3347	88.35	0.1984

Grid 1

B.9 Results using the first order correction for the data set bumpy.dat

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	6.6846	0.0471	0.0948	4.288	8.5061	0.249	0.1163	20.1473	94.66	0.2271
1	6.0539	0.0009	0.0034	0.1556	0.3387	0.198	0.0892	18.0355	94.55	0.2038
2	5.7904	0.0001	0.0004	0.0797	0.0626	0.1754	0.0803	17.9976	94.50	0.2035
3	5.3863	0.0001	-0.0001	0.0655	0.0097	0.1496	0.0461	17.9687	94.44	0.2035
4	5.3863	0.0001	-0.0001	0.0658	0.0105	0.1496	0.0469	17.969	94.44	0.2035
5	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
6	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
7	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
8	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
9	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
10	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
11	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
12	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
13	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
14	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
15	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
16	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
17	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
18	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035
19	5.3863	0.0001	-0.0001	0.0658	0.0104	0.1493	0.0473	17.969	94.44	0.2035

Grid 6

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	5.3863	-0.0029	0.0033	0.7847	0.3671	2.961	0.3437	14.6817	86.67	0.1974
1	6.2736	0.0009	0.0026	1.0125	0.1304	2.7082	-0.3192	15.5121	89.15	0.1971
2	5.9306	0.0000	0.0034	0.9591	0.2186	2.7981	-0.2094	15.1762	88.21	0.197
3	5.9451	0.0000	0.0033	0.9628	0.2159	2.795	-0.2088	15.1906	88.25	0.197
4	5.9426	0.0000	0.0033	0.9623	0.2165	2.7956	-0.2093	15.1883	88.24	0.197
5	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2095	15.1884	88.24	0.197
6	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
7	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
8	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
9	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
10	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
11	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
12	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
13	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
14	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
15	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
16	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
17	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
18	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197
19	5.9428	0.0000	0.0033	0.9623	0.2165	2.7955	-0.2094	15.1884	88.24	0.197

Grid 5

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	5.9428	-0.0035	0.0089	1.5324	0.0598	5.2973	-1.8202	14.1279	84.10	0.2017
1	6.3406	-0.0007	0.0064	1.5101	-0.0452	4.8316	-1.7141	14.6358	86.20	0.1989
2	6.5338	-0.0000	0.006	1.5071	-0.0588	4.5735	-1.7344	14.905	87.12	0.1984
3	6.6486	-0.0000	0.006	1.5094	-0.0487	4.3859	-1.7044	15.078	87.63	0.1983
4	6.6535	-0.0000	0.006	1.5091	-0.0492	4.3814	-1.6894	15.0877	87.65	0.1983
5	6.6512	-0.0000	0.0059	1.5089	-0.0507	4.389	-1.6779	15.0849	87.64	0.1983
6	6.655	-0.0000	0.0059	1.5089	-0.0508	4.3833	-1.676	15.091	87.66	0.1983
7	6.6556	-0.0000	0.0059	1.5089	-0.0506	4.3823	-1.6756	15.092	87.66	0.1983
8	6.6556	-0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
9	6.6556	-0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
10	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
11	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
12	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
13	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
14	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
15	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
16	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
17	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
18	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983
19	6.6556	0.0000	0.0059	1.5089	-0.0507	4.3824	-1.6756	15.0918	87.66	0.1983

Grid 4

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	6.6556	-0.0017	0.008	1.5697	-0.0603	4.9147	-2.0978	14.9865	86.79	0.201
1	6.8693	-0.0000	0.0071	1.5843	-0.0211	4.5661	-1.855	15.2505	87.82	0.1997
2	6.872	-0.0004	0.0074	1.5699	0.0093	4.6075	-1.8394	15.2275	87.84	0.1993
3	6.9205	-0.0001	0.0071	1.5519	0.0104	4.489	-1.7669	15.301	88.06	0.1993
4	6.9353	-0.0000	0.007	1.5386	-0.0082	4.4319	-1.7824	15.3344	88.12	0.1994
5	6.9424	0.0000	0.0067	1.5351	-0.0339	4.3995	-1.8079	15.3555	88.15	0.1996
6	6.9377	0.0000	0.0067	1.5382	-0.053	4.404	-1.8448	15.3543	88.13	0.1996
7	6.9305	0.0001	0.0067	1.5429	-0.0609	4.4177	-1.8659	15.3452	88.10	0.1997
8	6.9226	0.0000	0.0067	1.5466	-0.0597	4.434	-1.8806	15.3324	88.07	0.1997
9	6.9174	0.0000	0.0068	1.5483	-0.0542	4.4449	-1.8855	15.3233	88.04	0.1997
10	6.9154	0.0000	0.0069	1.5487	-0.0488	4.4495	-1.8845	15.3196	88.03	0.1996
11	6.9152	0.0000	0.0069	1.5488	-0.0449	4.4502	-1.8802	15.3194	88.03	0.1996
12	6.9152	0.0000	0.0069	1.5491	-0.0425	4.4505	-1.8757	15.3196	88.03	0.1996
13	6.915	0.0000	0.0069	1.5494	-0.0409	4.4511	-1.8722	15.3194	88.03	0.1997
14	6.9146	0.0000	0.0069	1.5496	-0.0398	4.452	-1.8699	15.3187	88.03	0.1997
15	6.9143	0.0000	0.0069	1.5497	-0.039	4.4527	-1.8683	15.3182	88.03	0.1997
16	6.9142	0.0000	0.0069	1.5497	-0.0386	4.4529	-1.8669	15.3181	88.03	0.1996
17	6.9143	-0.0000	0.0069	1.5496	-0.0385	4.4528	-1.8659	15.3182	88.03	0.1996
18	6.9144	-0.0000	0.0069	1.5496	-0.0385	4.4527	-1.8652	15.3184	88.03	0.1996
19	6.9145	-0.0000	0.0069	1.5496	-0.0386	4.4526	-1.8649	15.3185	88.03	0.1996

Grid 3

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	6.9145	0.0000	0.0069	1.5683	-0.0558	4.5046	-1.9144	15.2991	87.91	0.1999
1	6.9138	-0.0001	0.007	1.5704	-0.0563	4.5286	-1.9343	15.2945	87.88	0.2
2	6.925	-0.0000	0.0071	1.5711	-0.0554	4.5212	-1.9428	15.3095	87.91	0.2001
3	6.9317	-0.0000	0.0071	1.5729	-0.0555	4.5189	-1.9467	15.3166	87.93	0.2001
4	6.9342	0.0000	0.0071	1.5746	-0.0555	4.5225	-1.9488	15.3177	87.94	0.2001
5	6.936	-0.0000	0.0071	1.5759	-0.0553	4.5256	-1.9495	15.3179	87.94	0.2
6	6.9372	0.0000	0.0071	1.5772	-0.0551	4.5273	-1.9484	15.3173	87.95	0.2
7	6.9368	0.0000	0.0071	1.5784	-0.0548	4.5308	-1.9469	15.3143	87.95	0.2
8	6.9359	0.0000	0.0071	1.5793	-0.0544	4.5345	-1.9457	15.311	87.94	0.2
9	6.9352	0.0000	0.0071	1.58	-0.0539	4.5375	-1.9448	15.3082	87.94	0.1999
10	6.9348	0.0000	0.0071	1.5806	-0.0533	4.54	-1.9444	15.3059	87.94	0.1999
11	6.9344	0.0000	0.0071	1.5811	-0.0528	4.5422	-1.9442	15.3038	87.94	0.1999
12	6.9343	0.0000	0.0071	1.5814	-0.0522	4.5441	-1.9441	15.3023	87.94	0.1999
13	6.9344	-0.0000	0.0071	1.5817	-0.0517	4.5455	-1.944	15.3014	87.94	0.1999
14	6.9347	-0.0000	0.0071	1.5819	-0.051	4.5463	-1.9437	15.3008	87.94	0.1998
15	6.9351	-0.0000	0.0071	1.582	-0.0504	4.5469	-1.9431	15.3004	87.94	0.1998
16	6.9354	-0.0000	0.0071	1.5821	-0.0499	4.5472	-1.9424	15.3	87.94	0.1998
17	6.9358	-0.0000	0.0071	1.5821	-0.0493	4.5474	-1.9415	15.2998	87.94	0.1998
18	6.9361	-0.0000	0.0071	1.5821	-0.0487	4.5474	-1.9406	15.2995	87.94	0.1998
19	6.9364	-0.0000	0.0071	1.582	-0.0482	4.5474	-1.9397	15.2994	87.94	0.1998

Grid 2

q	θ_q	$\frac{dGCV}{d\theta}$	$\frac{d^2GCV}{d\theta^2}$	$\frac{dR}{d\theta}$	$\frac{d^2R}{d\theta^2}$	$\frac{dTr}{d\theta}$	$\frac{d^2Tr}{d\theta^2}$	R	Tr	GCV
0	6.9364	-0.0000	0.0071	1.5842	-0.0502	4.5542	-1.9465	15.2975	87.93	0.1998
1	6.9374	-0.0000	0.0071	1.5852	-0.0512	4.5559	-1.9495	15.2976	87.93	0.1998
2	6.9374	0.0000	0.0071	1.5859	-0.0518	4.5576	-1.951	15.2969	87.93	0.1998
3	6.9373	0.0000	0.0071	1.5864	-0.0522	4.5589	-1.9518	15.2961	87.92	0.1998
4	6.937	0.0000	0.0071	1.5868	-0.0525	4.5604	-1.9525	15.2951	87.92	0.1998
5	6.9368	-0.0000	0.0071	1.5871	-0.0527	4.5616	-1.9533	15.2945	87.92	0.1998
6	6.9369	-0.0000	0.0071	1.5873	-0.0529	4.5624	-1.954	15.2943	87.92	0.1999
7	6.9371	-0.0000	0.0071	1.5874	-0.0529	4.5628	-1.9546	15.2944	87.92	0.1999
8	6.9373	-0.0000	0.0071	1.5875	-0.053	4.563	-1.955	15.2946	87.92	0.1999
9	6.9376	-0.0000	0.0071	1.5876	-0.053	4.5631	-1.9555	15.2948	87.92	0.1999
10	6.9378	-0.0000	0.0071	1.5876	-0.053	4.5632	-1.9558	15.2951	87.92	0.1999
11	6.9381	-0.0000	0.0071	1.5876	-0.0529	4.5633	-1.9562	15.2953	87.92	0.1999
12	6.9383	-0.0000	0.0071	1.5877	-0.0529	4.5633	-1.9565	15.2957	87.92	0.1999
13	6.9386	-0.0000	0.0071	1.5877	-0.0529	4.5633	-1.9568	15.296	87.92	0.1999
14	6.9389	-0.0000	0.0071	1.5877	-0.0528	4.5632	-1.9571	15.2963	87.92	0.1999
15	6.9391	-0.0000	0.0071	1.5877	-0.0528	4.5632	-1.9573	15.2966	87.92	0.1999
16	6.9394	-0.0000	0.0071	1.5878	-0.0527	4.5632	-1.9575	15.2969	87.92	0.1999
17	6.9396	-0.0000	0.0071	1.5878	-0.0527	4.5632	-1.9577	15.2971	87.92	0.1999
18	6.9398	-0.0000	0.0071	1.5878	-0.0527	4.5632	-1.9579	15.2973	87.92	0.1999
19	6.94	-0.0000	0.0071	1.5878	-0.0526	4.5632	-1.9581	15.2975	87.92	0.1999

Grid 1

B.10 Differentiating Tr with respect to θ and λ

θ	$dGCV/d\theta$	$d^2GCV/d\theta^2$	$dR/d\theta$	$d^2R/d\theta^2$
10.936	0.0007	0.0027	0.1791	0.2054
10.8534	0.0001	0.0017	0.1361	0.1181
10.8471	0.0000	0.0015	0.1272	0.0983
10.847	0.0000	0.0015	0.1266	0.0969
10.847	0.0000	0.0015	0.1266	0.0969
10.847	0.0000	0.0015	0.1266	0.0969
10.847	0.0000	0.0015	0.1266	0.0969
10.847	0.0000	0.0015	0.1266	0.0969
$dTr/d\theta$	$d^2Tr/d\theta$	R	Tr	GCV
1.795	-0.3764	3.0002	90.79	0.0368
1.9	-0.3967	2.958	90.29	0.0366
1.9329	-0.401	2.9471	90.13	0.0366
1.9355	-0.4013	2.9463	90.12	0.0366
1.9355	-0.4013	2.9463	90.12	0.0366
1.9355	-0.4013	2.9463	90.12	0.0366
1.9355	-0.4013	2.9463	90.12	0.0366
1.9355	-0.4013	2.9463	90.12	0.0366

Table B.1: Results generated by the MINGCV algorithm, differentiating Tr with respect to θ, for grid no. 6

θ	$dGCV/d\theta$	$d^2GCV/d\theta^2$	$dR/d\theta$	$d^2R/d\theta^2$
10.936	0.0007	0.0027	0.1791	0.2054
10.8534	0.0001	0.0017	0.1361	0.1181
10.8471	0.0000	0.0015	0.1272	0.0983
10.847	0.0000	0.0015	0.1266	0.0969
10.847	0.0000	0.0015	0.1266	0.0969
10.847	0.0000	0.0015	0.1266	0.0969
10.847	0.0000	0.0015	0.1266	0.0969
10.847	0.0000	0.0015	0.1266	0.0969
$dTr/d\theta$	$d^2Tr/d\theta$	R	Tr	GCV
1.795	-0.3764	51.4591	101.00	0.5095
1.9	-0.3967	2.8876	89.02	0.0368
1.9329	-0.401	3.0002	90.79	0.0368
1.9355	-0.4013	2.958	90.29	0.0366
1.9355	-0.4013	2.9471	90.13	0.0366
1.9355	-0.4013	2.9463	90.12	0.0366
1.9355	-0.4013	2.9463	90.12	0.0366
1.9355	-0.4013	2.9463	90.12	0.0366

Table B.2: Results generated by the MINGCV algorithm, differentiating Tr with respect to λ, for grid no. 6

B.11 Results generated by the MINGCV algorithm for the data set sine.dat, with finite difference calculation of $d^2GCV/d\theta^2$ and the convergence criteria emplaced.

No. of θ updates	θ	$dGCV/d\theta$ $\times 10^3$	Finite difference $d^2GCV/d\theta^2$ $\times 10^3$	$d^2GCV/d\theta^2$ $\times 10^3$	GCV
0	10.309	1.8073		0.1753	0.0323
1	9.809	0.6021	2.4104	2.4104	0.0313
2	9.5592	0.5679	0.1367	0.1367	0.0312
3	5.4049	-0.4968	0.2563	0.2563	0.0315
4	7.3433	-0.2027	0.1517	0.1517	0.0305
5	8.6795	0.4308	0.4741	0.4741	0.0307
6	7.7709	0.0312	0.4398	0.4398	0.0305
7	7.7	-0.0081	0.4398	0.5543	0.0305
8	7.7185	0.0019	0.4398	0.5447	0.0305
9	7.7141	-0.0004	0.4398	0.5372	0.0305
10	7.715	0.0001	0.4398	0.5435	0.0305

Grid 6

No. of θ updates	θ	$dGCV/d\theta$ $\times 10^3$	Finite difference $d^2GCV/d\theta^2$ $\times 10^3$	$d^2GCV/d\theta^2$ $\times 10^3$	GCV
0	7.7148	-0.3427		-0.444	0.03
1	8.2148	-0.0656	0.5542	0.5542	0.0309
2	8.3332	0.0158	0.6878	0.687	0.0308
3	8.3102	0.0061	0.6878	0.4224	0.0309
4	8.3013	0.0009	0.6878	0.5919	0.0309
5	8.3001	-0.0001	0.6878	0.7602	0.0309
6	8.3002	-0.0001	0.6878	0.2096	0.0309

Grid 5

B.12 Results generated by the modified MINGCV algorithm for the data set bumpy.dat, with the convergence criteria emplaced.

No. of θ updates	θ	$dGCV/d\theta$ $\times 10^3$	Finite difference $d^2GCV/d\theta^2$ $\times 10^3$	$d^2GCV/d\theta^2$ $\times 10^3$	GCV
0	10.309	50.4926		4.8979	0.2803
1	9.809	35.7445	29.4963	35.4624	0.2577
2	8.5971	14.8687	17.2267	20.1801	0.2274
3	7.734	5.4827	10.8745	13.7326	0.2191
4	7.2298	2.6509	5.6166	6.9181	0.2171
5	6.7578	1.0937	3.2993	3.9367	0.2162
6	6.4263	0.2964	2.4051	2.5639	0.2159
7	6.3031	0.0266	2.1891	2.19	0.2159
8	6.2909	-0.0009	2.1891	2.1377	0.2159
9	6.2914	-0.0007	2.1891	2.1551	0.2159
10	6.2917	-0.0002	2.1891	2.1613	0.2159
11	6.2918	-0.0002	2.1891	2.1613	0.2159

Grid 6

No. of θ updates	θ	$dGCV/d\theta$ $\times 10^3$	Finite difference $d^2GCV/d\theta^2$ $\times 10^3$	$d^2GCV/d\theta^2$ $\times 10^3$	GCV
0	6.2918	1.8887		0.3002	0.1981
1	5.7918	-0.2116	4.2006	3.5893	0.198
2	5.8422	-0.0845	4.2006	3.2069	0.198
3	5.8623	-0.0181	4.2006	3.0625	0.198
4	5.8666	-0.0035	4.2006	3.0696	0.198
5	5.8674	-0.0007	4.2006	3.0576	0.198

Grid 5

No. of θ updates	θ	$dGCV/d\theta$ $\times 10^3$	Finite difference $d^2GCV/d\theta^2$ $\times 10^3$	$d^2GCV/d\theta^2$ $\times 10^3$	GCV
0	5.8676	-9.3486		-1.5933	0.2063
1	6.3676	-3.3763	11.9446	12.039	0.2016
2	6.6503	-1.5679	6.3978	9.3063	0.2004
3	6.8953	-0.2309	5.4555	7.324	0.2001
4	6.9376	-0.0616	5.4555	6.6035	0.2002
5	6.9489	0.1668	5.4555	6.6313	0.2002
6	6.9184	-0.0068	5.4555	6.6077	0.2002
7	6.9196	0.0439	5.4555	6.6578	0.2002
8	6.9116	-0.0361	5.4555	6.6817	0.2002
9	6.9182	0.0177	5.4555	6.6771	0.2002
10	6.9149	-0.0172	5.4555	6.6674	0.2002

Grid 4

Appendix C

Results for Chapter 12

C.1 Results generated by the bivariate MINGCV algorithm, for the data set 121.dat

The tables in this appendix are discussed in Chapter 12. They report updates of various quantities relevant to the OPTRSS and MINGCV algorithms described in Chapter 8. The notation is explained as follows:

θ: the logarithm of the smoothing parameter.

R: the estimate of the residual sum of squares.

Tr: the estimate of $tr(I - A)$, where A is the influence matrix.

GCV: the estimate of the generalised cross validation.

Sobolev Norm: as discussed in Chapter 12, the Sobolev Norm is the square root of the curvature, $\sqrt{\alpha^T Z \alpha}$.

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-1.94	9.67E-02	0.141	0.0000	101	0.26	12.8	14.8
1	-2.44	8.32E-02	2.18E-02	0.239	98.7	0.235	2.49	14.7
2	-2.53	8.22E-02	1.39E-02	0.239	98.1	0.232	1.84	14.7
3	-2.59	8.11E-02	1.24E-02	0.239	97.8	0.23	1.71	14.8
4	-2.64	8.06E-02	1.18E-02	1.07E-02	97.5	0.229	1.65	14.8
5	-3.14	7.63E-02	6.67E-03	1.03E-02	94.4	0.216	1.2	15.2
6	-3.64	7.35E-02	4.10E-03	5.14E-03	91.5	0.205	0.943	15.7
7	-4.14	7.21E-02	1.80E-03	4.59E-03	88.8	0.197	0.709	16.3
8	-4.53	7.18E-02	2.60E-04	3.93E-03	86.8	0.192	0.546	16.9
9	-4.6	7.18E-02	2.60E-04	3.93E-03	86.8	0.192	0.546	16.9
10	-4.59	7.18E-02	-3.22E-05	3.93E-03	86.5	0.192	0.516	17
11	-4.52	7.18E-02	-2.81E-04	3.93E-03	86.7	0.192	0.494	16.9
12	-4.47	7.17E-02	-1.73E-04	3.93E-03	87.1	0.193	0.506	16.8
13	-4.5	7.16E-02	8.23E-05	3.93E-03	87.3	0.193	0.527	16.8

Grid 6

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-4.5	7.60E-02	-8.48E-03	3.93E-03	71	0.162	0.785	17.2
1	-4.02	7.32E-02	-3.64E-03	9.69E-03	77.1	0.172	0.937	16.3
2	-3.64	7.26E-02	-5.83E-04	8.14E-03	81.4	0.181	1.07	15.8
3	-3.57	7.26E-02	4.56E-05	8.14E-03	82.2	0.183	1.1	15.7
4	-3.58	7.26E-02	5.42E-05	8.14E-03	82.1	0.183	1.1	15.7
5	-3.58	7.26E-02	-6.51E-06	8.14E-03	82	0.183	1.1	15.7

Grid 5

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-3.58	7.28E-02	-1.15E-04	8.14E-03	77.3	0.172	1.31	15.8
1	-3.57	7.27E-02	-6.01E-06	8.14E-03	77.6	0.173	1.28	15.8
2	-3.57	7.27E-02	-1.02E-05	8.14E-03	77.6	0.173	1.28	15.8
3	-3.57	7.27E-02	5.53E-05	8.14E-03	77.6	0.173	1.29	15.8
4	-3.57	7.28E-02	-2.98E-05	8.14E-03	77.5	0.173	1.28	15.8
5	-3.57	7.27E-02	5.83E-06	8.14E-03	77.6	0.173	1.28	15.8

Grid 4

C.2 Results generated by the bivariate MINGCV algorithm, for the data set franke1.dat

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-5.52	0.222	-1.29E-02	0	81.9	0.385	0.904	14.3
1	-5.02	0.216	-1.13E-02	3.23E-03	84.2	0.391	0.764	10.7
2	-4.52	0.211	-7.09E-03	8.38E-03	86.3	0.397	0.843	8.11
3	-4.02	0.208	-6.19E-03	1.79E-03	88.1	0.401	0.762	6.1
4	-3.52	0.205	-5.96E-03	4.61E-04	89.6	0.406	0.666	4.62
5	-3.02	0.202	-5.19E-03	1.55E-03	91.1	0.41	0.592	3.59
6	-2.52	0.2	-4.16E-03	2.06E-03	92.5	0.413	0.542	2.88
7	-2.02	0.198	-2.74E-03	2.84E-03	93.7	0.417	0.557	2.33
8	-1.52	0.197	-5.35E-04	4.40E-03	94.7	0.42	0.662	1.89
9	-1.4	0.197	3.24E-04	7.07E-03	94.9	0.422	0.723	1.78
10	-1.45	0.197	2.68E-05	7.07E-03	94.8	0.421	0.723	1.81
11	-1.45	0.197	-3.21E-05	7.07E-03	94.8	0.421	0.723	1.81

Grid 6

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-1.45	0.197	5.11E-05	7.07E-03	94.7	0.421	0.756	1.83
1	-1.45	0.197	-1.41E-05	7.07E-03	94.6	0.421	0.752	1.84
2	-1.45	0.197	-5.21E-05	7.07E-03	94.6	0.421	0.754	1.84
3	-1.44	0.197	-3.43E-05	7.07E-03	94.6	0.421	0.759	1.83
4	-1.44	0.197	-1.31E-05	7.07E-03	94.6	0.421	0.762	1.83
5	-1.44	0.197	4.82E-08	7.07E-03	94.7	0.421	0.762	1.83

Grid 5

C.3 Results generated by the bivariate MINGCV algorithm, for the data set franke2.dat

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-5.52	6.63E-03	1.95E-03	0	81.9	6.67E-02	0.184	8.1
1	-6.02	5.50E-03	1.72E-03	4.68E-04	79.7	5.91E-02	0.153	9.66
2	-6.52	4.84E-03	8.90E-04	1.66E-03	76.9	5.35E-02	9.33E-02	11.2
3	-7.02	4.58E-03	4.28E-04	9.25E-04	74.1	5.01E-02	6.26E-02	12.5
4	-7.48	4.47E-03	4.72E-05	8.23E-04	71.5	4.78E-02	3.95E-02	13.7
5	-7.54	4.45E-03	2.58E-05	8.23E-04	71.3	4.75E-02	3.69E-02	13.9
6	-7.57	4.45E-03	1.01E-05	8.23E-04	71.1	4.74E-02	3.57E-02	13.9
7	-7.58	4.45E-03	-2.74E-07	8.23E-04	71	4.74E-02	3.53E-02	14

Grid 6

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-7.58	0.451E-02	-0.344E-03	8.23E-04	62.2	4.18E-02	4.35E-02	13.7
1	-7.17	0.446E-02	-0.883E-04	6.12E-04	66	4.41E-02	5.28E-02	12.5
2	-7.02	0.445E-02	0.243E-04	7.80E-04	67.4	4.50E-02	5.71E-02	12.1
3	-7.05	0.445E-02	0.652E-05	7.80E-04	67.2	4.48E-02	5.61E-02	12.2
4	-7.06	0.445E-02	-0.402E-06	7.80E-04	67.1	4.47E-02	5.58E-02	12.2
5	-7.06	0.445E-02	0.831E-07	7.80E-04	67.1	4.47E-02	5.59E-02	12.2

Grid 5

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-7.06	0.453E-02	-0.107E-03	7.80E-04	63.3	4.26E-02	6.38E-02	12.3
1	-6.92	0.452E-02	0.146E-05	7.91E-04	64.8	4.36E-02	6.60E-02	11.9
2	-6.92	0.451E-02	0.157E-04	7.91E-04	64.9	4.36E-02	6.59E-02	11.9
3	-6.94	0.451E-02	0.628E-06	7.91E-04	64.7	4.34E-02	6.54E-02	12
4	-6.95	0.451E-02	0.999E-06	7.91E-04	64.7	4.34E-02	6.55E-02	12
5	-6.95	0.451E-02	-0.157E-05	7.91E-04	64.7	4.35E-02	6.54E-02	12

Grid 4

C.4 Results generated by the bivariate MINGCV algorithm, for the data set franke3.dat

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-5.52	3.65E-03	2.13E-03	0	81.9	4.95E-02	0.172	8.13
1	-6.02	2.42E-03	1.92E-03	4.07E-04	79.7	3.92E-02	0.141	9.61
2	-6.52	1.64E-03	1.21E-03	1.43E-03	76.9	3.11E-02	8.51E-02	11
3	-7.02	1.23E-03	7.11E-04	9.90E-04	74.1	2.60E-02	4.95E-02	12.1
4	-7.52	9.64E-04	3.76E-04	6.69E-04	71.2	2.21E-02	2.71E-02	13.2
5	-8.02	8.14E-04	2.00E-04	3.53E-04	68.5	1.95E-02	1.56E-02	14.1
6	-8.52	7.45E-04	8.81E-05	2.23E-04	65.9	1.80E-02	8.91E-03	14.9
7	-8.92	7.28E-04	2.95E-05	1.49E-04	64	1.73E-02	5.64E-03	15.4
8	-9.11	7.27E-04	7.63E-06	1.10E-04	63.1	1.70E-02	4.50E-03	15.6
9	-9.18	7.27E-04	4.54E-07	1.10E-04	62.7	1.69E-02	4.15E-03	15.7
10	-9.19	7.28E-04	2.95E-05	1.49E-04	64	1.73E-02	5.64E-03	15.4

Grid 6

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-9.19	0.170E-03	0.444E-04	1.10E-04	44.7	5.82E-03	2.57E-03	14
1	-9.59	0.152E-03	0.172E-04	6.75E-05	41.2	5.07E-03	1.55E-03	14.3
2	-9.85	0.154E-03	-0.322E-05	8.01E-05	38.5	4.78E-03	1.11E-03	14.5
3	-9.81	0.151E-03	0.329E-05	8.01E-05	39.1	4.80E-03	1.18E-03	14.5
4	-9.85	0.152E-03	-0.448E-06	8.01E-05	38.6	4.76E-03	1.12E-03	14.5
5	-9.84	0.152E-03	0.144E-06	8.01E-05	38.7	4.76E-03	1.13E-03	14.5

Grid 5

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-9.84	1.79E-04	-4.15E-05	8.01E-05	27.8	3.73E-03	8.50E-04	14
1	-9.34	1.71E-04	8.74E-06	1.00E-04	33.9	4.42E-03	1.56E-03	13.7
2	-9.43	1.71E-04	-8.18E-07	1.00E-04	32.7	4.28E-03	1.39E-03	13.7
3	-9.42	1.71E-04	-8.70E-07	1.00E-04	32.8	4.29E-03	1.39E-03	13.7
4	-9.41	1.71E-04	1.51E-07	1.00E-04	32.9	4.31E-03	1.41E-03	13.7
5	-9.42	1.71E-04	6.52E-08	1.00E-04	32.9	4.31E-03	1.41E-03	13.7

Grid 4

C.5 Results generated by the bivariate MINGCV algorithm, for the data set franke4.dat

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-5.03	6.33E-03	2.50E-03	0	86	6.84E-02	0.222	6.6
1	-5.53	4.70E-03	2.86E-03	0	83.8	5.75E-02	0.23	8.3
2	-5.53	4.75E-03	2.47E-03	7.81E-04	83.8	5.77E-02	0.203	8.24
3	-5.78	4.18E-03	2.28E-03	7.72E-04	82.9	5.36E-02	0.183	9.01
4	-6.03	3.65E-03	1.97E-03	1.24E-03	81.9	4.95E-02	0.155	9.83
5	-6.28	3.19E-03	1.70E-03	1.08E-03	81	4.57E-02	0.132	10.6
6	-6.53	2.82E-03	1.37E-03	1.29E-03	80	4.24E-02	0.106	11.4
7	-6.78	2.51E-03	1.08E-03	1.16E-03	79	3.95E-02	8.36E-02	12.2
8	-7.03	2.27E-03	8.22E-04	1.05E-03	78	3.71E-02	6.42E-02	12.9
9	-7.28	2.09E-03	5.95E-04	9.07E-04	77	3.53E-02	4.81E-02	13.6
10	-7.53	1.97E-03	4.14E-04	7.28E-04	76.1	3.38E-02	3.55E-02	14.2
11	-7.78	1.89E-03	2.77E-04	5.48E-04	75.1	3.26E-02	2.62E-02	14.8
12	-8.03	1.83E-03	1.79E-04	3.90E-04	74.2	3.18E-02	1.96E-02	15.3
13	-8.49	1.77E-03	7.84E-05	2.19E-04	72.8	3.06E-02	1.22E-02	16.1
14	-8.94	1.75E-03	4.55E-05	9.19E-05	71.7	2.98E-02	9.53E-03	17

Grid 6

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-8.94	1.78E-04	7.57E-05	9.19E-05	52.4	6.99E-03	4.03E-03	14.4
1	-9.44	1.43E-04	4.19E-05	6.77E-05	49	5.86E-03	2.21E-03	14.8
2	-9.94	1.33E-04	1.66E-05	5.07E-05	44.9	5.18E-03	1.29E-03	15.2
3	-10.3	1.30E-04	5.46E-06	3.40E-05	42.5	4.79E-03	9.27E-04	15.5

Grid 5

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-10.3	2.35E-05	2.60E-05	3.40E-05	26	1.26E-03	2.97E-04	14.2
1	-10.8	1.67E-05	1.64E-05	1.92E-05	20.4	8.34E-04	1.27E-04	14.3
2	-11.3	1.06E-05	8.91E-06	1.49E-05	16.5	4.08E-04	4.93E-05	14.4

Grid 4

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-11.3	1.23E-05	9.35E-06	1.49E-05	8.74	3.06E-04	1.98E-05	14.1
1	-11.8	1.23E-05	9.35E-06	1.49E-05	8.74	3.06E-04	1.98E-05	14.1
2	-12.2	1.12E-05	7.21E-06	8.56E-06	7.23	2.42E-04	1.19E-05	14.1
3	-12.5	8.16E-06	7.69E-06	8.56E-06	6.06	1.73E-04	7.15E-06	14.1
4	-12.7	7.76E-06	6.03E-06	6.64E-06	5.03	1.40E-04	4.47E-06	14.1
5	-12.8	9.00E-06	1.98E-06	1.62E-05	4.14	1.24E-04	2.76E-06	14.1

Grid 3

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-12.8	2.13E-06	2.55E-05	1.62E-05	2.28	3.33E-05	1.53E-06	14
1	-13.3	9.71E-06	1.17E-05	5.55E-05	1.76	5.47E-05	9.15E-07	14
2	-13.6	8.90E-06	1.03E-05	6.62E-06	1.44	4.29E-05	5.59E-07	14
3	-13.8	8.98E-07	2.56E-05	6.62E-06	1.13	1.07E-05	3.49E-07	14
4	-14.1	2.27E-06	2.32E-05	9.37E-06	0.884	1.33E-05	2.17E-07	14
5	-14.3	1.09E-05	6.91E-06	6.53E-05	0.692	2.28E-05	1.37E-07	14
6	-14.4	3.34E-06	1.90E-05	6.53E-05	0.635	5.03E-05	1.03E-07	14

Grid 2

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-14.4	2.19E-05	8.74E-06	6.53E-05	0.479	2.24E-05	1.30E-07	13.2
1	-14.6	5.56E-05	-7.03E-05	5.91E-04	0.434	3.24E-05	8.69E-08	13.2
2	-14.5	1.72E-05	-3.50E-07	5.88E-04	0.489	2.03E-05	8.47E-08	13.2
3	-14.5	9.75E-05	-1.65E-04	5.88E-04	0.494	4.88E-05	8.61E-08	13.3
4	-14.2	2.66E-05	-1.95E-05	5.18E-04	0.651	3.35E-05	1.47E-07	13.3
5	-14.2	5.85E-06	2.25E-05	5.18E-04	0.677	1.64E-05	1.58E-07	13.3
6	-14.2	4.73E-05	-6.13E-05	5.18E-04	0.652	4.48E-05	1.45E-07	13.3
7	-14.1	1.61E-05	1.53E-06	5.31E-04	0.732	2.93E-05	1.81E-07	13.3
8	-14.1	6.06E-06	2.15E-05	5.31E-04	0.731	1.80E-05	1.80E-07	13.3
9	-14.1	3.32E-05	-3.29E-05	5.31E-04	0.704	4.05E-05	1.67E-07	13.3
10	-14.1	2.66E-05	-1.98E-05	5.31E-04	0.749	3.86E-05	1.87E-07	13.3
11	-14	5.22E-06	2.28E-05	5.31E-04	0.776	1.77E-05	2.00E-07	13.3
12	-14.1	3.49E-05	-3.65E-05	5.31E-04	0.745	4.41E-05	1.85E-07	13.2
13	-14	2.24E-05	-1.16E-05	5.31E-04	0.797	3.77E-05	2.10E-07	13.3
14	-14	2.23E-05	-1.13E-05	5.31E-04	0.814	3.85E-05	2.19E-07	13.3
15	-14	6.23E-06	2.06E-05	5.31E-04	0.832	4.45E-05	2.28E-07	13.3

Grid 1

C.6 Results generated by the bivariate MINGCV algorithm, for the data set peaks.dat

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-0.552	0.399	8.28E-02	0	168	0.558	15.6	9
1	-1.05	0.354	6.63E-02	3.31E-02	165	0.517	12.5	9.79
2	-1.55	0.336	4.06E-02	5.13E-02	163	0.497	8.41	10.5
3	-2.05	0.319	3.01E-02	2.10E-02	161	0.477	6.59	11.5
4	-2.55	0.301	3.02E-02	2.10E-02	158	0.458	6.22	12.9
5	-2.8	0.293	3.16E-02	2.10E-02	157	0.448	6.23	13.8
6	-3.05	0.286	3.05E-02	4.72E-03	156	0.44	5.91	14.7
7	-3.3	0.279	2.78E-02	1.05E-02	155	0.432	5.4	15.8
8	-3.55	0.274	2.46E-02	1.29E-02	154	0.425	4.84	16.8
9	-3.8	0.269	2.15E-02	1.23E-02	153	0.419	4.31	18.1
10	-4.05	0.264	1.90E-02	1.02E-02	153	0.412	3.86	19.4
11	-4.46	0.259	1.71E-02	7.76E-03	152	0.403	3.5	21.9

Grid 6

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-4.46	6.09E-02	1.52E-02	1.13E-02	135	0.175	2.17	20.6
1	-4.96	6.03E-02	4.83E-03	2.08E-02	130	0.168	1.11	21.3
2	-5.2	5.66E-02	1.52E-03	1.42E-02	128	0.16	0.776	22.4
3	-5.3	5.50E-02	2.13E-03	1.42E-02	127	0.157	0.788	23
4	-5.45	5.42E-02	3.06E-03	1.42E-02	126	0.155	0.822	23.5
5	-5.67	5.40E-02	2.21E-03	3.98E-03	125	0.151	0.72	25

Grid 5

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-5.67	6.23E-02	-1.13E-02	7.38E-02	94	0.123	0.627	23.8
1	-5.7	5.50E-02	-4.36E-03	4.54E-02	101	0.125	0.635	23.3
2	-5.6	5.55E-02	-4.60E-03	4.54E-02	102	0.126	0.66	22.9
3	-5.5	5.47E-02	-3.56E-03	1.02E-02	104	0.128	0.703	22.5
4	-5.15	5.40E-02	-2.02E-03	4.43E-03	109	0.133	0.839	21.2
5	-4.7	5.41E-02	6.43E-04	5.85E-03	116	0.141	1.11	19.8
6	-4.81	5.38E-02	1.52E-04	4.47E-03	114	0.139	1.02	20.1
7	-4.84	5.38E-02	-2.32E-05	4.47E-03	114	0.139	1	20.2
8	-4.83	5.38E-02	8.50E-05	4.47E-03	114	0.138	1.01	20.3

Grid 4

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-4.83	5.26E-02	-2.27E-04	4.47E-03	102	0.123	1.15	20.5
1	-4.8	5.28E-02	1.00E-04	4.47E-03	103	0.124	1.17	20.3
2	-4.83	5.28E-02	-4.71E-04	4.47E-03	102	0.124	1.13	20.3
3	-4.72	5.30E-02	4.48E-04	8.72E-03	104	0.126	1.21	20
4	-4.77	5.29E-02	2.76E-05	8.72E-03	103	0.125	1.17	20.1
5	-4.77	5.29E-02	-2.23E-05	8.72E-03	103	0.125	1.17	20.1

Grid 3

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-4.77	5.28E-02	2.32E-04	8.72E-03	97.6	0.118	1.3	20.3
1	-4.8	5.27E-02	-7.49E-05	8.72E-03	97.2	0.117	1.26	20.4
2	-4.79	5.27E-02	-6.88E-05	8.72E-03	97.4	0.118	1.26	20.4
3	-4.78	5.27E-02	-1.75E-05	8.72E-03	97.6	0.118	1.26	20.3
4	-4.78	5.28E-02	-5.40E-06	8.72E-03	97.6	0.118	1.26	20.3
5	-4.78	5.28E-02	-3.08E-05	8.72E-03	97.6	0.118	1.26	20.3

Grid 2

C.7 Results generated by the bivariate MINGCV algorithm, for the data set peaks15.dat

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-0.552	1.53	0.112	0	168	1.09	29.5	10.9
1	-1.05	1.51	4.49E-02	0.135	165	1.07	19.2	11.8
2	-1.39	1.5	1.69E-02	8.41E-02	164	1.06	14.6	12.5
3	-1.59	1.49	9.41E-03	3.73E-02	163	1.05	13.2	13
4	-1.84	1.49	7.52E-04	3.43E-02	162	1.04	11.8	13.7
5	-1.86	1.49	2.97E-03	3.43E-02	162	1.04	12	13.7
6	-1.95	1.49	-8.86E-04	3.43E-02	161	1.04	11.4	13.9
7	-1.92	1.49	-2.42E-03	3.43E-02	161	1.04	11.2	13.9
8	-1.85	1.49	-2.24E-03	3.43E-02	162	1.04	11.2	13.7
9	-1.79	1.49	9.63E-04	3.43E-02	162	1.04	11.6	13.5
10	-1.81	1.49	1.29E-03	3.43E-02	162	1.04	11.7	13.6
11	-1.85	1.49	3.73E-04	3.43E-02	162	1.04	11.6	13.7

Grid 6

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-1.85	1.4	1.55E-02	3.43E-02	155	0.965	19	14.9
1	-2.31	1.4	-1.30E-02	6.31E-02	152	0.945	15.7	16.8
2	-2.11	1.4	-2.24E-03	5.22E-02	153	0.955	16.3	15.8
3	-2.06	1.4	1.14E-03	5.22E-02	153	0.957	16.8	15.6
4	-2.09	1.4	-1.86E-04	5.22E-02	153	0.956	16.7	15.7
5	-2.08	1.4	7.54E-06	5.22E-02	153	0.956	16.7	15.7

Grid 5

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-2.08	1.37	-1.51E-02	5.22E-02	149	0.922	19.7	16.2
1	-1.79	1.38	6.37E-03	7.43E-02	152	0.939	21.3	14.9
2	-1.88	1.38	-3.30E-03	7.43E-02	151	0.934	20.2	15.3
3	-1.83	1.38	2.73E-04	7.43E-02	152	0.936	20.5	15.1
4	-1.84	1.38	-1.28E-04	7.43E-02	152	0.936	20.4	15.1
5	-1.84	1.38	5.30E-06	7.43E-02	152	0.936	20.4	15.1

Grid 4

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-1.84	1.38	-4.05E-03	7.43E-02	149	0.924	23.2	15.4
1	-1.78	1.38	-1.80E-04	7.43E-02	150	0.928	23.5	15.1
2	-1.78	1.38	-1.79E-04	7.43E-02	150	0.928	23.3	15.1
3	-1.78	1.38	2.25E-05	7.43E-02	150	0.928	23.3	15.1
4	-1.78	1.38	-1.08E-04	7.43E-02	150	0.928	23.2	15.1
5	-1.78	1.38	-6.98E-05	7.43E-02	150	0.928	23.2	15.1

Grid 3

C.8 Results generated by the bivariate MINGCV algorithm, for the data set peaks0.dat

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-0.552	0.386	8.93E-02	0	168	0.549	16.5	9.05
1	-1.05	0.343	6.75E-02	4.35E-02	165	0.51	12.6	9.8
2	-1.55	0.324	4.24E-02	5.03E-02	163	0.488	8.55	10.5
3	-2.05	0.303	3.45E-02	1.57E-02	161	0.465	7.06	11.6
4	-2.55	0.283	3.57E-02	1.57E-02	158	0.443	6.81	13.1
5	-2.8	0.273	3.71E-02	1.57E-02	157	0.433	6.8	14.1
6	-3.05	0.265	3.56E-02	6.16E-03	156	0.423	6.42	15.1
7	-3.3	0.258	3.25E-02	1.24E-02	155	0.415	5.84	16.2
8	-3.55	0.251	2.87E-02	1.49E-02	154	0.407	5.2	17.3
9	-3.8	0.245	2.52E-02	1.42E-02	153	0.399	4.61	18.6
10	-4.05	0.239	2.22E-02	1.19E-02	153	0.392	4.11	20
11	-4.46	0.233	1.99E-02	9.32E-03	152	0.382	3.69	22.6

Grid 7

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-4.46	1.54E-02	1.83E-02	9.64E-03	135	8.80E-02	1.93	20.4
1	-4.96	1.22E-02	6.98E-03	2.26E-02	130	7.52E-02	0.754	20.8
2	-5.27	7.59E-03	4.08E-03	9.41E-03	127	5.85E-02	0.434	21.7
3	-5.71	5.18E-03	2.84E-03	2.87E-03	124	4.58E-02	0.284	23.1

Grid 6

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-5.71	2.06E-03	1.98E-03	5.92E-03	94	2.24E-02	0.13	20.3
1	-6.18	1.21E-03	1.44E-03	1.61E-03	93.4	1.71E-02	8.39E-02	20.7
2	-6.68	8.32E-04	7.78E-04	1.33E-03	86.2	1.31E-02	4.03E-02	21
3	-7.18	5.55E-04	4.58E-04	6.39E-04	79.9	9.65E-03	2.09E-02	21.4

Grid 5

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-7.18	5.09E-04	6.77E-04	6.39E-04	55.2	6.56E-03	1.63E-02	20.5
1	-7.64	3.72E-04	4.69E-04	4.16E-04	47	4.77E-03	8.37E-03	20.7
2	-8.14	2.71E-04	3.40E-04	2.59E-04	39.8	2.72E-03	4.41E-03	21

Grid 4

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-8.14	2.34E-04	5.55E-04	2.59E-04	19.3	1.56E-03	1.69E-03	20.6
1	-8.57	2.31E-04	4.87E-04	2.71E-04	16.6	1.33E-03	1.13E-03	20.6
2	-9.02	2.18E-04	4.45E-04	1.68E-04	14.2	1.11E-03	7.72E-04	20.7
3	-9.37	2.00E-04	4.16E-04	1.16E-04	12.1	9.01E-04	5.22E-04	20.7
4	-9.62	1.92E-04	3.74E-04	1.69E-04	10.3	7.53E-04	3.51E-04	20.7
5	-9.87	1.79E-04	3.51E-04	9.13E-05	8.77	6.50E-04	2.39E-04	20.7

Grid 3

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-9.87	1.63E-04	8.55E-04	9.13E-05	5.16	3.47E-04	1.62E-04	20.8
1	-10.3	1.45E-04	8.83E-04	9.13E-05	4.19	2.66E-04	1.06E-04	20.8
2	-10.5	2.03E-05	1.13E-03	9.13E-05	3.35	2.75E-04	6.89E-05	20.8

Grid 2

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-10.5	7.36E-05	1.56E-03	9.13E-05	2.13	9.60E-05	4.09E-05	23.2
1	-11	1.62E-04	1.30E-03	1.04E-03	1.75	1.17E-04	2.63E-05	23.2
2	-11.3	6.14E-04	3.61E-04	3.75E-03	1.41	1.84E-04	1.68E-05	23.1
3	-11.4	2.36E-04	1.07E-03	3.75E-03	1.29	1.04E-04	1.35E-05	23.1
4	-11.6	2.19E-03	-2.81E-03	1.36E-02	0.996	2.45E-04	8.14E-06	23.2
5	-11.4	1.56E-03	-1.65E-03	5.59E-03	1.18	2.45E-04	1.13E-05	23.2
6	-11.1	4.52E-04	5.90E-04	7.59E-03	1.57	1.76E-04	1.96E-05	23.2
7	-11.2	1.23E-04	1.25E-03	7.59E-03	1.47	8.59E-05	1.71E-05	23.2
8	-11.4	5.40E-04	4.03E-04	5.13E-03	1.27	1.55E-04	1.26E-05	23.1
9	-11.5	2.12E-04	1.07E-03	5.13E-03	1.18	9.02E-05	1.09E-05	23.2
10	-11.7	3.00E-03	-4.48E-03	2.66E-02	0.971	2.80E-04	7.40E-06	23.2
11	-11.5	1.33E-04	1.29E-03	3.42E-02	1.12	6.77E-05	1.02E-05	23.2
12	-11.5	2.64E-03	-3.77E-03	3.42E-02	1.09	2.94E-04	9.57E-06	23.2
13	-11.4	1.87E-03	-2.23E-03	1.39E-02	1.2	2.74E-04	1.18E-05	23.2
14	-11.3	4.12E-04	7.16E-04	1.84E-02	1.4	1.50E-04	1.60E-05	23.2
15	-11.3	4.50E-06	1.52E-03	1.84E-02	1.36	1.52E-05	1.49E-05	23.2
16	-11.4	3.17E-04	8.97E-04	1.84E-02	1.26	1.18E-04	1.28E-05	23.2
17	-11.4	7.49E-05	1.39E-03	1.84E-02	1.2	5.48E-05	1.17E-05	23.2
18	-11.5	7.49E-05	1.39E-03	1.84E-02	1.12	1.48E-04	1.17E-05	23.2

Grid 1

No. of upates	θ	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$	Tr	R	$dR/d\theta$	Sobolev Norm
0	-11.5	3.09E-03	-4.49E-03	7.79E-02	0.977	2.86E-04	1.25E-05	46.5
1	-11.5	2.68E-03	-3.66E-03	7.79E-02	1.04	2.84E-04	1.31E-05	46.5
2	-11.4	9.75E-04	1.26E-04	7.79E-02	1.1	1.81E-04	1.45E-05	46.9
3	-11.4	1.86E-03	-1.81E-03	7.79E-02	1.11	2.52E-04	1.45E-05	46.4
4	-11.4	2.35E-03	-2.90E-03	7.79E-02	1.14	2.91E-04	1.50E-05	46.5
5	-11.3	2.13E-03	-2.38E-03	7.79E-02	1.18	1.93E-04	1.64E-05	46.7

Grid 0

Appendix D

Results for Chapter 13

The tables in this appendix are discussed in Chapter 12. They report updates of various quantities relevant to the OPTRSS and MINGCV algorithms described in Chapter 8. The notation is explained as follows:

θ: the logarithm of the smoothing parameter.

θ_q: the q^{th} update of the logarithm of the smoothing parameter.

R: the estimate of the residual sum of squares.

Signal: the estimate of $tr(A)$, where A is the influence matrix.

GCV: the estimate of the generalised cross validation.

Sobolev Norm: as discussed in Chapter 12, the Sobolev Norm is the square root of the curvature, $\sqrt{\alpha^T Z \alpha}$.

D.1 Results generated by the bivariate MINGCV algorithm, for the African temperature data set.

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	3.24	3.77	0.447	0
1	2.74	3.55	0.215	0.464
2	2.28	3.42	0.22	0.464
3	1.8	3.34	0.168	0.109
4	1.55	3.31	0.146	8.90E-02
5	1.3	3.27	0.13	6.35E-02
6	1.05	3.24	0.114	6.46E-02
7	0.804	3.21	9.66E-02	6.87E-02
8	0.554	3.19	8.03E-02	6.53E-02
9	0.304	3.17	6.54E-02	5.97E-02
10	5.38E-02	3.15	5.20E-02	5.33E-02
11	-0.196	3.14	4.05E-02	4.62E-02
12	-0.446	3.13	3.07E-02	3.90E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.49E+03	4.31	0	0	0.806
1.47E+03	1.91	687	6.15	24.3
1.47E+03	1.85	342	6.92	25.7
1.46E+03	1.81	344	7.94	27.1
1.46E+03	1.79	266	8.9	28.9
1.46E+03	1.78	232	9.47	29.7
1.46E+03	1.77	208	10.1	30.5
1.46E+03	1.76	183	10.9	31.3
1.46E+03	1.75	157	11.6	32.1
1.46E+03	1.75	133	12.4	32.8
1.46E+03	1.74	110	13.2	33.5
1.46E+03	1.74	90.3	14.1	34.2
1.46E+03	1.73	72.7	15	34.8

Grid 6

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-0.446	2.32	0.11	3.90E-02
1	-0.919	2.27	0.106	1.39E-02
2	-1.17	2.24	8.93E-02	6.76E-02
3	-1.42	2.21	7.49E-02	5.77E-02
4	-1.67	2.19	6.37E-02	4.48E-02
5	-1.92	2.17	5.16E-02	4.82E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.46E+03	1.72	57.7	16.4	35.4
1.42E+03	1.45	186	25.1	75.9
1.41E+03	1.43	176	27.5	78
1.41E+03	1.42	150	29.2	79.3
1.41E+03	1.41	127	31.4	80.4
1.41E+03	1.4	110	33.8	81.7

Grid 5

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-1.92	1.76	9.02E-02	4.82E-02
1	-2.31	1.72	8.25E-02	3.11E-02
2	-2.56	1.7	6.77E-02	5.92E-02
3	-2.81	1.68	5.76E-02	4.01E-02
4	-3.06	1.67	5.09E-02	2.70E-02
5	-3.31	1.66	4.34E-02	3.01E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.41E+03	1.39	92.2	37.1	83
1.30E+03	1.15	191	46	193
1.30E+03	1.14	165	49.2	195
1.29E+03	1.13	138	52.7	198
1.29E+03	1.12	122	56.9	203
1.28E+03	1.11	110	61.6	208

Grid 4

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-3.31	1.53	-7.47E-02	3.01E-02
1	-3.19	1.49	-3.61E-02	0.154
2	-2.96	1.48	-1.82E-02	7.65E-02
3	-2.72	1.48	2.50E-03	8.70E-02
4	-2.75	1.48	-1.06E-03	8.70E-02
5	-2.74	1.48	6.45E-04	8.70E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.28E+03	1.1	97.7	69	212
1.05E+03	0.873	118	81.5	440
1.08E+03	0.883	124	75.3	412
1.10E+03	0.894	145	70.8	397
1.11E+03	0.907	170	66	379
1.11E+03	0.905	166	66.5	382

Grid 3

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-2.74	1.42	-1.63E-02	8.70E-02
1	-2.56	1.42	6.62E-03	0.122
2	-2.61	1.42	1.15E-03	0.122
3	-2.62	1.42	-4.37E-04	0.122
4	-2.62	1.42	-1.51E-04	0.122
5	-2.62	1.42	-1.31E-04	0.122
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.11E+03	0.906	167	66.4	381
981	0.782	251	74.3	511
1.01E+03	0.802	267	69.5	487
999	0.796	261	70.9	493
997	0.795	260	71.1	495
998	0.796	260	71	494

Grid 2

D.2 Results generated by the bivariate MINGCV algorithm for the African temperature data set, with an initial grid resolution of 25.6^o.

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	4.18	4.99	0.812	0
1	3.68	4.53	0.41	0.804
2	3.18	4.41	0.207	0.405
3	2.68	4.34	0.108	0.198
4	2.18	4.27	7.16E-02	7.28E-02
5	1.68	4.2	5.60E-02	3.13E-02
6	1.18	4.15	4.13E-02	2.94E-02
7	0.683	4.12	2.59E-02	3.08E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.48E+03	4.31	0	0	0.875
1.46E+03	2.21	1.20E+03	5.56	13.5
1.46E+03	2.11	614	6.08	15.2
1.46E+03	2.08	317	6.59	16.2
1.46E+03	2.06	170	6.97	16.9
1.46E+03	2.04	115	7.45	17.5
1.46E+03	2.03	91	8.11	18.1
1.46E+03	2.01	68.7	8.86	18.7

Grid 8

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	0.683	2.83	0.18	3.08E-02
1	.102	2.77	0.136	0.175
2	-0.229	2.75	9.78E-02	0.152
3	-0.479	2.75	6.16E-02	0.145
4	-0.905	2.73	3.06E-02	7.29E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.46E+03	2	45.6	9.98	19.2
1.44E+03	1.64	274	18	37.8
1.44E+03	1.62	212	19.1	41
1.43E+03	1.61	158	19.8	42.5
1.43E+03	1.61	107	20	43.7

Grid 7

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-.905	2.19	6.39E-02	7.29E-02
1	-1.37	2.17	5.68E-02	2.81E-02
2	-1.62	2.15	4.72E-02	3.87E-02
3	-1.87	2.13	4.62E-02	3.83E-03
4	-2.12	2.12	4.33E-02	1.18E-02
5	-2.37	2.11	3.87E-02	1.82E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.43E+03	1.6	61	21.7	44.9
1.38E+03	1.38	139	27.4	99.3
1.38E+03	1.37	123	28.4	102
1.37E+03	1.36	104	30	103
1.37E+03	1.36	97.4	32.3	105
1.37E+03	1.35	90.5	34.8	107

Grid 6

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-2.37	1.68	4.77E-02	1.82E-02
1	-2.75	1.65	4.31E-02	1.84E-02
2	-3	1.63	3.97E-02	1.34E-02
3	-3.25	1.63	3.05E-02	3.71E-02
4	-3.5	1.62	2.26E-02	3.14E-02
5	-3.75	1.62	1.58E-02	2.73E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.37E+03	1.34	82	39	109
1.22E+03	1.07	166	54.8	261
1.21E+03	1.05	137	58.3	264
1.21E+03	1.04	122	63	270
1.20E+03	1.04	105	67.4	277
1.19E+03	1.03	90.6	72.2	284

Grid 5

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-3.75	1.63	-0.164	2.73E-02
1	-3.64	1.54	-0.115	0.195
2	-3.14	1.5	-7.66E-02	7.66E-02
3	-2.64	1.48	-5.35E-02	4.62E-02
4	-2.14	1.49	-0.121	4.62E-02
5	-1.89	1.51	-0.211	4.62E-02
6	-1.64	1.54	-0.429	4.62E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.19E+03	1.02	78.4	80.1	290
918	0.793	111	95.3	559
953	0.802	111	88.6	524
998	0.828	148	76.8	479
1.05E+03	0.862	175	66.4	429
1.10E+03	0.907	130	58.8	379
1.12E+03	0.934	53.1	56.3	354

Grid 4

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-1.64	1.57	-1.12	4.62E-02
1	-1.14	1.67	-2.51	4.62E-02
2	-0.888	1.83	-5.43	4.62E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.15E+03	1.01	-143	59.2	329
1.12E+03	0.952	-636	60.3	357
1.15E+03	1.01	-1.93E+03	68.2	327

Grid 3

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-0.888	2.13	-12.3	4.62E-02
1	-0.388	2.63	-26.6	4.62E-02
2	-0.138	3.45	-56.8	4.62E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.18E+03	1.18	-4.75E+03	104	299
1.18E+03	1.16	-1.11E+04	104	299
1.19E+03	1.31	-2.49E+04	134	285

Grid 2

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	0.138	4.91	-149	4.62E-02
1	0.362	7.35	-320	4.62E-02
2	0.612	11.4	-687	4.62E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.21E+03	1.81	-5.49E+04	251	271
1.21E+03	1.81	-1.46E+05	251	270
1.21E+03	2.22	-3.14E+05	324	267

Grid 1

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	0.612	11.4	-687	4.62E-02
1	-0.693	13.2	-3.30E+03	4.62E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.21E+03	2.99	-6.79E+05	906	264
1.21E+03	1.15	-3.28E+06	251	264

Grid 0

D.3 Results generated by the bivariate MINGCV algorithm, for the Australian temperature data set.

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	1.86	1.2	0.257	0
1	1.36	1.14	9.42E-02	0.326
2	1.07	1.11	6.43E-02	0.103
3	0.567	1.08	5.42E-02	2.03E-02
4	6.74E-02	1.05	4.81E-02	1.21E-02
5	-0.433	1.02	4.46E-02	7.08E-03
6	-0.933	0.998	4.24E-02	4.35E-03
7	-1.43	0.978	4.03E-02	4.27E-03
8	-1.93	0.959	3.90E-02	2.60E-03
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.13E+03	4.96	0	0	1.12
1.11E+03	1.07	294	6.5	21.6
1.11E+03	1.05	112	6.7	23.2
1.11E+03	1.03	79.1	7.23	24.3
1.11E+03	1.01	67.2	8.37	26.2
1.11E+03	0.998	59.7	9.79	28.2
1.10E+03	0.984	55	11.6	30
1.10E+03	0.971	51.8	13.9	31.7
1.10E+03	0.96	48.7	16.7	33.3

Grid 6

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-1.93	0.960	4.00E-02	2.60E-03
1	-2.06	0.79	3.19E-02	2.60E-03
2	-2.56	0.775	2.33E-02	1.70E-02
3	-3.06	0.766	1.42E-02	1.83E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.10E+03	0.947	46.7	21.2	34.8
1.06E+03	0.831	46.1	24	73.5
1.06E+03	0.819	34.4	27.9	79

Grid 5

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-3.06	0.606	6.51E-03	1.83E-02
1	-3.56	0.6	4.68E-03	5.13E-03
2	-4.0	0.598	7.70E-04	7.83E-03
3	-4.4	0.598	-3.11E-04	7.83E-03
4	-4.36	0.597	-8.73E-04	7.83E-03
5	-4.24	0.597	2.28E-04	9.87E-03
6	-4.27	0.597	3.90E-04	9.87E-03
7	-4.31	0.597	2.98E-04	9.87E-03
Tr	R	$dR/d\theta$	Sobolev norm	Signal
1.05E+03	0.809	23.3	34.1	81.8
962	0.661	36.8	42.7	172
955	0.652	29.5	46.5	179
945	0.645	22.9	52.9	189
944	0.643	21.3	53.8	190
945	0.644	20.7	53.1	189
947	0.645	21.9	51.6	187
947	0.645	22	51.9	187

Grid 4

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-4.31	0.552	-1.99E-02	9.87E-03
1	-4.09	0.542	-9.76E-03	4.05E-02
2	-3.85	0.539	-4.91E-03	2.02E-02
3	-3.6	0.539	-5.40E-05	1.99E-02
4	-3.6	0.538	1.08E-04	1.99E-02
5	-3.61	0.538	-3.70E-06	1.99E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
946	0.644	21.8	53	188
820	0.537	34.6	64.7	314
838	0.544	37.8	59.5	296
850	0.55	42.3	55.2	284
864	0.559	47.3	50.8	270
864	0.559	47.2	50.8	270

Grid 3

No. of updates	θ_q	GCV	$dGCV/d\theta$	$d^2GCV/d\theta^2$
0	-3.61	0.528	8.85E-05	1.99E-02
1	-3.61	0.529	4.37E-04	1.99E-02
2	-3.63	0.529	-5.15E-04	1.99E-02
3	-3.61	0.529	1.82E-04	1.99E-02
4	-3.61	0.529	-1.61E-04	1.99E-02
5	-3.61	0.529	2.63E-05	1.99E-02
Tr	R	$dR/d\theta$	Sobolev norm	Signal
864	0.559	47	50.9	270
798	0.512	68.7	56.3	336
798	0.511	67.9	56.4	336
796	0.51	67.1	56.8	338
798	0.512	67.6	56.2	336
797	0.511	67.4	56.4	337

Grid 2