Software Support for Metrology
Best Practice Guide No. 4

Discrete Modelling and Experimental Data Analysis

R M Barker, M G Cox, A B Forbes and P M Harris

April 2004

Version 2.0

Software Support for Metrology
Best Practice Guide No. 4

Discrete Modelling and Experimental Data Analysis

R M Barker, M G Cox, A B Forbes and P M Harris
Centre for Mathematics and Scientific Computing

April 2004

ABSTRACT

Metrology, the science of measurement, involves the determination from experiment of estimates of the values of physical quantities, along with the associated uncertainties. In this endeavour, a mathematical model of the measurement system is required in order to extract information from the experimental data. Modelling involves *model building*: developing a mathematical model of the measurement system in terms of equations involving parameters that describe all the relevant aspects of the system, and *model solving*: determining estimates of the model parameters from the measured data by solving the equations constructed as part of the model.

This best-practice guide covers all the main stages in experimental data analysis: construction of candidate models, model parameterisation, uncertainty structure in the data, uncertainty of measurements, choice of parameter estimation algorithms and their implementation in software, with the concepts illustrated by case studies.

The Guide looks at validation techniques for the main components of discrete modelling: building the functional and statistical model, model solving and parameter estimation methods, goodness of fit of model solutions and experimental design and measurement strategy. The techniques are illustrated in detailed case studies.

Version 2.0

ISSN 1471–4124

Authorised by Dr Dave Rayner,
Head of the Centre for Mathematics and Scientific Computing

National Physical Laboratory,
Queens Road, Teddington, Middlesex, United Kingdom TW11 0LW

Contents

Chapter 1

Introduction

1.1 Mathematical modelling in metrology

Metrology, the science of measurement, involves the determination of quantitative estimates of physical quantities from experiment, along with the associated uncertainties. This process involves the following components:

Model building. Developing a mathematical model of the experimental system in terms of mathematical equations involving parameters that describe all the relevant aspects of the system. The model will need to specify how the system is expected to respond to input data and the nature of the uncertainties associated with the data.

Model solving. Determining estimates of the model parameters from the measured data by solving the mathematical equations constructed as part of the model. In general, this involves developing an algorithm that will determine the values for the parameters that best explain the data. These algorithms are often referred to as *estimators*.

Software implementation of solution algorithms. Practically all calculations of fitted parameters are performed by software.

Model validation. Determining whether the results produced are consistent with the input data, theoretical results, reference data, etc. All stages need to be examined. Does the model adequately encapsulate what is known about the system? Does the method of solution produce unbiased estimates of the parameters and valid uncertainties? If information about the model is determined by software, then it is important that the software is valid to ensure that conclusions are based on reliable calculations.

Generally, these steps are revisited as the model is refined and the experimental design is evolved, resulting in a better explanation of the observed behaviour and more dependable uncertainties associated with the quantities of interest.

[0]This document: http://www.npl.co.uk/ssfm/download/bpg.html#ssfmbpg4

1.2 Scope and structure of this Guide

It is useful to classify the types of data arising in metrology into two categories: i) *discrete* and ii) *continuous*.

Example: the distribution of heat in a rectangular plate

Modelling discrete data. In a measurement experiment, the temperatures T_i are measured simultaneously at a fixed number m of locations (x_i, y_i) on a rectangular plate in a steady state. The data can be represented in a finite array whose ith row is (x_i, y_i, T_i). The temperature $t(x, y, \mathbf{a})$ is modelled as a function of location and model parameters $\mathbf{a}$. For example, $\mathbf{a}$ could be the coefficients of a bivariate polynomial surface. The data analysis problem is to find the values of the parameters $\mathbf{a}$ so that $t(x, y, \mathbf{a})$ best explains the data. For instance, a least-squares estimate of $\mathbf{a}$ is found by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} \left(T_i - t(x_i, y_i, \mathbf{a})\right)^2 .$$

The measurement strategy is discrete in the sense that only a finite number of measurements are taken. The data analysis problem is discrete in the sense that the function to be minimised is a discrete sum based on algebraic equations. However, the model involves continuous phenomena: the temperature is modelled as a function t of location, even though the data representing the temperature are given at a finite number of points.

Modelling continuous data. Two adjacent edges of the plate are held at temperatures $g(x)$ and $h(y)$ where g and h are known functions defined at distances x and y along the edges. The data analysis problem is to determine the steady-state temperature $t(x, y)$ at each point on the plate, given the coefficient ν of heat conduction of the material. The data analysis problem will involve the solution of the heat equation, a partial differential equation, subject to the boundary conditions. The data is continuous in the sense that g and h are defined at each point along the edge, not at a finite number of points. In practice, these functions will be specified by a finite amount of information, for example, the coefficients of polynomial representations of the functions. The numerical solution will also involve a discretisation of the equations to be solved. ♯

This Guide is concerned with modelling discrete data and experimental data analysis. In chapters 2 and 3, we describe the main components of model building and model solving and are meant to give an overview of discrete modelling in metrology. Chapter 4 discusses the data analysis methods used in metrology, while chapter 5 is concerned with important empirical models used in metrology. These two chapters present tutorial material on estimation methods and model types.

Chapters 6 to 10 discusses the main aspects of model validation. A number of case studies are given in section 11 illustrating the use of model types, parameter estimation and validation methods. The chapters on data analysis methods, model types and case studies will be further expanded in future revisions of the Guide.

A summary of the main issues is given in section 12.

Version 1.0 (March 2000)

Version 1.1 revision of this guide (January 2002)

The main changes introduced in version 1.1 were:

- Correction of typographical errors;
- Correction to formulæ concerning Chebyshev polynomials on page 75;
- Minor changes to the text;
- Expanded index section.

Version 2.0 (April 2004)

The main changes introduced in this revision are

- Incorporation (as chapters 6 to 10) of SS*f*M Best Practice Guide No. 10: *Discrete Model Validation* [11];
- Review of statistical concepts;
- More explicit description of statistical models in terms of random variables;
- Tutorial material on generalised Gauss-Markov regression, asymptotic least squares, maximum likelihood estimation, Bayesian parameter estimation;
- Tutorial material on Fourier series, asymptotic polynomials, tensor product surfaces, wavelets, radial basis functions, neural networks, and nonuniform rational B-splines;
- Additional case studies.

1.3 Discrete modelling resources

1.3.1 Reference books

Discrete modelling draws on a number of disciplines, including data approximation, optimisation, numerical analysis and numerical linear algebra, and statistics. Although aspects of discrete modelling are technically difficult, much

of it relies on a few underlying concepts covered in standard text books; see, for example, [94, 117, 119, 142, 150, 169, 183, 187, 204, 207]. Many text books and reference books have explicit descriptions of algorithms; see e.g. [110, 119, 165, 207], and a number of books also supply software on a disk, including the *Numerical Recipes* family of books [184] which give reasonably comprehensive guidance on algorithm design and further reading.

1.3.2 Conference series

While standard textbooks are valuable for understanding the basic concepts, few are concerned with metrology directly. The main objective of the conference series *Advanced Mathematical and Computational Tools in Metrology* is to discuss how these mathematical, numerical and computational techniques can be used in metrology. Collected papers associated with the conferences are published; see [44, 45, 46, 47, 48] and the forthcoming [181]. Many of the papers present survey or tutorial material directly relevant to discrete modelling; see, for example, [19, 21, 22, 23, 37, 43, 59, 60, 61, 62, 64, 81, 88, 101, 103, 118, 148, 149, 153, 186].

The conference series *Algorithms for Approximation* [78, 145, 151, 152] deals with more general aspects of data approximation, many of which have direct relevance to metrology.

1.3.3 Software sources

The last four decades have been ones of great success in terms of the development of reliable algorithms for solving the most common computational problems. In the fields of numerical linear algebra – linear equations, linear least squares, eigenvalues, matrix factorisations — and optimisation — nonlinear equations, nonlinear least squares, minimisation subject to constraints, linear programming, nonlinear programming — there is now a substantial core of software modules which the metrologist can exploit.

The scientist has a range of sources for software: i) specialist software developers/collectors such as the NAG library in the UK and IMSL in the US, ii) National laboratories, for example NPL, Harwell, Argonne, Oakridge, iii) universities, iv) industrial laboratories, v) software houses and vi) instrument manufacturers. Library software, used by many scientists and continually maintained, provides perhaps the best guarantee of reliability.

Library software. Below is a list of some of the libraries which have routines relevant to the metrologist.

NAG: A large Fortran Library[1] covering most of the computational disciplines including quadrature, ordinary differential equations, partial differential

[1] The NAG Library is available in other languages

equations, integral equations, interpolation, curve and surface fitting, optimisation, linear algebra (simultaneous linear equations, matrix factorisations, eigenvalues), correlation and regression analysis, analysis of variance and non-parametric statistics. [173]

IMSL: International Mathematical and Statistical Libraries, Inc. Similar to NAG but based in the US. [203]

LINPACK: A Fortran library for solving systems of linear equations, including least-squares systems, developed at Argonne National Laboratory (ANL), USA. See [87], and Netlib (below).

EISPACK: A companion library to LINPACK for solving eigenvalue problems also developed at ANL. See [193], and Netlib (below).

LAPACK: A replacement for, and further development of, LINPACK and EISPACK. LAPACK also appears as a sub-chapter of the NAG library. See [190], and Netlib (below).

Harwell: Optimisation routines including those for large and/or sparse problems. [128]

DASL: Data Approximation Subroutine Library, developed at NPL, for data interpolation and approximation with polynomial and spline curves and surfaces. [8]

MINPACK: Another Fortran Library developed at ANL for function minimisation. MINPACK contains software for solving nonlinear least-squares problems, for example. See [115], and Netlib (below).

A number of journals also publish the source codes for software. In particular the *ACM Transactions on Mathematical Software* has published over 700 algorithms for various types of computation. *Applied Statistics* publishes software for statistical computations.

Most library software has been written in Fortran 77, a language well suited to numerical computation but in other ways limited in comparison with more modern languages. The situation has changed radically with the advent of new versions of the language — Fortran 90/95 [148, 157] — which have all the features that Fortran 77 was perceived as lacking while maintaining full backwards compatibility. Using Fortran 90/95 to create dynamically linked libraries (DLLs), it is relatively straightforward to interface the numerical library software with spreadsheet packages on a PC, for example, or to software written in other languages. Many library subroutines now also appear in Fortran 90/95 implementations, e.g. [12, 174]; see also [185].

Scientific software packages. There are a number of scientific software packages, including Matlab, Mathematica, MathCad and S-Plus that are widely used by numerical mathematicians, scientists and engineers [154, 155, 156, 166, 208, 209]. The online documentation associated with these packages includes extensive tutorial material.

Netlib. A comprehensive range of mathematical software can be obtained over the Internet through *Netlib* [86]. For example, the LINPACK, EISPACK, LAPACK and MINPACK libraries are available through Netlib along with the later algorithms published in *ACM Transactions on Mathematical Software* [188]. The system is very easy to use and there are also browsing, news and literature search facilities.

Statlib. Statlib is similar to Netlib but covers algorithms and software for statistical calculations. [197]

Guide to Available Mathematical Software - GAMS. The Guide to Available Mathematical Software [170] developed and maintained by the National Institute of Standards and Technology (NIST), Gaithersburg, MD, provides a comprehensive listing of mathematical software classified into subject areas such as linear algebra, optimisation, etc. It includes the software in Netlib and the NAG and IMSL libraries. Using the search facilities the user can quickly identify modules in the public domain or in commercial libraries.

e-Handbook of Statistical Methods. NIST/SEMATECH also publishes, online, a Handbook of Statistical Methods [171].

1.3.4 SS*f*M and EUROMETROS

The resources we have listed so far relate to science in general rather than metrology in particular. Certainly, many of the problems in metrology are generic and it is sensible to apply general solutions where they are appropriate. The SS*f*M programme as a whole aims to bridge the gap between the best computational techniques and the needs of metrology with the main focus of bringing appropriate technology to the metrologist in a usable form. The SS*f*M website [167] continues to provide an access point to a range of resources in the form of best-practice guides, reports, etc., and has assembled a large number of documents.

In particular, the SS*f*M programme provides not only information but also access to metrology software. While, in principle, many of the computations required by metrologist can be performed by assembling software from appropriate library modules found in Netlib or GAMS, the path from a list of library modules to software that solves the metrologist's problem in the preferred environment may be long and complicated, for a number of reasons:

1. The identification of the correct library modules and their integration may require detailed technical knowledge of the algorithms.
2. The library software may be in an unfamiliar language and/or not easily interfaced with the metrologist's application software.
3. The available software is not portable to the metrologist's computer.

4. The software documentation is unfriendly, written in generic mathematics rather than the language of the application domain.

The EUROMETROS[2] metrology software repository (formerly METROS) provides the metrologist easy access to modules performing the key functions for metrology. METROS was developed as part of the *Software re-use* project of the first SS*f*M programme; in the second SS*f*M programme, it was renamed as EUROMETROS to reflect its relationship with the EUROMET organisation [91] and *AMCTM* European thematic network [2], and was re-implemented with new user interfaces [9].

The current content of EUROMETROS is predominantly routines for fitting empirical and geometric models to data, and routines associated with those models or with the fitting problems. These are mainly implementations developed by NPL in SS*f*M and previous programmes. EUROMETROS also describes functions and algorithms for metrology problems, test data sets for testing solutions to those problems, and reference material (best/good practice guides, papers, case study reports, etc.) related to software for metrology.

The implementations of metrological functions in EUROMETROS will be written and documented in the language of the metrological domain rather than that of the underlying numerical technology. Interface modules will allow the metrologist to use the key functions in the preferred computing environment, for example, as part of a spreadsheet calculation. EUROMETROS will both make existing software visible and accessible, and promote the development of new functions, algorithms and implementations to meet new requirements. As part of the SS*f*M programme, NPL have developed procedures for the testing and validation of algorithms and software contributed to EUROMETROS [10], which will help ensure the quality of contributions.

1.4 General notation

See table 1.1.

[2] `www.eurometros.org`

$\sharp$	denotes the end of text concerning an example.
m	number of measurements.
n	number of model parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$.
$\mathbf{a}$	vector of model parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$.
$N(\mu, \sigma^2)$	univariate Gaussian or normal distribution with mean μ and standard deviation σ.
$U(a, b)$	rectangular (uniform) distribution, constant on $[a, b]$ and zero outside this interval.
p	number of model variables $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$.
$\mathbf{x}$	vector of model variables $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$.
$\mathbb{R}$	the set of real numbers.
$\mathbb{R}^n$	the set of n-vectors $\mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$ of real numbers.
$\{x_i\}_1^m$	set of m elements indexed by $i = 1, 2, \ldots, m$.
$\mathbf{z}$	data set of measurements, often comprising of data points $\{(\mathbf{x}_i, y_i)\}_1^m$ each stored as a column vector.
$[a, b]$	set of numbers $\{x : a \leq x \leq b\}$.
$\mathbf{a}^{\mathrm{T}}$, A^{T}	transpose of a vector or matrix.
$\mathbf{a}^{\mathrm{T}}\mathbf{b}$	inner product of two vectors $\mathbf{a}^{\mathrm{T}}\mathbf{b} = a_1 b_1 + \ldots + a_n b_n$.
I	identity matrix with 1s on the diagonal and 0s elsewhere.
J	Jacobian matrix associated with a set of functions $f_i(\mathbf{a})$ of parameters: $J_{ij} = \frac{\partial f_i}{\partial a_j}$.
$\mathcal{A}(\mathbf{z})$	parameter estimate determined from data $\mathbf{z}$ by estimator $\mathcal{A}$.
$D(\boldsymbol{\alpha})$	Distribution with parameters from the vector $\boldsymbol{\alpha}$ (section 2.4.2).
$D(\mathbf{u})$	Generalised distance (section 4.3.2).
$E(\boldsymbol{X})$	expectation of the vector of random variables $\boldsymbol{X}$.
$V(\boldsymbol{X})$	variance or uncertainty matrix of the vector of random variables $\boldsymbol{X}$.
$u(x)$	standard uncertainty associated with the estimate, x, of a random variable.
$\mathrm{std}_i\{x_i\}$	standard deviation of the data set $\{x_i\}_{i=1}^m$.
D, E	scaling matrices (section 2.3.1), random variables (section 2.4.5); and see $D(\boldsymbol{\alpha})$, $D(\mathbf{u})$, $E(\boldsymbol{X})$ above.
X, Y, etc.,	random variables.
$\nabla_{\mathbf{a}} f$	vector of partial derivatives $(\frac{\partial f}{\partial a_1}, \ldots, \frac{\partial f}{\partial a_n})^{\mathrm{T}}$ for a function $f(\mathbf{a})$ with respect to the parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$.
$\sum_{i=1}^{m} x_i$	sum of elements: $\sum_{i=1}^m x_i = x_1 + \ldots + x_m$.
$\prod_{i=1}^{m} x_i$	product of elements: $\prod_{i=1}^m x_i = x_1 \times \ldots \times x_m$.

Table 1.1: General notation used in this Guide.

Chapter 2

Model building

2.1 Model types

Mathematical modelling, in general, involves the assignment of mathematical terms for all the relevant components of a (measurement) system and the derivation of equations giving the relationships between these mathematical entities. In these models, we can distinguish between terms that relate to quantities that are known or measured and those that are unknown or to be determined from the measurement data. We will in general call the former terms *model variables* and use $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$, $\mathbf{y}$, etc., to denote them and call the latter *model parameters* and denote them by $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$, $\mathbf{b}$, etc.

A *physical model* is one in which there is a theory that defines how the variables depend on each other.

An *empirical model* is one in which a relationship between the variables is expected or observed but with no supporting theory. Many models have both empirical and physical components.

An *explicit model* is one in which one or more of the variables is given as a directly computable function of the remaining variables. We write $y = \phi(\mathbf{x}, \mathbf{a})$ to show that y is a function of the model variables $\mathbf{x}$ and parameters $\mathbf{a}$. If $\mathbf{x}$ and $\mathbf{a}$ are known, then the corresponding value for y can be calculated. The variable y is known as the *response* or *dependent* variable and the variables $\mathbf{x}$ are known as the *covariates*, *stimulus* or *explanatory variables*.[1]

An *implicit model* is one in which the variables are linked through a set of equations. We write, for example, $f(\mathbf{x}, \mathbf{a}) = 0$ to show that the components of $\mathbf{x}$ are related implicitly. It is often possible to write one variable as a function of the others, e.g.,

$$x_1 = \phi_1(x_2, \ldots, x_p, \mathbf{a}).$$

[1] The term *independent variable* is sometimes used but the use of the word 'independent' can be confused with the notion of statistical independence.

Example: implicitly and explicitly defined circle

The equation for a circle centred at (a_1, a_2) with radius a_3 can be written implicitly as

$$f(\mathbf{x}, \mathbf{a}) = (x_1 - a_1)^2 + (x_2 - a_2)^2 - a_3^2 = 0.$$

We can solve for x_1 explicitly in terms of x_2,

$$x_1 = a_1 \pm \sqrt{[a_3^2 - (x_2 - a_2)^2]},$$

or for x_2 in terms of x_1,

$$x_2 = a_2 \pm \sqrt{[a_3^2 - (x_1 - a_1)^2]}.$$

We can rewrite these two equations in parametric form $\mathbf{x} = \boldsymbol{\phi}(u, \mathbf{a})$ as

$$\begin{aligned} (x_1, x_2) &= (a_1 \pm \sqrt{[a_3^2 - (u - a_2)^2]}, u), \quad \text{or} \\ (x_1, x_2) &= (u, a_2 \pm \sqrt{[a_3^2 - (u - a_1)^2]}). \end{aligned}$$

The first equation becomes problematical when $|x_2 - a_2| \approx a_3$ while the second when $|x_1 - a_1| \approx a_3$. It is often the case that when going from an implicit expression to an explicit expression there is a preferred choice (depending on the particular circumstances) and that some choices are excluded because the equations become singular in some way. Sometimes an implicit form is preferable even when an explicit form can be deduced from it because the former has better numerical stability.

Alternatively, we can express the circle parametrically $\mathbf{x} = \boldsymbol{\phi}(u, \mathbf{a})$ as

$$x_1 = a_1 + a_3 \cos u, \quad x_2 = a_2 + a_3 \sin u.$$

This form is valid for all values of u. ♯

A *linear model* is one in which the parameters $\mathbf{a}$ appear linearly. For explicit models, it takes the form

$$y = \phi(\mathbf{x}, \mathbf{a}) = a_1\phi_1(\mathbf{x}) + \ldots + a_n\phi_n(\mathbf{x}),$$

where the functions $\phi_j(\mathbf{x})$ are *basis functions* depending on the variables $\mathbf{x}$.

A *nonlinear model* is one in which one or more of the parameters $\mathbf{a}$ appear nonlinearly.

Example: exponential decay

The function

$$y = a_1 e^{-a_2 x}$$

is an example of a nonlinear (explicit) model since the parameter a_2 appears nonlinearly. ♯

Many (but by no means all) of the models that occur in practice such as polynomials (section 5.1) and splines (section 5.2) are linear. They have the advantage that when it comes to determining best estimates of the model parameters from data (chapter 3) the equations that arise are easier to solve.

2.2 Space of models

Consider an experimental set up in which a response variable y depends on a number of covariates $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$. We make the assumption that the system is deterministic in that the same values of the variables gives rise to the same response, i.e., if $\mathbf{x}_1 = \mathbf{x}_2$ then correspondingly $y_1 = y_2$. With this assumption, we can say that the response y is a *function* of the variables $\mathbf{x}$ and write

$$y = \phi(\mathbf{x}),$$

to denote this relationship. If we assume that the response y depends continuously on each of the variables x_k, then we can restrict the choices for ϕ to be continuous functions. Further assumptions will in turn limit the possible choices for ϕ.

The goal of the modelling process is to include enough information about the system so that the range of choices for the function ϕ is determined by specifying a finite number of additional parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$. Each set of values of these parameters determines uniquely a response function $y = \phi(\mathbf{x}, \mathbf{a})$. We call the collection of all such functions $\phi(\mathbf{x}, \mathbf{a})$ the *space of models*. Ideally, the actual response function ϕ is specified by one such function $\phi(\mathbf{a}, \mathbf{x})$, i.e., the space of models is large enough to model the actual behaviour. On the other hand we do not want the space of models to include functions that represent system behaviour that is physically impossible, i.e., the space of models should not be too large.

Example: linear response

One of the most common types of model is one in which the response variable depends linearly on a single variable x:

$$y = \phi(x, a_1, a_2) = a_1 + a_2 x,$$

specified by intercept a_1 and slope a_2. Here the space of models is the collection of linear functions $\{y = a_1 + a_2 x\}$. ♯

The term *linear response* model should not be confused with a linear model (defined in section 2.1), although linear response models are linear because $a_1 + a_2 x$ is linear in (a_1, a_2).

Example: exponential decay

Suppose the response y is an exponential decay depending on the single variable x (time, say). Then y can be modelled as

$$y = \phi(x, a_1, a_2) = a_1 e^{-a_2 x}$$

depending on two parameters a_1 and a_2. Here, the space of models is the collection of functions $\{y = a_1 e^{-a_2 x}\}$. ♯

Example: circles

In dimensional metrology, the nominal shape of the cross section of a shaft is modelled as a circle. A circle (in a given Cartesian co-ordinate system) can be

specified by three parameters, its centre coordinates (a_1, a_2) and radius a_3. To each set of parameters (a_1, a_2, a_3), we associate the circle

$$\{(x, y) : (x - a_1)^2 + (y - a_2)^2 = a_3^2\}.$$

♯

Example: water density

A number of models for the density of water y as a function of temperature x have been proposed, e.g.

$$\begin{aligned}
\frac{y}{y_0} &= \phi_1(x, a_1, \ldots, a_4) = 1 - \frac{a_2(x - a_1)^2(x + a_3)}{x + a_4}, \\
\frac{y}{y_0} &= \phi_2(x, a_1, \ldots, a_6) = 1 - \frac{a_2(x - a_1)^2(x + a_3)(x + a_5)}{(x + a_4)(x + a_6)}, \\
\frac{y}{y_0} &= \phi_3(x, a_1, \ldots, a_6) = 1 - \sum_{j=1}^{5} a_{j+1}(x - a_1)^j, \\
\frac{y}{y_0} &= \phi_4(x, a_1, \ldots, a_9) = 1 - \sum_{j=1}^{9} a_j x^{j-1},
\end{aligned}$$

where y_0 represents the maximum density. These models are empirical in that there is no theory to define exactly the space of models. Note that the number of parameters (4, 6, 6 and 9) used to specify the functions differs from model to model. This is often the case with empirical models. ♯

In some sense, the essence of model building is being able to define the right number and type of parameters that are required to characterise the behaviour of the system adequately.

2.3 Model parameterisation

Model parameterisation is concerned with how we specify members of the space of models. Given a space of models, a *parameterisation* assigns to a set of values of the parameters $\mathbf{a}$ a unique member of the space of models, e.g., a particular curve from a family of curves.

Example: straight lines

The equation

$$L_1 : (a_1, a_2) \mapsto \{y = a_1 + a_2 x\}$$

associates to the pair of parameters (a_1, a_2) the linear function $y = a_1 + a_2 x$. Consider, also,

$$L_2 : (a_1, a_2) \mapsto \{y = a_1 + a_2(x - 100)\}.$$

These two methods are mathematically equivalent in the sense that given any pair (a_1, a_2) it is possible to find a unique pair (a_1', a_2') such that L_2 assigns the

same line to (a_1', a_2') as L_1 assigns to (a_1, a_2), and vice versa. From a numerical point of view, the parameterisation L_2 may be preferable if the variable x is likely to have values around 100. However, the parameterisation

$$L_3 : (a_1, a_2) \mapsto \{x = a_1 + a_2 y\}$$

is not equivalent to L_1 since there is no pair (a_1, a_2) that L_3 can assign to the line $y = 0$. Similarly, L_1 cannot represent the line $x = 0$.

Note that the parameterisation

$$L_4 : (a_1, a_2) \mapsto \{-x \sin a_1 + y \cos a_1 = a_2\}$$

can be used to represent all lines. ♯

Example: circles

The assignment

$$C_1 : (a_1, a_2, a_3) \mapsto \{(x, y) : (x - a_1)^2 + (y - a_2)^2 = a_3^2\},$$

parameterises circles in terms of their centre coordinates and radius. Consider also

$$\begin{aligned} C_2 : (a_1, a_2, a_3) &\mapsto \{(x, y) : x^2 + y^2 + a_1 x + a_2 y + a_3 = 0\}, \\ C_3 : (a_1, a_2, a_3) &\mapsto \{(x, y) : a_1(x^2 + y^2) + a_2 x + y = a_3\}. \end{aligned}$$

The parameterisations C_1 and C_2 are equivalent to each other in that they can represent exactly the same set of circles but not to C_3. The parameterisation C_3 can be used to model arcs of circle approximately parallel to the $x-$axis in a stable way. Indeed, lines (in this context, circles with infinite radius) correspond to circles with $a_1 = 0$ in this parameterisation. ♯

2.3.1 Centring and scaling

Model parameterisations that are equivalent from a mathematical point of view may have different characteristics numerically. For example, we can scale or translate the variables and parameters and still define the same model space.

Example: variable transformation for a quadratic curve

Suppose in an experiment, the response y is modelled as a quadratic function of the variable x,

$$y = a_1 + a_2 x + a_3 x^2,$$

and x is expected to lie in the range $[95,\ 105]$. Using this equation, the quadratic curves are specified by the coefficients a_1, a_2 and a_3. We can instead parameterise these curves in terms of the transformed variable z

$$y = b_1 + b_2 z + b_3 z^2,$$

where $z = (x - 100)/5$ is expected to lie in the range $[-1,\ 1]$. ♯

More generally, given a model of the form $y = \phi(\mathbf{x}, \mathbf{a})$, we can reparameterise it as $y = \phi(\mathbf{z}, \mathbf{b})$ where

$$\mathbf{z} = D(\mathbf{x} - \mathbf{x}_0), \quad \mathbf{b} = E(\mathbf{a} - \mathbf{a}_0),$$

and D and E are $p \times p$ and $n \times n$ nonsingular scaling matrices and $\mathbf{x}_0$ and $\mathbf{a}_0$ fixed p- and n-vectors. Typically, we set $\mathbf{x}_0$ to be the centroid of the data:

$$\mathbf{x}_0 = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i,$$

$\mathbf{a}_0$ to be middle of the expected range for the parameters $\mathbf{a}$ and set the scaling matrices such that

$$\frac{\partial \phi}{\partial z_k}, \frac{\partial \phi}{\partial b_j} \approx \pm 1 \quad \text{near} \quad \mathbf{z} = \mathbf{0}, \quad \mathbf{b} = \mathbf{0}.$$

These transformations will generally improve the numerical performance of algorithms operating with the model. Often, the improvements are very significant.

2.3.2 Choice of basis functions

Suppose we have a linear model defined in terms of the basis functions ϕ_j as

$$y = \phi(\mathbf{x}, \mathbf{a}) = a_1 \phi_1(\mathbf{x}) + \ldots + a_n \phi_n(\mathbf{x}).$$

Given a nonsingular $n \times n$ matrix D whose jth column is $\mathbf{d}_j$, we can define new basis functions ψ_j according to

$$\psi_j(\mathbf{x}) = d_{1j} \phi_1(\mathbf{x}) + \ldots + d_{nj} \phi_n(\mathbf{x}),$$

and reparameterise the model as

$$y = \psi(\mathbf{x}, \mathbf{b}) = b_1 \psi_1(\mathbf{x}) + \ldots + b_n \psi_n(\mathbf{x}),$$

in order to improve the stability of the model. Such considerations are particularly important for polynomial or spline models (sections 5.1, 5.2).

2.3.3 Resolving constraints

Often the natural 'parameters' used to describe a model give rise to degrees of freedom that need to be resolved.

Example: parameters describing the geometry of targets on a planar artefact

In dimensional metrology, artefacts such as a hole plate can be modelled as a set of targets lying in a plane. The location of these targets can be described by their coordinates $\mathbf{a} = (a_1, b_1, a_2, b_2, \ldots, a_n, b_n)^{\mathrm{T}}$ where $\mathbf{a}_j = (a_j, b_j)^{\mathrm{T}}$ is the location of the jth target. However, the parameters $\mathbf{a}$ do not specify the frame of reference for the targets and three constraints have to be introduced to fix

the three degrees of freedom (two translational and one rotational) associated with the system.

For example, suppose there are four points nominally at the corners of a square. We can eliminate the translational degrees of freedom by constraining the centroid $(\bar{a}, \bar{b})^{\mathrm{T}}$ to be at $(0, 0)^{\mathrm{T}}$:

$$\bar{a} = \frac{1}{n}\sum_{j=1}^{n} a_j = 0, \quad \bar{b} = \frac{1}{n}\sum_{j=1}^{n} b_j = 0, \quad \text{where } n = 4 \text{ for a square}$$

Similarly, we can fix the orientation of the targets by constraining one of the targets to lie on the line $y = x$: $a_1 = b_1$, say. These three constraints can be written in the form

$$D\mathbf{a} = \mathbf{0},$$

where D is the 3×8 matrix

$$D = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 4 & -4 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

♯

2.4 Uncertainty structure in measurement data

In this section, we review briefly some of the statistical concepts used to represent our uncertainty or degree of belief in measurement data. See, for example, [71, 72, 139].

2.4.1 Probability

The probability $P(A)$ of a statement (proposition, event) A is a real number between 0 and 1, with 0 meaning the statement must be false and 1 that it must be true. The larger the probability, the more likely the statement is to be true. The probability of A and B being true is denoted by $P(A, B)$. The notation $P(A|B)$ means the probability of A given that statement B is true. There are two basic rules that define how probabilities are combined.[2] If $\bar{A}$ represents the statement 'A is false' then

$$P(A) + P(\bar{A}) = 1.$$

This is called the *sum rule*. The *product rule* states that

$$P(A, B) = P(A|B) \times P(B) = P(B|A) \times P(A),$$

in words, the probability that both A and B are true is equal to the probability that A is true given that B is true times the probability that B is true. Two

[2]The work of R. T. Cox [80] showed that these rules are essentially unique and that any useful theory of probability would have to obey them.

statements A and B are independent if $P(A|B) = P(A)$ and $P(B|A) = P(B)$, i.e., the probability of one being true does not depend on our knowledge of the other. For independent A and B, the product rule is $P(A, B) = P(A)P(B)$.

Bayes' Theorem arises from a rearrangement of the product rule:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}. \tag{2.1}$$

If we regard A as a statement about parameters and B as a statement about measurement data, Bayes' Theorem allows us to update our original information $P(A)$ about A in the light of the measurements B; see section 4.11.

2.4.2 Random variables and distributions

We use a random variable X to represent a quantity about which we have uncertain knowledge. The quantity may be discrete, for example, the number of pills in a bottle taken from a production line in a pharmaceutical plant, or continuous, for example, the volume of liquid in a medicine bottle from another production line. We associate to a random variable X a probability distribution which allows us to assign probabilities to statements about X.

Discrete random variables

A *discrete random variable* X is a variable that can take only a finite number of possible values. The *frequency function* $p(x)$ states the probabilities of occurrence of the possible outcomes:

$$p(x) = P(X = x),$$

the probability that the outcome is x. The *distribution function* $G(x)$ gives the probability that a random variable takes a value no greater than a specified value:

$$G(x) = P(X \leq x), \qquad -\infty < x < \infty.$$

The distribution function varies from zero to one throughout its range, never decreasing.

Continuous random variables

A *continuous random variable* X is a variable that can take any value in its range (which may be infinite). For a continuous random variable X, the counterpart of the frequency function (for a discrete random variable) is the *probability density function* (PDF) $g(x)$. This function has the property that the probability that the value of X lies between a and b is

$$P(a < X < b) = \int_a^b g(x)\, dx.$$

In order that the sum rule is obeyed, PDFs must have unit area, i.e.,

$$P(-\infty < X < \infty) = \int_{-\infty}^{\infty} g(x)\,dx = 1.$$

For example, the rectangular PDF is a density function that describes the fact that the value of X is equally likely to lie anywhere in an interval $[a,\ b]$:

$$g(x) = \begin{cases} \frac{1}{b-a}, & a \le x \le b, \\ 0, & \text{otherwise.} \end{cases}$$

We use the notation $X \sim U(a, b)$ to indicate that X has a rectangular distribution defined on the interval $[a, b]$.

The *distribution function* $G(x)$ gives the probability that a random variable takes a value no greater than a specified value, and is defined as for a discrete random variable:

$$G(x) = P(X \le x), \qquad -\infty < x < \infty.$$

The distribution function can be expressed in terms of the probability density function as

$$G(x) = \int_{-\infty}^{x} g(t)\,dt.$$

Conversely, $g(x) = G'(x)$, the derivative of G. A continuous probability distribution can therefore be defined in terms of either the distribution function G or the probability density function g.

A function $Y = h(X)$ of a random variable X is also random variable and its distribution is determined by h and the PDF of X.

Probability density functions used in practice are usually determined by a small number of parameters. One of the most important distributions is the *normal* or *Gaussian distribution* whose PDF is

$$g(x) = g(x|\mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ -\frac{1}{2}\left(\frac{x-\mu}{\sigma} \right)^2 \right\}.$$

We use the notation $X \sim N(\mu, \sigma^2)$ to indicate X is a random variable associated with a normal distribution defined by parameters μ and σ. More generally, $X \sim D(\boldsymbol{\alpha})$ means that X is associated with a probability distribution D whose PDF is defined in terms of parameters $\boldsymbol{\alpha}$.

A vector $\boldsymbol{X} = (X_1, \ldots, X_n)^{\mathrm{T}}$ of random variables has a multivariate distribution defined in terms of a nonnegative multivariate function $g(\mathbf{x})$. Two random variables $(X, Y)^{\mathrm{T}}$ are *independently distributed* if the associated PDF $g(x, y)$ can be factored as $g(x, y) = g_x(x)g_y(y)$.

2.4.3 Operations on distributions

Measures of location and dispersion

For a given distribution, it is usual to calculate, if possible, quantities that provide a useful summary of its properties. A measure of location $L(X)$ is such that $L(X + c) = L(X) + c$ and is used to determine a representative value for X. A measure of dispersion (spread) $S(X)$ is such that $S(cX) = cS(X)$ and gives an estimate of the size of the likely range of values of X.

Expectations

Summarising quantities are often derived in terms of expectations.
If $X \sim D$ has associated PDF $g(x)$ and $h(X)$ is a function of X, then the expectation $E(h(X))$ of $h(X)$ is

$$E(h(X)) = \int_{-\infty}^{\infty} h(x)g(x)\,dx.$$

(If may be that this integral is not finite in which case $E(h(X))$ is said not to exist.)

Mean, variance and standard deviation

Of particular importance are the *mean* $\mu = E(X)$,

$$\mu = \int_{-\infty}^{\infty} xg(x)\,dx,$$

and the *variance* $V(X) = E((X - E(X))^2)$:

$$V(X) = \int_{-\infty}^{\infty} (x-\mu)^2 g(x)\,dx, \quad \mu = E(X).$$

The positive square root of the variance is known as the *standard deviation* and is usually denoted by σ so that $\sigma^2 = V(X)$. The mean is a measure of location of X and the standard deviation is a measure of dispersion. We note that if $X \sim N(\mu, \sigma^2)$ then $E(X) = \mu$ and $V(X) = \sigma^2$. If X has a rectangular distribution, $X \sim U(a, b)$, then $E(X) = (a + b)/2$ and $V(X) = (b - a)^2/3$.

Expectations can also be applied to multivariate distributions. For example, the *covariance* $C(X, Y)$ of a pair (X, Y) of random variables with joint PDF $g(x, y)$ is defined to be

$$\begin{aligned} C(X,Y) &= E((X - E(X))(Y - E(Y))), \\ &= \int (x-\mu_X)(y-\mu_Y)g(x,y)\,dx\,dy, \\ \mu_X &= E(X) = \int xg(x,y)\,dx\,dy, \quad \mu_Y = E(Y) = \int yg(x,y)\,dx\,dy, \end{aligned}$$

and $V(X) = C(X, X)$. If $\boldsymbol{X} = (X_1, \ldots, X_n)^{\mathrm{T}}$ is a vector of random variables then the *variance matrix* $V(\boldsymbol{X})$, also known as the *uncertainty matrix* , *covariance* or *variance-covariance matrix*, is the $n \times n$ matrix with $V_{ij} = C(X_i, X_j)$.

Example: multivariate normal (Gaussian) distribution

The multivariate normal (Gaussian) distribution for n variables $N(\boldsymbol{\mu}, V)$ is defined by its mean $\mu = (\mu_1, \ldots, \mu_n)^{\mathrm{T}}$ and $n \times n$ variance matrix V and has PDF

$$p(\mathbf{x}|\boldsymbol{\mu}, V) = \frac{1}{|2\pi V|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} V^{-1} (\mathbf{x} - \boldsymbol{\mu})\right\}, \tag{2.2}$$

where $|V|$ denotes the determinant of V. ♯

The importance of the mean and variance are that there are simple rules for calculating them for linear combinations of random variables. If X and Y are random variables and c and d are two constants, then

$$\left.\begin{array}{rcl} E(cX + dY) & = & cE(X) + dE(Y), \\ V(cX + dY) & = & c^2 V(X) + d^2 V(Y) + 2cdC(X, Y) \end{array}\right\}. \tag{2.3}$$

2.4.4 Propagation of uncertainties

The law of the propagation of uncertainties (LPU), see [24] and [71, chapter 6] is rooted in the rules for the means and variances expressed in (2.3). Suppose first that Y is a linear combination of n random variables $X = (X_1, \ldots, X_n)^{\mathrm{T}}$,

$$Y = c_1 X_1 + \ldots + c_n X_n = \mathbf{c}^{\mathrm{T}} \boldsymbol{X}, \tag{2.4}$$

where $\mathbf{c} = (c_1, \ldots, c_n)^{\mathrm{T}}$ are known constants. Suppose that X_j are associated with distributions with means x_j and standard deviations $u_j = u(x_j)$ and that the X_j are independently distributed. A simple extension of (2.3) shows that Y is associated with a distribution with mean

$$y = E(Y) = c_1 E(X_1) + \ldots + c_n E(X_n) = c_1 x_1 + \ldots + c_n x_n,$$

and variance

$$u^2(y) = V(Y) = c_1^2 V(X_1) + \ldots + c_n^2 V(X_n) = c_1^2 u_1^2 + \ldots + c_n^2 u_n^2.$$

This is true no matter the distributions for X_j (so long as their means and standard deviations are defined).

Now suppose Y is defined as a function $Y = f(X_1, \ldots, X_n)$ with the X_j distributed as before. The random variable Y is associated with a distribution and we wish to know its mean and standard deviation. We can find an approximate answer by linearising the function Y about $y = f(\mathbf{x})$. In (2.4) the constant c_j represents the *sensitivity* of Y with respect to changes in X_j: if X_j changes by Δ_j then Y changes by $c_j \Delta_j$. For a nonlinear function f, the sensitivity of Y with respect to a change in X_j is given by the partial derivative $c_j = \partial f / \partial X_j$ evaluated at x_j. (This partial derivative is simply the slope at $X_j = x_j$ of the

function f regarded as a function of X_j alone with all other variables held fixed.) The linear approximation can then be written as

$$Y - y \approx c_1(X_1 - x_1) + \ldots + c_n(X_n - x_n),$$

or

$$\tilde{Y} \approx c_1\tilde{X}_1 + \ldots + c_n\tilde{X}_n, \tag{2.5}$$

with new random variables $\tilde{Y} = Y - y$ and $\tilde{X}_j = X_j - x_j$, $j = 1, \ldots, n$.

Equation (2.5) is of the same form as (2.4) and so

$$E(Y - y) = E(Y) - y \approx c_1(E(X_1) - x_1) + \ldots + c_n(E(X_n) - x_n) = 0,$$

i.e., $E(Y) \approx y$, and

$$\begin{aligned} u_y^2 &= V(Y - y) = V(Y) \\ &\approx c_1^2 V(X - x_1) + \ldots + c_n^2 V(X_n - x_n) \\ &= c_1^2 u_1^2 + \ldots c_n^2 u_n^2. \end{aligned}$$

Here, we have used the rule $V(X - x) = V(X)$. In summary, for nonlinear functions $Y = f(\boldsymbol{X})$ we use the same rule as for linear functions but with the sensitivities c_j calculated as partial derivatives. We must be aware, however, that the resulting estimate of the standard deviation is derived from a linearisation and may be different from the actual value (section 7.2).

2.4.5 Measurement model

The space of models represents the mathematical relationship between the various variables and parameters. In practice, the values of the variables are inferred from measurements subject to random effects, that are difficult to characterise. These effects are modelled as random variables, generally with expectation zero. The actual measured values are regarded as observations of the associated random variable drawn from the corresponding statistical distribution.

Suppose that the response y is modelled as a function $y = \phi(\mathbf{x}, \mathbf{a})$ depending on variables $\mathbf{x}$ and model parameters $\mathbf{a}$ and that measurements of y are subject to random effects. The measurement model is of the form

$$Y = \phi(\mathbf{x}, \mathbf{a}) + E, \qquad E(E) = 0.$$

We note that since $E(E) = 0$, $E(Y) = \phi(\mathbf{x}, \mathbf{a})$, i.e., the value of $\phi(\mathbf{x}, \mathbf{a})$ predicted by the model $\phi(\mathbf{x}, \mathbf{a})$ is equated with the expected value of the random variable Y. Suppose measurements y_i are gathered with $y_i \in Y$, i.e., y_i is an observation of the random variable Y_i where

$$Y_i = \phi(\mathbf{x}_i, \mathbf{a}) + E_i, \qquad E(E_i) = 0.$$

We can then write

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \qquad \epsilon_i \in E_i,$$

where $\epsilon_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$ represents the observed value of the random variable E_i and can be thought of the *deviation* between the measurement value and the model prediction. (In data approximation, we sometimes refer to ϵ_i as the *approximation* or *residual error*.)

In many situations the measurements of two or more variables are subject to significant random effects. In this case the measurement model has a more general form such as

$$\begin{aligned} &\boldsymbol{X} = \mathbf{x}^* + \boldsymbol{D}, \quad E(\boldsymbol{D}) = \mathbf{0}, \quad E(\boldsymbol{X}) = \mathbf{x}^*; \\ &Y = \phi(\mathbf{x}^*, \mathbf{a}) + E, \quad E(E) = 0. \end{aligned}$$

Measurements $(\mathbf{x}_i, y_i)$ are regarded as observations of the random variables $(\boldsymbol{X}, Y)$, $i = 1, \ldots, m$, and we write

$$y_i = \phi(\mathbf{x}^* + \boldsymbol{\delta}_i, \mathbf{a}) + \epsilon_i, \quad i = 1, \ldots, m,$$

with $\boldsymbol{\delta} \in \boldsymbol{D}$ and $\epsilon_i \in E$.

For implicit models $f(\mathbf{x}, \mathbf{a}) = 0$, the corresponding model equations are written as

$$\begin{aligned} &\boldsymbol{X} = \mathbf{x}^* + \boldsymbol{E}, \quad E(\boldsymbol{E}) = \mathbf{0}, \quad E(\boldsymbol{X}) = \mathbf{x}^*; \\ &f(\mathbf{x}^* + \boldsymbol{\epsilon}_i, \mathbf{a}) = 0, \ \boldsymbol{\epsilon}_i \in \boldsymbol{E} \quad i = 1, \ldots, m. \end{aligned}$$

Example: refractive index of air

The refractive index of air is modelled as a function of air temperature, pressure and humidity (and other variables such as carbon dioxide content) with all three subject to significant random effects. ♯

2.4.6 Statistical models for random effects

The uncertainty structure has to describe not only which measurements are subject to random effects but also the statistical nature of these effects. Measurements $\mathbf{z} = (z_1, \ldots, z_m)^{\mathrm{T}}$ are regarded as observations associated with random variables $\boldsymbol{Z} = (Z_1, \ldots, Z_m)^{\mathrm{T}}$ and the statistical model is described by information about the multivariate statistical distribution for $\boldsymbol{Z}$. Often the information about the multivariate PDF is summarised in terms of the mean $E(\boldsymbol{Z})$ and variance (uncertainty) matrix $V(\boldsymbol{Z})$ rather than specifying the complete PDF.

If measurement z_i is associated with random variable Z_i, then the *standard uncertainty* $u(z_i)$ associated with z_i is the standard deviation of Z_i, i.e.,

$$u^2(z_i) = V(Z_i) = (V(\boldsymbol{Z}))_{ii},$$

the ith diagonal element of uncertainty matrix $V(\boldsymbol{Z})$.

Example: standard experiment model

We will refer to the following model as the *standard experiment model*. A response variable y is modelled as a function $y = \phi(\mathbf{x}, \mathbf{a})$ of variables $\mathbf{x}$ and

parameters $\mathbf{a}$ and a set $\{(\mathbf{x}_i, y_i)\}_1^m$ of measurements gathered with each y_i subject to independent random effects described by a normal distribution with zero mean and standard deviation σ. The model equations are

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \qquad i = 1, \ldots, m,$$

with $\boldsymbol{\epsilon} \in N(\mathbf{0}, \sigma^2 I)$. These equations represent a complete statement of the model of the experiment.

The standard uncertainty $u(y_i)$ associated with y_i is σ. ♯

There are common variations in this standard model. For example, the standard uncertainties may vary with the measurements, in which case $\epsilon_i \in N(0, \sigma_i^2)$. If the random variables E_i are correlated, with uncertainty matrix V, the vector $\boldsymbol{\epsilon}$ is modelled as belonging to a multinormal distribution: $\boldsymbol{\epsilon} \in N(\mathbf{0}, V)$.

There is further guidance on statistical modelling for experimental error models in [72].

Chapter 3

Model solving and estimators

3.1 Approximation from a space of models

The space of models attempts to characterise all possible (or probable) behaviour of a particular type of system, e.g., the ways in which a response variable could vary with its covariates. Model solving is the process of determining from data gathered from a measurement system, a particular model that adequately represents the system behaviour. Constructing the model space is concerned with defining where we should look to explain the behaviour; model solving is concerned with selecting the best candidate from the options defined by the model space.

If the members of the model space are described by parameters $\mathbf{a}$ and the measurement data $\mathbf{z}$ is regarded as being generated by a system specified by parameters $\mathbf{a}^*$, then model solving amounts to providing an estimate of $\mathbf{a}^*$ from $\mathbf{z}$. A scheme for determining such an estimate from data we term an *estimator*. We use the symbols $\mathcal{A}$, $\mathcal{B}$, etc., to denote estimators; $\mathbf{a} = \mathcal{A}(\mathbf{z})$ means the estimate of the model parameters provided by estimator $\mathcal{A}$ from data $\mathbf{z}$.

3.2 Error functions and approximation norms

In general, estimators are defined using an *error function* $F(\mathbf{a}|\mathbf{z})$ that provides some measure of how well the data $\mathbf{z}$ matches the model behaviour specified by $\mathbf{a}$. The estimate of $\mathbf{a}^*$ is provided by (the estimate of) the minimiser of $F(\mathbf{a}|\mathbf{z})$, i.e., a point at which F takes a minimum value. Different estimators are associated with different error functions.

Error functions are usually constructed to provide an aggregate measure of goodness of fit taking into account all the measurement data. These error functions

are often related to approximation norms and the least-squares estimator is one of a family of estimators derived from such norms.

Example: approximation norms

In a standard experiment with model $y = \phi(\mathbf{x}, \mathbf{a})$ and data $\mathbf{z} = \{(\mathbf{x}_i, y_i)\}_1^m$, the quantity

$$f_i = f_i(\mathbf{x}_i, \mathbf{a}) = y_i - \phi(\mathbf{x}_i, \mathbf{a})$$

is a measure of the deviation of the model specified by $\mathbf{a}$ from the data point $(\mathbf{x}_i, y_i)$. An aggregate measure of the fit is given by a norm of the vector $\mathbf{f} = (f_1, \ldots, f_m)^{\mathrm{T}}$ such as the p-norm

$$F_p(\mathbf{a}|\mathbf{z}) = \|\mathbf{f}\|_p = \left\{ \sum_{i=1}^{m} |f_i|^p \right\}^{1/p},$$

for any prescribed value of p satisfying $1 \leq p \leq \infty$. In this guide, the p-norm is denoted by L_p; elsewhere ℓ_p is also used for the discrete case.

Of particular importance are the L_1-norm

$$F_1(\mathbf{a}|\mathbf{z}) = \sum_{i=1}^{m} |f_i|,$$

the L_2-norm (least squares)

$$F_2(\mathbf{a}|\mathbf{z}) = \left\{ \sum_{i=1}^{m} f_i^2 \right\}^{1/2},$$

and the L_∞ or Chebyshev norm

$$F_\infty(\mathbf{a}|\mathbf{z}) = \max_{1 \leq i \leq m} |f_i|.$$

♯

3.3 Estimator properties

Suppose that an experimental system is specified by parameters $\mathbf{a}^*$, measurements $\mathbf{z}$ have been gathered, resulting in parameter estimates $\mathbf{a} = \mathcal{A}(\mathbf{z})$. Regarding $\mathbf{z}$ as a set of observations of a vector of random variables $\boldsymbol{Z}$ with multivariate PDF $p(\mathbf{x})$, then $\mathbf{a}$ is an observation of the vector of random variables $\boldsymbol{A} = \mathcal{A}(\boldsymbol{Z})$. In principle, the PDF $g_{\mathcal{A}}$ associated with $\boldsymbol{A}$ is determined by that for $\boldsymbol{Z}$, and has a mean $E(\boldsymbol{A})$ and variance $V(\boldsymbol{A})$. We would like $g_{\mathcal{A}}$ to be concentrated in a region close to $\mathbf{a}^*$ so that the probability of observing an estimate $\mathcal{A}(\mathbf{z})$ that is is close to $\mathbf{a}^*$ is high. One measure of how good an estimator is given by the *mean squared error* (MSE) defined by

$$\mathrm{MSE}(\mathcal{A}) = E((\boldsymbol{A} - \mathbf{a}^*)^2) = \int (\mathbf{x} - \mathbf{a}^*)^2 g_{\mathcal{A}}(\mathbf{x})\, d\mathbf{x} = \int (\mathcal{A}(\mathbf{x}) - \mathbf{a}^*)^2 p(\mathbf{x})\, d\mathbf{x},$$

and the *root mean squared error*, $\mathrm{RMSE}(\mathcal{A}) = (\mathrm{MSE})^{1/2}$. The RMSE is a measure of the likely distance of the estimate from $\mathbf{a}^*$. An estimate $\mathcal{A}$ is *unbiased* if $E(\boldsymbol{A}) = \mathbf{a}^*$, in which case $\mathrm{MSE}(\mathcal{A}) = V(\boldsymbol{A})$. An unbiased estimator with a small variance is *statistically efficient*. Efficiency is used in a relative sense to compare estimators with each other (or with certain theoretical bounds; see e.g., [150, chapter 4]). The MSE depends on both the *bias* $E(\boldsymbol{A}) - \mathbf{a}^*$ and the variance $V(\boldsymbol{A})$. An estimator $\mathcal{A}$ is *consistent* if the more data points we take in each data set $\mathbf{z}$, the closer $\mathbf{a} = \mathcal{A}(\mathbf{z})$ gets to $\mathbf{a}^*$ (in a stochastic sense).

3.4 Maximising the likelihood

Maximum likelihood estimation uses the fact that in a complete statement of a model, the deviations ϵ_i are modelled as belonging to statistical distributions defined in terms of probability density functions (section 2.4.6). These distributions can be used to define a likelihood function. Suppose the measurement model is of the form

$$Y_i = \phi(x_i, \mathbf{a}) + E_i,$$

where $\boldsymbol{E} = (E_1, \ldots, E_m)^{\mathrm{T}}$ has multivariate PDF $g(\mathbf{x})$. Let

$$\boldsymbol{\phi}(\mathbf{a}) = (\phi(x_1, \mathbf{a}), \ldots, \phi(x_m, \mathbf{a}))^{\mathrm{T}},$$

a vector function of $\mathbf{a}$. The probability $p(\mathbf{y}|\mathbf{a})$ of observing the data $\mathbf{y}$ given that the model specified by parameters $\mathbf{a}$ is represented by $g(\mathbf{a}) = g(\mathbf{y} - \boldsymbol{\phi}(\mathbf{a}))$, which we can regard as function of $\mathbf{a}$. In general, if $\mathbf{z}$ are observations of random variables $\boldsymbol{Z}$, the likelihood $l(\mathbf{a}|\mathbf{z})$ of $\mathbf{a}$ giving rise to data $\mathbf{z}$ is the same as the probability $p(\mathbf{z}|\mathbf{a})$, i.e., $l(\mathbf{a}|\mathbf{z}) = p(\mathbf{z}|\mathbf{a})$. The notation is used to indicate that we regard the likelihood as a function of the parameters $\mathbf{a}$ with the observed data $\mathbf{z}$ fixed, while $p(\mathbf{z}|\mathbf{a})$ is a function of $\mathbf{z}$ with $\mathbf{a}$ regarded as fixed.[1] The maximum likelihood estimate of $\mathbf{a}$ is that which maximises $l(\mathbf{a}|\mathbf{z})$, i.e., that which provides the most probable explanation of the data $\mathbf{z}$. Maximum likelihood estimates enjoy favourable properties with respect to bias and statistical efficiency and usually represent an appropriate method for determining parameter estimates. Many standard parameter estimation methods can be formulated as maximum likelihood estimation for particular statistical models for the random effects.

Example: standard experiment and least squares

In the standard experiment, the model equations are of the form

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad i = 1, \ldots, m, \ \boldsymbol{\epsilon} \in N(0, \sigma^2 I).$$

Regarding $f_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$ as having the probability density function specified for ϵ_i, the associated likelihood function is (proportional to)

$$\prod_{i=1}^{m} \exp\left\{-\frac{1}{2}\frac{f_i^2}{\sigma^2}\right\} = \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{m} f_i^2\right\},$$

[1] More correctly, the data $\mathbf{z}$ are observations of random variables $\boldsymbol{Z}$ whose PDF is the function $\mathbf{x} \mapsto p(\mathbf{x}|\mathbf{a})$.

so that the likelihood is maximised when

$$\sum_{i=1}^{m} f_i^2 = \sum_{i=1}^{m} (y_i - \phi(\mathbf{x}_i, \mathbf{a}))^2$$

is minimised with respect to $\mathbf{a}$. ♯

The importance of least-squares estimation derives from the fact that it represents a maximum likelihood estimation for models subject to normally distributed random effects in the response variable. For linear models, it can be shown that it is unbiased and optimally efficient; see section 4.10.

Example: uniform distributions and Chebyshev approximation

Suppose, in an experiment, the model equations are of the form

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i,$$

where $\epsilon_i \in U(-S, S)$ are modelled as belonging to uniform (rectangular) distribution on the interval $[-S, S]$. This situation can arise, for example, if the measurements y_i are read from a digital indicating device, in which case S is half the last displayed unit. If all other random effects are negligible, a uniform distribution is appropriate. Regarding $f_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$ as having the probability density function specified for ϵ_i, the associated likelihood function is (proportional to) 1 if $|f_i| \leq S$, $i = 1, \ldots, m$, and 0 otherwise. The likelihood is maximised by any $\mathbf{a}$ such that

$$\max_i |f_i| = \max_i |y_i - \phi(\mathbf{x}_i, \mathbf{a}))| \leq S.$$

Such an $\mathbf{a}$, if it exists, can be found by solving the L_∞ (i.e., Chebyshev or *minimax*) optimisation problem

$$\min_{\mathbf{a}} \max_i |y_i - \phi(\mathbf{x}_i, \mathbf{a}))|.$$

In this way we can think of Chebyshev approximation as a maximum likelihood estimator for uniformly distributed random effects. ♯

Example: exponential power distributions and p-norms

Just as least squares and Chebyshev correspond to maximum likelihood estimation associated with Gaussian and rectangular sampling distributions, respectively, approximation in a p-norm (section 3.2) corresponds to an exponential power distribution (see e.g., [33, section 3.2.1]) with PDF

$$g(x) = \frac{k}{\alpha_3} \exp\left\{ -\frac{1}{2} \left| \frac{x - \alpha_1}{\alpha_3} \right|^{2/(1+\alpha_2)} \right\},$$

where $-\infty < \alpha_1 < \infty$, $-1 < \alpha_2 \leq 1$, is such that $p = 2/(1 + \alpha_2)$, $\alpha_3 > 0$, and the normalising constant is given by

$$k^{-1} = \Gamma\left(1 + \frac{1 + \alpha_2}{2}\right) 2^{1+(1+\alpha_2)/2}.$$

The parameter α_2 controls the kurtosis or 'peakedness' of the distribution. The value of $\alpha_2 = 0$ gives the normal distribution, as α_2 approaches -1 the distribution becomes more rectangular, and towards $+1$ the peak becomes narrower. ♯

3.5 Parameter estimation as optimisation problems

Estimators are usually defined in terms of minimising an error function $F(\mathbf{a}|\mathbf{z})$ defined in terms of the data $\mathbf{z}$ and the parameters $\mathbf{a}$. These optimisation problems are generally solved by determining a set of *optimality conditions* for the parameters $\mathbf{a}$ that must necessarily hold at the solution and then employing an algorithm designed to produce a solution satisfying these conditions. The following are some of the optimisation problems that are relevant to discrete modelling (in roughly decreasing order of importance in metrology) and for which mature and reliable algorithms and software implementations are available. Throughout, C is an $m \times n$ matrix, $m \geq n$, with rows $\mathbf{c}_i^{\mathrm{T}}$, $\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$ an m-vector of observations, and $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$ a vector of optimisation parameters.

3.5.1 Linear least squares

Solve

$$\min_{\mathbf{a}} \sum_{i=1}^{m} (y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a})^2 = \sum_{i=1}^{m} (y_i - (c_{i1}a_1 + \ldots + c_{in}a_n))^2.$$

In matrix form, this problem is written as

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|_2^2.$$

The matrix C is often referred to as the *observation matrix* or *design matrix*.

3.5.2 Nonlinear least squares

Given m functions $f_i(\mathbf{a})$ of parameters $\mathbf{a}$, solve

$$\min_{\mathbf{a}} \sum_{i=1}^{m} f_i^2(\mathbf{a}),$$

where the functions f_i usually depend on $\mathbf{y}$.

3.5.3 Linear least squares subject to linear equality constraints

Given C, $\mathbf{y}$, an $n \times p$ matrix D, $p < n$, and a p-vector $\mathbf{z}$, solve

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|_2^2$$

subject to the constraints

$$D\mathbf{a} = \mathbf{z}.$$

3.5.4 Nonlinear least squares subject to linear equality constraints

Given m functions $f_i(\mathbf{a})$ of parameters $\mathbf{a}$, an $n \times p$ matrix D, $p < n$ and a p-vector $\mathbf{z}$, solve

$$\min_{\mathbf{a}} \sum_{i=1}^{m} f_i^2(\mathbf{a})$$

(where the functions f_i usually depend on $\mathbf{y}$), subject to the constraints

$$D\mathbf{a} = \mathbf{z}.$$

3.5.5 Linear L_1

Given C and $\mathbf{y}$, solve

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|_1 \equiv \min_{\mathbf{a}} \sum_{i=1}^{m} |y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|.$$

3.5.6 Linear Chebyshev (L_∞)

Given C and $\mathbf{y}$, solve

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|_\infty \equiv \min_{\mathbf{a}} \max_i |y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|.$$

3.5.7 Linear programming

Given n-vectors $\mathbf{c}$ and $\mathbf{d}_i$, $i = 1, \ldots, m$, and $\mathbf{y}$, solve

$$\min_{\mathbf{a}} \mathbf{c}^{\mathrm{T}}\mathbf{a}$$

subject to the linear inequality constraints

$$\mathbf{d}_i^{\mathrm{T}}\mathbf{a} \geq y_i, \quad i = 1, \ldots, m.$$

3.5.8 Unconstrained minimisation

Given a function $F(\mathbf{a})$ of parameters $\mathbf{a}$, solve

$$\min_{\mathbf{a}} F(\mathbf{a}).$$

3.5.9 Nonlinear Chebyshev (L_∞)

Given m functions $f_i(\mathbf{a})$ of parameters $\mathbf{a}$, solve

$$\min_{\mathbf{a}} \max_i |f_i(\mathbf{a})|,$$

where the functions f_i usually depend on $\mathbf{y}$.

This formulation can be reposed as

$$\min_{\mathbf{a},s} s$$

subject to the constraints

$$-s \le f_i(\mathbf{a}) \le s, \quad i = 1, \ldots, m.$$

This is a special case of the following.

3.5.10 Mathematical programming

Given functions $F(\mathbf{a})$ and $g_k(\mathbf{a})$, $k = 1, \ldots, K$, of parameters $\mathbf{a}$, n-vectors $\mathbf{d}_i$, $i = 1, \ldots, m$, and $\mathbf{y}$, solve

$$\min_{\mathbf{a}} F(\mathbf{a})$$

subject to the linear constraints

$$\mathbf{d}_i^{\mathrm{T}} \mathbf{a} \ge y_i, \quad i = 1, \ldots, m$$

and nonlinear constraints

$$g_k(\mathbf{a}) \ge 0, \quad k = 1, \ldots, K.$$

3.6 Minimisation of a function of several variables

Let $F(\mathbf{a})$ be a general (smooth) function of n variables $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$: F is the *objective function* of the minimisation problem.

Let $\mathbf{g}(\mathbf{a})$ be the *gradient* of F, with components $g_j = \partial F/\partial a_j$, and H the *Hessian matrix* of second partial derivatives,

$$H_{jk} = \partial^2 F/\partial a_j \partial a_k.$$

At a minimum $\mathbf{a}^*$ of F, $\mathbf{g}(\mathbf{a}^*) = \mathbf{0}$. If $\mathbf{a}$ is an approximate solution we wish to find a step $\mathbf{p}$ such that $\mathbf{g}(\mathbf{a} + \mathbf{p}) = \mathbf{0}$. To first order,

$$\mathbf{g}(\mathbf{a} + \mathbf{p}) = \mathbf{g} + H\mathbf{p},$$

suggesting that $\mathbf{p}$ should be chosen so that

$$H\mathbf{p} = -\mathbf{g}. \tag{3.1}$$

In the Newton algorithm, an estimate of the solution $\mathbf{a}$ is updated according to $\mathbf{a} := \mathbf{a} + t\mathbf{p}$, where $\mathbf{p}$ solves (3.1) and t is a step length chosen to ensure a sufficient decrease in F. Near the solution, the Newton algorithm converges quadratically, i.e., if at the kth iteration the distance of the current estimate $\mathbf{a}_k$ from the solution $\mathbf{a}^*$ is $\|\mathbf{a}_k - \mathbf{a}^*\|$, then the distance of the subsequent estimate $\mathbf{a}_{k+1}$ from the solution is $\|\mathbf{a}_{k+1} - \mathbf{a}^*\| = O(\|\mathbf{a}_k - \mathbf{a}\|^2)$, so that the distance to the solution is squared approximately at each iteration.

3.6.1 Nonlinear least squares

For nonlinear least-squares problems, the objective function is of the form[2]

$$F(\mathbf{a}) = \frac{1}{2}\sum_{i=1}^{m} f_i^2(\mathbf{a})$$

and has gradient

$$\mathbf{g} = J^{\mathrm{T}}\mathbf{f},$$

where J is the Jacobian matrix

$$J_{ij} = \frac{\partial f_i}{\partial a_j}, \tag{3.2}$$

and Hessian matrix

$$H = J^{\mathrm{T}}J + G, \qquad G_{jk} = \sum_{i=1}^{m} f_i \frac{\partial^2 f_i}{\partial a_j \partial a_k}.$$

3.6.2 Large scale optimisation

The main computational step in the Newton algorithm is the formulation and solution of the equations (3.1) for the search direction $\mathbf{p}$ which generally takes $O(n^3)$ operations where n is the number of parameters. For very large problems, this may not be feasible (usually because too much time is required).

The conjugate gradient approach [117] is one of the main tools in general purpose large scale optimisation, particularly because it requires only a few vectors to be stored. Suppose we wish to find the minimum of $F(\mathbf{a})$, given an initial estimate $\mathbf{a}_0$. For nonlinear problems, the algorithm takes the form

I Set $k = 0$, $\mathbf{g}_0 = \nabla_{\mathbf{a}} F(\mathbf{a}_0)$.

II While $\|\mathbf{g}_k\| > \tau$ (where $\tau > 0$ is a small constant),

[2]The fraction $\frac{1}{2}$ is sometimes included to simplify related expressions.

i Set $k = k + 1$.

ii Determine a search direction. If $k = 1$ set $\mathbf{p}_1 = -\mathbf{g}_0$. If k is a multiple of n, set $\mathbf{p}_k = -\mathbf{g}_{k-1}$. Otherwise, set

$$\beta_k = \|\mathbf{g}_{k-1}\|^2 / \|\mathbf{g}_{k-2}\|^2, \quad \mathbf{p}_k = -\mathbf{g}_{k-1} + \beta_k \mathbf{p}_{k-1}.$$

iii Determine the step length. Find α_k to minimise $F(\mathbf{a}_{k-1} + \alpha_k \mathbf{p}_k)$.

iv Update

$$\mathbf{a}_k = \mathbf{a}_{k-1} + \alpha_k \mathbf{p}_k, \quad \mathbf{g}_k = \nabla_{\mathbf{a}} F(\mathbf{a}_k).$$

III Set $\mathbf{a} = \mathbf{a}_k$ and finish.

There has been much research in developing efficient, large-scale optimisation algorithms; see e.g., [51, 161, 210]. One of the main approaches is to use a limited memory quasi-Newton algorithm [117, section 4.8]. In a quasi-Newton algorithm, the update step (3.1) is determined from an approximation to the Hessian matrix H of second partial derivatives of the objective function $F(\mathbf{a})$ or its inverse. Starting from the identity matrix, this approximation is built up from successive estimates $\mathbf{g}_k$ of the function gradients. If F is a quadratic function of n parameters, then after n steps the approximation to the Hessian is exact (in exact arithmetic). For large n, memory and computation constraints may prohibit any attempt to approximate H. Instead, the Hessian matrix is approximated by limited number of quasi-Newton updates and can be stored by a correspondingly limited number of n-vectors.

3.7 Problem conditioning

The numerical accuracy of the solution parameters $\mathbf{a}$ will depend on the conditioning of the problem. A problem is *well-conditioned* if a small change in the data corresponds to a small change in the solution parameters, and conversely.

3.7.1 Condition of a matrix, orthogonal factorisation and the SVD

The condition of a discrete modelling problem can usually be analysed in terms of the condition of a matrix associated with the problem, for example, the observation matrix for linear least-squares problems or the Jacobian matrix for nonlinear problems.

An $m \times n$ matrix Q is *orthogonal* if $Q^{\mathrm{T}}Q = I$, the $n \times n$ identity matrix. If $m = n$ then we have in addition $QQ^{\mathrm{T}} = I$. Any two columns $\mathbf{q}_j$, $\mathbf{q}_k$, $j \neq k$, of an orthogonal matrix are at right angles to each other in the sense that $\mathbf{q}_j^{\mathrm{T}}\mathbf{q}_k = 0$. Orthogonal matrices have the property of preserving the Euclidean (2-norm) length of a vector: $\|Q\mathbf{x}\| = \|\mathbf{x}\|$.

Given two vectors $\mathbf{x} = (x_1, x_2, x_3)^{\mathrm{T}}$ and $\mathbf{y} = (y_1, y_2, y_3)^{\mathrm{T}}$ in $\mathbb{R}^3$, they can be rotated by a rotation matrix Q so that one lies along the x-axis and one lies in

the xy-plane:

$$Q^{\mathrm{T}} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \\ 0 & 0 \end{bmatrix} \text{ or } \quad \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} = Q \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \\ 0 & 0 \end{bmatrix},$$

expressing the matrix with columns $\mathbf{x}$ and $\mathbf{y}$ as a product of an orthogonal matrix and an upper-triangular matrix. More generally, any $m \times n$ matrix C can be factored as

$$C = QR = [Q_1 \; Q_2] \begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix} = Q_1 R_1, \tag{3.3}$$

where Q is $m \times m$ orthogonal, Q_1 (Q_2) is the submatrix comprising the first n (last $m - n$) columns of Q, and R_1 is $n \times n$ upper triangular. Any $m \times n$ matrix C can also be factored as the product

$$C = USV^{\mathrm{T}} = [U_1 \; U_2] \begin{bmatrix} S_1 \\ \mathbf{0} \end{bmatrix} V^{\mathrm{T}} = U_1 S_1 V^{\mathrm{T}}, \tag{3.4}$$

where U is an $m \times m$ orthogonal matrix, U_1 (U_2) is the submatrix comprising the first n (last $m-n$) columns of U, S_1 an $n \times n$ diagonal matrix with diagonal entries $s_1 \geq s_2 \geq \ldots \geq s_n$, and V an $n \times n$ orthogonal matrix. This factorisation is known as the singular value decomposition (SVD). The columns of U (V) are the *left (right) singular vectors* and the s_j are known as the *singular values*.

The SVD shows that C maps the orthonormal vectors $\mathbf{v}_j$ onto the vectors $s_j \mathbf{u}_j$. If C has singular values all equal to one then it is an orthogonal matrix and, conversely; C is full rank if and only if $s_n > 0$.

The ratio $\kappa = s_1/s_n$ of the largest singular value of a matrix to the smallest is known as the *condition number* of the matrix. There are high quality public domain software implementations of reliable algorithms to determine the SVD [87, 190].

If $C = USV^{\mathrm{T}}$ then the eigenvalue decomposition of $C^{\mathrm{T}}C$ is given by

$$C^{\mathrm{T}}C = VS^2V^{\mathrm{T}},$$

showing that the eigenvalues λ_j of $C^{\mathrm{T}}C$ are the squares of the singular values of C: $\lambda_j = s_j^2$ and the eigenvectors of $C^{\mathrm{T}}C$ are precisely the right singular vectors of C.

The singular values have a geometrical interpretation. The matrix C maps the unit sphere $\{\mathbf{x} : \|\mathbf{x}\| = 1\}$ in $\mathbb{R}^n$ into a hyper-ellipsoid in $\mathbb{R}^m$. The singular values are the lengths of the semi-axes of the ellipsoid. In particular, the largest singular value s_1 is such that

$$s_1 = \|C\mathbf{v}_1\| = \max_{|\mathbf{v}|=1} \|C\mathbf{v}\|,$$

and the smallest s_n such that

$$s_n = \|C\mathbf{v}_n\| = \min_{|\mathbf{v}|=1} \|C\mathbf{v}\|. \tag{3.5}$$

The condition number is the ratio of the length of the largest semi-axis to that the smallest. An ill-conditioned matrix is one which maps the sphere into a long thin ellipsoid. Orthogonal matrices map the unit sphere to a unit sphere.

The unwelcome numerical consequences of working with ill-conditioned matrices are due to the fact that computation will involve relatively large numbers leading to cancellation errors. The value of orthogonal matrices is that no large numbers are introduced unnecessarily into the computations.

The conditioning of a problem depends on the parameterisation of the model. Often, the key to being able to determine accurate solution parameters is in finding an appropriate parameterisation.

Example: basis vectors for $\mathbb{R}^3$

Suppose we take as basis vectors for three dimensional space $\mathbb{R}^3$ the vectors $\mathbf{e}_1 = (1, 0, 0)^{\mathrm{T}}$, $\mathbf{e}_2 = (1, 0.001, 0)^{\mathrm{T}}$ and $\mathbf{e}_3 = (1, 0, 0.001)^{\mathrm{T}}$. Any point in $\mathbf{y}$ in $\mathbb{R}^3$ can be written as a linear combination

$$\mathbf{y} = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3.$$

For example,

$$\begin{aligned} (0.0, 1.0, 1.0)^{\mathrm{T}} &= -2000\mathbf{e}_1 + 1000\mathbf{e}_2 + 1000\mathbf{e}_3, \\ (0.0, 1.1, 1.1)^{T} &= -2200\mathbf{e}_1 + 1100\mathbf{e}_2 + 1100\mathbf{e}_3, \end{aligned}$$

showing that a change of the order of 0.1 in the point $\mathbf{y}$ requires a change of order 100 in the parameter values $\mathbf{a}$. This type of ill-conditioning means that up to three significant figures of accuracy could be lost using these basis vectors.

If $E = [\ \mathbf{e}_1\ \mathbf{e}_2\ \mathbf{e}_3\ ]$, the orthogonal factorisation of $E = QR$ produces the standard basis vectors $\mathbf{q}_1 = (1, 0, 0)^{\mathrm{T}}$, $\mathbf{q}_2 = (0, 1, 0)$ and $\mathbf{q}_3 = (0, 0, 1)^{\mathrm{T}}$ from the columns of Q. In many situations, an analysis using QR factorisations can lead a better choice of basis vectors (or functions). ♯

3.8 Numerical stability of algorithms

One factor affecting the numerical accuracy of the parameter estimates is the conditioning of the problem. A second is the numerical stability of the algorithm used to solve the computational problem associated with finding the parameter estimates. A numerically stable algorithm is one that introduces no unnecessary additional ill-conditioning into a problem. Many of the numerical difficulties in solving computational problems arise because the calculations introduce large numbers leading to large cancellation errors. A very simple example is the calculation of the difference of two squares $c = a^2 - b^2$. If $a = 101$ and $b = 100$, then $c = 201$; all three numbers are of the order of 100. If we calculate a^2 and b^2, we introduce numbers of the order of 10^4. If instead we calculate $a - b$ and $a + b$ and set

$$c = (a - b)(a + b),$$

all the intermediate quantities remain of the order of 100 or smaller. A floating-point error analysis shows that the latter method is numerically superior. The calculation of a^2 and b^2 can also lead to overflow problems.

Analysing the stability of an algorithm generally requires a specialist in numerical analysis. Many of the algorithms implemented in high quality library numerical software have a supporting error analysis demonstrating their favourable behaviour (which is why the algorithms appear in the library in the first place).

Issues concerning the numerical stability of algorithms are covered in the companion best-practice guide on *Numerical analysis for algorithm design in metrology* [68].

3.9 Uncertainty associated with parameter estimates

A key aspect of metrological data analysis is the requirement to produce estimates of the uncertainties associated with the fitted parameters $\mathbf{a} = \mathcal{A}(\mathbf{z})$ derived from data $\mathbf{z}$. These estimates can be in the form of standard uncertainties

$$u_j = u(a_j), \quad j = 1, \ldots, n,$$

associated with the parameter estimates or the $n \times n$ uncertainty (covariance) matrix $V_{\mathbf{a}}$ associated with $\mathbf{a}$. (The standard uncertainties are given by $u_j = (V_{\mathbf{a}}(j,j))^{1/2}$). Often we are interested in the standard uncertainty associated with a function $h(\mathbf{a})$ of the parameters. From the law for the propagation of uncertainty (section 2.4.4), if $\mathbf{g} = (\frac{\partial h}{\partial a_1}, \ldots, \frac{\partial h}{\partial a_n})^{\mathrm{T}}$ are the partial derivatives of h (sensitivity coefficients), then the variance V_h and standard uncertainty $u(h)$ associated with h are obtained by

$$V_h = \mathbf{g}^{\mathrm{T}} V_{\mathbf{a}} \mathbf{g}, \quad u(h) = V_h^{1/2}.$$

Note that for these calculations, the complete uncertainty matrix $V_{\mathbf{a}}$ is required, not just the standard uncertainties associated with $\mathbf{a}$. For this reason, it is recommended that in designing estimator algorithms, the calculation of the uncertainty matrix associated with the parameters estimates is included (at least as an option).

There are three basic approaches to estimating these statistics. For some estimators, including least squares, the uncertainty matrix can be calculated from the data and the solution estimate (see sections 4.1 and 4.2). A second approach is to apply a number of random perturbations Δ_q, determined in accordance with the statistical model for the random effects, to the data X producing data sets $X_q = X + \Delta_q$ and corresponding solution estimates $\mathbf{a}_q = \mathcal{A}(X_q)$, $q = 1, \ldots, N$. The covariance of the sample of estimates $\{\mathbf{a}_q\}$ is an estimate of $V_{\mathbf{a}}$. This is the *Monte Carlo* approach (see, for example, [64, 71, 191]. Importantly, Monte Carlo simulations can be used to validate the uncertainties calculated by analytical methods. Both the analytical and Monte Carlo approaches depend explicitly or implicitly on a statistical model for the

random effects. The third approach, based on resampling methodologies, is similar to Monte Carlo simulations except that the perturbations Δ_q of the data are determined from the distribution of the residuals associated with the original fit rather than from a statistical model. See, for example, [43, 75, 90].

3.10 Numerical simulation and experimental design

The calculation of the uncertainty matrix, etc., is obviously important in uncertainty estimation. These calculations can also be used in numerical simulations to determine the effectiveness of different measurement strategies and experiment configurations. In this situation, exact measurement data $\mathbf{x}_i^*$ are generated according to a specified measurement strategy and the data then perturbed according to the statistical model for the random effects. By changing the measurement strategy and monitoring the effect on the variances of the parameters of interest, it is often possible to improve the experimental efficiency. Importantly, this can be achieved by using the same data analysis modules which are required to determine estimates of the parameters and their uncertainties to actual data. In other words, with very little additional effort the model solving tools can be used to improve experimental strategy.

Chapter 4

Estimators

In this chapter, we describe in more detail some of the common estimators and associated algorithms.

4.1 Linear least squares (LLS)

4.1.1 Description

Given data $\{(\mathbf{x}_i, y_i)\}_1^m$ and the linear model

$$y = \phi(\mathbf{x}, \mathbf{a}) = a_1\phi_1(\mathbf{x}) + \ldots + a_n\phi_n(\mathbf{x}), \quad n \leq m,$$

the linear least-squares estimate of the parameters $\mathbf{a}$ is the one which solves

$$\min_{\mathbf{a}} \sum_{i=1}^{m} (y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a})^2,$$

where $\mathbf{c}_i = (\phi_1(\mathbf{x}_i), \ldots, \phi_n(\mathbf{x}_i))^{\mathrm{T}}$.

Let C be the matrix whose ith row is $\mathbf{c}_i^{\mathrm{T}}$, $\mathbf{y}$ the vector whose ith element is y_i and $\mathbf{f}(\mathbf{a}) = \mathbf{y} - C\mathbf{a}$. The problem can be reposed as

$$\min_{\mathbf{a}} F(\mathbf{a}) = \mathbf{f}^{\mathrm{T}}\mathbf{f} = \|\mathbf{y} - C\mathbf{a}\|_2^2.$$

At the solution, it is known that the partial derivatives of F with respect to the parameters are zero, i.e.,

$$\frac{\partial F}{\partial a_j} = 0, \quad j = 1, \ldots, n,$$

and this leads to the system of linear equations of order n,

$$C^{\mathrm{T}}C\mathbf{a} = C^{\mathrm{T}}\mathbf{y}, \tag{4.1}$$

known as the *normal equations*. If C is full rank, so that $C^{\mathrm{T}}C$ is invertible, the solution parameters are given (mathematically) by

$$\mathbf{a} = (C^{\mathrm{T}}C)^{-1}C^{T}\mathbf{y}. \tag{4.2}$$

Linear least-squares estimators are the most common of the estimators used in metrology. They are optimal for linear models in which the measurements of a single response variable are subject to uncorrelated normally distributed random effects:

$$y_i = a_1\phi_1(\mathbf{x}_i) + \ldots + a_n\phi_n(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2), i = 1, \ldots, m \geq n.$$

They are suitable for any system for which the main random effects are associated with the response variables and these effects are symmetrically distributed about a zero mean; see section 4.1.9.

Linear least squares are less suitable for data in which more than one variable is subject to significant random effects or for data which contains outliers or rogue points or where the random effects are modelled as being governed by long tailed distributions (section 4.7).

4.1.2 Algorithms to find the linear least-squares estimate

There are two basic approaches to determining a least-squares solution to a set of over-determined equations.

Solving the normal equations. Although equation (4.2) suggests that the linear least-squares estimate is found by inverting the $n \times n$ matrix $H = C^{\mathrm{T}}C$, as in the case of practically all matrix equation problems, matrix inversion is far from the best option. If the normal equations are to be solved, the preferred approach exploits the fact that H is symmetric and, assuming it is full rank, has a Cholesky decomposition

$$H = LL^{\mathrm{T}},$$

where L is an $n \times n$ lower triangular matrix (so that $L(i, j) = 0$ if $i > j$). With this factorisation, the parameters $\mathbf{a}$ are determined by solving, in sequence, two triangular systems

$$L\mathbf{b} = C^{\mathrm{T}}\mathbf{y}, \quad L^{\mathrm{T}}\mathbf{a} = \mathbf{b}.$$

The Cholesky factorisation and the solution of the triangular systems are easily implemented in software, requiring only a few lines of code [119].

Orthogonal factorisation methods. If C has orthogonal factorisation (section 3.7.1)

$$C = QR = [Q_1\ Q_2]\begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \tag{4.3}$$

then, using the fact that $\|Q\mathbf{x}\| = \|\mathbf{x}\|$, we have

$$\|\mathbf{y} - C\mathbf{a}\| = \|Q^{\mathrm{T}}\mathbf{y} - Q^{\mathrm{T}}C\mathbf{a}\| = \left\| \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix} - \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \mathbf{a} \right\|,$$

where $\mathbf{q}_1$ is the first n and $\mathbf{q}_2$ the last $m-n$ elements of $Q^{\mathrm{T}}\mathbf{y}$, i.e., $\mathbf{q}_1 = Q_1^{\mathrm{T}}\mathbf{y}$, etc. From this it is seen that $\|\mathbf{y} - C\mathbf{a}\|$ is minimised if $\mathbf{a}$ solves the upper triangular system

$$R_1\mathbf{a} = \mathbf{q}_1.$$

In practice, the orthogonalisation is applied to the augmented matrix

$$Q^{\mathrm{T}} \begin{bmatrix} C & \mathbf{y} \end{bmatrix} = \begin{bmatrix} R_1 & \mathbf{q}_1 \\ \mathbf{0} & \|\mathbf{f}\| \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

to produce simultaneously the upper triangular factor R_1, the right-hand side vector $\mathbf{q}_1$ and the norm $\|\mathbf{f}\|$ of the residuals $\mathbf{f} = \mathbf{y} - C\mathbf{a}$.

As with the Cholesky factorisation, orthogonal factorisations are easy to construct [119].

The main advantage of the orthogonal factorisation method over the normal equations method is one of numerical accuracy. If due to ill-conditioning in the matrix C the orthogonal factorisation method potentially loses p decimal digits of accuracy, then the normal equations method potentially loses $2p$ decimal digits.

Taking into account sparsity structure in the observation matrix. There are a number of applications in which the observation matrix has a large number of zero entries. This *sparsity structure* can be exploited to increase the efficiency of the solution process; some of these techniques are described in [56, 58, 67, 180].

4.1.3 Uncertainty associated with the fitted parameters

The uncertainty matrix $V_{\mathbf{a}}$ associated with the fitted parameters is obtained using the fact that the linear least-squares solution $\mathbf{a}$ is a linear combination of the data values $\mathbf{y}$. If $\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$ has associated uncertainty matrix[1] matrix $V_{\mathbf{y}}$ and $\mathbf{a}(\mathbf{y}) = G\mathbf{y}$ are n linear functions of $\mathbf{y}$, then the uncertainty matrix associated with $\mathbf{a}$ is given by[2]

$$V_{\mathbf{a}} = GV_{\mathbf{y}}G^{\mathrm{T}}.$$

The normal equations (4.1) define the linear least-squares solution (from equation (4.2)), as

$$\mathbf{a} = C^{\dagger}\mathbf{y},$$

[1] That is, $\mathbf{y}$ is an observation of a vector of random variables $\boldsymbol{Y}$ whose multivariate distribution has variance matrix $V_{\mathbf{y}}$.

[2] That is, $\mathbf{a}$ is an observation of a vector of random variables $\boldsymbol{A}$ whose multivariate distribution has variance matrix $V_{\mathbf{a}}$.

where

$$C^\dagger = (C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}} \tag{4.4}$$

is the *pseudo-inverse* of C [119, section 5.5.4] and is such that $CC^\dagger C = C$, $C^\dagger CC^\dagger = C^\dagger$ and $C^\dagger \left(C^\dagger\right)^{\mathrm{T}} = (C^{\mathrm{T}}C)^{-1}$. Therefore,

$$V_{\mathbf{a}} = C^\dagger V_{\mathbf{y}} \left(C^\dagger\right)^{\mathrm{T}}.$$

If $V_{\mathbf{y}} = \sigma^2 I$ (as in the case for the standard experiment), this expression simplifies to

$$V_{\mathbf{a}} = C^\dagger \sigma^2 I \left(C^\dagger\right)^{\mathrm{T}} = \sigma^2 (C^{\mathrm{T}}C)^{-1}.$$

If C has orthogonal factorisation given in (4.3) then, using the fact that $Q^{\mathrm{T}}Q = I$ for an orthogonal matrix, $V_{\mathbf{a}}$ can be calculated from the triangular factor R_1 and σ:

$$V_{\mathbf{a}} = \sigma^2 \left(R_1^{\mathrm{T}} R_1\right)^{-1} = \sigma^2 R_1^{-1} R_1^{-\mathrm{T}}.$$

If $h = \mathbf{h}^{\mathrm{T}}\mathbf{a}$, a linear combination of the parameters, then

$$u(h) = \sigma \|\tilde{\mathbf{h}}\|,$$

where $\tilde{\mathbf{h}}$ solves

$$R_1^{\mathrm{T}} \tilde{\mathbf{h}} = \mathbf{h}.$$

This means that the standard uncertainties associated with the fitted parameters, or linear combinations of those parameters, can be determined efficiently by solving such triangular systems.

These calculations assume that the standard deviation σ associated with the random effects[3] in the data is already known. If this is not the case, then for overdetermined systems a *posterior estimate* $\hat{\sigma}$ of σ can be determined from the vector $\mathbf{r} = \mathbf{y} - C\mathbf{a}$ of residuals:

$$\hat{\sigma} = \|\mathbf{r}\|/(m-n)^{1/2}. \tag{4.5}$$

Details. This estimate is justified as follows.

If $X_i \sim N(0,1)$, $i = 1, \ldots, m$, are independent normal variates then $\sum_{i=1}^m X_i^2$ has a χ^2_m distribution with mean m and variance $2m$. Let $\boldsymbol{R}$ be the random vector of residuals so that

$$\boldsymbol{R} = \boldsymbol{Y} - C\boldsymbol{A} = \boldsymbol{Y} - CC^\dagger \boldsymbol{Y} = (I - CC^\dagger)\boldsymbol{Y}.$$

If $C = Q_1 R_1$ as in (4.3), then $CC^\dagger = Q_1 Q_1^{\mathrm{T}}$ and $I - Q_1 Q_1^{\mathrm{T}} = Q_2 Q_2^{\mathrm{T}}$, so that

$$S^2 = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{R} = \left(Q_2^{\mathrm{T}}\boldsymbol{Y}\right)^{\mathrm{T}} Q_2^{\mathrm{T}}\boldsymbol{Y}.$$

Now Q is orthogonal so setting $\tilde{\boldsymbol{Y}} = Q\boldsymbol{Y}$ we have $\mathrm{Var}(\tilde{\boldsymbol{Y}}) = I$ also. Therefore, $S^2 = \sum_{i=n+1}^m \tilde{Y}_i^2$ is a sum of squares of $m-n$ independent, normal variates and has a χ^2_ν distribution with $\nu = m-n$ degrees of freedom, with $E(S^2) = \nu$, $\mathrm{Var}(S^2) = 2\nu$. From this analysis, we see that given a least-squares solution $\mathbf{a}$, a posterior estimate of σ is $\hat{\sigma}$ in (4.5).

While this estimate is derived under the assumption that the random effects are governed by a Gaussian distribution, it is likely to be a good approximation for distributions with similar features, e.g., unimodal (that is, having one peak).

[3] That is, $\boldsymbol{\epsilon} \in \boldsymbol{E}$ with $V(\boldsymbol{E}) = \sigma^2 I$.

4.1.4 Calculation of other quantities associated with the model fit

We summarise here the quantities associated with a linear least-squares fit that are often useful to calculate.

- Estimates of the solution parameters $\mathbf{a} = (C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}\mathbf{y} = C^{\dagger}\mathbf{y}$.
- The model predictions $\hat{\mathbf{y}} = C\mathbf{a} = C(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}\mathbf{y} = CC^{\dagger}\mathbf{y}$, i.e., the predicted responses $\hat{y}_i$ at values $\mathbf{x}_i$ of the covariates.
- The residual vector

  $$\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - C\mathbf{a} = (I - C(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}})\mathbf{y} = (I - CC^{\dagger})\mathbf{y},$$

  where I is the $m \times m$ identity matrix.
- The posterior estimate of the standard deviation of the random effects

  $$\hat{\sigma} = \|\mathbf{r}\|/(m-n)^{1/2}.$$

- The uncertainty (covariance) matrix associated with the fitted parameters. If an estimate of σ is available

  $$V_{\mathbf{a}} = \sigma^2(C^{\mathrm{T}}C)^{-1},$$

  otherwise, $V_{\mathbf{a}}$ can be obtained from

  $$V_{\mathbf{a}} = \hat{\sigma}^2(C^{\mathrm{T}}C)^{-1},$$

  where $\hat{\sigma}$ is given by (4.5).
- The standard uncertainties associated with the fitted parameters $u(a_j) = (V_{\mathbf{a}}(j,j))^{1/2}$, i.e., the square roots of the diagonal elements of the uncertainty matrix $V_{\mathbf{a}}$.
- The correlation matrix associated with the fitted parameters defined by

  $$C_R(i,j) = \frac{V_{\mathbf{a}}(i,j)}{(V_{\mathbf{a}}(i,i)V_{\mathbf{a}}(j,j))^{1/2}}.$$

  Note that C_R is independent of the value of σ used to define the uncertainty matrix.
- The uncertainty (covariance) matrix $V_{\hat{\mathbf{y}}}$ associated with the model predictions $\hat{\mathbf{y}}$

  $$V_{\hat{\mathbf{y}}} = CV_{\mathbf{a}}C^{\mathrm{T}} = \sigma^2 C(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}.$$

- The standard uncertainties associated with the model predictions $u(\hat{y}_i) = (V_{\hat{\mathbf{y}}}(i,i))^{1/2}$.
- The uncertainty matrix $V_{\mathbf{r}}$ associated with the residuals

  $$V_{\mathbf{r}} = \sigma^2(I - C(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}).$$

- The standard uncertainties associated with the residual errors $u(r_i) = (V_{\mathbf{r}}(i,i))^{1/2}$.
- If $(\mathbf{z}, w)$ represents a new data point (generated from the same model but not used in defining the model fit) then the predicted model value at $\mathbf{z}$ is

$$\hat{w} = \phi(\mathbf{z}, \mathbf{a}) = \mathbf{d}^{\mathrm{T}}\mathbf{a},$$

where $\mathbf{d} = (d_1, \ldots, d_n)^T = (\phi_1(\mathbf{z}, \mathbf{a}), \ldots, \phi_n(\mathbf{z}, \mathbf{a}))^{\mathrm{T}}$, the standard uncertainty associated with $\hat{w}$ is

$$u(\hat{w}) = (\mathbf{d}^{\mathrm{T}} V_{\mathbf{a}} \mathbf{d})^{1/2},$$

the predicted residual error is $t = w - \hat{w} = w - \mathbf{d}^{\mathrm{T}}\mathbf{a}$ and its variance is

$$V_t = \sigma^2 + \mathbf{d}^{\mathrm{T}} V_{\mathbf{a}} \mathbf{d}.$$

More generally, if $Z = \{\mathbf{z}_q\}_{q=1}^{m_Z}$ is a range of values for the covariates and D is the corresponding matrix of basis functions evaluated at $\mathbf{z}_q$, i.e.,

$$D_{q,j} = \phi_j(\mathbf{z}_q),$$

then the uncertainty matrix $V_{\mathbf{w}}$ associated with the model values $\mathbf{w} = (w_1, \ldots, w_{m_Z})^{\mathrm{T}}$, $w_q = \phi(\mathbf{z}_q, \mathbf{a})$, is

$$V_{\mathbf{w}} = D V_{\mathbf{a}} D^{\mathrm{T}},$$

and the standard uncertainty $u(w_q)$ is

$$u(w_q) = (V_{\mathbf{w}}(q,q))^{1/2}.$$

We note that if the observation matrix has QR factorisation $C = Q_1 R_1$ where Q_1 is an $m \times n$ orthogonal matrix and R_1 is an $n \times n$ upper triangular matrix and singular value decomposition (SVD) $C = U_1 S_1 V^{\mathrm{T}}$ where U_1 is an $m \times n$ orthogonal matrix, S_1 is $n \times n$ diagonal matrix and V is $n \times n$ orthogonal matrix, then

$$\begin{aligned}
C^T C &= R_1^{\mathrm{T}} R_1 = V S_1^2 V^{\mathrm{T}}, \\
(C^T C)^{-1} &= R_1^{-1} R_1^{-\mathrm{T}} = V S_1^{-2} V^{\mathrm{T}}, \\
(C^{\mathrm{T}} C)^{-1} C^{\mathrm{T}} &= C^{\dagger} = R_1^{-1} Q_1^{\mathrm{T}} = V S_1^{-1} U_1^{\mathrm{T}}, \quad \text{and} \\
C (C^{\mathrm{T}} C)^{-1} C^{\mathrm{T}} &= C C^{\dagger} = Q_1 Q_1^{\mathrm{T}} = U_1 U_1^{\mathrm{T}}.
\end{aligned}$$

These relations show that all the model outputs listed above can be calculated from QR factorisation or SVD of C. All the statistical information can be derived from $V_{\mathbf{a}}$.

4.1.5 Weighted linear least-squares estimator

If the random effects ϵ_i are uncorrelated but drawn from distributions with different standard deviations, e.g., $\epsilon_i \in N(0, \sigma_i^2)$ then the appropriate estimator is a weighted linear least-squares estimator which estimates $\mathbf{a}$ by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} w_i^2 (y_i - \mathbf{c}_i^{\mathrm{T}} \mathbf{a})^2,$$

with $w_i = 1/\sigma_i$. Algorithms for the unweighted linear least squares problem can be easily adapted to deal with the weighted case by applying them to

$$\tilde{y}_i = w_i y_i, \quad \tilde{C}(i,j) = w_i C(i,j).$$

4.1.6 Gauss-Markov estimator

More generally, if the vector of random effects are modelled as belonging to a multivariate distribution with uncertainty (covariance) matrix $V_{\mathbf{y}}$, assumed to be full rank, the Gauss-Markov estimator which solves

$$\min_{\mathbf{a}} \ (\mathbf{y} - C\mathbf{a})^{\mathrm{T}} V^{-1} (\mathbf{y} - C\mathbf{a}), \tag{4.6}$$

is appropriate. If V has a Cholesky decomposition $V = LL^{\mathrm{T}}$, then the Gauss-Markov estimate can be determined by applying the linear least-squares estimator to

$$\tilde{\mathbf{y}} = L^{-1}\mathbf{y}, \quad \tilde{C} = L^{-1}C.$$

The generalised QR decomposition can be employed to solve this problem in a numerically stable way [27, 70, 122, 190].

Details. For a general (full rank) uncertainty matrix V with a factorisation $V = LL^{\mathrm{T}}$, where L is an $m \times m$ matrix, also necessarily full rank, the least-squares estimate is given by

$$\mathbf{a} = \tilde{C}^{\dagger}\tilde{\mathbf{y}}, \quad \tilde{C} = L^{-1}C, \ \tilde{y} = L^{-1}\mathbf{y},$$

where $\tilde{C}^{\dagger}$ is the pseudo-inverse of $\tilde{C}$. For well conditioned V and L, this approach is satisfactory. However, if L is poorly conditioned the formation and use of $\tilde{C}$, etc., can be expected to introduce numerical errors. The *generalised QR factorisation* [70, 122, 179, 190] approach avoids this potential numerical instability. Suppose $V = LL^{\mathrm{T}}$, where L is $m \times p$. (Often $p = m$ but the approach applies in the more general case.) The estimate $\mathbf{a}$ can be found by solving

$$\min_{\mathbf{a},\mathbf{e}} \ \mathbf{e}^{\mathrm{T}}\mathbf{e} \quad \text{subject to constraints} \mathbf{y} = C\mathbf{a} + L\mathbf{e}. \tag{4.7}$$

Note that if L is invertible,

$$\mathbf{e} = L^{-1}(\mathbf{y} - C\mathbf{a}), \quad \mathbf{e}^{\mathrm{T}}\mathbf{e} = (\mathbf{y} - C\mathbf{a})^{\mathrm{T}} V^{-1} (\mathbf{y} - C\mathbf{a}).$$

We factorise $C = QR$ and $Q^T L = TU$ where R and T are upper-triangular and Q and U are orthogonal. Multiplying the constraints by Q^{T}, we have

$$\begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \tilde{\mathbf{y}}_2 \end{bmatrix} = \begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix} \mathbf{a} + \begin{bmatrix} T_{11} & T_{12} \\ & T_{22} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{e}}_1 \\ \tilde{\mathbf{e}}_2 \end{bmatrix}, \tag{4.8}$$

where $\tilde{\mathbf{y}} = Q^{\mathrm{T}}\mathbf{y}$, and $\tilde{\mathbf{e}} = U\mathbf{e}$.

From the second set of equations, $\tilde{\mathbf{e}}_2$ must satisfy $\tilde{\mathbf{y}}_2 = T_{22}\tilde{\mathbf{e}}_2$.

Given any $\tilde{\mathbf{e}}_1$, the first set of equations is satisfied if $R_1\mathbf{a} = \tilde{\mathbf{y}}_1 - T_{11}\tilde{\mathbf{e}}_1 - T_{12}\tilde{\mathbf{e}}_2$.

We choose $\tilde{\mathbf{e}}_1 = 0$ in order to minimise

$$\mathbf{e}^{\mathrm{T}}\mathbf{e} = \tilde{\mathbf{e}}^{\mathrm{T}}\tilde{\mathbf{e}} = \tilde{\mathbf{e}}_1^{\mathrm{T}}\tilde{\mathbf{e}}_1 + \tilde{\mathbf{e}}_2^{\mathrm{T}}\tilde{\mathbf{e}}_2,$$

so that $\mathbf{a}$ solves $R_1\mathbf{a} = \tilde{\mathbf{y}}_1 - T_{12}\tilde{\mathbf{e}}_2$.

Public-domain library software for solving (4.7) and, more generally, computing generalised QR factorisations is available [190].

Uncertainty matrix associated with the Gauss-Markov estimate. If $\mathbf{a}$ is the Gauss-Markov estimate, the associated uncertainty matrix $V_{\mathbf{a}}$ is given by

$$V_{\mathbf{a}} = (C^{\mathrm{T}} V^{-1} C)^{-1}.$$

Details. In terms of the generalised QR factorisation [70],

$$V_{\mathbf{a}} = KK^{\mathrm{T}} \quad \text{where } K \text{ solves } R_1 K = T_{11}.$$

4.1.7 Linear least squares subject to linear equality constraints

Linear equality constraints of the form $D\mathbf{a} = \mathbf{z}$ where D is a $p \times n$ matrix, $p < n$, can be reduced using an orthogonal factorisation approach. Such constraints arise in the application of resolving constraints to remove degrees of freedom from the model (section 2.3.3).

Suppose D^{T} is of full column rank and has the QR factorisation

$$D^{\mathrm{T}} = U \begin{bmatrix} S_1 \\ 0 \end{bmatrix}, \tag{4.9}$$

and form U_1 from the first n and U_2 from the last $n-p$ columns of the orthogonal factor U. If $\mathbf{a}_0$ is any solution of $D\mathbf{a} = \mathbf{z}$, then for any $(n-p)$-vector $\tilde{\mathbf{a}}$, $\mathbf{a} = \mathbf{a}_0 + U_2\tilde{\mathbf{a}}$ automatically satisfies the constraints:

$$D\mathbf{a} = D\mathbf{a}_0 + DU_2\tilde{\mathbf{a}} = \mathbf{z} + S_1^{\mathrm{T}} U_1^{\mathrm{T}} U_2 \tilde{\mathbf{a}} = \mathbf{z},$$

since $U_1^{\mathrm{T}} U_2 = \mathbf{0}$. The optimisation problem

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\| \quad \text{subject to} D\mathbf{a} = \mathbf{z},$$

can be reformulated as the unconstrained linear least-squares problem

$$\min_{\tilde{\mathbf{a}}} \ \|\mathbf{y} - C(\mathbf{a}_0 + U_2\tilde{\mathbf{a}})\| = \min_{\tilde{\mathbf{a}}} \ \|\tilde{\mathbf{y}} - \tilde{C}\tilde{\mathbf{a}}\|,$$

where

$$\tilde{\mathbf{y}} = \mathbf{y} - C\mathbf{a}_0, \quad \tilde{C} = CU_2.$$

This approach to linear equality constraints is quite general and can be applied to different types of optimisation problems.

Uncertainty matrix associated with the linearly constrained LLS estimate. The constrained solution parameters are given by $\tilde{\mathbf{a}} = \tilde{C}^{\dagger}\tilde{\mathbf{y}}$ where $\tilde{C}^{\dagger} = (\tilde{C}^{\mathrm{T}}\tilde{C})^{-1}\tilde{C}^{\mathrm{T}}$ is the pseudo-inverse of $\tilde{C}$ (section 4.1.3). If $\mathbf{y}$ is associated with uncertainty matrix $V_{\mathbf{y}}$, then the uncertainty matrix $V_{\tilde{\mathbf{a}}}$ associated with $\tilde{\mathbf{a}}$ is given by

$$V_{\tilde{\mathbf{a}}} = \tilde{C}^{\dagger} V_{\mathbf{y}} (\tilde{C}^{\dagger})^{\mathrm{T}}.$$

This also requires that $V_{\tilde{\mathbf{y}}} = V_{\mathbf{y}}$, which follows from $\tilde{\mathbf{y}} = \mathbf{y} - C\mathbf{a}_0$.

In particular if $V_{\mathbf{y}} = \sigma^2 I$, then

$$V_{\tilde{\mathbf{a}}} = \sigma^2 (\tilde{C}^T \tilde{C})^{-1}.$$

Since the unconstrained parameters $\mathbf{a} = \mathbf{a}_0 + U_2\tilde{\mathbf{a}}$, the uncertainty matrix $V_{\mathbf{a}}$ associated with $\mathbf{a}$ is

$$V_{\mathbf{a}} = U_2 V_{\tilde{\mathbf{a}}} U_2^{\mathrm{T}}.$$

4.1.8 Using linear least-squares solvers

Software for solving linear least-squares systems is generally straightforward to use. The user has to supply the observation matrix C and the right hand side vector $\mathbf{y}$ as inputs. The software will calculate the solution parameters $\mathbf{a}$ and the residual vector $\mathbf{r} = \mathbf{y} - C\mathbf{a}$. If the software uses an orthogonal factorisation approach (as can be recommended) then the triangular factor R_1 of the observation matrix is useful output as all necessary statistics can be determined efficiently using R_1 and $\mathbf{r}$.

4.1.9 The Gauss-Markov theorem

Least-squares methods are the most common estimators implemented and are appropriate for many practical model fitting problems. For linear models the following *Gauss-Markov Theorem* [150, chapter 6] can be used to justify their use:

> Gauss-Markov Theorem *For models of the form*
>
> $$\mathbf{y} = C\mathbf{a} + \boldsymbol{\epsilon},$$
>
> *where C is an $m \times n$ full rank matrix, $m \geq n$, and for which the random effects modelled by $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_m)^{\mathrm{T}}$ are observations of a vector of random variables $\boldsymbol{E}$ with variance $V(\boldsymbol{E}) = \sigma^2 I$, the linear least-squares estimator*
>
> $$\mathcal{A}(\mathbf{y}) = (C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}\mathbf{y}$$
>
> *is unbiased, i.e., $\mathcal{A}(\mathbf{y})$ is an observation of a vector of random variables $\boldsymbol{A}$ with expectation $E(\boldsymbol{A}) = \mathbf{a}$, and has a smaller variance matrix $V(\boldsymbol{A})$ than that for any other linear estimator.*

From this point of view, least-squares estimation is optimal for these models.

Note that there is no assumption that the random effects are normally or even symmetrically distributed, only that they are uncorrelated and have equal variance. This generality supports the use of least-squares methods.

Assumptions about normality are usually only invoked when it is required to provide coverage interval associated with the fitted parameters. A consequence of the Gauss-Markov theorem is that if the uncertainty matrix associated with the data is $V_{\mathbf{y}}$ then the corresponding Gauss-Markov estimator (4.6) is optimal.

4.1.10 Bibliography and software sources

Algorithms for solving linear least-squares systems are described in detail in [27, 119, 142, 207]. There are linear least-squares solvers in the NAG and IMSL libraries, LINPACK, MINPACK, LAPACK, DASL and Matlab, for example [8, 87, 115, 156, 173, 190, 203]. See also [126, 180].

4.2 Nonlinear least squares

4.2.1 Description

The nonlinear least-squares problem is: given m functions $f_i(\mathbf{a})$ of parameters $\mathbf{a} = (a_1, \ldots, a_n)$, $m \geq n$, solve

$$\min_{\mathbf{a}} \; F(\mathbf{a}) = \frac{1}{2} \sum_{i=1}^{m} \; f_i^2(\mathbf{a}). \tag{4.10}$$

(The fraction $\frac{1}{2}$ is used so that related expressions are simpler.) Necessary conditions for $\mathbf{a}$ to be a solution are that

$$\frac{\partial F}{\partial a_j} = \sum_{i=1}^{m} f_i \frac{\partial f_i}{\partial a_j} = 0, \quad j = 1, \ldots, n.$$

Defining the *Jacobian matrix* $J = J(\mathbf{a})$ by

$$J_{ij} = \frac{\partial f_i}{\partial a_j}(\mathbf{a}), \tag{4.11}$$

this condition can be written as $J^{\mathrm{T}}(\mathbf{a})\mathbf{f}(\mathbf{a}) = \mathbf{0}$.

Nonlinear least-squares estimators are used widely in metrology in situations where the response variable is modelled as a nonlinear function $y = \phi(\mathbf{x}, \mathbf{a})$ of the model parameters $\mathbf{a}$ and covariates $\mathbf{x}$. They have good bias and efficiency properties for models in which the measurements of the response variable are subject to uncorrelated random effects:

$$\begin{aligned} &y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad i = 1, \ldots, m \geq n, \\ &\boldsymbol{\epsilon} \in \boldsymbol{E}, \quad E(\boldsymbol{E}) = \mathbf{0}, \quad V(\boldsymbol{E}) = \sigma^2 I. \end{aligned}$$

If $\boldsymbol{E} \in N(\mathbf{0}, \sigma^2 I)$, then the nonlinear least-squares estimate is the maximum likelihood estimate of $\mathbf{a}$. Nonlinear least-squares estimators are suitable for any

system for which the random effects are associated with the measurements of the response variable and these random effects are independently distributed with zero mean and approximately equal standard deviations.

Nonlinear least squares are less suitable (without modification) for data in which more than one variable is subject to significant random effects (section 4.3), data which contains outliers (section 4.7) or where there is significant correlation associated with the random effects (section 4.2.5).

4.2.2 Algorithms for nonlinear least squares

Gauss-Newton algorithm for minimising a sum of squares. The Gauss-Newton algorithm is a modification of Newton's algorithm for minimising a function. Let

$$F(\mathbf{a}) = \frac{1}{2}\sum_{i=1}^{m} f_i^2(\mathbf{a})$$

and let $J(\mathbf{a})$ be the Jacobian matrix $J = \partial f_i/\partial a_j$.

Then (in the notation of section 3.6) $\mathbf{g} = J^{\mathrm{T}}\mathbf{f}$ and $H = J^{\mathrm{T}}J + G$, where

$$G_{jk} = \sum_{i=1}^{m} f_i \frac{\partial^2 f_i}{\partial a_j \partial a_k}.$$

The Gauss-Newton (GN) algorithm follows the same approach as the Newton algorithm (section 3.6), only that in determining the update step, H is approximated by $J^{\mathrm{T}}J$, i.e., the term G is ignored and $\mathbf{p}$ is found by solving $J^{\mathrm{T}}J\mathbf{p} = -J^{\mathrm{T}}\mathbf{f}$. This is corresponds to the linear least-squares problem $J\mathbf{p} = -\mathbf{f}$ and can be solved using an orthogonal factorisation approach, for example; see section 4.1. The Gauss-Newton algorithm in general converges linearly at a rate that depends on the condition of the approximation problem, the size of the residuals $\mathbf{f}$ near the solution and the curvature. If the problem is well-conditioned, the residuals are small and the summand functions f_i are nearly linear, then $J^{\mathrm{T}}J$ is a good approximation to the Hessian matrix H and convergence is fast.

Gauss-Newton with line search. In practice, the update step is often of the form $\mathbf{a} = \mathbf{a} + t\mathbf{p}$ where the step length parameter t is chosen to ensure there is a sufficient decrease in the value of the objective function $F(\mathbf{a})$ at each iteration. If $\xi(t) = F(\mathbf{a} + t\mathbf{p})$, then $\xi'(0) = \mathbf{g}^{\mathrm{T}}\mathbf{p}$ and

$$\rho(t) = \frac{\xi(t) - \xi(0)}{t\xi'(0)} \tag{4.12}$$

is the ratio of the actual decrease to the predicted decrease. If ξ is quadratic and t^* minimises F along $\mathbf{a} + t\mathbf{p}$, then $\rho(t^*) = 1/2$. A line search strategy is designed to find a value t^* such that

$$1 - \eta > \rho(t^*) > \eta \tag{4.13}$$

for some pre-assigned $0 < \eta < 1/2$. A simple one-step strategy is to evaluate $\rho(1)$ and if it does not satisfy (4.13), set

$$t = \frac{1}{2(1 - \rho(1))},$$

the minimum of the parabola defined from $\xi(0)$, $\xi(1)$ and $\xi'(0)$.

Gauss-Newton with trust regions. The introduction of a line search is designed to improve the convergence characteristics of the Gauss-Newton algorithm. Another approach to help make the algorithm more robust is based on the concept of a trust region. In this approach, the step taken at each stage is restricted to a region in which a quadratic approximation centred at the current solution estimate to the function being minimised is judged to be valid. The size of the trust region is adjusted depending on the progress of the algorithm. See, for example, [94, 160]. A Levenberg-Marquardt trust region algorithm for nonlinear least squares is implemented in MINPACK [115].

Termination criteria. A second practical issue is concerned with convergence criteria usually involving i) the change in the objective function $\Delta F = F(\mathbf{a}) - F(\mathbf{a} + \mathbf{p})$, ii) the norm $\|\mathbf{p}\|$ of the step, and iii) the norm $\|\mathbf{g}\|$ of the gradient. Ideally, the criteria should be invariant with respect to changes of scale in the objective function and parameters.

A Gauss-Newton algorithm works well for problems where i) a good initial guess of the solution parameters is available, ii) the Jacobian matrix at the solution is reasonably well-conditioned, and iii) the functions f_i are not highly nonlinear. Well-designed least-squares optimisation algorithms will still work satisfactorily even if not all of these conditions apply.

Taking into account sparsity structure in the Jacobian matrix. Since the main step in the Gauss-Newton algorithm is the solution of a linear least-squares system, structured or sparse matrix techniques can be used in nonlinear least-squares problems [67].

4.2.3 Uncertainty associated with the fitted parameters

In the context of model fitting, suppose $f_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$ and that the uncertainty matrix associated with $\mathbf{y}$ is $V_\mathbf{y}$. Then the uncertainty matrix associated with the fitted parameters is approximated by

$$V_\mathbf{a} = H^{-1} J^{\mathrm{T}} V_\mathbf{y} J H^{-1} \tag{4.14}$$

where H is the Hessian matrix evaluated at the solution.

Details. Since $f_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$, we can regard $\mathbf{f} = \mathbf{f}(\mathbf{a}, \mathbf{y})$ as a function of both $\mathbf{y}$ and $\mathbf{a}$. The n equations $\mathbf{g}(\mathbf{a}, \mathbf{y}) = J^{\mathrm{T}}(\mathbf{a})\mathbf{f}(\mathbf{a}, \mathbf{y}) = \mathbf{0}$ which must hold at a minimum implicitly define $\mathbf{a} = \mathbf{a}(\mathbf{y})$ as a function of $\mathbf{y}$. In order to calculate the uncertainty

matrix $V_{\mathbf{a}}$ we need to calculate the sensitivity matrix K with $K_{ji} = \partial a_j / \partial y_i$. Taking derivatives of the equation $\mathbf{g}(\mathbf{a}(\mathbf{y}), \mathbf{y}) = \mathbf{0}$ with respect to $\mathbf{y}$ yields

$$HK + J^{\mathrm{T}} = \mathbf{0}$$

so that $K = -H^{-1}J^{\mathrm{T}}$ as appears in (4.14).

If $V_{\mathbf{y}} = \sigma^2 I$ and using the approximation $H \approx J^{\mathrm{T}}J$, we have

$$V_{\mathbf{a}} \approx (J^{\mathrm{T}}J)^{-1}J^{\mathrm{T}}\sigma^2 I J (J^{\mathrm{T}}J)^{-1} = \sigma^2 (J^{\mathrm{T}}J)^{-1}. \tag{4.15}$$

If J has QR factorisation $J = Q_1 R_1$ at the solution where R_1 is an $n \times n$ upper-triangular matrix then $V_{\mathbf{y}} \approx \sigma^2 (R_1^{\mathrm{T}} R_1)^{-1}$. A posterior estimate $\hat{\sigma}$ of σ can be determined from the vector $\mathbf{f}$ of residuals at the solution according to

$$\hat{\sigma} = \frac{\|\mathbf{f}\|}{(m-n)^{1/2}}.$$

Both (4.14) and (4.15) are based on linearisations and therefore can only provide an estimate of the variance associated with the fitted parameters. For highly nonlinear models (with relatively large curvature) these estimates may be significantly different from the true variances. Monte Carlo simulation techniques, for example, can be used either to validate these estimates (section 8.2) or provide alternative estimates that do not involve any linearising approximations.

4.2.4 Weighted nonlinear least-squares estimator

If the functions f_i relate to random effects ϵ_i with differing variances σ_i^2, then the appropriate estimator is a weighted nonlinear least-squares estimator which estimates $\mathbf{a}$ by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} w_i^2 f_i^2(\mathbf{a}),$$

with $w_i = 1/\sigma_i$. Algorithms for the unweighted nonlinear least squares can be easily adapted to deal with the weighted case by applying them to $\tilde{f}_i = w_i f_i$.

4.2.5 Nonlinear Gauss-Markov estimator

If the covariance matrix associated with $\boldsymbol{\epsilon}$ is V, assumed to be full rank, the appropriate estimate of the model parameters is the one that solves

$$\min_{\mathbf{a}} \ \mathbf{f}^{\mathrm{T}}(\mathbf{a}) V^{-1} \mathbf{f}(\mathbf{a}).$$

As in the linear case, we can use the Cholesky decomposition of $V = LL^{\mathrm{T}}$ to convert this problem to a standard nonlinear least-squares problem applied to

$$\tilde{\mathbf{f}} = L^{-1}\mathbf{f}.$$

As for the case of linear least squares, if V and hence L is poorly conditioned the formation and use of L^{-1} could lead to numerical instability. The Gauss-Newton

algorithm can be adapted so that at each iteration the Gauss-Newton step is found by solving

$$\min_{\mathbf{a},\mathbf{e}} \mathbf{e}^{\mathrm{T}}\mathbf{e} \quad \text{subject to constraints} \mathbf{y} = -J\mathbf{p} + L\mathbf{e},$$

using, for example, the generalised QR decomposition (section 4.1.6).

4.2.6 Nonlinear least squares subject to linear constraints

Algorithms for nonlinear least squares can also be adapted to problems with p linear constraints $D\mathbf{a} = \mathbf{z}$ on the parameters, $p < n$. As described in section 4.1.7, the optimisation problem can be reposed as an unconstrained problem of the form

$$\min_{\tilde{\mathbf{a}}} \sum_{i=1}^{m} \tilde{f}_i^2(\tilde{\mathbf{a}}) \tag{4.16}$$

where $\tilde{f}_i(\tilde{\mathbf{a}}) = f_i(\mathbf{a}_0 + U_2\tilde{\mathbf{a}})$. Here $\mathbf{a}_0$ is any set of parameters satisfying the constraints, i.e., $D\mathbf{a}_0 = \mathbf{z}$, $\tilde{\mathbf{a}}$ represents the reduced set of $(n-p)$ parameters and U_2 is an $n \times (n-p)$ orthogonal matrix (derived from the QR factorisation of D^{T}) such that $DU_2 = \mathbf{0}$. Note that if J is the Jacobian matrix of partial derivatives $J_{ij} = \frac{\partial f_i}{\partial a_j}$, then the Jacobian matrix associated with (4.16) is given by $\tilde{J} = JU_2$.

4.2.7 Using nonlinear least-squares solvers

Software for solving nonlinear least-squares systems is in principle straightforward to use. The user has to supply a software module to calculate the vector of function values $\mathbf{f}$ and the Jacobian matrix J of partial derivatives for a given value of the optimisation parameters $\mathbf{a}$. For complicated models, the correct calculation of these derivatives can involve a lot of effort both in deriving the correct formulæ and in their subsequent implementation in software. For this reason, many optimisation packages offer versions of the algorithms for which only function values are required and use finite difference approximations of the form

$$\frac{\partial f}{\partial a_j}(\mathbf{a}) \approx \frac{f(a_1, \ldots, a_j + \Delta_j, a_{j+1}, \ldots, a_n) - f(a_1, \ldots, a_n)}{\Delta_j}$$

to estimate the derivatives. This is done at the cost of accuracy of the solution and usually efficiency of the underlying algorithm. There is much current research on finding better ways of estimating derivatives. Automatic differentiation techniques, including forward and reverse accumulation, and the complex step method and their use in metrology are described in [30]. The complex step method is particularly easy to implement in languages such as Matlab or Fortran 90/95 that support complex arithmetic.

The user has also to supply an initial estimate of the optimisation parameters. For most metrology applications, this is not a usually a problem but there are situations where this is a major difficulty.

The optimisation software will calculate the solution parameters $\mathbf{a}$ and the vector of function values $\mathbf{f}$ at the solution. If the software uses an orthogonal factorisation approach in the iterative step then the triangular factor R_1 of the Jacobian matrix at the solution is useful output as all necessary statistics can be determined efficiently using R_1 and $\mathbf{f}$.

4.2.8 Bibliography and software sources

There are a number of nonlinear least-squares solvers in MINPACK and the NAG and IMSL libraries [115, 173, 203]. Nonlinear least-squares algorithms are described in [94, 117], for example. See also [161]. For more on automatic differentiation, see for example, [20, 30, 121, 196].

4.3 Generalised distance regression (GDR)

4.3.1 Description

Linear and nonlinear least-squares estimators are appropriate if only one measured variable is subject to significant random effects. However, in many metrological situations, there is significant uncertainty associated with more than one of the measured variables and it is important to take this into account in determining parameter estimates that are free from significant bias.

In a generalised distance regression (GDR) formulation, it is assumed that each set of measurements $\mathbf{x}_i$ is subject to random effects so that $\mathbf{x}_i = \mathbf{x}_i^* + \boldsymbol{\epsilon}_i$, where $\mathbf{x}_i^*$ satisfies the model constraints $f(\mathbf{x}_i^*, \mathbf{a}) = 0$ for some unknown $\mathbf{a}$. The set of measurements $\mathbf{x}$ subsumes both the stimulus variables and the response variable (y). In this formulation, y is treated on the same footing as the other components of $\mathbf{x}$.

It is assumed that the effects modelled by $\boldsymbol{\epsilon}_i$ associated with the components of $\mathbf{x}_i$ can be correlated with each other, but that the ith and jth sets are uncorrelated, $i \neq j$. (More general uncertainty structures are considered in [100], for example.) If V_i is the uncertainty (covariance) matrix associated with $\boldsymbol{\epsilon}_i$[4] (assumed to be full rank), then maximum likelihood estimates of the model parameters $\mathbf{a}$ can be found by solving

$$\min_{\mathbf{a},\{\mathbf{x}_i^*\}} \sum_{i=1}^{m} (\mathbf{x}_i - \mathbf{x}_i^*)^{\mathrm{T}} V_i^{-1} (\mathbf{x}_i - \mathbf{x}_i^*) \tag{4.17}$$

subject to the model constraints $f(\mathbf{x}_i^*, \mathbf{a}) = 0$. This is an implicit formulation of the problem. If the surface $f(\mathbf{x}, \mathbf{a}) = 0$ can be represented explicitly (i.e., parametrically) as $\mathbf{x} = \boldsymbol{\phi}(\mathbf{u}, \mathbf{a})$, then (4.17) can be reformulated as

$$\min_{\mathbf{a},\{\mathbf{u}_i^*\}} \sum_{i=1}^{m} (\mathbf{x}_i - \boldsymbol{\phi}(\mathbf{u}_i^*, \mathbf{a}))^{\mathrm{T}} V_i^{-1} (\mathbf{x}_i - \boldsymbol{\phi}(\mathbf{u}_i^*, \mathbf{a})), \tag{4.18}$$

[4] That is, $\boldsymbol{\epsilon}_i \in \boldsymbol{E}_i$ and $V(\boldsymbol{E}_i) = V_i$.

an unconstrained optimisation problem. If each $V_i = I$, the identity matrix, the GDR problem is known as *orthogonal regression*. Orthogonal regression for linear models is sometimes termed *total least squares*.

Generalised distance regression methods have not been used extensively until recent years. A typical situation for which they are appropriate is where the response $y = \phi(x, \mathbf{a})$ is modelled as a function of the variable x and parameters $\mathbf{a}$, and both y and x are measured subject to random effects, giving rise to observation equations of the form

$$x_i = u_i^* + \delta_i, y_i = \phi(x_i^*) + \epsilon_i, \quad \delta_i \in N(0, \sigma_x^2), \epsilon_i \in N(0, \sigma_y^2).$$

The maximum likelihood estimate of the parameters is found by solving

$$\min_{\mathbf{a}, \{u_i^*\}} \sum_{i=1}^{m} \left\{ \left(\frac{x_i - u_i^*}{\sigma_x} \right)^2 + \left(\frac{y_i - \phi(u_i^*, \mathbf{a})}{\sigma_y} \right)^2 \right\}.$$

Orthogonal regression is used extensively in co-ordinate metrology.

4.3.2 Algorithms for generalised distance regression

Separation of variables approaches. At first sight, both generalised regression formulations (4.17) and (4.18) represent significantly more challenging optimisation problems than standard nonlinear least-squares problems as they have to take into account the additional parameters, etc. However, using a separation-of-variables approach, it is possible to convert them to standard nonlinear least-squares problems in the parameters $\mathbf{a}$. We consider the explicit case (4.18) first.

We assume that V is a symmetric, strictly positive definite matrix with inverse $M = V^{-1}$. Denote by $\mathbf{u}_i^* = \mathbf{u}_i^*(\mathbf{a})$ the solution of the footpoint problem (FPP)

$$\min_{\mathbf{u}} D(\mathbf{u}) = (\mathbf{x}_i - \boldsymbol{\phi}(\mathbf{u}, \mathbf{a}))^{\mathrm{T}} M (\mathbf{x}_i - \boldsymbol{\phi}(\mathbf{u}, \mathbf{a})), \tag{4.19}$$

and define $\langle\ ,\ \rangle_M$ and $\|\ \|_M$ by

$$\langle \mathbf{x}, \mathbf{y} \rangle_M = \mathbf{x}^{\mathrm{T}} M \mathbf{y}, \quad \|\mathbf{x}\|_M = \langle \mathbf{x}, \mathbf{x} \rangle_M^{1/2}.$$

If $\mathbf{n}_i$ is the Euclidean unit normal to the surface at $\mathbf{x}_i^* = \boldsymbol{\phi}(\mathbf{u}_i^*, \mathbf{a})$, then it is straightforward to show that if we define the *generalised distance* $d_i = d_i(\mathbf{a})$ by

$$d_i = \langle \mathbf{x}_i - \mathbf{x}_i^*, \mathbf{n}_i \rangle / \|\mathbf{n}_i\|_M, \tag{4.20}$$

then

$$d_i^2 = \|\mathbf{x}_i - \mathbf{x}_i^*\|_M^2 = D(\mathbf{u}_i^*),$$

and

$$\frac{\partial d_i}{\partial a_j} = -\langle \frac{\partial \mathbf{f}}{\partial a_j}, \mathbf{n}_i \rangle / \|\mathbf{n}_i\|_M. \tag{4.21}$$

In this way, the explicit generalised distance regression problem can be posed as a standard nonlinear least-squares problem $\min_{\mathbf{a}} d_i^2(\mathbf{a})$ where each function and

its gradient is calculated as in (4.20) and (4.21) with all quantities evaluated at the solution $\mathbf{u}_i^*$ of the appropriate footpoint problem.

Example: simple GDR for parametric curves

The simple GDR problem for parametric curves can be stated as: given data points $\{(x_i, y_i)\}_1^m$ and strictly positive weights $\{(\alpha_i, \beta_i)\}_1^m$, minimise

$$\sum_{i=1}^{m} \left\{\alpha_i^2(x_i - \phi(u_i, \mathbf{a}))^2 + \beta_i^2(y_i - \psi(u_i, \mathbf{a}))^2\right\}$$

with respect to $\mathbf{a}$ and $\{u_i\}_1^m$ where $(\phi, \psi) = (\phi(u, \mathbf{a}), \psi(u, \mathbf{a}))$ is a parametric curve in $\mathbb{R}^2$. The theory above shows that this can be reformulated as:

$$\min_{\mathbf{a}} \sum_{i=1}^{m} d_i^2(\mathbf{a})$$

with

$$\begin{aligned} d_i &= \frac{1}{s_i}\left(-(x_i - \phi_i^*)\dot{\psi}_i + (y_i - \psi_i^*)\dot{\phi}_i\right), \\ \frac{\partial d_i}{\partial a_j} &= \frac{1}{s_i}\left(\frac{\partial \phi_i}{\partial a_j}\dot{\psi}_i - \frac{\partial \psi_i}{\partial a_j}\dot{\phi}_i\right), \end{aligned}$$

where

$$\begin{aligned} \dot{\phi}_i &= \frac{\partial \phi_i}{\partial u}, \quad \text{etc.}, \\ s_i &= \left(\frac{\dot{\psi}_i^2}{\alpha_i^2} + \frac{\dot{\phi}_i^2}{\beta_i^2}\right)^{1/2}, \end{aligned}$$

with all expressions evaluated at the solution u_i^* of the corresponding footpoint problem:

$$\min_u \left\{\alpha_i^2(x_i - \phi(u, \mathbf{a}))^2 + \beta_i^2(y_i - \psi(u, \mathbf{a}))^2\right\}.$$

♯

For the implicit case (4.17), denote by $\mathbf{x}_i^* = \mathbf{x}_i^*(\mathbf{a})$ the solution of the implicit footpoint problem

$$\min_{\mathbf{x}} \; D(\mathbf{x}) = (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}} M (\mathbf{x}_i - \mathbf{x}) \text{subject to} \quad f(\mathbf{x}, \mathbf{a}) = 0. \tag{4.22}$$

Then the generalised distance $d_i(\mathbf{a})$ is given by

$$d_i = \langle \mathbf{x}_i - \mathbf{x}_i^*, \nabla_{\mathbf{x}} f \rangle / \|\nabla_{\mathbf{x}} f\|_M, \text{with} \quad \frac{\partial d_i}{\partial a_j} = \frac{\partial f}{\partial a_j} / \|\nabla_{\mathbf{x}} f\|_M, \tag{4.23}$$

evaluated at $\mathbf{x} = \mathbf{x}_i^*$. Thus, the implicit generalised distance regression problem can also be posed as a standard nonlinear least-squares problem where each function evaluation involves the calculation of the optimal footpoints.

Example: simple GDR for implicit curves

The simple GDR problem for implicit curves can be stated as: given data points $\{(x_i, y_i)\}_1^m$ and strictly positive weights $\{(\alpha_i, \beta_i)\}_1^n$, minimise

$$\sum_{i=1}^{m} \alpha_i^2 (x_i - x_i^*)^2 + \beta_i^2 (y_i - y_i^*)^2$$

with respect to $\mathbf{a}$ and $\{(x_i^*, y_i^*)\}_1^m$ subject to the constraints $f(x_i^*, y_i^*, \mathbf{a}) = 0$, $i = 1, \ldots, m$. The theory above shows that this can be reformulated as:

$$\min_{\mathbf{a}} \sum_{i=1}^{m} d_i^2(\mathbf{a})$$

with

$$\begin{aligned} d_i &= \frac{1}{s_i} \left((x_i - x_i^*) f_x + (y_i - y_i^*) f_y \right), \\ \frac{\partial d_i}{\partial a_j} &= \frac{1}{s_i} \frac{\partial f}{\partial a_j}, \end{aligned}$$

where

$$\begin{aligned} f_x &= \frac{\partial f}{\partial x}, \quad \text{etc.}, \\ s_i &= \left(\frac{f_x^2}{\alpha_i^2} + \frac{f_y^2}{\beta_i^2} \right)^{1/2}, \end{aligned}$$

with all expressions evaluated at the solution (x_i^*, y_i^*) of the corresponding footpoint problem. ♯

Structured least-squares approaches for explicit models. The GDR problem for explicit models (4.18) can be solved directly if inefficiently using standard nonlinear least-squares algorithms. However, the fact that $p - 1$ parameters $\mathbf{u}_i^*$ only appear in n equations means that the associated Jacobian matrix of partial derivatives has a block-angular structure with the diagonal blocks corresponding to the footpoint parameters $\mathbf{u}_i^*$:

$$J = \begin{bmatrix} K_1 & & & J_1 \\ & K_2 & & J_2 \\ & & \ddots & \vdots \\ & & K_m & J_m \end{bmatrix}, \tag{4.24}$$

where K_i is the matrix of derivatives of the ith set of observation equations with respect to the transformation parameters $\mathbf{u}_i^*$, and the border blocks J_i store their derivatives with respect to $\mathbf{a}$. The form of J is illustrated in figure 4.1.

The upper-triangular factor R of the Jacobian matrix also has a block-angular

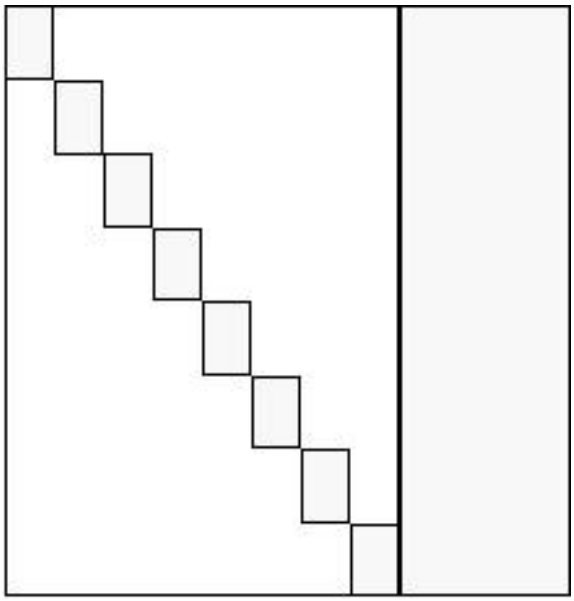

Figure 4.1: A block-angular Jacobian matrix J.

structure:

$$R = \begin{bmatrix} R_1 & & & & B_1 \\ & R_2 & & & B_2 \\ & & \ddots & & \vdots \\ & & & R_m & B_m \\ & & & & R_0 \end{bmatrix}, \tag{4.25}$$

where ${R_i}_1^m$, are $(p-1) \times (p-1)$ upper-triangular, $\{B_i\}_1^m$ are $(p-1) \times n$ border blocks and R_0 is the $n \times n$ upper-triangular factor corresponding to the parameters $\mathbf{a}$.

The use of structure exploiting algorithms for model fitting in metrology is discussed in [58, 67, 98, 104].

4.3.3 Approximate estimators for implicit models

We can find an approximate estimate of the solution parameters for the implicit GDR problem (4.17) by solving the least-squares problem

$$\min_{\mathbf{a}} \sum_{i=1}^{m} w_i^2 f(\mathbf{x}_i, \mathbf{a})^2,$$

where w_i are suitably chosen weights. Depending on the nature of the model and the uncertainty structure, this estimate may be fit for purpose or be used as an initial estimate in determining a refined estimate.

4.3.4 Orthogonal distance regression with linear surfaces

A linear surface in $\mathbb{R}^n$ (e.g., line in two dimensions, plane in three dimensions) is defined implicitly by an equation of the form

$$(\mathbf{x} - \mathbf{x}_0)^{\mathrm{T}}\mathbf{n} = 0,$$

where the n-vector $\mathbf{x}_0$ is a point lying in the surface and the n-vector $\mathbf{n}$ is a vector normal (orthogonal) to the surface. (Note that linear surfaces are not generally parameterised by this specification since the relationship is not one-to-one; for example any point $\mathbf{x}_0$ lying in the surface could be chosen.) The ODR problem for linear surfaces is: given data points $\{\mathbf{x}_i\}_1^m$ determine the linear surface which minimises $\sum_i d_i^2$ where $d_i = (\mathbf{x}_i - \mathbf{x}_0)^{\mathrm{T}}\mathbf{n}$ is the distance from $\mathbf{x}_i$ to the surface. It is straightforward to show that the best-fit surface passes through the centroid $\bar{\mathbf{x}}$

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i$$

of the data so its equation is of the form $(\mathbf{x} - \bar{\mathbf{x}})^{\mathrm{T}}\mathbf{n} = 0$. The normal vector $\mathbf{n}$ can be determined by solving

$$\min_{\mathbf{n},\|\mathbf{n}\|=1}\sum_{i=1}^{m}((\mathbf{x}_i - \bar{\mathbf{x}})^{\mathrm{T}}\mathbf{n})^2.$$

If $\bar{X}$ is the centred data matrix with ith row equal to $(\mathbf{x}_i - \bar{\mathbf{x}})^{\mathrm{T}}$, this problem can be posed as

$$\min_{\mathbf{n},\|\mathbf{n}\|=1}\|\bar{X}\mathbf{n}\|.$$

In other words, the solution $\mathbf{n}$ is the unit vector for which the norm of $\bar{X}\mathbf{n}$ takes its minimum value. From the definition of the singular value decomposition of a matrix (section 3.7.1), we see that the solution $\mathbf{n}$ is the right singular vector of $\bar{X}$ corresponding to the smallest singular value (equation (3.5)). Thus, if $\bar{X} = USV^{\mathrm{T}}$ is the singular value decomposition of $\bar{X}$ then $\mathbf{n} = \mathbf{v}_n$ specifies the normal vector to the ODR best-fit linear surface to the data points.

4.3.5 Bibliography and software sources

The case of orthogonal distance regression is considered in [3, 28, 39, 119, 130, 131, 133, 200, 201, 205], for example. The software package ODRPACK [29] provides a fairly comprehensive facility. Generalised distance regression is considered in [1, 19, 67, 70, 96, 98, 100, 104, 107, 108, 132]. The component XGENLINE for polynomial generalised distance regression is available for downloading from EUROMETROS [9, 92].

4.4 Generalised Gauss-Markov regression for curves

4.4.1 Description

Generalised Gauss-Markov regression combines generalised distance regression with nondiagonal uncertainty matrices. In this section, we describe it for models where the response y is a function of a single variable. The case for a response depending on more than one covariate is very similar. Suppose the model is $y = \phi(x, \mathbf{a})$ defined by parameters $\mathbf{a}$ and that we have data $\{(x_i, y_i)\}_{i=1}^m$ nominally lying on such a curve but subject to random effects with general uncertainty matrix V. We assume that V is full rank. Let $\mathbf{x} = (x_1, \ldots, x_m)^{\mathrm{T}}$, $\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$. The *generalised Gauss-Markov regression problem* [70] is

$$\begin{array}{c} \min \\ \mathbf{a}, \mathbf{x}^*, \mathbf{y}^* \\ y_i^* = \phi(x_i^*, \mathbf{a}) \end{array} \quad \left[\begin{array}{c} \mathbf{x} - \mathbf{x}^* \\ \mathbf{y} - \mathbf{y}^* \end{array} \right]^{\mathrm{T}} V^{-1} \left[\begin{array}{c} \mathbf{x} - \mathbf{x}^* \\ \mathbf{y} - \mathbf{y}^* \end{array} \right]. \tag{4.26}$$

4.4.2 Algorithms for generalised Gauss-Markov regression

The generalised Gauss-Markov problem is a type of nonlinear Gauss-Markov problem

$$\min_{\mathbf{a}, \mathbf{x}^*} \mathbf{f}^{\mathrm{T}} V^{-1} \mathbf{f}$$

where

$$\mathbf{f} = \mathbf{f}(\mathbf{x}^*, \mathbf{a}) = \left[\begin{array}{c} \mathbf{x} - \mathbf{x}^* \\ \mathbf{y} - \mathbf{y}^* \end{array} \right], \quad y_i^* = \phi(x_i^*, \mathbf{a}),$$

and can be solved using nonlinear least-squares algorithms (section 4.2.5). The Jacobian matrix associated with $\mathbf{f}$ is the $2m \times (m + n)$ matrix J with

$$J = - \left[\begin{array}{cc} I & \mathbf{0} \\ J_x & J_{\mathbf{a}}^* \end{array} \right], \tag{4.27}$$

where $J_{\mathbf{a}}^*$ is the $m \times n$ matrix with $J_{\mathbf{a}}^*(i, j) = \partial \phi(x_i^*, \mathbf{a}) / \partial a_j$ and J_x is the diagonal matrix with $J_x(i, i) = \partial \phi(x_i^*, \mathbf{a}) / \partial x$.

4.5 Linear Chebyshev (L_∞) estimator

4.5.1 Description

Given data $\{(\mathbf{x}_i, y_i)\}_1^m$ and the linear model

$$y = a_1 \phi_1(\mathbf{x}) + \ldots + a_n \phi_n(\mathbf{x}),$$

$n \leq m$, the Chebyshev estimate of the parameters $\mathbf{a}$ is the one which solves

$$\min_{\mathbf{a}} \; F(\mathbf{a}) = \max_i |y_i - \mathbf{c}_i^{\mathrm{T}} \mathbf{a}|,$$

where $\mathbf{c}_i = (\phi_1(\mathbf{x}_i), \ldots, \phi_n(\mathbf{x}_i))^{\mathrm{T}}$. If s is the minimum value of $F(\mathbf{a})$, at least $n+1$ of the terms $|y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|$ will be equal to s [183, 204]. Chebyshev estimates minimise the maximum approximation error rather than an error aggregated over all the data (as in least squares).

Chebyshev estimation is used widely in approximation where it is required to fit a curve or data set uniformly well across the range. In particular Chebyshev estimation can be regarded as a maximum likelihood estimator for linear models in which the measurements of a single response variable is subject to uncorrelated uniformly distributed random effects:

$$y_i = a_1\phi_1(\mathbf{x}_i) + \ldots + a_n\phi_n(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \in U(-S, S), i = 1, \ldots, m \geq n.$$

Chebyshev approximation (usually nonlinear) is used in dimensional metrology to estimate the maximum departure of an artefact/manufactured part from its nominal shape.

Linear Chebyshev estimators are less suitable for data in which more than one variable is subject to significant random effects and should not be used for data which contains outliers or rogue points.

Example: averaging

In the simple case of fitting a constant to a set of values, the Chebyshev solution is the midrange, i.e., the average of the maximum and minimum values. ♯

4.5.2 Algorithms for linear Chebyshev approximation

The Chebyshev approximation problem can be reformulated as

$$\min_{\mathbf{a},s} \; s$$

subject to the linear inequality constraints

$$-s \leq y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a} \leq s, \quad i = 1, \ldots, m.$$

This is a linear programming problem and can be solved by the simplex algorithm of Dantzig [83] (not to be confused with the simplex method of Nelder and Mead [168] for unconstrained minimisation). At the solution, at least $n+1$ of the inequalities hold as equalities so the solution can be found by determining the correct subset of $n+1$ constraints. From an initial choice of $n+1$ constraints, the simplex algorithm systematically updates this selection until the solution is found.

4.5.3 Bibliography and software sources

Linear Chebshev approximation is considered in [18, 183, 204], linear programming in [94, 117], for example. The algorithm of Barrodale and Philips [13] is widely used. There is a linear Chebyshev solver in the Matlab optimisation

Toolbox and the NAG library [156, 173] and linear programming software in the IMSL and NAG libraries [203, 173]; see also [161]. The use of Chebyshev approximation in coordinate metrology is discussed in [5, 6, 41, 42, 99, 133].

4.6 Linear L_1 estimation

4.6.1 Description

Given data $\{(\mathbf{x}_i, y_i)\}_1^m$ and the linear model

$$y = a_1\phi_1(\mathbf{x}) + \ldots + a_n\phi_n(\mathbf{x}),$$

$n \leq m$, the L_1 estimate of the parameters $\mathbf{a}$ is the one which solves

$$\min_{\mathbf{a}} \; F(\mathbf{a}) = \sum_{i=1}^{m} |y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|,$$

where $\mathbf{c}_i = (\phi_1(\mathbf{x}_i), \ldots, \phi_n(\mathbf{x}_i))^{\mathrm{T}}$. At the solution, at least n of the terms $|y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|$ will be zero and the L_1 estimate approximately balances the number and distribution of the vectors $\mathbf{c}_i$ associated with a positive residual with those associated with a negative [204]. Importantly, the magnitudes of the residuals are not important. For this reason, L_1 estimates are not particularly influenced by outliers or rogue points in the data.

Linear L_1 approximation methods are not commonly used in metrology. However, their ability to produce a good fit to the majority of the data in the presence of outliers can be very useful for systems that have normally distributed random effects in general but in which large, sporadic errors can occur, for example in measuring a surface in which there are a small number of cracks. For normally distributed random effects, the L_1 estimate can be expected to be reasonably close to a least-squares estimate.

Example: averaging

In the simple case of fitting a constant to a set of values, the L_1 solution is the median. ♯

Example: Comparing least-squares and L_1 line fits.

Figure 4.2 shows the least-squares and L_1 line fits to 12 data points with two 'outliers'. The L_1 fit (dotted line) completely ignores the large errors associated with points 3 and 11, well approximating the body of the data. In contrast, the least-squares fit is skewed towards the outliers. ♯

4.6.2 Algorithms for linear L_1 approximation

The L_1 approximation problem can be reformulated as

$$\min_{\mathbf{a},\{s_i\}} \; \sum_{i=1}^{m} s_i$$

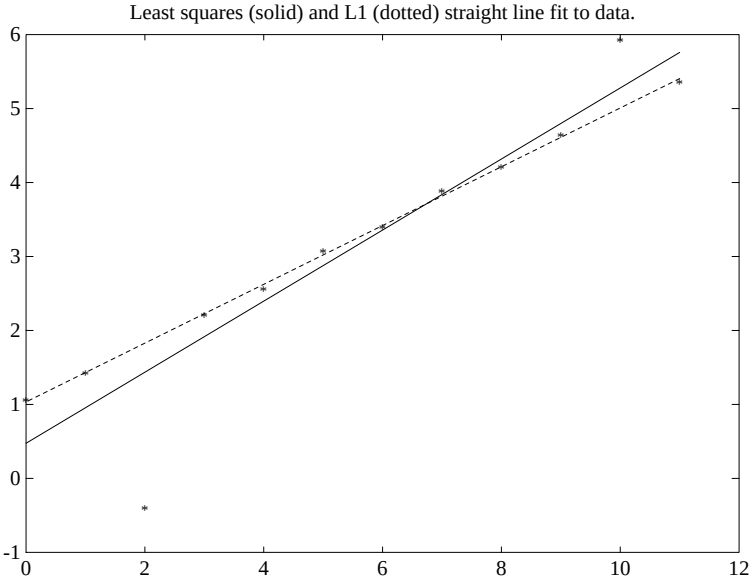

Figure 4.2: Least-squares and L_1 line fits to data with two outliers.

subject to the linear inequality constraints

$$-s_i \leq y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a} \leq s_i, \quad i = 1, \ldots, m.$$

At the solution $s_i = |y_i - \mathbf{c}_i^{\mathrm{T}}\mathbf{a}|$.

This is a linear programming problem and, as in the case of linear Chebyshev approximation (section 4.5), can be solved by the simplex algorithm [83]. The introduction of the potentially large number of parameters s_i means that a straightforward application of this algorithm would be inefficient. However, with modification the L_1 approximation problem can be solved effectively using a simplex-type method.

4.6.3 Bibliography and software sources

Linear L_1 approximation is considered in [14, 17, 143, 144, 183, 204], for example. The algorithms of Barrodale and Philips [15] and Bartels and Conn [16] are widely used.

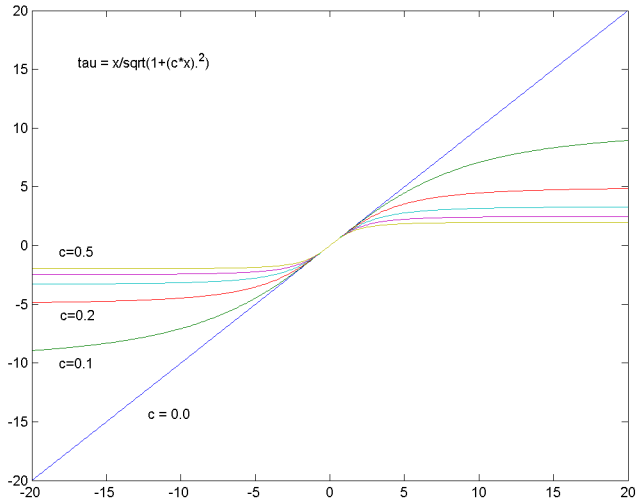

Figure 4.3: Graph of τ defined in (4.29) for different values of c.

4.7 Asymptotic least squares (ALS)

4.7.1 Description

Asymptotic least squares (ALS) is a form of nonlinear least-squares approximation in which a nonlinear transformation is applied in order to reduce the effect of large approximation errors associated with outliers or rogue data points. An asymptotic least-squares estimate minimises an objective function of the form

$$\tilde{F}(\mathbf{a}) = \frac{1}{2}\sum_{i=1}^{m} \tilde{f}_i(\mathbf{a})^2, \qquad \tilde{f}_i = \tau(f_i), \tag{4.28}$$

where $\tau(x)$ a transformation function having the following properties: i) τ has continuous second derivatives so that minimising $\tilde{F}$ is a smooth optimisation problem, ii) $\tau(0) = 0$, $\tau'(0) = 1$ and $\tau''(0) = 0$ so that for small f_i, $\tilde{F}$ has similar behaviour to a standard least-squares objective function, and iii) $\lim_{|x|\to\infty} \tau'(x) = 0$ so that increasing an already large approximation error will have a marginal effect on $\tilde{F}$. A simple function satisfying these criteria is

$$\tau(x) = x/(1 + c^2x^2)^{1/2}; \tag{4.29}$$

see figure 4.3. We note that $\lim_{x\to\pm\infty} \tau(x) = \pm 1/c$ and has the correct asymptotic behaviour

Asymptotic least squares is appropriate for models of the form

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i + \omega_i, \quad \boldsymbol{\epsilon} \in \boldsymbol{E}, \quad E(\boldsymbol{E}) = \mathbf{0}, \quad V(\boldsymbol{E}) = \sigma^2 I,$$

and $\omega_i = 0$ for most of the measurements but there is a possibility that for some of the data points ω_i could be large relative to σ. For this model, an appropriate form of τ is

$$\tau(x) = (x/\sigma)/(1 + c^2(x/\sigma)^2)^{1/2}. \tag{4.30}$$

The parameter c in (4.29) controls the level of ϵ at which the transform takes effect (figure 4.3). If $\boldsymbol{E} \sim N(\mathbf{0}, \sigma^2 I)$, we would expect approximately 95% of the deviations $y_i - \phi(\mathbf{x}_i, \mathbf{a})$ to lie in the interval$[-2\sigma, 2\sigma]$. In this region, we want τ to make a small change, suggesting a value of c in the region of $c = 1/4$.

4.7.2 Algorithms for asymptotic least squares

Even if f_i is linear in the parameters $\mathbf{a}$ the introduction of the nonlinear τ function makes the minimisation of $\tilde{F}$ a nonlinear least-squares problem.

To employ a Newton-type algorithm to minimise $\tilde{F}(\mathbf{a})$, we need to calculate

$$\mathbf{g} = \tilde{J}^{\mathrm{T}}\tilde{\mathbf{f}}, \quad \tilde{J}_{ij} = \dot{\tau}_i \frac{\partial f_i}{\partial a_j}, \quad \dot{\tau}_i = \frac{d\tau}{dx}(f_i),$$

and

$$\tilde{H} = \tilde{J}^{\mathrm{T}}\tilde{J} + \tilde{G}, \quad \tilde{G}_{jk} = \sum_i \tilde{f}_i \frac{\partial^2 \tilde{f}_i}{\partial a_j \partial a_k}.$$

We note that

$$\frac{\partial^2 \tilde{f}_i}{\partial a_j \partial a_k} = \ddot{\tau}_i \frac{\partial f_i}{\partial a_j}\frac{\partial f_i}{\partial a_k} + \dot{\tau}_i \frac{\partial^2 f_i}{\partial a_j \partial a_k}, \qquad \ddot{\tau}_i = \frac{d^2\tau}{d^2x}(f_i).$$

The first term on the right is the contribution due to the curvature in τ, the second, due to that in F. Even if the second term is small, the first term is likely to be significant. This means that in practice the Gauss-Newton algorithm implemented for ALS will have significantly slower convergence than a Newton algorithm. However, if f is linear with $\mathbf{f} = \mathbf{y} - C\mathbf{a}$, the second term is zero and a Newton algorithm can be implemented easily with $\tilde{J}$ and $\tilde{G}$ calculated using the following identities:

$$\tilde{J}_{ij} = -c_{ij}\dot{\tau}_i, \quad \tilde{G}_{jk} = \sum_i \tau_i \ddot{\tau}_i c_{ij} c_{ik}.$$

4.7.3 Uncertainty associated with the fitted parameters

Since the ALS method is a form of nonlinear least squares the approach given in section 4.2.3 is applicable. Since the τ function is likely to introduce significant curvature, $V_{\mathbf{a}}$ evaluated using the Hessian matrix (4.14), rather than its approximation (4.15), is recommended. As with all nonlinear estimation problems, the resulting $V_{\mathbf{a}}$ is based on a linearisation and could be significantly different from the true value. Monte Carlo techniques can be used to validate these estimates (section 8.2).

Example: assessment of aspheric surfaces

In determining the shape of high quality optical surfaces using measurements gathered by a coordinate measuring machine, care must be taken to ensure that the optical surface is not damaged by the contacting probe. However, using a low-force probing scheme, the presence of particles of dust on the artefact's surface introduces sporadic, large non-random effects into the measurement data. Figure 4.4 shows the residuals associated with an ALS fit of a hyperboloid surface to measurements of an aspheric mirror, a component in an earth observation camera. The spikes are due to particles of dust on the mirror or on the spherical probe. It is judged that 9 of the 401 measurements (i.e., approximately 2%) have been contaminated. Because the dust particles must necessarily have a positive diameter an asymmetric transform function τ was used in which only large, positive approximation errors are transformed. The standard noise associated with the measurements is of the order of 0.000 2 mm while the diameter of the dust particles is of the order of 0.002 mm. The difference between the ALS fitted surface and that generated using a standard (nonlinear) approach was of the order of 0.000 4 mm, and is seen to be significant relative to the standard noise. ♯

4.7.4 Bibliography and software sources

The ALS approach is described more fully in [140]. Nonlinear least-squares software can be used directly to provide ALS estimates (section 4.2.8).

4.8 Robust estimators

Because of there ability to cope with outliers, the L_1 and ALS estimators are termed *robust estimators*. There are other estimation algorithms designed to cope with outliers, including the Huber M-estimator [136, 137], which behaves like a least-squares estimator for small residuals and like L_1 for outliers. Aspects of robust estimation are considered in [63, 79, 189, 202, 205]. See also [161].

4.9 Nonlinear Chebyshev and L_1 approximation

The nonlinear Chebyshev optimisation problem is: given m functions $f_i(\mathbf{a})$, $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$, $n \leq m$, solve

$$\min_{\mathbf{a}} \; F(\mathbf{a}) = \max_i |f_i(\mathbf{a})|. \tag{4.31}$$

The Chebyshev optimisation problem arises in data approximation with nonlinear models. Given data $\{(\mathbf{x}_i, y_i)\}_1^m$ and the nonlinear model

$$y = \phi(\mathbf{x}, \mathbf{a}),$$

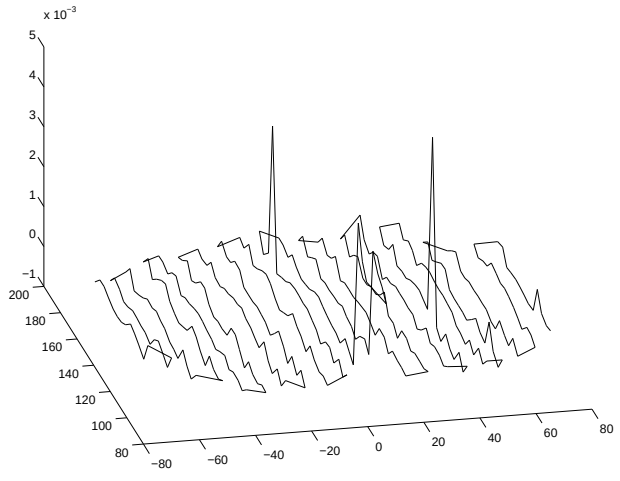

Figure 4.4: Residuals associated with an ALS fit of a hyperboloid surface to measurements of an aspheric mirror. The spikes are due to particles of dust on the mirror or on the spherical probe. The units for each axis are millimetres.

the nonlinear Chebyshev estimate of the parameters $\mathbf{a}$ is the one that solves the optimisation problem (4.31) with $f_i(\mathbf{a}) = y_i - \phi(\mathbf{x}_i, \mathbf{a})$. Chebyshev approximation problems occur frequently in dimensional metrology in which a geometric element is fitted to co-ordinate data according to the Chebyshev or related criteria.

The problem can be reformulated as

$$\min_{\mathbf{a},s} \; s$$

subject to the nonlinear constraints

$$-s \leq f_i(\mathbf{a}) \leq s, \quad i = 1, \ldots, m.$$

General purpose optimisation software can be used to solve this problem effectively.

The nonlinear L_1 optimisation problem is: given m functions $f_i(\mathbf{a})i,\ n \leq m$, solve

$$\min_{\mathbf{a}} \; F(\mathbf{a}) = \sum_{i=1}^{m} |f_i(\mathbf{a})|. \tag{4.32}$$

The L_1 optimisation problem arises in data approximation involving nonlinear models with $f_i(\mathbf{a}) = y_i - \phi(\mathbf{x}_i, \mathbf{a})$.

The problem (4.32) can be formulated as

$$\min_{\mathbf{a},\{s_i\}} \sum_{i=1}^{m} \; s_i$$

subject to the constraints

$$-s_i \leq f_i(\mathbf{a}) \leq s_i, \quad i = 1, \ldots, m,$$

and solved using general purpose optimisation software. However, unlike the nonlinear Chebyshev problem, this is not a very efficient approach due to the introduction of the extra parameters s_i. An approach designed to overcome this disadvantage is described in [164].

A simpler approach to these nonlinear approximation problems is to use a Gauss-Newton strategy (section 4.2) in which at each major iteration a linear Chebyshev or L_1 problem is solved [177, 178, 204]. These algorithms can work work well on some problems, but can exhibit slow convergence on others.

4.9.1 Bibliography and software sources

Nonlinear Chebyshev and L_1 approximation are considered in [163, 164, 177, 178, 204]. There are Chebyshev (minimax) optimisation modules in the Matlab optimisation Toolbox and the NAG library [173]. There are general purpose optimisation modules that can be applied to these problems in the NAG and IMSL libraries [173, 203]. Chebyshev approximation with geometric elements is considered in [5, 6, 41, 99].

4.10 Maximum likelihood estimation (MLE)

4.10.1 Description

Suppose $Z_i \sim D_i(\mathbf{a})$, $i = 1, \ldots, m$, are m independently distributed random variables associated with distributions $D_i(\mathbf{a})$ with PDFs $p_i(x|\mathbf{a})$ depending on n parameters $\mathbf{a} = (a_1, \ldots, a_n)$, and that $\mathbf{z}$ is a set of observations of $\boldsymbol{Z}$ (denoted $\mathbf{z} \in (Z)$. Let $l(\mathbf{a}|\mathbf{z})$ be the likelihood function defined by

$$l(\mathbf{a}|\mathbf{z}) = p(\mathbf{z}|\mathbf{a}) = \prod_{i=1}^{m} p_i(z_i|\mathbf{a}).$$

The maximum likelihood estimate $\hat{\mathbf{a}}$ of $\mathbf{a}$ maximises the likelihood function $l(\mathbf{z}|\mathbf{a})$.

MLE is a very general parameter estimation tool used widely across science. It requires that the PDFs $p_i(x|\mathbf{a})$ are fully specified. For normally distributed random variables, the MLE is the same as the least-squares estimate.

4.10.2 Algorithms for maximum likelihood estimation

Most maximum likelihood estimation algorithms determine an estimate by minimising the negative log likelihood function

$$F(\mathbf{a}|\mathbf{z}) = -\log l(\mathbf{a}|\mathbf{z}) = -\sum_{i=1}^{m} \log p_i(z_i|\mathbf{a}),$$

using a version of Newton's algorithm for function minimisation (section 3.6).

4.10.3 Uncertainty associated with the fitted parameters

Let $\mathbf{z}$ be an observation of random variables $\boldsymbol{Z}$ with associated multivariate PDF $p(\mathbf{x}|\mathbf{a})$. Let $\mathbf{z} \mapsto \mathcal{M}(\mathbf{z})$ be the maximum likelihood estimate associated with data $\mathbf{z}$. We regard the ML estimate $\hat{\mathbf{a}} = \mathcal{M}(\mathbf{z})$ as an observation of a vector of random variables $\hat{\boldsymbol{A}} = \mathcal{M}(\boldsymbol{Z})$ and the uncertainty matrix associated with $\hat{\mathbf{a}}$ is the variance matrix associated with $\hat{\boldsymbol{A}}$.

Asymptotic results (i.e., variants of the Central Limit Theorem [187]) can be used to show that if various regularity assumptions hold (to permit the interchange of integration and differentiation, for example, and ensure that various integrals are finite), then as the number of data points increases the distribution of $\hat{\boldsymbol{A}}$ approaches $N(\mathbf{a}, I^{-1}(\mathbf{a}))$ where $I(\mathbf{a})$, the *Fisher information matrix*, is the expectation of the Hessian matrix of second partial derivatives of $F(\mathbf{a}|\mathbf{x}) = -\log l(\mathbf{a}|\mathbf{x}) = -\log p(\mathbf{x}|\mathbf{a})$:

$$I(\mathbf{a}) = \int \frac{\partial^2 F}{\partial a_j \partial a_k}(\mathbf{a}|\mathbf{x}) p(\mathbf{x}|\mathbf{a})\, d\mathbf{x}.$$

This matrix can be approximated by the *observed Fisher information matrix*

$$\hat{I} = H = \frac{\partial^2 F}{\partial a_j \partial a_k}(\hat{\mathbf{a}}|\mathbf{z}).$$

We therefore take as an estimate of the uncertainty matrix $V_{\mathbf{a}}$ associated with the estimates $\hat{\mathbf{a}}$

$$V_{\mathbf{a}} = \hat{I}^{-1} = H^{-1}. \tag{4.33}$$

The asymptotic results show that as the number of measurements (information) increases the estimates $\hat{\mathbf{a}}$ approach $\mathbf{a}$, so that MLE is asymptotically unbiased. The inverse of the Fisher information matrix $I^{-1}(\mathbf{a})$ represents a lower bound on the variance of any unbiased estimator and the ML estimates attains this lower bound asymptotically. This means that as the number of measurements increases, the variance matrix associated with a maximum likelihood estimate will become at least as small as that for any other unbiased estimator.

The estimate in (4.33) is based on the asymptotic behaviour of the ML estimator as the number of measurements increases. We can instead use linearisation to provide an estimate of the uncertainty matrix associated with the ML estimates. At the minimum of $F(\mathbf{a}|\mathbf{z})$, the gradient $\mathbf{g}(\mathbf{a}|\mathbf{z}) = \nabla_{\mathbf{a}} F = \mathbf{0}$ and these n equations define $\mathbf{a} = \mathbf{a}(\mathbf{z})$ as functions of $\mathbf{z}$. If K is the sensitivity matrix

$$K_{ji} = \frac{\partial a_j}{\partial z_i}$$

and $V_{\mathbf{z}}$ is the uncertainty matrix associated with $\mathbf{z}$, i.e., the variance matrix associated with $\boldsymbol{Z}$, then

$$V_{\mathbf{a}} \approx K V_{\mathbf{z}} K^{\mathrm{T}}.$$

Taking differentials of the equation $\mathbf{g}(\mathbf{a}(\mathbf{z}), \mathbf{z}) = \mathbf{0}$, we have

$$HK + H_{\mathbf{z}} = 0, \quad H_{\mathbf{z}}(j,i) = \frac{\partial^2 F}{\partial a_j \partial z_i},$$

so that

$$K = -H^{-1} H_{\mathbf{z}},$$

and

$$V_{\mathbf{a}} \approx H_{\mathbf{a}}^{-1} H_{\mathbf{z}} V_{\mathbf{z}} H_{\mathbf{z}}^{\mathrm{T}} H^{-1}.$$

Example: MLE for the standard experimental model

Suppose the model equations are

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad \boldsymbol{\epsilon} \in N(\mathbf{0}, \sigma^2 I).$$

The likelihood $p(y_i|\mathbf{a}, \sigma)$ of observing y_i given parameters $\mathbf{a}$ and σ is

$$p(y_i|\mathbf{a}, \sigma) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ -\frac{1}{2\sigma^2}(y_i - \phi(\mathbf{x}_i, \mathbf{a}))^2 \right\},$$

and the likelihood of observing $\mathbf{y}$ is

$$\begin{aligned} l(\mathbf{a},\sigma|\mathbf{y}) &= p(\mathbf{y}|\mathbf{a},\sigma) = \prod_{i=1}^{m} p(y_i|\mathbf{a},\sigma) \\ &= \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i-\phi(\mathbf{x}_i,\mathbf{a}))^2\right\}. \end{aligned} \tag{4.34}$$

The log likelihood function $L(\mathbf{a}|\mathbf{y})$ is given by

$$-L(\mathbf{a},\sigma|\mathbf{y}) = m\log\sigma + \frac{m}{2}\log(2\pi) + \frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i-\phi(\mathbf{x}_i,\mathbf{a}))^2. \tag{4.35}$$

The likelihood is maximised by $\hat{\mathbf{a}}$ and $\hat{\sigma}$ if $\hat{\mathbf{a}}$ minimises

$$\sum_{i=1}^{m}(y_i-\phi(\mathbf{x}_i,\mathbf{a}))^2,$$

and $\hat{\sigma}$ is such that

$$\hat{\sigma}^2 = \sqrt{\frac{\mathbf{r}^{\mathrm{T}}\mathbf{r}}{m}}, \tag{4.36}$$

where $r_i = y_i - \phi(\mathbf{x}_i,\hat{\mathbf{a}})$ are the residuals at the solution. We note that the ML estimate of $\mathbf{a}$ for normally distributed random effects is the same as the least-squares estimate while (4.36) differs (slightly) from that derived from the expectation of the χ^2 distribution in (4.5). In fact the expected value of $\hat{\sigma}^2$ is

$$E(\hat{\sigma}^2) = \sigma^2\frac{m-n}{m} = \sigma^2\left(1-\frac{n}{m}\right) \neq \sigma^2.$$

However, as $m \longrightarrow \infty$, $E(\hat{\sigma}^2) \longrightarrow \sigma^2$, showing that the ML estimate is asymptotically unbiased.

Figures 4.5 and 4.6 graph the negative likelihood surfaces associated with determining a constant α and standard deviation σ from 20 and 100 data points sampled from a normal distribution. The surface for 20 points is flatter than that for 100, so that the minimum is less well defined for 20 points. ♯

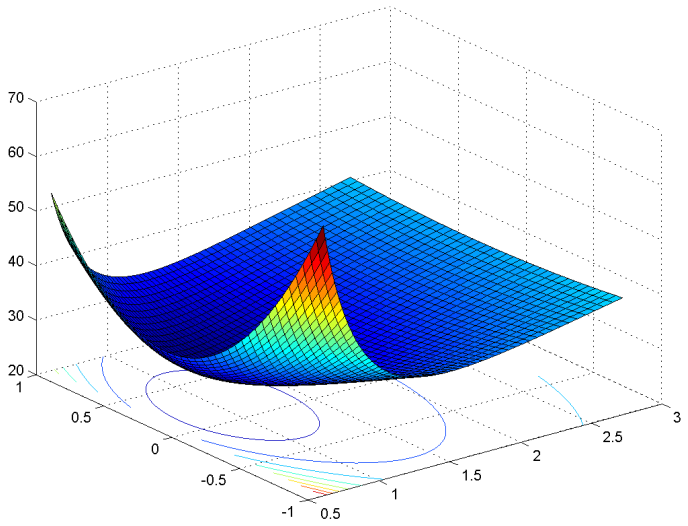

Figure 4.5: Log likelihood surface $-L(\alpha, \sigma|\mathbf{y})$ associated with the estimation of α and σ for data $y_i \in N(0,1)$ with 20 data points, plotted as a function of α and $\log \sigma^2$.

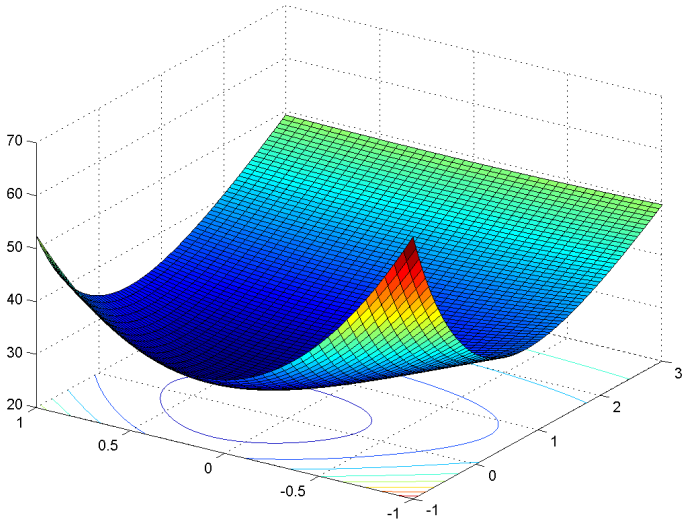

Figure 4.6: As figure 4.5 but with 100 data points.

4.11 Bayesian parameter estimation

4.11.1 Description

Both least-squares and maximum-likelihood methods, for example, are based on a so-called classical approach to statistical inference. In this paradigm, the parameters $\mathbf{a}$ we are trying to determine are fixed but unknown. The measurements $\mathbf{z}$ are assumed to have been generated according to a statistical model whose behaviour depends on $\mathbf{a}$. On the basis of the measurements $\mathbf{z}$ estimates $\hat{\mathbf{a}}$ are found for $\mathbf{a}$. These estimates are regarded as a sample from a vector of random variables $\boldsymbol{A}$ and the uncertainty associated with $\hat{\mathbf{a}}$ is determined from the distribution associated with this random vector.

In a Bayesian formulation [33, 158, 192], knowledge about $\mathbf{a}$ is encoded in a probability distribution $p(\mathbf{a}|I)$ derived from the information I we have to hand. As more information is gathered through measurement experiments, for example, these distributions are updated.

In the context of data analysis, we assume a *prior* distribution $p(\mathbf{a})$ and that data $\mathbf{z}$ has been gathered according to a sampling distribution depending on $\mathbf{a}$ from which we can calculate the probability $p(\mathbf{z}|\mathbf{a})$ of observing $\mathbf{z}$. This probability is the same as the likelihood function $l(\mathbf{a}|\mathbf{z})$ used in maximum likelihood estimation. Bayes' Theorem (2.1) states that the *posterior* distribution $p(\mathbf{a}|\mathbf{z})$ for $\mathbf{a}$ after observing $\mathbf{z}$ is related to the likelihood and the prior distribution by

$$p(\mathbf{a}|\mathbf{z}) = kp(\mathbf{z}|\mathbf{a})p(\mathbf{a}) = kl(\mathbf{a}|\mathbf{z})p(\mathbf{a}), \tag{4.37}$$

where the constant k is chosen to ensure that the posterior distribution integrates to 1, i.e.,

$$\int p(\mathbf{a}|\mathbf{z})\, d\mathbf{a} = 1.$$

In this form, Bayes' theorem says that the posterior distribution is the likelihood weighted by the prior distribution.

4.11.2 Parameter estimates and their associated uncertainties

The posterior distribution represents all the information about $\mathbf{a}$ taking into account the measurement data $\mathbf{z}$ and the prior information. In practice, summary information about this distribution is required and in metrology it is usual to provide parameter estimates along with associated uncertainties. Ideally, this would be in the form of the mean and variance of the posterior distribution. However, both these quantities require integration of multivariate functions and for problems involving even a modest number of parameters, 10 say, this integration is computationally expensive. For large problems it becomes impractical.

An alternative to providing estimates that require global knowledge of the distribution is to provide an approximation to the distribution on the basis of

local knowledge. This is the approach taken in generalised maximum likelihood estimation (GMLE), also known as maximising the posterior (MAP) [158]. The main idea is to determine a quadratic approximation to the negative logarithm $-\log p(\mathbf{a}|\mathbf{z})$ of the posterior distribution about its mode $\hat{\mathbf{a}}$:

$$-\log p(\mathbf{a}|\mathbf{z}) \approx -\log p(\hat{\mathbf{a}}|\mathbf{z}) + \frac{1}{2}(\mathbf{a}-\hat{\mathbf{a}})^{\mathrm{T}} H (\mathbf{a}-\hat{\mathbf{a}}), \tag{4.38}$$

where

$$H_{jk} = -\frac{\partial^2 \log p(\hat{\mathbf{a}}|\mathbf{z})}{\partial \alpha_j \partial \alpha_k}$$

is the Hessian matrix of second partial derivatives of $-\log p(\mathbf{z}|\mathbf{a})$ evaluated at the minimum $\hat{\mathbf{a}}$. (The linear term in this approximation is absent since $\partial \log p(\mathbf{a}|\mathbf{z})/\partial \alpha_j = 0$ at $\mathbf{a} = \hat{\mathbf{a}}$.) Taking exponentials of (4.38), we approximate the posterior distribution by

$$p(\mathbf{a}|\mathbf{z}) \approx k \exp\left\{-\frac{1}{2}(\mathbf{a}-\hat{\mathbf{a}})^{\mathrm{T}} H (\mathbf{a}-\hat{\mathbf{a}})\right\},$$

where k is a normalising constant. Recognising this as a multivariate normal distribution, setting $V = H^{-1}$, we have

$$p(\mathbf{a}|\mathbf{z}) \approx \frac{1}{|2\pi V|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{a}-\hat{\mathbf{a}})^{\mathrm{T}} V^{-1} (\mathbf{a}-\hat{\mathbf{a}})\right\},$$

i.e., $\mathbf{a} \sim N(\hat{\mathbf{a}}, V)$. (The notation $|V|$ denotes the determinant of V.) This approach provides parameter estimates $\hat{\mathbf{a}}$ and associated uncertainty matrix V using standard nonlinear optimisation techniques. We note that we can determine these terms without knowing the constant of proportionality in (4.37).

As with most approximating methods, this approach has to be used with some care. The multivariate normal distribution is unimodal and symmetric. If the true posterior distribution is multimodal or skewed, then the approximation could well provide poor information. (There may also be numerical difficulties in implementing the approach in these circumstances.)

4.11.3 Algorithms for GMLE

As for maximum likelihood estimation, GMLE algorithms can be implemented using standard unconstrained optimisation algorithms (section 3.6).

Example: GMLE for the standard experimental model

Suppose the model equations are

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad \boldsymbol{\epsilon} \in N(\mathbf{0}, \sigma^2 I).$$

The log likelihood function is given by (4.35). The prior $p(\mathbf{a}, \sigma)$ should reflect what is known before the experiment takes place. If nothing is known, then a *non-informative prior* should be assigned which is essentially constant so that the posterior distribution is proportional to the likelihood. In metrological

examples it is likely that some prior information is available, based on nominal values or previous experience using the measuring instrument, for example. In these circumstances, we may propose a prior distribution for $\mathbf{a}$ of the form $p(\mathbf{a}) = N(\mathbf{0}, \tau^2 I)$ and one for σ of the form

$$\log \sigma^2 \sim N(\log \sigma_0^2, (\log \rho)^2), \quad \rho \geq 1,$$

where $\mathbf{a}_0$, τ, σ_0 and ρ are specified. Roughly, the prior for σ^2 encodes the belief that we are 95% certain that $\sigma_0^2/\rho^2 \leq \sigma^2 \leq \sigma_0^2\rho^2$. Assuming $\mathbf{a}$ and σ are independently distributed, the logarithm of the prior distribution is given by

$$\begin{aligned} -\log p(\mathbf{a}, \sigma) &= \frac{1}{2}\log\left(2\pi\tau^2\right) + \frac{1}{2\tau^2}\sum_{j=1}^{n}(a_j - a_{0,j})^2 + \\ &\quad \frac{1}{2}\log\left(2\pi(\log\rho)^2\right) + \frac{1}{2(\log\rho)^2}(\log\sigma^2 - \log\sigma_0^2)^2. \end{aligned}$$

The generalised ML estimate is found by minimising

$$\begin{aligned} F(\mathbf{a}, \sigma|\mathbf{y}) &= \frac{m}{2}\log\sigma^2 + \frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \phi(\mathbf{x}_i, \mathbf{a}))^2 + \\ &\quad \frac{1}{2\tau^2}\sum_{j=1}^{n}(a_j - a_{0,j})^2 + \frac{1}{2(\log\rho)^2}(\log\sigma^2 - \log\sigma_0^2)^2, \end{aligned}$$

with respect to $\mathbf{a}$ and σ.

The quantities τ and ρ control the weight given to the prior information relative to the information represented in the data. As τ and ρ become larger (corresponding to weaker prior information) the posterior distribution becomes dominated by the likelihood function and the GML estimates approach the ML estimates. ♯

Chapter 5

Discrete models in metrology

In this chapter we describe some common models used in metrology.

5.1 Polynomial curves

5.1.1 Description

Polynomials provide a class of linear models that are used extensively as empirical models for experimental data. A polynomial of degree n can be written as

$$f_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n = \sum_{j=0}^{n} a_j x^j = \sum_{j=0}^{n} a_j \phi_j(x),$$

where $\phi_j(x) = x^j$ are the *monomial basis* functions. (The indexing starts at zero so that the index matches the exponent.) A polynomial of degree 1 is a straight line, degree 2 a quadratic curve, etc. The immediate appeal of polynomials is that computation with polynomials requires only addition and multiplication.

5.1.2 Advantages and disadvantages

Polynomials are good for:

- Representing a smooth curve $y = \phi(x)$ or data generated from a smooth curve over a fixed interval $[x_{\min}, x_{\max}]$. They are extremely flexible and from the mathematical point of view can be used to approximate any smooth curve to a given accuracy by choosing a high enough degree. They are used, for example, to represent calibration curves of sensors.

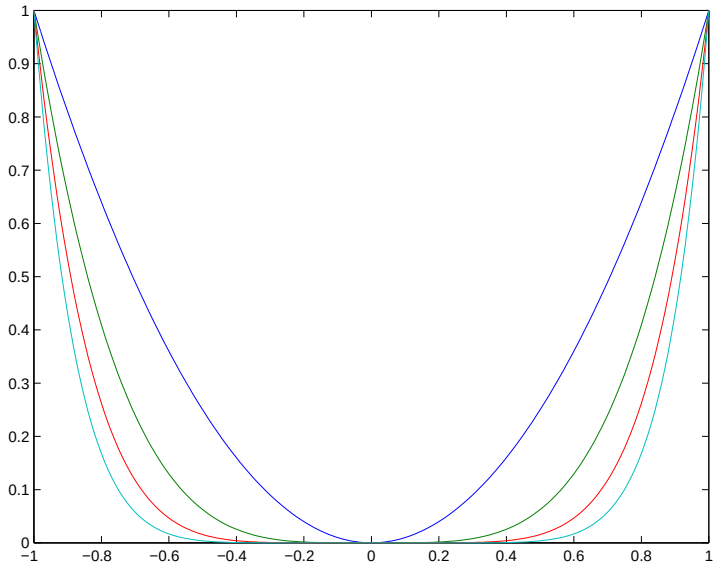

Figure 5.1: Graphs of x^{2j}, $j = 1, 2, 3, 4$, on the interval $[-1,\ 1]$.

Polynomials are not good for:

- Representing curves or data with sharp discontinuities in value or slope.
- Describing asymptotic behaviour where the curve approaches a straight line as the variable x gets larger in magnitude (section 4.7).

5.1.3 Working with polynomials

While the description of polynomials in terms of the monomial basis functions makes clear the nature of polynomial functions, the use of the monomial basis in numerical computation leads to severe numerical difficulties. A first difficulty is that for values of the variable x significantly greater than one in absolute value, the terms x^j become very large as j increases. This problem is solved by working with a normalised variable z. If x varies within the range $[x_{\min}, x_{\max}] = \{x : x_{\min} \leq x \leq x_{\max}\}$, then

$$z = \frac{(x - x_{\min}) - (x_{\max} - x)}{x_{\max} - x_{\min}} = \frac{x - (x_{\max} + x_{\min})/2}{(x_{\max} - x_{\min})/2}, \tag{5.1}$$

and all its powers lie in the range $[-1, 1]$. (The first expression for evaluating z above has better numerical properties [68].) For small degree polynomials ($n \leq 4$, say), this normalisation is sufficient to remove most numerical difficulties.

The second difficulty arises from the fact that, especially for large j, the basis function ϕ_j looks very similar to ϕ_{j+2} in the range $[-1, 1]$. Figure 5.1 presents

n	$[-1,1]$	$[0,2]$	$[4,6]$	$[19,21]$
5	2	4	9	15
10	4	9	16	24
20	10	18	*	*

Table 5.1: Estimates of the number of decimal digits lost using the monomial basis functions for different degrees and intervals. An entry $*$ indicates the system was too ill-conditioned for the calculation to be made.

the graphs of $\phi_{2j} = x^{2j}$ $j = 1, 2, 3, 4$. We can regard polynomial functions defined on $[-1, 1]$ as members of a vector space of functions. In this vector space, the angle between two polynomials $p(x)$ and $q(x)$ can be determined in terms of integrals involving their product, e.g.,

$$\int_{-1}^{1} p(x)q(x)w(x)dx,$$

where $w(x)$ is a weighting function. With this definition of angle, it is straightforward to show that the monomial basis functions ϕ_j and ϕ_{j+2} point in the roughly the same direction (in the sense that the angle between them is small), leading to ill-conditioning. This ill-conditioning worsens rapidly as the degree increases and the variable values move further from zero. Table 5.1 gives an estimate of the number of decimal digits lost using the monomial basis functions generated by 31 values $\{x_i\}_1^{31}$ randomly distributed in the interval $[-1, 1]$ and subsequently translated to the intervals $[0, 2]$, $[4, 6]$, $[19, 21]$. From the table, it is easy to see why polynomials are sometimes thought to be of very limited use because of numerical stability problems. In fact, it is their representation (i.e., parameterisation) in terms of the monomial basis functions which leads to instability, not polynomials *per se*.

Alternative representations can be derived by finding basis functions with better properties.

The *Chebyshev* polynomials $T_j(x)$ are one such set of basis functions and have the property that they are orthogonal to each other on the interval $[-1,\ 1]$ with respect to the weighting function $w(x) = 1/(1 + x^2)^{1/2}$. They are defined by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \quad j \geq 2.$$

Chebyshev polynomials can also be defined using the trigonometrical relationship

$$T_j(\cos\theta) = \cos j\theta, \quad \cos\theta = x.$$

Figure 5.2 presents the graphs of T_2 to T_5. Conventionally, T_0 is replaced by $T_0/2$ in the basis, so that

$$f_n(x) = \frac{1}{2}a_0T_0(x) + a_1T_1(x) + \ldots + a_nT_n(x) = {\sum_{j=1}^{n}}' a_jT_j(x);$$

the notation $\sum'$ indicates that the first term is halved.

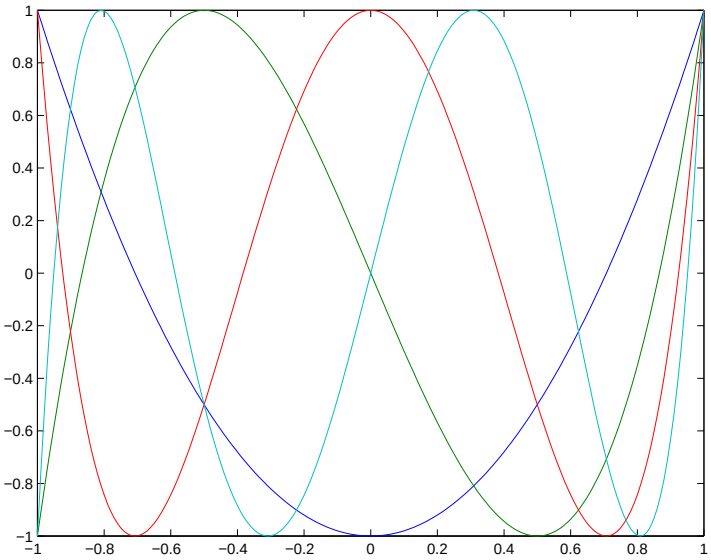

Figure 5.2: Chebyshev polynomials T_i, $i = 2, \ldots, 5$.

Using orthogonal polynomials in conjunction with the variable transformation formula (5.1) it is possible to use high degree polynomial models over any interval in a numerically stable way [109, 198]. Algorithms based on Chebyshev polynomials have been implemented in NPL's Data Approximation Subroutine Library — DASL [8] — (and other libraries) and used successfully for many years. It is disappointing that there are still many polynomial regression packages available for PCs that implement algorithms based on the standard monomial representation and are therefore prone to produce unreliable results. It should be emphasised that operations with a Chebyshev representation are, in essence, no more complicated than those using a monomial basis.

Example: evaluating a polynomial from a Chebyshev representation

A Chebyshev representation of a polynomial $p = p(x, \mathbf{a})$ of degree n ($n > 0$) is given in terms of the Chebyshev parameters (coefficients) $\mathbf{a} = (a_0, \ldots, a_n)^{\mathrm{T}}$ and constants $x_{\min}$ and $x_{\max}$ giving the range. The following scheme can be used to evaluate p at x.

I Calculate the normalised variable

$$z = \frac{(x - x_{\min}) - (x_{\max} - x)}{x_{\max} - x_{\min}}.$$

II Set $p = a_0/2 + a_1 z$, $t_0 = 1$, $t_1 = z$.

III for $j = 2 : n$

$$t_j \quad = \quad 2zt_{j-1} - t_{j-2},$$

$$p \quad = \quad p + a_j t_j.$$

DASL uses Clenshaw's recurrence to evaluate a polynomial from its Chebyshev representation: it requires fewer multiplications and has slightly superior floating-point error properties. [49, 54, 76, 116] ♯

Example: least-squares regression with polynomials using a Chebyshev representation

The following steps determine the least-squares best-fit polynomial of degree n ($n > 0$) to data $\{(x_i, y_i)\}_{i=1}^m$ using a Chebyshev representation. It follows the same approach as the general method described in section 4.1 for fitting a linear model to data, forming the observation matrix C whose jth column is the jth basis function evaluated at x_i, i.e., in this case, $C(i, j) = T_{j+1}(x_i)$.

I Calculate $x_{\min} = \min_i x_i$ and $x_{\max} = \max_i x_i$.

II Calculate the normalised variables

$$z_i = \frac{(x_i - x_{\min}) - (x_{\max} - x_i)}{x_{\max} - x_{\min}}, \quad i = 1, \ldots, m.$$

III Calculate the $m \times (n + 1)$ observation matrix C, column by column using the recurrence relationship. For each i:

III.1 $C(i, 1) = 1$, $C(i, 2) = z_i$,

III.2 for $j = 3 : n + 1$, $C(i, j) = 2z_i C(i, j - 1) - C(i, j - 2)$.

III.3 Adjust the first column: $C(i, 1) = C(i, 1)/2$.

IV Solve in the least-squares sense

$$C\mathbf{a} = \mathbf{y}.$$

If the linear least-squares problem is solved using a QR factorisation of the augmented matrix $[C\ \mathbf{y}]$ as described in section 4.1.2, it is possible to determine from the same orthogonal factorisation the least-squares polynomials of all degrees up to n (and the norms of the corresponding residual error vectors). This makes it very efficient to determine a range of polynomial fits to the data from which to select a best fit and is extremely useful in model validation; see, for example, [60, 61]. ♯

Other operations such as calculating the derivative of a polynomial are straightforward using a Chebyshev representation.

Example: derivative of a polynomial using a Chebyshev representation

If p is an n-degree polynomial with Chebyshev coefficients $\mathbf{a} = (a_0, \ldots, a_n)^{\mathrm{T}}$ defined on the range $[x_{\min}, x_{\max}]$ then its derivative $p' = \partial p/\partial x$ is a degree $n - 1$ polynomial on the same range and can therefore be represented in terms of Chebyshev coefficients $\mathbf{b} = (b_0, \ldots, b_{n-1})^{\mathrm{T}}$. The coefficients $\mathbf{b}$ are calculated directly from $\mathbf{a}$ and $x_{\min}$ and $x_{\min}$:

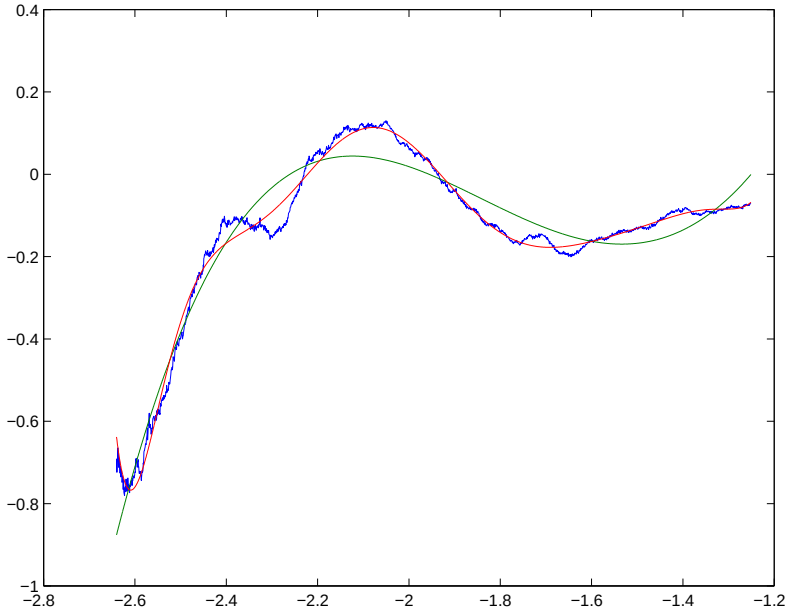

Figure 5.3: Least-squares polynomials of degrees 4 and 10 to 2000 data points.

I Set $b_{n+1} = b_n = 0$.

II for $j = n, n-1, \ldots, 2, 1$,

$$b_{j-1} = b_{j+1} + \frac{4ja_j}{x_{\max} - x_{\min}}.$$

♯

Example: polynomial fits to data

As an example of polynomial fits, figure 5.3 shows the least-squares polynomials of degrees 4 and 10 to 2000 data points, while figure 5.4 shows the least-squares polynomial of degree 18. ♯

There are other numerical approaches to polynomial regression. Given data $\{(x_i, y_i)\}_1^m$ and weights $\{w_i\}_1^m$ the Forsythe method implicitly determines a set of basis functions ϕ_j that are orthogonal with respect to the inner product defined by

$$\sum_{i=1}^{m} w_i f(x_i) g(x_i).$$

The method of solution exploits this orthogonality, using the fact that the observation matrix C that is generated is orthogonal, so that $C^{\mathrm{T}}C$ is a diagonal matrix and the normal equations can thus be solved very simply. The use of the normal equations is numerically safe since C is perfectly well conditioned. The

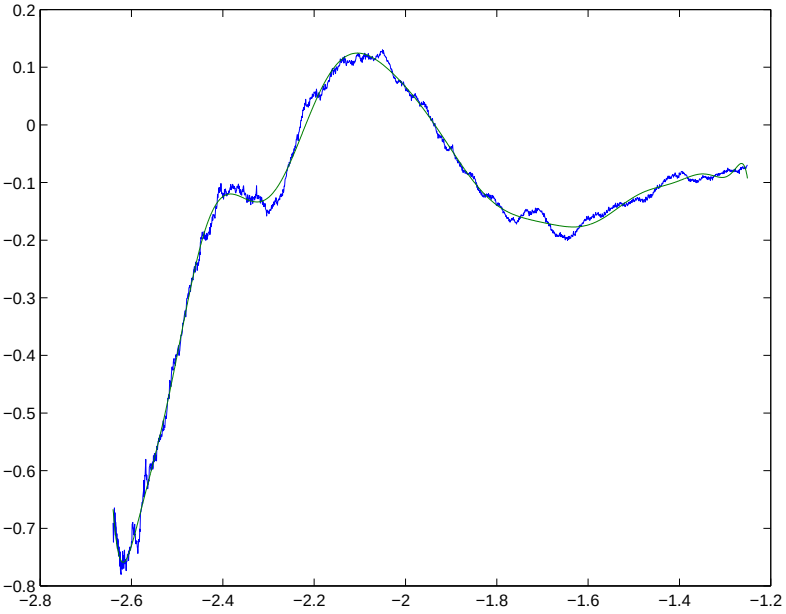

Figure 5.4: Least-squares polynomial of degree 18 to data.

set of orthogonal polynomial are generated specifically for the data $\{x_i\}$ and $\{w_i\}$. By contrast, the Chebyshev polynomials are much more versatile since they are defined in the same way for all data sets.

5.1.4 Bibliography and software sources

Approximation with polynomials is one of the main topics in data and function approximation. See, for example, [50, 109, 111, 123, 183, 198, 204]. Software for polynomial approximation appears in the NAG and IMSL libraries [173, 203] and there are a large number of software routines associated with polynomials available through Netlib [86]. NPL's Data Approximation Subroutine Library (DASL) and NPLFit package have extensive facilities for polynomial approximation [8, 172]. NPLFit, in particular, is aimed at metrological applications and has easy-to-use facilities for determining polynomial fits and associated uncertainties. NPLFit available as a package for downloading from EUROMETROS [9, 92].

5.2 Polynomial spline curves

5.2.1 Description

Like polynomials, polynomial spline curves — splines for short — are a class of linear models widely used for modelling discrete data. A spline $s(x)$ of order n defined over an interval $[x_{\min}, x_{\max}]$ is composed of sections of polynomial curves $p_k(x)$ of degree $n-1$ joined together at fixed points $\{\lambda_k\}_1^N$ in the interval.

Consider the case where there is one knot, at λ:

$$x_{\min} < \lambda < x_{\max},$$

and suppose we wish to build a continuous curve using two cubic polynomial curves

$$\begin{aligned} s(x) &= p_1(x, \mathbf{a}) = a + a_1 x + a_2 x^2 + a_3 x^3, \quad x \in [x_{\min}, \lambda], \\ &= p_2(x, \mathbf{b}) = b + b_1 x + b_2 x^2 + b_3 x^3, \quad x \in [\lambda, x_{\max}]. \end{aligned}$$

We impose smoothness constraints by insisting that the function values for both curves are equal at λ and so are the first and second derivatives. (If, in addition, we were to insist that the third derivatives are equal we would force $\mathbf{a} = \mathbf{b}$.) We can show that if s satisfies these three continuity constraints, it can be written in the form

$$s(x, \mathbf{a}, c) = p_1(x, \mathbf{a}) + c(x - \lambda)_+^3,$$

where $(x-\lambda)_+ = x - \lambda$ if $x > \lambda$ and 0 otherwise.

In general, if s is a spline of order n with continuity up to the $(n-2)$nd derivative on a set of N knots $\{\lambda_k\}_1^N$ with

$$x_{\min} < \lambda_1 < \lambda_2 < \ldots < \lambda_N < x_{\max}$$

then s can be written uniquely as

$$s(x, \mathbf{a}, \mathbf{c}) = p(x, \mathbf{a}) + \sum_{k=1}^{N} c_k (x - \lambda_k)_+^{n-1}, \tag{5.2}$$

where $p(x, \mathbf{a})$ is a polynomial of degree $n-1$. The number of parameters required to define s is $n + N$ (order + number of interior knots) and s is a linear combination of the polynomial basis functions and the *truncated power functions*

$$\phi_k(x) = (x - \lambda_k)_+^{n-1}.$$

B-spline basis functions. The representation (5.2) can be used to define an explicit method of constructing a polynomial spline. In practice, using this representation can give rise to severe numerical problems (because of ill-conditioning) and, in addition, has major efficiency drawbacks. Practically all

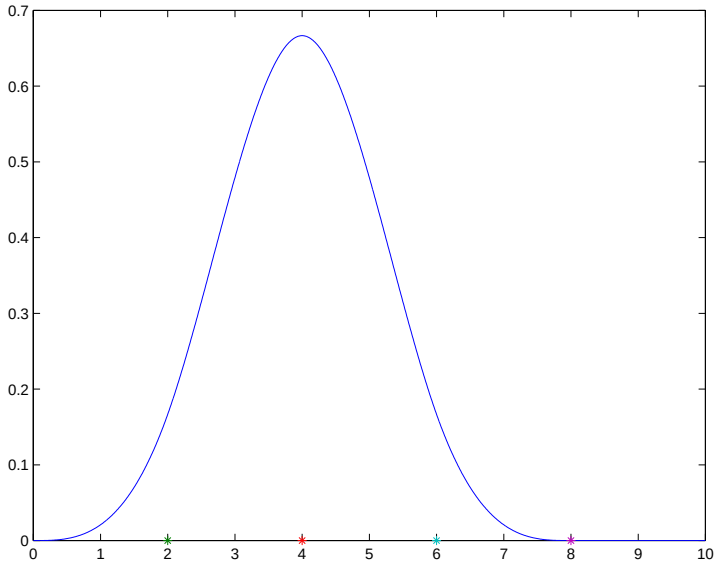

Figure 5.5: B-spline basis function $N_{4,4}(x, \boldsymbol{\lambda})$ defined on the interval $[0, 10]$ with knot set $\boldsymbol{\lambda} = (2, 4, 6, 8)^{\mathrm{T}}$.

calculations using spline functions are performed using a B-spline representation of the form

$$s(x, \mathbf{a}) = \sum_{j=1}^{n+N} a_j N_{n,j}(x, \boldsymbol{\lambda}), \tag{5.3}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^{\mathrm{T}}$ is the interior knot set satisfying

$$x_{\min} = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N < \lambda_{N+1} = x_{\max}, \tag{5.4}$$

and $N_{n,j}(x, \boldsymbol{\lambda})$ are the B-spline basis functions of order n (i.e., degree $n-1$). The basis functions $N_{n,j}(x, \boldsymbol{\lambda})$ are specified by the interior knot set $\boldsymbol{\lambda} = \{\lambda_k\}_1^N$, range limits

$$x_{\min} = \lambda_0, \text{and} \quad x_{\max} = \lambda_{N+1},$$

and the additional exterior knots, λ_j, $j = 1-n, \ldots, -1$ and $j = N+2, \ldots, N+n$. These exterior knots are usually assigned to be

$$\lambda_j = \begin{cases} x_{\min}, & j < 0, \\ x_{\max}, & j > N+1. \end{cases}$$

With this choice, the basis functions are defined by the interior knots $\boldsymbol{\lambda}$ and the range constants $x_{\min}$ and $x_{\max}$. The use of coincident knots with $\lambda_j = \ldots = \lambda_{j+k}$ allows us a greater degree of discontinuity at λ_j. We use $q = n + N$ to denote the number of basis functions.

A common choice of order is $n = 4$, splines constructed from cubic polynomials — *cubic splines* — because they give sufficient smoothness for most metrology

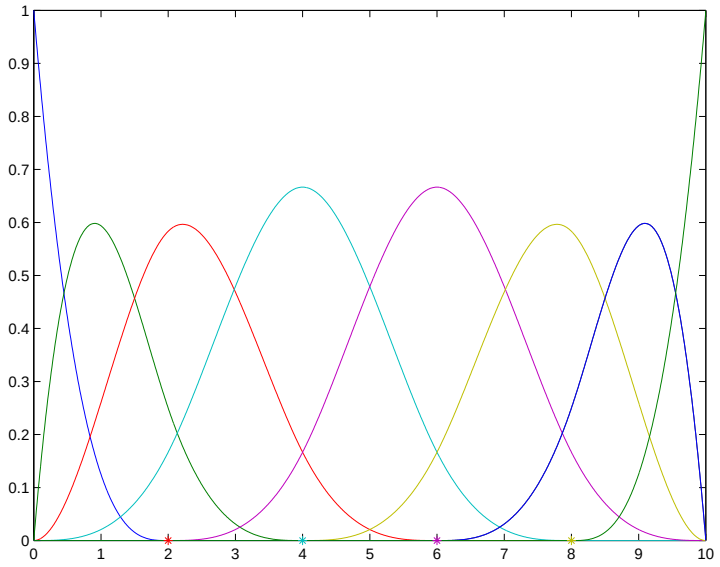

Figure 5.6: B-spline basis functions $N_{4,j}(x, \boldsymbol{\lambda})$ defined on the interval $[0, 10]$ with knot set $\boldsymbol{\lambda} = (2, 4, 6, 8)^{\mathrm{T}}$.

applications. Figure 5.5 graphs a B-spline basis function for a cubic spline defined on the interval $[0, 10]$ with knot set $\boldsymbol{\lambda} = (2, 4, 6, 8)^{\mathrm{T}}$. Figure 5.6 graphs all eight $(= n + N)$ basis functions for this knot set.

The B-spline basis functions have a number of valuable properties including:

$$\begin{aligned} N_{n,j}(x) &\geq 0, \\ N_{n,j}(x) &= 0, x \notin [\lambda_{j-n}, \lambda_j] \quad \text{(compact support)}, \\ \sum_j N_{n,j}(x) &\equiv 1, x \in [x_{\min}, x_{\max}]. \end{aligned} \tag{5.5}$$

Using a B-spline basis, calculations with splines can be performed in a numerically stable way.

5.2.2 Typical uses

Splines are used in much the same way as polynomials, but have additional capabilities. Splines are good for:

- Representing a smooth curve $y = \phi(x)$ or data generated from a smooth curve over a fixed interval $x \in [x_{\min}, x_{\max}]$. They are extremely flexible and from the mathematical point of view can be used to approximate any smooth curve to a given accuracy by choosing sufficient number of knots

or a high enough order (degree). They are used, for example, to represent calibration curves of sensors.

- Because spline approximation can be made computationally very efficient, splines are used to represent very large sets of data.
- Splines can be use to represent curves with varying characteristics and sharp changes in shape or discontinuities, provided a suitable set of knots is used.

Splines are less good for:

- Describing asymptotic behaviour where the curve approaches a straight line as the variable x gets larger in magnitude.

Because of their flexibility, splines are used in many applications areas of mathematical modelling.

5.2.3 Working with splines

As with polynomials, it is essential to use an appropriate set of basis functions. The representation using B-splines (equation (5.3), above) is strongly recommended. Since, for a specified set of knots, splines form a linear model, calculations involving splines centre around evaluating the basis functions $N_{n,j}(x, \boldsymbol{\lambda})$. Like Chebyshev polynomials, the basis function $N_{n,j}$ can be evaluated using a three-term recurrence relationship. The first order B-spline basis functions $N_{1,j}$ $j = 1, \ldots, N+1$ are defined by

$$\begin{aligned} N_{1,j}(x) &= \left\{ \begin{array}{ll} 1, & x \in [\lambda_{j-1}, \lambda_j), \\ 0, & \text{otherwise}, \end{array} \right\} j = 1, \ldots, N, \\ N_{1,N+1}(x) &= \left\{ \begin{array}{ll} 1, & x \in [\lambda_N, \lambda_{N+1}], \\ 0, & \text{otherwise}, \end{array} \right. \end{aligned}$$

and, for $n > 1$,

$$N_{n,j}(x) = \left\{ \begin{array}{ll} \frac{\lambda_j - x}{\lambda_j - \lambda_{j-n+1}} N_{n-1,j}(x), & j = 1, \\ \frac{x - \lambda_{j-n}}{\lambda_{j-1} - \lambda_{j-n}} N_{n-1,j-1}(x) + \frac{\lambda_j - x}{\lambda_j - \lambda_{j-n+1}} N_{n-1,j}(x), & 1 < j < N+n, \\ \frac{x - \lambda_{j-n}}{\lambda_{j-1} - \lambda_{j-n}} N_{n-1,j-1}(x), & j = N+n. \end{array} \right.$$

The first order B-spline basis functions equal one on a knot interval $[\lambda_{j-1}, \lambda_j)$ and zero elsewhere. An order n B-spline basis function is the weighted convex combination of two "adjacent" order $n-1$ B-spline basis functions.

Once the basis functions have been defined, spline evaluation and data fitting with splines can be performed following the general scheme for linear models.

Example: evaluating a spline in terms of its B-spline basis

A spline $s = s(x, \mathbf{a})$ of order n can be defined in terms of the B-spline coefficients (parameters) $\mathbf{a} = (a_1, \ldots, a_q)$, the interior knot set $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^{\mathrm{T}}$ and constants $x_{\min}$ and $x_{\max}$ giving the range. The following scheme can be used to evaluate s at x.

I Evaluate the B-spline basis functions $N_{n,j}(x)$, $j = 1, \ldots, q = n + N$, using the recurrence relations.

II Set

$$s(x) = \sum_{j=1}^{q} a_j N_{n,j}(x). \tag{5.6}$$

s is usually evaluated by a recurrence involving the a_j, see [55]. ♯

Example: least-squares regression with splines using a B-spline representation

The following steps determine the least-squares best-fit spline of order n with a given knot set $\boldsymbol{\lambda}$ and range $[x_{\min}, x_{\max}]$ to data $\{(x_i, y_i)\}_{i=1}^{m}$ using a B-spline representation. It is assumed that the knots satisfy

$$x_{\min} < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N < x_{\max},$$

and that $x_{\min} \leq x_i \leq x_{\max}$, $i = 1, \ldots, m$.

I Evaluate the B-spline basis functions $N_{n,j}(x_i)$, $j = 1, \ldots, q = n + N$, $i = 1, \ldots, m$, using the recurrence relations.

II Evaluate the $m \times q$ observation matrix C defined by $C(i, j) = N_{n,j}(x_i)$.

III Solve in the least-squares sense

$$C\mathbf{a} = \mathbf{y}.$$

♯

Other operations such as calculating the derivative of a spline are equally straightforward using a B-spline representation.

Example: derivative of a spline using a B-spline representation

Let $s = s(x, \mathbf{a})$ be a spline of order n defined in terms of the B-spline coefficients (parameters) $\mathbf{a} = (a_1, \ldots, a_q)^{\mathrm{T}}$, $q = n + N$, the interior knot set $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)^{\mathrm{T}}$ and range $[x_{\min}, x_{\max}]$. Its derivative $s' = \partial s / \partial x$ is an $(n-1)$th order spline defined by coefficients $\mathbf{b} = (b_1, \ldots, b_{q-1})^{\mathrm{T}}$, with

$$b_j = \begin{cases} (n-1)\frac{a_{j+1} - a_j}{\lambda_j - \lambda_{j-n+1}}, & \lambda_j > \lambda_{j-n+1}, \\ a_{j+1} - a_j, & \lambda_j = \lambda_{j-n+1}, \end{cases} \qquad j = 1, \ldots, q-1.$$

♯

Two features arise in working with splines that do not appear in approximation with general linear models. The first is the banded structure in the observation matrix and the second is the choice of knot set.

Banded structure in the observation matrix. The compact support property (equation (5.5)) of the B-spline basis functions means that for any $x \in [x_{\min}, x_{\max}]$ at most n of the basis functions $N_{n,j}(x)$ will be nonzero at x. More specifically, if $x \in [\lambda_{j-1}, \lambda_j)$, then only $N_{n,j}$, $N_{n,j+1}, \ldots, N_{n,j+n-1}$ can be nonzero. Thus, to evaluate an order n spline at any given point, only n basis functions need to be evaluated (and the inner product step (5.6) involves at most n nonzero contributions.) More importantly, any row of the observation matrix C has at most n nonzero elements appearing contiguously, i.e., adjacent to each other along the row, giving the observation matrix a *banded structure*. Figure 5.7 shows schematically (a) the structure of the observation matrix C for fitting a cubic spline (i.e., $n = 4$) with four (i.e., $N = 4$) interior knots to 11 ordered data points $(x_i, y_i)_1^{11}$, $x_i \leq x_{i+1}$ and (b) the structure of the triangular factor R determined from a QR factorisation of C (section 4.1).

The banded structure can be exploited effectively in solving the linear least squares system that arises using an orthogonal factorisation approach. The main consequence of this is that the fitting procedure can be accomplished in $O(mn^2)$ steps (i.e., in a number of steps proportional to mn^2) rather than $O(m(N+n)^2)$ if a general, full matrix approach is used. In other words, for a fixed order of spline ($n = 4$ a common choice), the computation time using a structure-exploiting approach is essentially proportional to the number m of data points and independent of the number of knots N. Using a full-matrix approach, the computation time is approximately proportional to mN^2 for a large number of knots. This efficiency saving is significant, particularly for large knot sets and is one of the reasons why splines are so popular and effective in modelling data.

Choice of knot set. In approximation using polynomials, the main choice that a user has is fixing the degree of the polynomial. In spline approximation, the user has to fix the order (usually set at a small number with four the most common choice) and also has the much greater flexibility in fixing the number and location of the interior knots $\boldsymbol{\lambda}$ (subject to the constraints on ordering (5.4)). The knot placement can have a considerable effect on the quality of the fit, but there is no usable set of criteria that can be used to determine an optimal placement strategy (although there is much research in addressing aspects of this problem). However, there are a number of guidelines that help the user to arrive at a good set of knots. We assume that we wish to fit an nth order spline to m data points $\{(x_i, y_i)\}_1^m$.

- The number of knots N must be less than or equal to $m - n$ (i.e. $q = n + N \leq m$) in order to be able to determine all the coefficients (otherwise the observation matrix C would be rank deficient). Generally, we are looking for the smallest number of knots that provides a good fit.

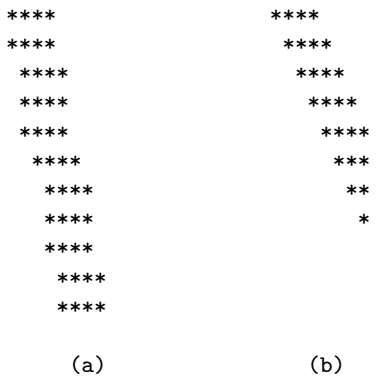

Figure 5.7: Schematic representation of (a) the structure of the observation matrix C for fitting a cube spline ($n = 4$) with four ($N = 4$) interior knots to 11 ordered data points $(x_i, y_i)_1^{11}$, $x_i \leq x_{i+_1}$ and (b) the structure of the triangular factor R determined from a QR factorisation of C.

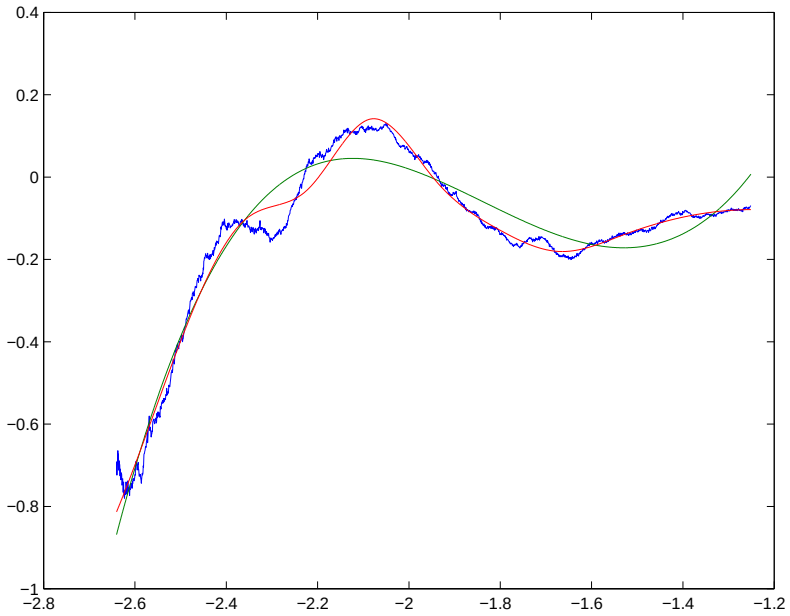

Figure 5.8: Least-squares cubic splines ($n = 4$) with one and seven interior knots to 2000 data points.

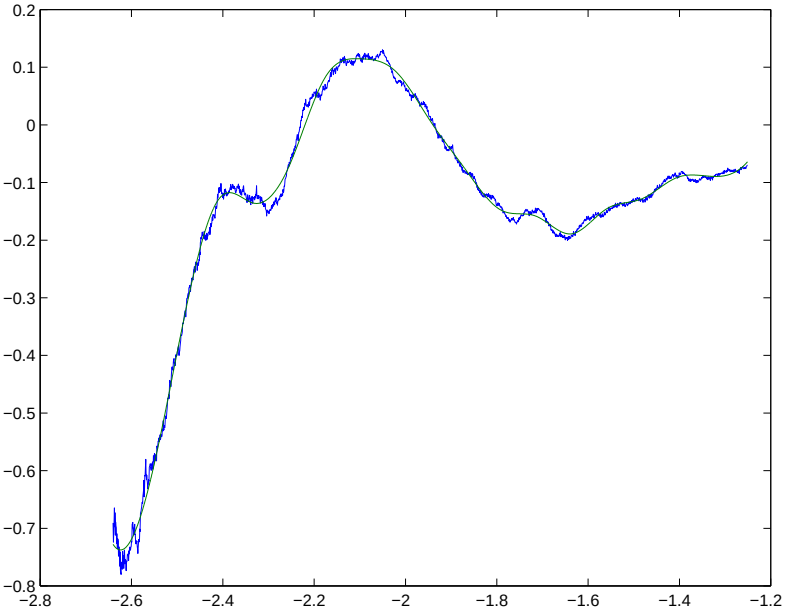

Figure 5.9: Least-squares cubic spline ($n = 4$) with 15 interior knots to data.

- The knots λ_j should be interspersed with the abscissae $\{x_i\}$. One set of conditions (Schoenberg-Whitney) state that there should be a subset $\{t_1, \ldots, t_q\} \subset \{x_1, \ldots, x_m\}$ such that

$$t_j < \lambda_j < t_{j+n}, \quad j = 1, \ldots, N.$$

- More knots are needed in regions where the curve underlying the data is rapidly changing, fewer knots where the curve is relatively smooth.

The goodness of fit is, naturally, a qualitative attribute often assessed from a visual examination of the fit to the data. If the fit does not follow the data adequately in a region, more knots should be added, perhaps adjusting nearby knots. If the fit seems to be following the noise in the data in some regions, then knots should be removed from those regions and the remaining knots possibly adjusted. After say three or four passes, a satisfactory fit can often be attained.

Example: spline fit to data

As an example of spline fits, figure 5.8 shows the least-squares cubic splines ($n = 4$) with one and seven interior knots to 2000 data points, while figure 5.9 shows the cubic spline least-squares fit with 15 interior knots. In figure 5.10, we can compare this latter fit with a polynomial fit of degree 18 to the same data. Note that both the polynomial and spline are defined by 19 basis functions. The spline is seen to be more flexible and able to follow the shape of the data more closely. ♯

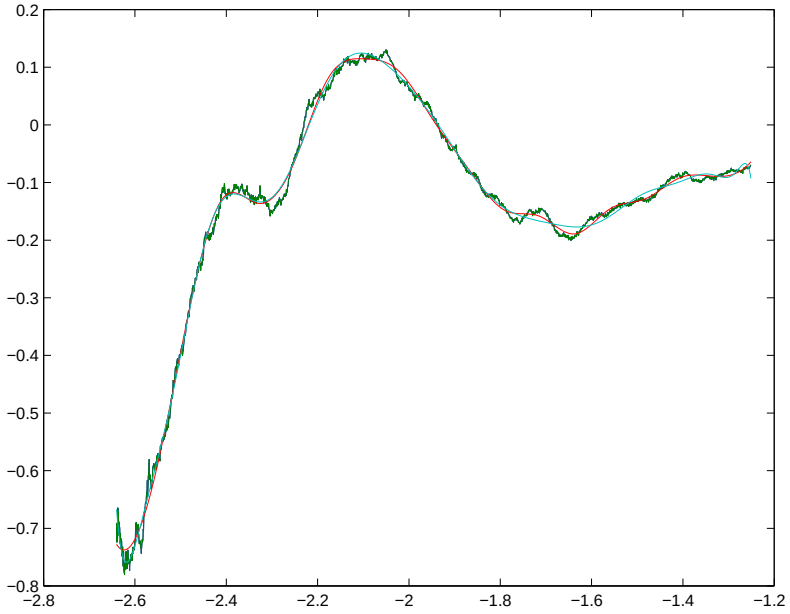

Figure 5.10: Least-squares cubic spline ($n = 4$) with 15 interior knots and the degree 18 least-squares polynomial to data.

5.2.4 Bibliography and software sources

Algorithms for working with splines in terms of their B-spline representation are given in [53, 55, 56, 57, 85]. Software for spline interpolation and approximation appear in the NAG and IMSL libraries [173, 203], the Matlab spline toolbox [156], and various spline packages available through Netlib [86]. Algorithms for knot placement are described in [73, 74, 147].

Because of the computational efficiency gains to be made using structured solvers, it is recommended that special purpose spline approximation packages are used rather that standard optimisation software. DASL and the NPLFit package have extensive facilities for spline approximation [8, 172]. NPLFit, in particular is aimed at metrological applications and has easy-to-use facilities for calculating spline fits, knot choice, and associated uncertainties. NPLFit is available as a package for downloading from EUROMETROS [9, 92].

5.3 Fourier series

5.3.1 Description

A *Fourier series* of degree n is generally written as

$$\phi(x, \mathbf{a}) = a_0 + \sum_{j=1}^{n} a_j \cos jx + \sum_{j=1}^{n} b_j \sin jx,$$

where $\mathbf{a} = (a_0, a_1, \ldots, a_n, b_1, \ldots, b_n)^{\mathrm{T}}$. We note that $\phi(x + 2\pi, \mathbf{a}) = \phi(x, \mathbf{a})$. To model functions with period $2L$, we modify the above to

$$\phi(x, \mathbf{a}|L) = a_0 + \sum_{j=1}^{n} a_j \cos j\pi x/L + \sum_{j=1}^{n} b_j \sin j\pi x/L.$$

Since

$$\int_{-\pi}^{\pi} \cos jx \cos kx \, dx = \int_{-\pi}^{\pi} \sin jx \sin kx \, dx = 0, \quad j \neq k,$$

and

$$\int_{-\pi}^{\pi} \cos jx \sin kx \, dx = \int_{-\pi}^{\pi} \cos jx \, dx = \int_{-\pi}^{\pi} \sin jx \, dx = 0,$$

the basis functions 1, $\cos jx$ and $\sin jx$ are orthogonal with respect to the unit weighting function over any interval of length 2π.

If $f(x)$ is a periodic function with $f(x + 2\pi) = f(x)$ then its representation as a Fourier series is given by

$$f(x) = a_0 + \sum_{j=1}^{\infty} (a_j \cos jx + b_j \sin jx),$$

where

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \, dx,$$

and

$$a_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos jx \, dx, \quad b_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin jx \, dx, j = 1, 2, \ldots$$

Fourier series are used to model periodic functions and to analyse the frequency component or spectral characteristics of data. The Fourier transform and its inverse are important in signal processing and filtering. Fourier series are less successful in analysing data arising from responses $y(x)$ where the frequency component of y changes with location x (see section 5.6).

5.3.2 Working with Fourier series

For fixed period L, $\phi(x, \mathbf{a})$ is a linear model and fitting a Fourier series to data follows the same general scheme for fitting linear models to data $\{(x_i, y_i)\}_{i=1}^{m}$:

I Fix period L and degree n with $2n + 1 \leq m$.

II Form $m \times (2n+1)$ observation matrix C. For $i = 1, \ldots, m$, set $C(i, 1) = 1$, and for $j = 1, \ldots, n$, $C(i, 2j) = \cos(2\pi j/L)$ and $C(i, 2j+1) = \sin(2\pi j/L)$.

III Solve the linear least-squares system

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|^2,$$

for parameters $\mathbf{a}$.

Uncertainties associated with the fitted parameters can be estimated using the general approach described in section 4.1.

It has been assumed that the period L is known. If this is not the case then we can regard L as an unknown, in which case the observation matrix $C = C(L)$ is now a nonlinear function of L[1] and the fitting problem becomes

$$\min_{\mathbf{a},L} \|\mathbf{y} - C(L)\mathbf{a}\|^2,$$

a nonlinear least-squares problem (section 4.2). This problem can be solved using the Gauss-Newton algorithm for example. Alternatively, let $\mathbf{a}(L)$ solve the linear least-squares problem

$$\min_{\mathbf{a}} \|\mathbf{y} - C(L)\mathbf{a}\|^2,$$

and set $\mathbf{r}(L) = \mathbf{y} - C(L)\mathbf{a}(L)$ and $F(L) = \|\mathbf{r}(L)\|$, the norm of the residuals for period L. A univariate minimisation algorithm can be applied to $F(L)$ to find an optimal or at least better estimate of the period.

5.3.3 Fast Fourier Transform (FFT)

For data $(x_j, y_j)_{j=1}^m$ where the abscissae $\{x_j\}$ are uniformly spaced in an interval of length one period, e.g.,

$$x_j = j2L/m,$$

the coefficients $\mathbf{a} = (a_0, a_1, \ldots, a_n, b_1, \ldots, b_n)^{\mathrm{T}}$ for the best-fit Fourier series can be calculated using the discrete Fourier transform (DFT). For any integer $m > 0$ the explicit discrete Fourier transform matrix F is the complex valued matrix defined by

$$F_{jk} = \exp\{-2\pi i(j-1)(k-1)/m\},$$

where $i = \sqrt{-1}$. Its inverse is given by

$$F_{jk}^{-1} = \frac{1}{m} \exp\{2\pi i(j-1)(k-1)/m\}.$$

The DFT of an m-vector $\mathbf{y}$ is simply $\mathbf{w} = F\mathbf{y}$. Since F is complex valued, $\mathbf{w}$ is also. The coefficients a_0, $\mathbf{a}$ and $\mathbf{b}$ of the degree n Fourier series approximation to $\mathbf{y}$ is found from $\mathbf{w}$ as follows

$$a_0 = w_1/m, \quad a_j = 2\Re(w_j)/m, \quad b_j = 2\Im(w_j)/m, \quad j = 1, \ldots, n,$$

[1] Or we could work with K=1/L instead.

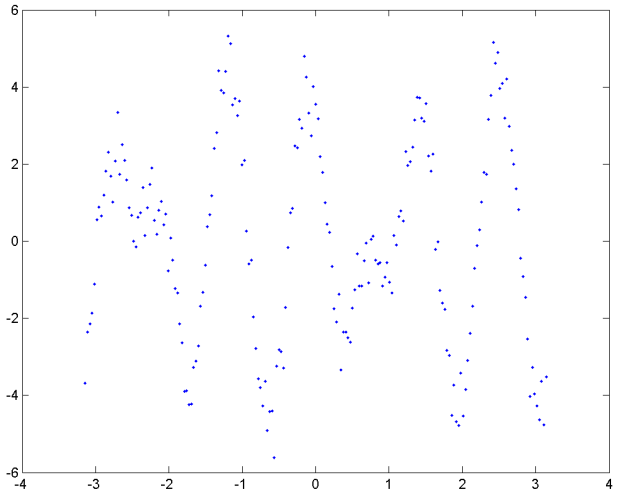

Figure 5.11: Data generated according to the model (5.7).

where $\Re(w_j)$ and $\Im(w_j)$ are the real and imaginary parts of the jth element of $\mathbf{w}$, respectively. The fitted values $\hat{\mathbf{y}}$ can be determined using the inverse DFT:

$$\hat{\mathbf{y}} = \Re\left(F^{-1} \begin{bmatrix} \mathbf{w}(1:n+1) \\ \mathbf{0} \\ \mathbf{w}(m-n+1:m) \end{bmatrix} \right).$$

Instead of working with the explicit transform matrices, the fast Fourier transform uses matrix factorisation techniques to recursively divide the calculations into smaller subproblems and attains a computational efficiency of $O(m \log m)$ rather than $O(m^2)$.

Example: fitting data generated from three Fourier components

Figure 5.11 plots data generated according to the model

$$y_j = 3\cos 5x - 2\sin 7x + 0.5\cos 9x + \epsilon_j, \qquad \boldsymbol{\epsilon} \in N(\mathbf{0}, 0.25I). \tag{5.7}$$

For this data $L = \pi = 3.1416$. Figure 5.12 graphs best-fit Fourier series of degree $n = 10$ with the estimate $\hat{L} = 3.1569$ of L found by a univariate minimisation algorithm. ♯

5.3.4 Bibliography and software sources

Fourier series and transforms are discussed in [34, 35, 84, 141, 162], for example. The fast Fourier transform was developed by Cooley and Tukey [52]. Further developments include [112], for example.

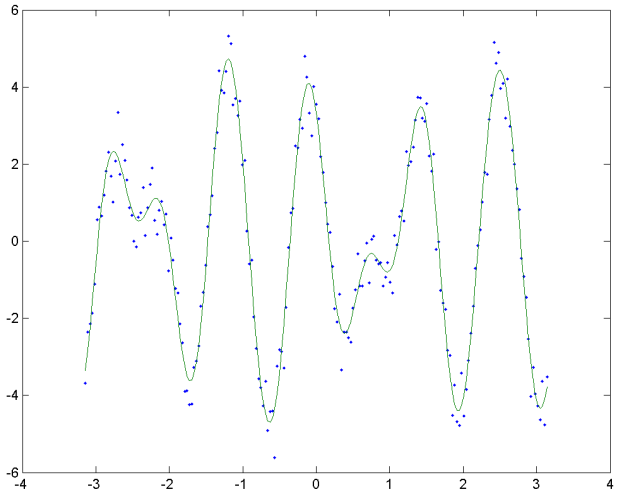

Figure 5.12: Fitted Fourier series of degree $n = 10$ to data in figure 5.11.

5.4 Asymptotic polynomials

Asymptotic behaviour associated with physical systems is quite common. For example, a response may decay to a constant as time passes. However empirical models such as polynomials, splines and Fourier series do not lend themselves to modelling asymptotic behaviour. In this section we describe a simple class of modified polynomial basis functions that can be used to model a range of asymptotic behaviour.

5.4.1 Description

Let $\{\phi_j(x)\}_{j=0}^n$ be a set of polynomial basis functions defined on $[-1, 1]$, such as Chebyshev polynomials (section 5.1). Define

$$w(x) = w(x|x_0, c, k) = \frac{1}{(1 + c^2(x - x_0)^2)^{k/2}}, \quad c > 0.$$

$w(x)$ is smooth and, for c large, $w(x)$ behaves like $|x|^{-k}$ as $|x| \to \infty$.
Defining

$$\tilde{\phi}_j = w(x)\phi_j(x),$$

then

$$\tilde{\phi}(x, \mathbf{a}) = \sum_{j=0}^{n} a_j \tilde{\phi}_j(\mathbf{x})$$

behaves like x^{n-k} as $|x| \to \infty$ and c gets large. In particular, if $k = n$, then ϕ can model asymptotic behaviour of approaching a constant. The constant c controls the degree to which asymptotic behaviour is imposed on the model.

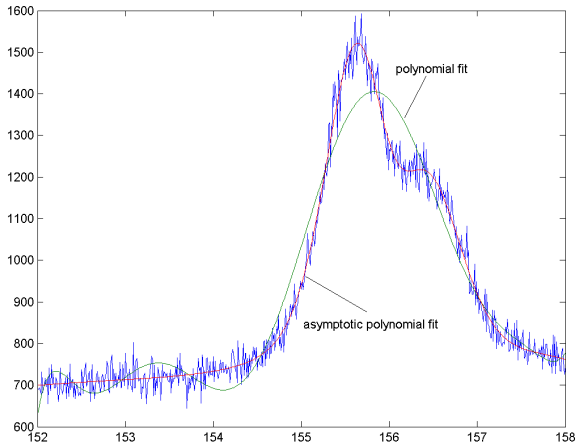

Figure 5.13: Asymptotic and standard polynomial fits of degree 9 to measurements of material properties (for aluminium).

The weighting function w can be modified to provide different asymptotic behaviour as x approaches ∞ and $-\infty$:

$$\begin{aligned} w(x) = w(x|x_0, c, k, l) &= \frac{1}{(1 + c^2(x - x_0)^2)^{k/2}}, \quad x \geq x_0, \\ &= \frac{1}{(1 + c^2(x - x_0)^2)^{l/2}}, \quad x < x_0. \end{aligned}$$

5.4.2 Working with asymptotic polynomials

With x_0 and c fixed, the function $\tilde{\phi}$ is a linear combination of basis functions and so the general approach to model fitting can be adopted:

I Fix x_0, c, k and degree n.

II Form $m \times (n+1)$ observation matrix C for $\{\phi_j\}$: for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, $C(i, j) = \phi_j(x_i)$ and weight vector $w_i = w(x_i|x_0, c, k)$. Normalise weight vector $w_i := w_i/M$ where $M = \max_i |w_i|$.

III Form modified observation matrix $\tilde{C}_{ij} = w_i C_{ij}$.

IV Solve the linear least-squares system

$$\min_{\mathbf{a}} \|\mathbf{y} - \tilde{C}\mathbf{a}\|^2$$

for parameters $\mathbf{a}$.

Uncertainties associated with the fitted parameters can be estimated using the general approach described in section 4.1. Using the Forsythe method [109], the

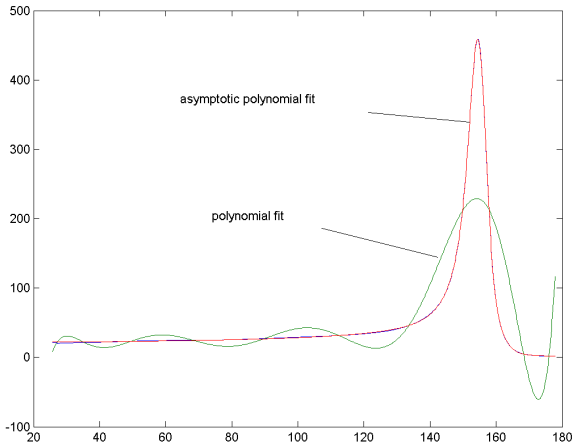

Figure 5.14: Asymptotic and standard polynomial fits of degree 9 to measurements of thermo-physical properties.

modified basis functions $\tilde{\phi}_j$ can be determined so that the observation matrix $\tilde{C}$ is orthogonal, leading to better numerical properties.

It has been assumed above that constants x_0 and c are fixed. However, we can regard one or both as additional parameters to be determined in which case the observation matrix $\tilde{C} = \tilde{C}(x_0, c)$ is now a nonlinear function of x_0 and c and the fitting problem becomes

$$\min_{\mathbf{a},x0,c} \|\mathbf{y} - \tilde{C}(x_0, c)\mathbf{a}\|^2,$$

a nonlinear least-squares problem (section 4.2). This problem can be solved using the Gauss-Newton algorithm for example. Note that at each iteration only $\tilde{C}$ has to be formed from C; there is no need to recalculate C.

Alternatively, let $\mathbf{a}(x_0, c)$ solve the linear least-squares problem

$$\min_{\mathbf{a}} \|\mathbf{y} - \tilde{C}(x_0, c)\mathbf{a}\|^2,$$

and set $\mathbf{r}(x_0, c) = \mathbf{y} - \tilde{C}(x_0, c)\mathbf{a}(x_0, c)$ and $F(x_0, c) = \|\mathbf{r}(x_0, c)\|$, the norm of the residuals. A multivariate minimisation algorithm can be applied to $F(x_0, c)$ to find an optimal or at least better estimate of these parameters.

Example: asymptotic polynomial and (standard) polynomial fits compared

In figures 5.13–5.16, asymptotic polynomial and standard polynomial fits of the same degree have been fitted to data portraying asymptotic behaviour. In each case, the asymptotic polynomial fit gives a better representation of the data. In figures 5.14 and 5.16 the asymptotic polynomial fit is barely distinguishable from the data. ♯

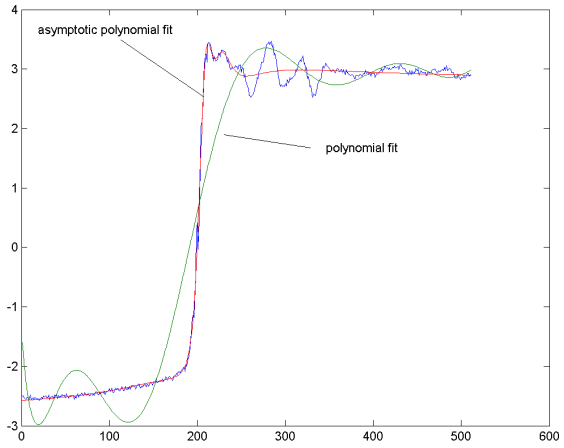

Figure 5.15: Asymptotic and standard polynomial fits of degree 9 to oscilloscope response measurements.

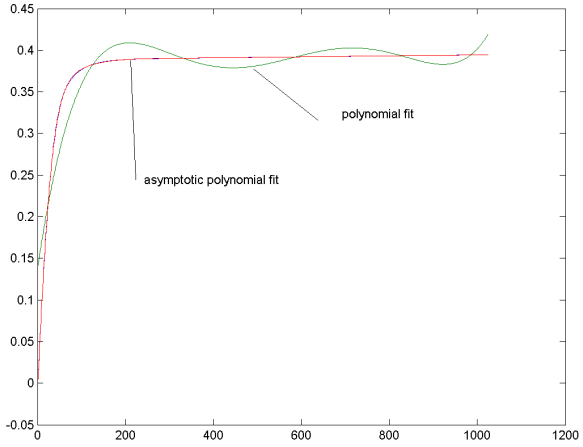

Figure 5.16: Asymptotic and standard polynomial fits of degree 5 to photodiode response measurements.

5.5 Tensor product surfaces

5.5.1 Description

The simplest way to generate linear empirical models for surfaces is to construct them from linear empirical models for curves. Suppose

$$\begin{aligned} \phi(x, \mathbf{a}) &= a_1\phi_1(x) + \ldots + a_{n_x}\phi_{n_x}(x) \quad \text{and} \\ \psi(y, \mathbf{b}) &= b_1\psi_1(y) + \ldots + b_{n_y}\psi_{n_y}(y) \end{aligned}$$

are two linear models for curves. Then the functions $\gamma_{k\ell}(x, y) = \phi_k(x)\psi_\ell(y)$, $k = 1, \ldots, n_x$, $\ell = 1, \ldots, n_y$, form the *tensor product* set of basis functions for defining linear models for representing surfaces of the form

$$z = \gamma(x, y, \mathbf{a}) = \sum_{k=1}^{n_x} \sum_{\ell=1}^{n_y} a_{jk}\gamma_{k\ell}(x, y). \tag{5.8}$$

In particular, tensor products of Chebyshev polynomials and B-spline basis functions are used extensively: see below.

Tensor products are particularly useful representations for data (x_i, y_i, z_i) in which the behaviour of the surface is similar across the domain. They are less efficient in representing generally bland surfaces with local areas of large variations. A second (and related) disadvantage is that the number of basis functions is $n_x \times n_y$, so that to capture variation in both x and y a large number of basis functions can be required. On the positive side, if the data points (x_i, y_i) lie on or near a rectangular grid, the computations can be performed very efficiently [4]: see below.

Tensor product surfaces have been proposed [69] for modelling the kinematic behaviour of coordinate measuring machines (CMMs). An empirical model is used to describe the motion of the probe stylus assembly of the CMM (its location and orientation) in terms of three functions specifying a positional correction and three a rotational correction. Each correction is a function of three independent variables, the scale readings returned by the CMM, and is represented by a tensor product of polynomial spline curves.

Tensor product spline surfaces have also been used in the modelling of a photodiode response [127], in which the independent variables are time and active layer thickness. A spline surface approximation is used to smooth measurements of the response, represent concisely the very large quantities of measurements that are made, and permit effective manipulation of the underlying function including obtaining derivatives and evaluating convolutions.

5.5.2 Working with tensor products

Orthogonality of tensor products

If $\{\phi_k\}$ and $\{\psi_l\}$ are orthonormal[2] with respect to inner products

$$\langle p, q\rangle_u = \int_a^b p(x)q(x)u(x)\,dx, \quad \langle p, q\rangle_v = \int_c^d p(x)q(x)v(x)\,dx,$$

respectively, then $\{\gamma_{kl}(x, y) = \phi_k(x)\psi_l(y)\}$ are orthonormal with respect to the inner product

$$\langle p, q\rangle_w = \int_a^b \int_c^d p(x, y)q(x, y)w(x, y)\,dy\,dx,$$

where $w(x, y) = u(x)v(y)$.

Data approximation using tensor product surfaces

Given data points (x_i, y_i, z_i), $i = 1, \ldots, m$, the least-squares best-fit tensor product surface is found by solving

$$\min_{\mathbf{a}} \quad \sum_{i=1}^{m} (z_i - \gamma(x_i, y_i, \mathbf{a}))^2,$$

with $\gamma(x, y, \mathbf{a})$ defined by (5.8). In matrix terms, we solve

$$\min_{\mathbf{a}} \|\mathbf{z} - \Gamma\mathbf{a}\|^2,$$

where $\mathbf{z} = (z_1, \ldots, z_m)^{\mathrm{T}}$, Γ is an $m \times n_x n_y$ matrix of elements $\gamma_{k\ell}(x_i, y_i)$, and $\mathbf{a}$ is an $n_x n_y \times 1$ vector of elements $a_{k\ell}$. In this formulation, the order of the elements $a_{k\ell}$ in $\mathbf{a}$ (and the order of the corresponding columns of Γ) comes from a choice of ordering of the $n_x n_y$ basis functions $\gamma_{k\ell}(x, y)$.

In the case that the data points relate to measurements on a *grid* in the xy-domain, an alternative linear algebraic formulation is possible that exploits *separability* of the tensor product basis functions and leads to a problem that can be solved significantly faster. Let the data points be (x_i, y_j, z_{ij}), $i = 1, \ldots, m_x$, $j = 1, \ldots, m_y$, and let matrices Φ, Ψ, A and Z be defined by

$$\begin{aligned} (\Phi)_{ik} &= \phi_k(x_i), \qquad i = 1, \ldots, m_x, \qquad k = 1, \ldots, n_x, \\ (\Psi)_{j\ell} &= \psi_\ell(y_j), \qquad j = 1, \ldots, m_y, \qquad \ell = 1, \ldots, n_y, \end{aligned}$$

and

$$\begin{aligned} (Z)_{ij} &= z_{ij}, \qquad i = 1, \ldots, m_x, \qquad j = 1, \ldots, m_y, \\ (A)_{k\ell} &= a_{k\ell}, \qquad k = 1, \ldots, n_x, \qquad \ell = 1, \ldots, n_y. \end{aligned}$$

[2] That is, for the appropriate inner product, $\langle p_k, p_l\rangle = 1$ if $k = l$, 0 otherwise.

Then, the surface approximation problem is to solve

$$\min_{A} \|Z - \Phi A \Psi^{\mathrm{T}}\|^2, \tag{5.9}$$

the solution to which is given (formally) by

$$\left(\Phi^{\mathrm{T}}\Phi\right) A \left(\Psi^{\mathrm{T}}\Psi\right) = \Phi^{\mathrm{T}} Z \Psi. \tag{5.10}$$

The solution to (5.10) may obtained in two stages: by solving

$$\left(\Phi^{\mathrm{T}}\Phi\right) \tilde{A} = \Phi^{\mathrm{T}} Z$$

for $\tilde{A}$, followed by solving

$$A \left(\Psi^{\mathrm{T}}\Psi\right) = \tilde{A}\Psi$$

for A. These relate, respectively, to least-squares solutions of

$$\min_{\tilde{A}} \|Z - \Phi \tilde{A}\|^2, \tag{5.11}$$

and

$$\min_{A} \|\tilde{A} - A\Psi^{\mathrm{T}}\|^2. \tag{5.12}$$

Consequently, the *surface* approximation problem (5.9) is solved by considering *curve* approximation problems (5.11) and (5.12) as follows. First, for each $j = 1, \ldots, m_y$, find the least-squares best-fit curve

$$f_j(x) = \sum_{k=1}^{n_x} \tilde{a}_{kj} \phi_k(x)$$

to the data $(x_i, z_{ij}),\ i = 1, \ldots, m_x$. Second, for each $i = 1, \ldots, n_x$, find the least-squares best-fit curve

$$f_i(y) = \sum_{\ell=1}^{n_y} a_{i\ell} \psi_\ell(y)$$

to the data $(y_j, \tilde{a}_{ij}),\ j = 1, \ldots, m_y$.

The least-squares best-fit surface is therefore obtained in $O(m_x m_y n_x^2 + m_y n_x n_y^2)$ operations compared with $O(m_x m_y n_x^2 n_y^2)$ that would apply if separability of the basis functions is ignored. For instance, if $m_x = m_y = 1000$ and $n_x = n_y = 100$, the number of operations differ by a factor of $O(10^4)$.

5.5.3 Chebyshev polynomial surfaces

We recall from section 5.1, that a polynomial curve $p_n(x)$ of degree n on the interval $x \in [x_{\min}, x_{\max}]$ has the representation[3]

$$p_n(x) = \frac{1}{2} a_0 T_0(\hat{x}) + a_1 \mathrm{T}_1(\hat{x}) + \ldots + a_n T_n(\hat{x}) = {\sum_{k=0}^{n}}' a_k T_k(\hat{x}),$$

[3]The notation $\sum'$ indicates that the first term in the sum is halved.
The normalised variable z, in section 5.1, has been replace by $\hat{x}$.

where $\hat{x} \in [-1, +1]$ is related to x by

$$\hat{x} = \frac{(x - x_{\min}) - (x_{\max} - x)}{x_{\max} - x_{\min}}$$

and $T_j(\hat{x}),\ j = 0, \ldots, n$, are Chebyshev polynomials. A tensor product polynomial surface $p_{n_x n_y}(x, y)$ of degree n_x in x and n_y in y on the rectangular domain $(x, y) \in [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ is therefore represented by

$$p_{n_x n_y}(x, y) = \sum_{k=0}^{n_x}{}' \sum_{\ell=0}^{n_y}{}' a_{k\ell} T_k(\hat{x}) T_\ell(\hat{y}), \tag{5.13}$$

where $\hat{x}$ and $\hat{y}$ are each normalised to lie in the interval $[-1, +1]$. We apply, here, the standard convention that coefficients in the above representation which have either k or ℓ zero are written as $a_{k\ell}/2$, and the coefficient with both k and ℓ zero is written as $a_{00}/4$.

The polynomial surface (5.13) has *total degree* $n_x + n_y$, the highest combined power of x and y of a basis function. Another way of representing a polynomial surface is to require that the *total degree* of the tensor product basis functions is specified as n. Such a polynomial surface has the representation

$$p_n(x, y) = \sum_{k=0, \ell=0}^{k+\ell \le n} a_{k\ell} T_k(\hat{x}) T_\ell(\hat{y}).$$

Advantages

- For data on regular grids, the solution algorithms are efficient and, with the use of orthogonal basis functions, numerically stable.
- Given polynomial approximation software components for one dimension (evaluation of Chebyshev basis functions, etc.) the implementation of algorithms for approximation with tensor product polynomials is straightforward, especially for data on regular grids.
- For data representing similar qualitative behaviour over the domain of interest, it is usually possible to determine good approximations.
- The order of the polynomials can be used to generate nested sequences of spaces from which to approximate the data.

Disadvantages

- For data representing different types of behaviour in different regions, a tensor product representation can be inefficient.
- For scattered data there is no easily tested criterion to determine *a priori* whether or not approximation with a particular order of polynomial will be well-posed.

5.5.4 Spline surfaces

Recalling section 5.2, a tensor product spline surface $s(x, y)$ of order n_x in x with knots $\boldsymbol{\lambda}$ and order n_y in y with knots $\boldsymbol{\mu}$ on the rectangular domain $(x, y) \in [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ is represented by

$$s(x, y) = s(x, y, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{k=1}^{n_x+N_x} \sum_{\ell=1}^{n_y+N_y} c_{k\ell} N_{n_x,k}(x, \boldsymbol{\lambda}) N_{n_y,\ell}(y, \boldsymbol{\mu}), \tag{5.14}$$

where the knot vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ satisfy, respectively,

$$x_{\min} = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{N_x-1} \leq \lambda_{N_x} < \lambda_{N_x+1} = x_{\max}$$

and

$$y_{\min} = \mu_0 < \mu_1 \leq \mu_2 \leq \ldots \leq \mu_{N_y-1} \leq \mu_{N_y} < \mu_{N_y+1} = y_{\max}.$$

The spline surface (5.14) is a piecewise bivariate polynomial of order n_x in x and n_y in y on $(\lambda_i, \lambda_{i+1}) \times (\mu_j, \mu_{j+1})$, $\quad i = 0, \ldots, N_x, \quad j = 0, \ldots, N_y$. The spline is $(n_x - k - 1)$-times continuously differentiable along the knot-line $x = \lambda_i$ if $\#(\lambda_\ell = \lambda_i, \ell \in \{1, \ldots, N_x\}) = k$ (and similarly for the knot-line $y = \mu_j$). So, for example, a spline surface of order four in x and y for which the λ_i and μ_j are distinct is a piecewise bicubic polynomial, that is twice continuously differentiable along the lines $x = \lambda_i$ and $y = \mu_j$.

Advantages

- For data on regular grids, the solution algorithms are extremely efficient and numerically stable. For scattered data, it is still possible to exploit sparsity structure in the observation matrix but the gain in efficiency is much less than that for the case of one dimension.
- Given spline approximation software components for one dimension (evaluation of B-spline basis functions, etc.) the implementation of algorithms for approximation with tensor product polynomials is straightforward for data on regular grids.
- For data representing similar qualitative behaviour over the domain of interest, it is usually possible to determine good approximations.
- The knot vectors can be chosen to generate nested sequence of spaces from which to approximate the data.
- For data on a rectangular grid, it is easy to check *a priori* whether a particular choice of knots will lead to a well-posed approximation problem.

Disadvantages

- Splines require the knot vectors to be chosen, for the problems to be linear. If the data or surface exhibits different behaviour in different regions, the choice of knots can affect significantly the quality of the spline representation [73].

- For data representing different types of behaviour in different regions, a tensor product representation can be inefficient.
- For scattered data, there is no easily tested criterion to determine *a priori* whether or not approximation with splines defined by a pair of knot sets will be well posed.

5.6 Wavelets

5.6.1 Description

Wavelets are now an important tool in data analysis and a survey of their application to metrology is given in [146].

In one dimension, wavelets are often associated with a multiresolution analysis (MRA). In outline, let $L^2(\mathbb{R})$ be the space of square integrable functions $f : \mathbb{R} \longrightarrow \mathbb{R}$ so that

$$\int_{-\infty}^{\infty} f^2(x)\,dx < \infty.$$

If $f, g \in L^2(\mathbb{R})$ we define

$$\langle f, g\rangle = \int_{-\infty}^{\infty} f(x)g(x)\,dx,$$

and $\|f\|^2 = \langle f, f\rangle$. This inner-product is used to define orthogonality for functions in $L^2(\mathbb{R})$.

A starting point for MRA is a function $\psi(x)$, the *mother wavelet*. From ψ we define a double sequence of functions

$$\psi_{j,k} = \frac{1}{2^{j/2}}\psi(2^{-j}x - k),$$

using translations and dilations. The mother wavelet is chosen so that $\{\psi_{j,k}\}$ forms an orthonormal basis for $L^2(\mathbb{R})$. Any $f \in L^2(\mathbb{R})$ can be expressed as

$$f(x) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \langle f, \psi_{j,k}\rangle \psi_{j,k}(x).$$

The functions $\{\psi_{j,k}\}$, $k \in \mathbb{Z}$, form an orthonormal basis for a subspace W_j of $L^2(\mathbb{R})$ and these subspaces are used to define a nested sequence of subspaces

$$\ldots \supset V_{j-1} \supset V_j \supset V_{j+1} \supset \ldots$$

where

$$V_{j-1} = V_j \oplus W_j,$$

i.e., any function $f_{j-1} \in V_{j-1}$ can be uniquely expressed as $f_{j-1} = f_j + g_j$, with $f_j \in V_j$ and $g_j \in W_j$. We regard f_j as a smoother approximation to f_{j-1} (since

$f(x) \in V_{j-1}$ if and only if $f(2x) \in V_j$) while g_j represents the difference in detail between f_{j-1} and f_j.

The orthogonality properties mean that computations using wavelets can be made very efficiently. In particular, the discrete wavelet transform is used to decompose a uniformly spaced finite set of discrete data points (j, f_j) into component functions at different frequencies (or scales). A major feature of a wavelet analysis is that (unlike Fourier analysis) it can describe different frequency behaviour at different locations.

Wavelets can also be used to analyse signals in higher dimensions. From the orthonormal wavelet basis for $L^2(\mathbb{R})$,

$$\{(\psi_{j,k}(x), j, k \in \mathbb{Z}\}$$

an orthonormal basis for $L^2(\mathbb{R}^2)$ is obtained by taking the tensor products (section 5.5) of two one-dimensional bases functions

$$\psi_{j_1,k_1,j_2,k_2}(x, y) = \psi_{j_1,k_1}(x)\psi_{j_2,k_2}(y).$$

and these functions can be used for MRA in two dimensions.

Advantages

- Wavelets are able to represent different types of behaviour in different regions.
- For data lying on a regular grid, algorithm implementations are efficient and numerically stable.
- Wavelets provide a nested sequence of spaces from which to approximate the data.
- Wavelets are important tools in filtering and data compression.
- Wavelets do not require the specification of subsidiary parameters (but a choice of mother wavelet is required).
- Many wavelet software packages are available.

Disadvantages

- Most wavelet implementations are concerned with data on a regular grid.
- The relationship between the choice of wavelet and the effectiveness of resulting analysis is not obvious.

5.7 Bivariate polynomials

5.7.1 Description

Tensor product surfaces (section 5.5) are especially computationally effective for approximating data where the xy-coordinates (x_i, y_i) are situated on a regular grid. If the locations of (x_i, y_i) are scattered, the tensor product approach is much less efficient. In the case of one dimension, given a set of data $\{(x_i, y_i)\}_{i=1}^m$, the Forsythe method generates, implicitly, a set of orthogonal polynomials $\phi_j(x)$ such that

$$\langle \phi_j, \phi_k \rangle = \sum_{i=1}^{m} \phi_j(x_i)\phi_k(x_i) = 0, \quad j \neq k.$$

Furthermore if there are at least n distinct x_i, then approximating the data with an order n (degree $n-1$) polynomial is a well-posed problem – the associated observation matrix has full rank. In two (or higher) dimensions conditions to guarantee a well conditioned approximation problem are much more complex. For example, if the data points (x_i, y_i, z_i) are such that (x_i, y_i) lie on a circle then the basis vectors corresponding to the basis functions x^2, y^2, x, y and 1 will be linearly dependent. More generally, if (x_i, y_i) lie on (or near to) an algebraic curve (i.e., one defined as the zeros of a polynomial), then the associated observation matrix will be rank deficient (or poorly conditioned).

In a paper by Huhtanen and Larsen [138], an algorithm is presented for generating bivariate polynomials that are orthogonal with respect to a discrete inner product. It is straightforward to implement and includes provision for the possibility of linear independency amongst the basis vectors. The algorithm also provides a recursive scheme to evaluate the polynomial where the length of the recursion is at most $2k+1$ where k is the degree of the polynomial. We illustrate the use of this algorithm in fitting data generated on the surface

$$z = x^4 - y^4 + xy^3 - x^3y + 2. \tag{5.15}$$

We have generated 101 data points (x_i^*, y_i^*) uniformly distributed around the circle $x^2+y^2 = 1$ and calculated z_i^* according to (5.15) so that (x_i^*, y_i^*, z_i^*) lie exactly on the surface; see figure 5.17. We have then added random perturbations to generate data points (x_i, y_i, z_i):

$$x_i = x_i^* + e_i, \quad y_i = y_i^* + f_i, \quad z_i = z_i^* + g_i, \quad e_i, f_i, g_i \in N(0, \sigma^2).$$

There are 15 basis functions associated with a bivariate polynomial of total degree 4. For the data points $\{(x_i^*, y_i^*)\}$ and degree $k = 4$ the algorithm generates 10 orthogonal vectors out of a possible 15, the remaining five being linear combinations of the other basis vectors. The maximum computed element $|(Q^*)^{\mathrm{T}}Q^* - I|$ was 1.5543×10^{-15}. For the data points, $\{(x_i, y_i)\}$, the random perturbations are enough to ensure that the basis functions are linearly independent and the algorithm produces all 15 orthogonal vectors. The maximum computed element of $|Q^{\mathrm{T}}Q - I|$ was 5.0774×10^{-14}.

This algorithm is certainly of interest for those who wish to approximate multivariate data with polynomials and it is likely there will be further developments.

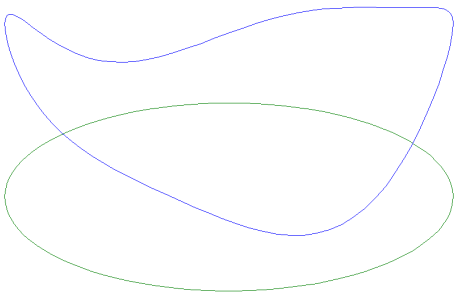

Figure 5.17: Curve defined by the quartic surface (5.15) intersected with the cylinder $x^2 + y^2 = 1$.

Multivariate orthogonal polynomials is an area of considerable research activity (see, e.g., [89]).

Advantages

- The Huhtanen and Larsen (HL) algorithm provides a method of approximating scattered data by bivariate polynomials.
- The algorithm is efficient compared to a full matrix approach and has favourable numerical properties.
- The algorithm copes with possible rank deficiency in the basis functions.
- The HL algorithm is reasonably straightforward to implement.
- The same approach can be applied in higher dimensions.
- The total order of the polynomial can be chosen to generate a nested sequence of spaces from which to chose an approximant.

Disadvantages

- Standard numerical tools for its implementation are not yet widely available.

5.7.2 Bibliography

Multivariate polynomials are discussed in [89, 138], for example.

5.8 RBFs: radial basis functions

5.8.1 Description

Let $\Lambda = \{\boldsymbol{\lambda}_j\}$, $j = 1, \ldots, n$, be a set of points in $\mathbb{R}^p$, and $\rho : \mathbb{R} \longrightarrow [0, \infty)$ a fixed function. A *radial basis function* (RBF) with centres Λ has the form

$$\phi(\mathbf{x}, \mathbf{a}) = \phi(\mathbf{x}, \mathbf{a}, \Lambda) = \sum_{j=1}^{m} a_j \rho(\|\mathbf{x} - \boldsymbol{\lambda}_j\|),$$

where $\|\mathbf{x}\| = (\mathbf{x}^{\mathrm{T}}\mathbf{x})^{1/2}$ is the Euclidean norm of a vector. Defining

$$\phi_j(\mathbf{x}) = \rho(\|\mathbf{x} - \boldsymbol{\lambda}_j\|),$$

then ϕ is seen to be a linear combination of basis functions. Therefore, approximation with RBFs follows the same general approach as with other empirical models defined in terms of basis functions. Given a set of data points $X = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^p \times \mathbb{R}\}$, $i = 1, \ldots, m$, the associated observation matrix has

$$C_{ij} = \rho(\|\mathbf{x}_i - \boldsymbol{\lambda}_j\|).$$

In least-squares approximation, estimates of the parameters $\mathbf{a}$ are found by solving

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|^2.$$

Common choices for the function ρ are i) $\rho(r) = r^3$, *cubic*, ii) $\rho(r) = e^{-r^2}$, *Gaussian*, iii) $\rho(r) = r^2 \log r$, *thin plate spline*, iv) $\rho(r) = (r^2 + \lambda^2)^{1/2}$, *multiquadric*, and v) $\rho(r) = (r^2 + \lambda^2)^{-1/2}$, *inverse multiquadric*. In practice, a scaling parameter μ_0 is required so that the RBF has the form

$$\phi(\mathbf{x}, \mathbf{a}|\mu_0, \Lambda) = \sum_{j=1}^{m} a_j \rho(\mu_0 \|\mathbf{x} - \boldsymbol{\lambda}_j\|).$$

If necessary, μ_0 can be regarded as a parameter to be determined as part of the fitting process, in which case the observation matrix $C = C(\mu_0)$ is now a nonlinear function of μ_0 and the optimisation problem becomes

$$\min_{\mathbf{a}, \mu_0} \|\mathbf{y} - C(\mu_0)\mathbf{a}\|^2,$$

a nonlinear least-squares problem (section 4.2). This problem can be solved using the Gauss-Newton algorithm for example. Alternatively, let $\mathbf{a}(\mu_0)$ solve the linear least-squares problem

$$\min_{\mathbf{a}} \|\mathbf{y} - C(\mu_0)\mathbf{a}\|^2,$$

and set $\mathbf{r}(\mu_0) = \mathbf{y} - C(\mu_0)\mathbf{a}(\mu_0)$ and $F(\mu_0) = \|\mathbf{r}(\mu_0)\|$, the norm of the residuals scaling parameter μ_0. A univariate minimisation algorithm can be applied to $F(\mu_0)$ to find an optimal estimate.

Advantages

- RBFs apply to scattered data.
- RBFs apply to multivariate data in any dimension. The computational cost is $O(mn(n+p)$, where m is the number of data points, n the number of centres and p the dimension.
- RBFs can represent different types of behaviour in different regions.
- It is generally possible to chose centres so that the data approximation problem is well-posed, i.e., there is no rank deficiency.
- RBF algorithms are easy to implement, involving only elementary operations and standard numerical linear algebra.
- By choosing the set of centres Λ appropriately, it is possible to generate a nested sequence of spaces from which to choose an approximant.

Disadvantages

- RBF basis functions have no natural orthogonality and can often lead to poorly conditioned observation matrices.
- RBFs give rise to full observation matrices with no obvious way of increasing computational efficiency.
- RBFs require the choice of subsidiary parameters, i.e., the centres and scaling parameter(s).

5.9 Neural networks

5.9.1 Description

Neural networks (NNs), see, e.g., [25, 26, 129], represent a broad class of empirical multivariate models. We present here two common types of network.

Multilayer perceptron

In a multilayer perceptron (MLP) [129, 159], a vector of inputs $\mathbf{x}$ is transformed to a vector of outputs $\mathbf{z}$ through a sequence of matrix-vector operations combined with the application of nonlinear *activation functions*. Often a network has three layers of nodes – input, hidden and output – and two transformations $\mathbb{R}^m \longrightarrow \mathbb{R}^l \longrightarrow \mathbb{R}^n$, $\mathbf{x} \longrightarrow \mathbf{y} \longrightarrow \mathbf{z}$ with

$$y_j = \psi(\mathbf{a}_j^{\mathrm{T}}\mathbf{x} + b_j), \quad z_k = \phi(\mathbf{c}_k^{\mathrm{T}}\mathbf{y} + d_k),$$

or, in matrix terms,

$$\mathbf{y} = \boldsymbol{\psi}(A\mathbf{x} + \mathbf{b}), \quad \mathbf{z} = \boldsymbol{\phi}(C\mathbf{y} + \mathbf{d}) = M(\mathbf{x}, A, \mathbf{b}, C, \mathbf{d}),$$

where A is an $l \times m$ matrix, C an $n \times l$ matrix, and $\mathbf{b}$ and $\mathbf{d}$ are l- and n-vectors, respectively. The activation function is often chosen to be the logistic sigmoid function $1/(1+e^{-x})$ or a hyperbolic tangent function $\tanh(x)$. These functions have unit gradient at zero and approach 1 as $x \to \infty$ and 0 or -1 as $x \to -\infty$. For classification problems, the network is designed to work as follows. The value of y_j indicates whether a feature specified by $\mathbf{a}_j$ is present ($y_j \approx 1$) or absent ($y_j \approx 0$ or -1) in the input $\mathbf{x}$. The output $\mathbf{z}$ completes the classification of the input according to the features identified in the hidden layer $\mathbf{y}$: the input is assigned to the qth class if $z_q \approx 1$ and $z_r \approx 0$ or -1, $r \neq q$. For empirical modelling, the second activation function is usually chosen to be the identity function $\phi(x) = x$, so that all values of output are possible, and

$$\mathbf{z} = M(\mathbf{x}, A, \mathbf{b}, C, \mathbf{d}) = C\boldsymbol{\psi}(A\mathbf{x} + \mathbf{b}) + \mathbf{d}, \tag{5.16}$$

a flexible multivariate function $M : \mathbb{R}^m \longrightarrow \mathbb{R}^n$.

Given training data comprising sets of inputs and required outputs $\{(\mathbf{x}_q, \mathbf{z}_q)\}$, an iterative optimisation process – the back-propagation algorithm – can be used to adjust the weighting matrices A and C and bias vectors $\mathbf{b}$ and $\mathbf{d}$ so that $M(\mathbf{x}_q, A, \mathbf{b}, C, \mathbf{d}) \approx \mathbf{z}_q$. Alternatively, standard large-scale optimisation techniques [67, 117, 120, 210] such as conjugate gradient methods can be employed. However, the optimisation problems are likely to be poorly conditioned or rank deficient and the optimisation algorithms need to cope with this possibility. Many algorithms therefore employ large-scale techniques combined with regularisation techniques [124, 125, 199].

MLP models are extremely flexible. Many of the problems associated with implementing them for a particular application are in deciding how to reduce the flexibility in order to produce a compact model while at the same time retaining enough flexibility in order to represent adequately the system being modelled.

RBF networks

Radial basis function (RBF) networks [36, 175, 176] have a similar design to multilayer perceptrons (MLPs) but the activation function is a radial basis function. Typically, we have

$$y_j = \rho_j(\|\mathbf{x} - \boldsymbol{\lambda}_j\|), \qquad \mathbf{z} = C\mathbf{y} + \mathbf{d},$$

where ρ_j is a Gaussian function, $\rho_j(x) = \exp\{-x^2/(2\sigma_j^2)\}$, for example. More generally, we can have

$$y_j = \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\lambda})^{\mathrm{T}} M_j (\mathbf{x} - \boldsymbol{\lambda})\right\},$$

where M_j is a symmetric, semi-positive definite matrix.

Advantages

- NNs can be used to approximate any continuous function $\mathbf{f} : \mathbb{R}^m \longrightarrow \mathbb{R}^n$ [113, 135].

- NNs can be used to perform nonlinear classification, in which data points belonging to different classes are separated by nonlinear hyper-surfaces.
- NN models are straightforward to evaluate and back-propagation algorithms, for example, are easy to implement.

Disadvantages

- The determination of optimal weights and biases is a nonlinear optimisation problem.
- The back-propagation algorithm can converge slowly to one of possibly many local minima.
- The behaviour of the model on training data can be a poor guide to its behaviour on similar data.
- The evaluation of the uncertainty associated with the fitted parameters is difficult.
- The effectiveness of the network can depend critically on its design (number and size of hidden layers).

5.10 Geometric elements

In this section we consider a class of models that have characteristics in many ways different from empirical models such as polynomials and splines. The most common geometric elements are lines in two and three dimensions, planes, circles in two and three dimensions, spheres, cylinders and cones. Less common but important in some fields are ellipses and ellipsoids, tori, aspherical surfaces and surfaces of revolution; see also section 5.11. Geometric elements generally can be defined in terms of two sets of parameters $\mathbf{a} = (\mathbf{s}^{\mathrm{T}}, \mathbf{t}^{\mathrm{T}})^{\mathrm{T}}$, those $\mathbf{s}$ defining their size and shape – *shape parameters* – and those $\mathbf{t}$ defining their location and orientation - *position parameters*. For example, a circle in the plane can be specified by one shape parameter describing its radius and two position parameters describing the location of its centre. In other parameterisations, there may be no such clear distinction.

Geometric elements are important in dimensional metrology, particularly coordinate metrology and in manufacturing and other engineering disciplines. They are used to represent the shape of manufactured parts and engineering components. They arise in many systems for which a geometrical description is required.

5.10.1 Working with geometrical elements

Most calculations with geometric elements involve the calculation of the distance $d(\mathbf{x}, \mathbf{a})$ from a data point $\mathbf{x}$ (in two or three dimensions, depending on the

element) to the profile or surface of the element in terms of its shape and position parameters $\mathbf{a}$. For example the least squares best-fit element to data $X = \{\mathbf{x}_i\}_1^m$ is found by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} d^2(\mathbf{x}_i, \mathbf{a}). \tag{5.17}$$

This type of regression is known as *orthogonal regression* since the error of fit at $\mathbf{x}_i$ is taken to be the smallest distance to the curve or surface rather that the distance calculated in a specific direction (such as parallel to the z-axis). This type of estimation is considered in section 4.3. The use of orthogonal regression is justified on the basis of maximum likelihood principles and/or on the basis of rotational invariance, since the properties of an artefact's shape determined from measurements should not be dependent on the orientation in which the artefact is measured, with respect to the co-ordinate system used.

Example: least-squares orthogonal regression with circles, implicit version

We model a circle implicitly as $f(\mathbf{x}, \mathbf{a}) = (x-a_1)^2 + (y-a_2)^2 - a_3^2 = 0$. Suppose the data points $\mathbf{x}_i = (x_i, y_i)^{\mathrm{T}}$ are generated by a co-ordinate measuring system with random effects modelled as

$$\mathbf{x}_i = \mathbf{x}_i^* + \boldsymbol{\epsilon}_i,$$

where $\mathbf{x}_i^* = (x_i^*, y_i^*)^{\mathrm{T}}$ is the data point lying on the circle $f(\mathbf{x}, \mathbf{a}) = 0$ and $\boldsymbol{\epsilon}_i$ represents a random effect. It is assumed that the components of $\boldsymbol{\epsilon}_i = (\epsilon_i, \delta_i)^{\mathrm{T}}$ are uncorrelated and drawn from a normal distribution $N(0, \sigma^2)$. The maximum likelihood estimate of the circle parameters $\mathbf{a}$ is found by minimising

$$\min_{\mathbf{a}, \{\boldsymbol{\epsilon}_i\}} \sum_{i=1}^{m} (\epsilon_i^2 + \delta_i^2) \quad = \quad \sum_{i=1}^{m} (x_i - x_i^*)^2 + (y_i - y_i^*)^2$$

subject to the constraints $f(\mathbf{x}_i^*, \mathbf{a}) = 0$. Given any $\mathbf{a}$, this sum is minimised by setting $\mathbf{x}_i^*$ equal to the point on the circle $f(\mathbf{x}, \mathbf{a}) = 0$ nearest $\mathbf{x}_i$:

$$\begin{aligned} x_i^* &= a_1 + a_3 \frac{x_i - a_1}{r_i}, \\ y_i^* &= a_2 + a_3 \frac{y_i - a_2}{r_i}, \quad \text{where} \\ r_i &= \{(x_i - a_1)^2 + (y_i - a_2)^2\}^{1/2}. \end{aligned}$$

For this $\mathbf{x}_i^*$,

$$\{(x_i - x_i^*)^2 + (y_i - y_i^*)^2\}^{1/2} = d(\mathbf{x}_i, \mathbf{a}) = r_i - a_3,$$

and the optimisation problem reduces to (5.17). ♯

Example: least-squares orthogonal regression with circles, parametric version

Alternatively, we model a circle parametrically as

$$x^* = a_1 + a_3 \cos u, \quad y_i^* = a_2 + a_3 \sin u.$$

The maximum likelihood estimation problem can then be posed as

$$\min_{\mathbf{a},\{u_i\}} \sum_{i=1}^{m}(\epsilon_i^2+\delta_i^2) \;=\; \sum_{i=1}^{m}(x_i-a_1-a_3\cos u_i)^2+(y_i-a_2-a_3\sin u_i)^2.$$

Given any $\mathbf{a}$, this sum is minimised by setting u_i according to

$$\begin{aligned}\cos u_i &= \frac{x_i-a_1}{r_i},\\ \sin u_i &= \frac{y_i-a_2}{r_i},\end{aligned}$$

so that the optimisation problem again reduces to (5.17). ♯

For the simpler geometric elements specified by parameters $\mathbf{a}$, the distance $d(\mathbf{x},\mathbf{a})$ from a point $\mathbf{x}$ to the element can be calculated as an explicit function of $\mathbf{x}$ and $\mathbf{a}$. For more complicated elements, a numerical approach is required to solve the associated *foot point problems*; see section 4.3.

Rotations and translations. Often the position parameters are defined in terms of rotations and translations. Let

$$R(\boldsymbol{\alpha}) = R_z(\gamma)R_y(\beta)R_x(\alpha)$$

be the composition of three plane rotations defined by

$$R_x(\alpha)=\begin{bmatrix}1&0&0\\0&\cos\alpha&-\sin\alpha\\0&\sin\alpha&\cos\alpha\end{bmatrix},\quad R_y(\beta)=\begin{bmatrix}\cos\beta&0&\sin\beta\\0&1&0\\-\sin\beta&0&\cos\beta\end{bmatrix}$$

and

$$R_z(\gamma)=\begin{bmatrix}\cos\gamma&-\sin\gamma&0\\\sin\gamma&\cos\gamma&0\\0&0&1\end{bmatrix}.$$

A roto-translation can be written in the form

$$\hat{\mathbf{x}} = T(\mathbf{x},\mathbf{t}) = R(\boldsymbol{\alpha})R_0(\mathbf{x}-\mathbf{x}_0),$$

and is specified by parameters $\mathbf{t}=(\mathbf{x}_0^{\mathrm{T}},\boldsymbol{\alpha}^{\mathrm{T}})^{\mathrm{T}}$ and fixed rotation R_0. The inverse transformation T^{-1} is

$$\mathbf{x} = \mathbf{x}_0 + R_0^{\mathrm{T}}R^{\mathrm{T}}(\boldsymbol{\alpha})\hat{\mathbf{x}}.$$

Example: orthogonal regression with cylinders I

Suppose we wish to fit a cylinder to data points $\{\mathbf{x}_i\}_1^m$. A cylinder is specified by a point on its axis $\mathbf{x}_0$, an axis direction vector $\mathbf{n}$ and its radius. If the cylinder axis is approximately coincident with the z-axis, we can parameterise the cylinder as follows:

$$\mathbf{x}_0(\mathbf{a})=\begin{bmatrix}a_1\\a_2\\0\end{bmatrix},\quad \mathbf{n}(\mathbf{a})=R_y^{\mathrm{T}}(a_4)R_x^{\mathrm{T}}(a_3)\mathbf{e}_z,\quad \mathbf{e}_z=(0,0,1)^{\mathrm{T}},$$

and radius a_5, five parameters in all. This parameterisation becomes less stable and eventually breaks down as the angle the cylinder axis makes with the z-axis approaches a right angle. A family of parameterisations generated from this parameterisation can used to describe cylinders in a general orientation and location. Let $\mathbf{n}_0$ be the approximate axis direction and R_0 a fixed rotation matrix such that $R_0^{\mathrm{T}}\mathbf{n}_0 = \mathbf{e}_z$. Similarly, let $\mathbf{z}_0$ be a point on the nominal axis. Then the cylinder is parameterised in terms of $\mathbf{x}_0(\mathbf{a})$, $\mathbf{n}(\mathbf{a})$ and its radius, where

$$\mathbf{x}_0(\mathbf{a}) = \mathbf{z}_0 + R_0^{\mathrm{T}} \begin{bmatrix} a_1 \\ a_2 \\ 0 \end{bmatrix}, \quad \mathbf{n}(\mathbf{a}) = R_0^{\mathrm{T}} R_y^{\mathrm{T}}(a_4) R_x^{\mathrm{T}}(a_3) \mathbf{e}_z.$$

Members of this family of parameterisations are specified by the extra constants determining the fixed translation vector and rotation matrix. In order to select an appropriate member of the family, an initial indication of the axis is required.

The distance $d(\mathbf{x}, \mathbf{a})$ to a cylinder parameterised in this way is given by

$$d(\mathbf{x}, \mathbf{a}) = \|(\mathbf{x} - \mathbf{x}_0(\mathbf{a})) \times \mathbf{n}(\mathbf{a})\| - a_5, \tag{5.18}$$

where $\mathbf{c} \times \mathbf{d}$ denotes the cross product of vectors. ♯

Example: orthogonal regression with cylinders II

We consider again orthogonal regression with cylinders, using a slightly different approach so that the position and shape parameters are separated. In the first approach described above, we think of moving and shaping the cylinder so that it lies as close as possible to the data. In this second approach we think of moving the data so that it is as close to possible to the cylinder.

A cylinder in *standard position* has its axis coincident with the z-axis. A cylinder has one shape parameter, its radius, and a cylinder in standard position is given by the equation

$$f(\mathbf{x}, s) = f(x, y, z, s) = x^2 + y^2 - s^2 = 0.$$

The distance from a point $\mathbf{x} = (x, y, z)^{\mathrm{T}}$ to a cylinder in standard position is given by $d(\mathbf{x}, s) = (x^2 + y^2)^{1/2} - s$.

Suppose, as before, we wish to fit a cylinder to data points $\{\mathbf{x}_i\}_1^m$. We assume that the data has been transformed by an initial translation and rotated so that the data approximately lies in the surface of the cylinder in standard position. Let T be the roto-translation defined by $\mathbf{t} = (a_1, a_2, a_3, a_4)^{\mathrm{T}}$, where

$$\hat{\mathbf{x}}(\mathbf{t}) = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = R_y(a_4) R_x(a_3) \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} a_1 \\ a_2 \\ 0 \end{bmatrix} \right).$$

The distance from a point $\mathbf{x}$ to the cylinder is given in terms of the position parameters $\mathbf{t}$ and shape parameters s by

$$d(\mathbf{x}, \mathbf{a}) = d(\hat{\mathbf{x}}(\mathbf{t}), s) = (\hat{x}^2 + \hat{y}^2)^{1/2} - s. \tag{5.19}$$

The advantages of this approach are firstly, the calculation of the distance and its derivatives is simpler (compare (5.19) with (5.18)) and, secondly and

more importantly, the calculations involving the transformation parameters are separated from the shape parameters and are largely generic, independent of the geometric element. ♯

5.10.2 Bibliography and software sources

Least-squares and Chebyshev regression with geometric elements and related form and tolerance assessment problems are considered in [5, 6, 7, 41, 42, 65, 95, 96, 97, 98, 102, 108, 114, 194, 211]. The package LSGE — least squares geometric elements — is available for download from EUROMETROS [9, 92].

5.11 NURBS: nonuniform rational B-splines

A nonuniform rational B-splines curve of order k is defined as a parametric curve $\boldsymbol{C} : \mathbb{R} \longrightarrow \mathbb{R}^2$ with

$$\boldsymbol{C}(u) = \frac{\sum_{j=0}^{n} N_{k,j}(u|\boldsymbol{\lambda}) w_j \boldsymbol{P}_j}{\sum_{j=0}^{n} N_{k,j}(u|\boldsymbol{\lambda}) w_j},$$

where $\boldsymbol{P}_j \in \mathbb{R}^2$ are the control points, w_j weights and $N_{k,j}(u|\boldsymbol{\lambda})$ B-spline basis functions defined on a knot set $\boldsymbol{\lambda}$ (section 5.2).

NURBS surfaces $\boldsymbol{S} : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ are generated using tensor products (section 5.5) of B-spline basis functions:

$$\boldsymbol{S}(u,v) = \frac{\sum_{j=0}^{n} \sum_{q=0}^{m} N_{k,j}(u|\boldsymbol{\lambda}) N_{l,q}(v|\boldsymbol{\mu}) w_{jq} \boldsymbol{P}_{jq}}{\sum_{j=0}^{n} \sum_{q=0}^{m} N_{k,j}(u|\boldsymbol{\lambda}) N_{l,q}(v|\boldsymbol{\mu}) w_{jq}},$$

where $N_{k,j}(u|\boldsymbol{\lambda})$ and $N_{l,q}(v|\boldsymbol{\mu})$ are the B-spline basis functions, $\boldsymbol{P}_{jq} \in \mathbb{R}^3$ are control points, and w_{jq} weights.

Nonuniform rational B-splines (NURBS) are used for computer graphics and extensively in computer-aid design for defining complex curves and surfaces and are therefore important in co-ordinate metrology.

Advantages

- NURBS can be used to model and modify highly complex curves and surfaces.
- The shape of the curve or surface is easily determined and modified by the location of the control points. NURBS provide local control, so that shifting one control point only affects the surface shape near that control point.
- NURBS are invariant under scaling, translation, shear, and rotation,

- NURBS can be used to define quadric surfaces, such as spheres and ellipsoids, commonly used in CAD exactly. Parametric B-spline surfaces can only approximate such surfaces and in doing so require many more control points.

Disadvantages

Although NURBS are in principle straightforward to implement, efficient and numerically stable approaches require appropriate use of the recurrence formulae associated with B-splines.

- Data approximation with NURBS (fitting a cloud of points with a NURBS curve or surface) is likely to give rise to rank deficient or poorly conditioned problems. However there are a number of ways of approaching approximation with parametric curves and surfaces, some of which give rise to well conditioned problems (see, e.g., [39, 103]).

5.11.1 Bibliography and software sources

Curve and surface representation in computer-aided design is described in [93, 182], for example. A number of software packages for NURBS are available for download including, for example, [195].

Chapter 6

Introduction to model validation

6.1 What is a valid model?

Having developed a mathematical model of a physical system, how do we know if the model is a good representation of reality?

There are two aspects of model validity. The first is *internal consistency*. To the extent that a model represents a set of mathematical statements, its validity can be checked for mathematical correctness. Typically, a model has a set of inputs described in terms of facts and assumptions about components of the physical system. It also has a set of outputs in terms of predicted behaviour of systems to which it applies. If the model is internally consistent then the outputs are valid so long as the inputs are valid.

The second aspect of validity is *external consistency* with prior information and/or experimental results. A model is valid if information extracted from it is not contradicted by other valid information. The validation status of a model is consolidated as it is shown to be consistent with more external information. Validation of an internally consistent model focuses on the extent to which the assumptions associated with the model apply, i.e., a validation of the applicability of the model.

The main output of the modelling process is usually a measurement result, i.e., an estimate of the value of a parameter, and a statement of its associated uncertainty.

Example: a simple spring balance.

Consider a simple spring balance used to estimate the weight W of an object based on the measurement of the extension of a spring. The linear extension is converted into an angular motion of an indicating needle that moves over a scale. The main inputs to the modelling process are:

- Hooke's Law for the extension $e = H(w, k)$ of a spring in terms of the applied load w and spring constant k.
- Hooke's constant k for the particular spring.
- The standard uncertainty σ_k associated with the estimate k.
- The functional dependence of the position $x = f(e)$ of the indicating needle on the extension e of the spring.
- Random effects with standard deviation σ_x associated with reading the position x of the needle on the scale.
- A specification of the minimum w_{min} and maximum w_{max} loads to be applied to the instrument.

The main outputs from the modelling process are:

- A function
$$w = F(x, k) \tag{6.1}$$
that converts the reading x of the indicator position into an estimate w of the artefact's weight.
- A rule to obtain the value
$$\sigma_w = \sigma_x \left(\frac{\partial F}{\partial x} \right) \tag{6.2}$$
of the standard uncertainty σ_w associated with the estimate w.

The validation of the internal consistency addresses the question: Assuming the inputs are valid, are the outputs valid? Such issues associated with this model include:

V1 The correctness of the function F derived from Hooke's law and the functional dependence $x = f(e)$.

V2 The correctness of the partial derivative of F.

V3 The expression for σ_w does not take into account the uncertainty in the value of Hooke's constant k and could therefore be overly optimistic.

V4 The expression for σ_w is based on a linear approximation to F at x. This linearisation will be invalid if the function F is highly nonlinear.

The internal consistency validation does not necessarily require any in depth knowledge of the metrological/physical system, only expertise in discrete modelling. Once the internal consistency of a model is established, the focus of the validation moves to the assumptions, explicit or implicit, associated with the model inputs. An expert in the area might wish to investigate:

V5 The applicability of Hooke's law to the type of spring.

V6 The assigned value for Hooke's constant.

V7 The functional form $x = f(e)$ relating the indicating needle to the spring extension.

V8 The dependence of the system on environmental factors such as temperature. The model as it stands depends only on the spring constant k.

In this activity, the expert is assessing the consistency of the model with prior information about the behaviour of similar systems. Finally, the validity of the model in terms of its outputs can be tested against experimental data. In this task, it is important that the statistical nature of the model outputs and the experimental data are taken into account.

6.2 Model validation as risk management

While the need for model validation seems self-evident, validation will generally require resources. It is therefore necessary to assess the *risks* associated with an invalid model and then design and implement *validation responses* to limit the risks to an acceptable level. In this, it is prudent to balance the risks incurred by an invalid model with the cost of validation. In the sections below we discuss the risks associated with each of the main components of the discrete modelling process and indicate suitable validation responses. The analysis of risks is discussed in SS*f*M Best Practice Guide No. 1: *Validation of software in measurement systems* [206].

Chapter 7

Validation of the model

In this chapter, we look at validation of how the model is built — the specification of the functional and statistical models.

7.1 Validation of the functional model

The functional model describes the mathematical relationships between the variables and parameters of the model. Typically, the functional model takes the form

$$y = \phi(\mathbf{x}, \mathbf{a}),$$

where y is the response variable and $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$ are the covariates or explanatory variables, and $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$ are the model parameters that specify the function ϕ. In the case of a linear response depending on a single variable we have

$$y = a_1 + a_2 x,$$

for example. There are two main types of functional model. A *physical model* is one for which there is a theory that defines how the variables depend on each other. An *empirical model* is one in which a relationship between the variables is expected or observed but with no supporting theory. Many models have both empirical and physical components.

Validation of the functional model concerns i) its comprehensiveness, i.e., the extent to which all the main variables and parameters are covered by the model and ii) the correctness of the functional relationship between the variables and parameters.

7.1.1 Comprehensiveness of the functional model

The specification of the functional model aims to achieve an appropriate level of comprehensiveness.

Risks. The risks associated with an overly comprehensive model are mainly concerned with the unnecessary commitment of resources to:

- Develop, understand and manipulate the model.
- Monitor or control variables that may have no significant impact on system behaviour.
- Solve computational problems associated with determining the model parameters from data.

The risks associated with a minimal model are mainly concerned with the impact on how well the model performs, including:

- Scope of the model outputs.
- Validity of the model outputs.
- Flexibility of the model to cope with different hardware components or operating conditions.

From knowledge of the physical system, all the factors and/or variables that could have a potential influence on the system can be listed in a table or datasheet and a status assigned to them according to their influence:

- *Unknown parameter.* The value of the variable is unknown and has to be determined from a fit of the model to data.
- *Parameter with prior estimate.* The value of the variable is unknown and has to be determined from a fit of the model to data. A prior estimate of the variable is available along the associated uncertainty.
- *Measured subject to uncertainty.* The variable is to be measured and the uncertainty associated with the measurement is likely to contribute to the uncertainty in the outputs.
- *Accurately measured.* The variable is to be measured but the uncertainty associated with the measurement is not significant compared to other sources of uncertainty.
- *Constant.* The variable can be set to its nominal value and treated as exact.
- *Null.* The value of the variable has no influence on the system and can be omitted from the model.

Assigning a variable to a status high on this list is a move towards increasing the comprehensiveness. In general, it is a good idea to start with a comprehensive model and then justify and document each step in its simplification. For example, a system may be moderately sensitive to temperature so that, on a first analysis, temperature should be included in the model and assigned a

status of 'measured subject to uncertainty'. However, if the system is kept in an accurately controlled temperature environment a status of 'constant' may be sufficient. This assignment can be validated by numerical simulation.

Suppose the temperature controlled environment guarantees that the temperature is kept within 1 °C of 20.0 °C and there is also available a temperature sensor that measures with a standard uncertainty 0.2 °C. If

$$y = \phi(t, \mathbf{x}, \mathbf{a})$$

represents the functional model in terms of temperature t, other variables $\mathbf{x}$ and parameters $\mathbf{a}$, then a quick estimate of the sensitivity of y to temperature can be determined by evaluating

$$\phi(t_0, \mathbf{x}, \mathbf{a}), \quad \phi(t_0 \pm \Delta t, \mathbf{x}, \mathbf{a})$$

for $t_0 = 20.0$ °C, $\Delta t = 1.0$ °C and 0.2 °C and for typical values of the variables $\mathbf{x}$ and $\mathbf{a}$. This provides an estimate of the variation of the model values due to variation in temperature in the ranges 20.0 °C ± 1.0 °C and 20 °C ± 0.2 °C. Then:

- *If* the variation in the model values for the case $\Delta t = 0.2$ °C is significant compared to the likely uncertainty associated with y, then t should be assigned a status of 'measured subject to uncertainty'; *otherwise*
- *If* the variation in the model values for the case $\Delta t = 1.0$ °C is significant compared to the likely measurement uncertainty associated with y, then t should be assigned a status of 'accurately measured'; *otherwise*
- The temperature t is assigned a status of 'constant'.

The interpretation of 'significant' will depend on circumstances. If the maximum variation of the model response due to a variable in its operating range is less than 1% of the uncertainty associated with the response, it may well be safe to regard the variable as a constant. However, if the system has a nonlinear response or in other ways can behave differently for different values of the variables, it is possible that a variable that has no major impact in one set of circumstances may well be significant in others. For this reason it is usually preferable to err on the side of a comprehensive model. In particular, quantities with prior calibrated values and associated uncertainties (section 4.11) can usually be included in the model without introducing major complications at the model solving stage (see, for example, [31, 106]).

Information about assumptions and status of the variables and parameters in a model can be recorded in a data sheet such as table 7.1. If the design of the system changes the data sheet can be updated to reflect the changes.

Validation responses.

- Design review by an expert in the metrology field, reviewing the data sheet of model variables and checking assumptions.

Variable/Parameter	v_1	v_2	...	v_p
Nominal Value				
Range				
Measurement uncertainty				
Status				
Notes				

Table 7.1: Example data sheet recording assumptions about the variables and parameters in a model. Not all cells will be applicable to all variables or parameters.

- Numerical simulation comparing the behaviour of comprehensive models with simpler models.

7.1.2 Correctness of the functional model

Risks. The risks associated with an incorrect functional model are concerned with functionality: the model gives invalid predictions. Often physical models are derived from models of subcomponents of the system which are aggregated, simplified or approximated. Errors can be introduced in a number of ways:

Typographical errors: errors in copying equations from source to where they are used;

Conceptualisation errors: incorrect understanding of the underlying physics;

Approximation errors: inappropriate simplifications, e.g., linearisations, significantly changing the behaviour of the model;

Scoping errors: models used outside the scope of their original application.

Validation responses for physical models. Appropriate techniques for this aspect of model validation are:

- Design review by an expert in the metrology area to check the modelling of the underlying physics.
- Design review by a modelling expert to check the mathematical derivation of equations.
- Numerical simulations to check the effect of approximations, simplifications, linearisations, etc., on the model values (relative to the likely measurement uncertainty).
- Evaluation of the model at variable/parameter values for which the physical response is known accurately.
- Numerical simulations to check the qualitative behaviour of the model against expected behaviour. For example, if a response is expected to

increase (or decrease) as a variable is increased, the model can be tested to verify that it exhibits this behaviour.

Empirical models are often used to represent observed or expected behaviour, for example, a calibration curve associated with a sensor's performance. Since empirical models are not derived from a physical theory their validation tends to focus on how well they represent measurement data; see chapter 10. Although there may be no explicit physical model available, there is often concrete knowledge about the type of behaviour to be expected and it is important that the empirical model is capable of reproducing this. Empirical models such as polynomial curves (section 5.1) and Fourier series (section 5.3) have been used successfully for many years. Part of their success is due to the fact that they have proven approximation properties and can represent any smooth curve. However, they are not appropriate for modelling systems in which the response approaches a limit asymptotically (as in saturation or decay with time). For these situations, an asymptotic polynomial model (section 5.4) would be more appropriate.

Validation responses for empirical models.

- Design review by a metrology and/or modelling expert to check the empirical model is appropriate for the expected type of behaviour.
- Numerical simulations to check the qualitative behaviour of the model against expected behaviour.

7.2 Validation of the statistical model

If a typical functional model is of the form $y^* = \phi(\mathbf{x}^*, \mathbf{a})$ and describes the relationship between y^* and $\mathbf{x}^*$ in the absence of random effects, a typical statistical model associated with measurements $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ is of the form

$$y_i = y_i^* + \epsilon_i, \quad \mathbf{x}_i = \mathbf{x}_i^*, \quad \epsilon_i \in N(0, \sigma^2),$$

which states that the variables $\mathbf{x}_i$ are measured accurately but that the measurements y_i are subject to independent, normally distributed random effects represented by ϵ_i with standard deviation σ. Thus, a complete statement of the statistical model specifies (i) the uncertainty structure, i.e., which variables are subject to random effects and what correlations exist between them, (ii) statistical distributions for the random effects and (iii) values of parameters associated with the distributions. While the derivation of the functional model can often rely on physical theory, the statistical model attempts to describe aspects of the system that are generally less well understood and, by their nature, inexact.

Risks. A poor statistical model can lead to:

Variable/Parameter	v_1	v_2	...	v_p
Nominal value				
Range				
Distribution				
Statistical parameters				
Status				
Notes				

Table 7.2: Data sheet as in table 7.1 expanded to record information about the functional *and* statistical model.

- Poor choice of solution method.
- Invalid model solutions.
- Invalid uncertainties associated with the model solution.

While an incorrect functional model will often be detected when the model is fitted to measurement data, a poor statistical model can produce estimates of the uncertainty associated with the fitted parameters that are out by a factor of two or more without anything being obviously wrong. Since decisions about the development or purchase of an instrument are made on the basis of an uncertainty statement, the impact of a poor statistical model can be large.

Assumptions about the main components of the statistical model – uncertainty structure, distributions and their parameter values – can be recorded in the datasheet used to summarise the functional model (table 7.2). Assumptions that need to be validated include the following.

Measurements of variables treated as exact. It is often assumed that only one variable (the response variable) is subject to random effects. In practice, many variables may be subject to random effects, some more significant than others. A decision to treat any one as exact has to be justified.

Uncertainties treated as having equal variance. While in many situations it is appropriate to assume that the random effects associated with a set of measurements are drawn from the same distribution (and hence have the same variance), the magnitude of these effects often have a dependence on the magnitude of the variable. For example, if the measurement of y is subject to a relative uncertainty then the statistical model

$$y = y^*(1+\epsilon), \quad \epsilon \in N(0,\sigma^2), \quad \text{or} \quad y = y^* + \epsilon, \quad \epsilon \in N(0,\sigma^2 y^2),$$

is appropriate. More generally, if estimates $y = y(\mathbf{x})$ are determined from measurements of variables $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$ with associated uncertainty (covariance) matrix $V_{\mathbf{x}}$, then the uncertainty associated with y is estimated by

$$\sigma_y^2 = (\nabla_{\mathbf{x}} y)^{\mathrm{T}} V_{\mathbf{x}} (\nabla_{\mathbf{x}} y), \tag{7.1}$$

where $\nabla_{\mathbf{x}} y = (\frac{\partial y}{\partial x_1}, \ldots, \frac{\partial y}{\partial x_p})^{\mathrm{T}}$ is the vector of partial derivatives of y with respect to x_j.

Random effects treated as independent. The random effects associated with a sequence of measurements by an individual sensor might well have a correlation structure and a time series analysis (or other methods) can be used to characterise this structure (see, e.g., [32]). Often, a more important (and more tractable) source of correlation is due to the dependence of estimates of variables on a common set of measurements. Suppose that estimates of $y = y(x, x_0)$ are determined from measurements of a variable x and constant x_0. (For example, y could be the length of a metal rod, x its temperature and x_0 the coefficient of thermal expansion for the metal.) If the measurements of x_i and x_0 are modelled as

$$x_i = x_i^* + \epsilon_i, \quad \epsilon_i \in N(0, \sigma^2), \qquad x_0 = x_0^* + \epsilon_0, \quad \epsilon_0 \in N(0, \sigma_0^2),$$

where ϵ_i and ϵ_0 represent normally distributed random effects, then the covariance of y_i with y_j is given by

$$\mathrm{cov}(y_i, y_j) = \sigma_0^2 \frac{\partial y}{\partial x_0}(x_i, x_0) \frac{\partial y}{\partial x_0}(x_j, x_0).$$

This effect is termed *structural correlation* because the correlation appears as a consequence of the structure in the functional relationship between the variables and can be quantified and subsequently taken into account at the model solving stage.

Probability distributions assumed to be normal. The assumption of normality can usually be justified for a number of reasons including (from an empirical view point) historical information on the behaviour of a sensor and (from a theoretical view point) the Central Limit Theorem (see, e.g., [187, chapter 5]). If the actual distribution is approximately symmetric, lightly tailed (vanishingly small probability far from the mean) and unimodal (has one peak), then an assumption of normality is usually safe. However, if it is known that the distribution is far from normal, then such an assumption could lead to an invalid model.

One area in which non-normality can be predicted is in the nonlinear transformation of a normally distributed variable. If $y = x_1 + x_2 t$ is a linear function of a variable t, and t is measured subject to random effects modelled as

$$t = t^* + \epsilon, \quad \epsilon \in N(0, \sigma_t^2),$$

then the statistical model for y is

$$y^* = x_1 + x_2 t^*, \quad y = y^* + \delta, \quad \delta \in N(0, x_2^2 \sigma_t^2).$$

This model is exact: if the variance of t is σ_t^2, then the variance of y is $x_2^2 \sigma_t^2$ (section 2.4). However, if $y = y(t)$ is a *nonlinear* function of t then the variance σ_y^2 of y is *approximated* by

$$\sigma_y^2 = \left(\frac{\partial y}{\partial t} \right)^2 \sigma_t^2. \tag{7.2}$$

This approximation will be good so long as the curvature of y at t is small compared with σ_t. This approximation needs to be validated as the following two examples show.

Figure 7.1 shows the curve $y = 1/t$ and 200 data points (t_i, y_i) generated according to the model

$$y_i = 1/t_i, \quad t_i = 3 + \epsilon_i, \quad \epsilon_i \in N(0, 1).$$

The estimate of the standard deviation of y_i derived from (7.2) is $\sigma_y = 1/9 \approx 0.11$ while the value obtained from the sample is 0.20. The estimate of the variance of the y_i given by (7.2) is derived using the approximating tangent line to the curve at (3,1/3). The linearised estimate of the y_i's are shown on the right hand vertical axis while the actual sample is shown on the left hand vertical axis. The linearisation fails to account for the nonlinear behaviour of the function.

Figure 7.2 presents the same type of information for the curve $y = t^2$ and 200 data points (t_i, y_i) generated according to the model

$$y_i = t_i^2, \quad t_i = 1 + \epsilon_i, \quad \epsilon_i \in N(0, 1).$$

In this case, the predictions derived from the linearisation of the model implicit in (7.2) are completely misleading: compare the spread of values of the two sides of the figure.

Monte Carlo simulations (see section 8.2 and [71]), such as those used to generate figures 7.1 and 7.2, can be used to check the actual variation of a function $y(\mathbf{t})$ of measured quantities compared with those predicted from a formula such as (7.2) or the more general (7.1).

Validation responses for statistical models. Appropriate techniques for this aspect of model validation are:

- Design review by an expert in the metrology field, reviewing the data sheet for the statistical model for the measurement data and checking assumptions.
- Design review by modelling expert to check the statistical models for derived quantities.
- Numerical simulation to check the effect of approximations, simplifications, linearisations, etc., associated with the statistical model.
- Monte Carlo simulations to check the variation in derived quantities against the predicted variation.

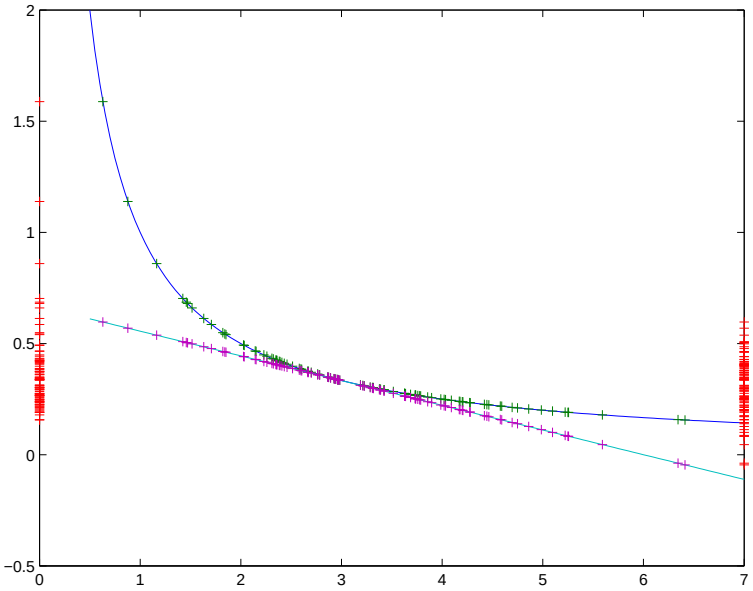

Figure 7.1: Actual and estimated variation of $y_i = 1/t_i$ where t_i are normally distributed.

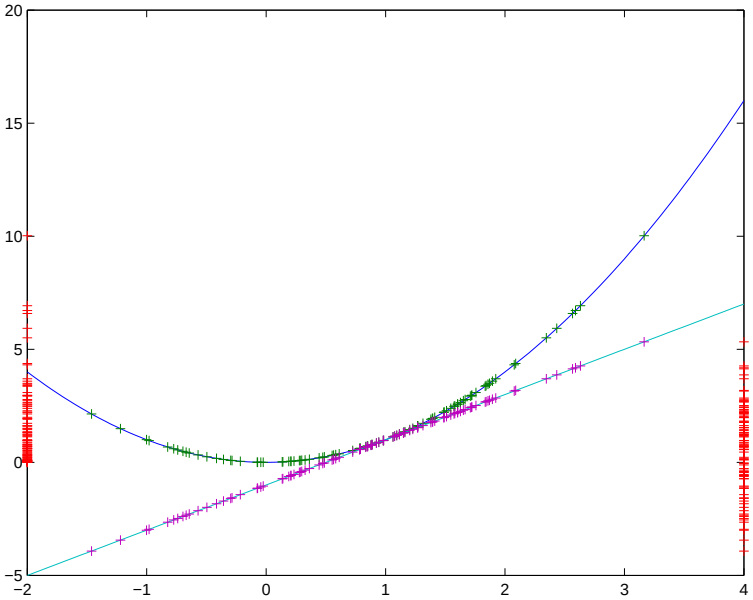

Figure 7.2: Actual (left hand vertical axis) and estimated variation (right hand vertical axis) of $y_i = t_i^2$ where t_i are normally distributed.

Chapter 8

Estimator validation

See also [10] on algorithm testing.

The previous chapter has been concerned with validating how the model has been built. This chapter addresses how the model is solved, i.e., the choice of estimator. It looks at the estimation of parameters and the evaluation of the associated standard uncertainties.

Risks. A poor choice of estimator can lead to

- Biased estimates of the model parameters.
- Overly optimistic or pessimistic uncertainties associated with the fitted parameters.

The measurement results and the associated uncertainties are the outputs from the estimation process and the quality of these outputs is limited directly by the estimator. The functional and statistical models may characterise the measurement system comprehensively but this comprehensiveness will only be translated into the measurement results if the estimation process also embodies this characterisation. An (inefficient) estimator that makes poor use of the data will produce parameter estimates that have a larger associated uncertainty than warranted from the uncertainty associated with the data (section 3.3, section 4.10). A possible consequence of this is that, unnecessarily, additional measurements are undertaken or hardware upgrades are implemented in order to meet the system performance specification.

8.1 Estimator validation issues

Models of experimental systems can be complicated for a number of reasons. For example, the equations describing relationships between the variables and

parameters may be quite involved or the data may be highly correlated. For such systems, estimators which may be expected to perform well (from maximum likelihood principles) could well require the solution of difficult optimisation problems. In these circumstances, we may wish or be forced to simplify the model and/or the optimisation problem in order to define an *approximate estimator* that is easier to implement.

The following paragraphs outline typical situations in which estimator validation is required.

Approximations in the functional model. As considered in section 7.1, the functional model is often simplified to make it more tractable. Any such change to the functional model will have an effect on the estimates of the fitted parameters and consequently issues considered in the validation of the functional model will be directly relevant to the choice of estimator. However, two additional aspects of functional model validation are important at the model solving stage.

Firstly, while at the model building stage, we may be quite comfortable with moderately complex functional models, when it comes to the practical step of implementing methods to solve the equations derived from these complex models a further degree of simplification may seem necessary over and above those already adopted.

Secondly, for well-conditioned, linear problems (section 3.7, [68, 134]) we can expect that a relative change in the functional model will have a similarly sized relative change in the model solution. For poorly conditioned and/or nonlinear problems the effect of an approximation of the functional model on the solution parameters is not so easy to predict and it may be that a seemingly insignificant change to the model in terms of the function values can have an unacceptably large change in the model solution.

A linear approximation to a nonlinear model can have a significantly different behaviour, qualitatively and quantitatively, from the nonlinear model. Figure 8.1 shows the linear approximation to an exponential model of the form $y = Ae^{-Bx}$. The linear model predicts that the response variable is zero at $x = 3$. For the exact model, the response is never zero.

Approximations in the statistical model. Similarly to the case of the functional model, at the model-solving stage additional simplifications to the statistical model are made, either explicitly or implicitly, in order to produce a simpler computational problem to be solved. Examples include (i) measurements of variables treated as exact, (ii) input quantities treated as exact, (iii) random effects in observation equations treated as independent, (iv) random effects in observation equations treated as having equal variance and (v) random effects treated as normally distributed.

The consequences of such approximations are in two areas. The first is that the information in the data is not used to maximum advantage and the resulting parameter estimates are not as good, in terms of their bias and variability (section 3.3). Secondly, for most estimators, the estimate of the uncertainty associated with the fitted model parameters is derived on the basis of a set statistical model for the data. If the statistical model does not apply, the

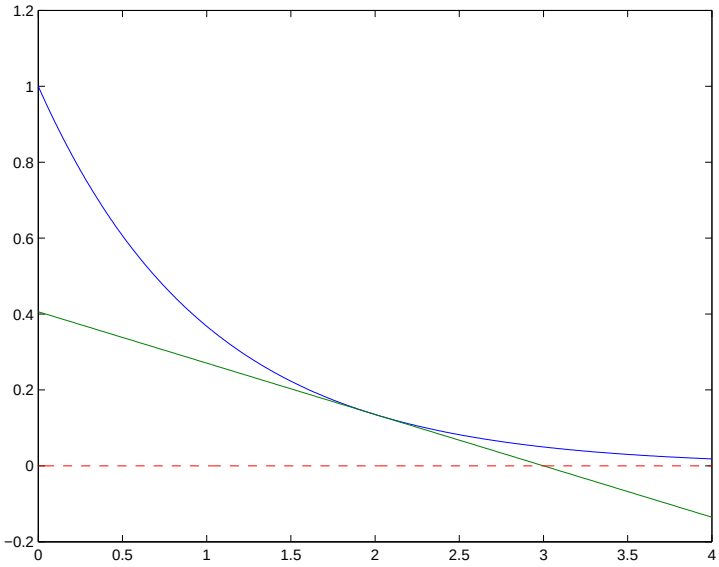

Figure 8.1: Linear approximation to an exponential model.

uncertainty estimates could be completely invalid. While it is acceptable to apply a suboptimal estimator to data as long as the uncertainty calculations reflect the true statistical model and estimator behaviour, uncertainty estimates based on an approximate statistical model for any estimator can give misleading information.

Approximate solution method. Approximations in the functional and statistical model are generally reflected in changes to the error function to be minimised. If an approximate solution method is employed to minimise the error function then, again, the parameter estimates can be biased and/or the uncertainty estimates invalid.

For these reasons, it is important that the choice of parameter estimation method is validated. We describe three classes of validation techniques (i) Monte Carlo simulation, (ii) null space benchmarking and (iii) estimator analysis.

8.2 Monte Carlo simulations

Monte Carlo simulation (see, e.g., [71]) is an important, general tool in modelling and analysis and can be used effectively in estimator validation. The general approach is as follows.

I Given a choice of model parameters $\mathbf{a}^* = (a_1, \ldots, a_n)^{\mathrm{T}}$, generate $\mathbf{x}^*$ satisfying the functional model.

II For $q = 1, 2, \ldots, N$,

II.1.q Generate perturbed data sets $\mathbf{x}_q = \mathbf{x}^* + \Delta_q$ where Δ_q is generated according to the statistical model.

II.2.q Apply the estimator $\mathcal{A}$ to the data set $\mathbf{x}_q$ to determine an estimate $\mathbf{a}_q = (a_{1,q}, \ldots, a_{n,q})^{\mathrm{T}}$ of the model parameters. Store $\mathbf{a}_q^{\mathrm{T}}$ as the qth row of an $N \times n$ matrix A.

III Compare statistics associated with A with $\mathbf{a}^*$.

In step II.1.q, we need to be able to generate data from the appropriate statistical distributions using corresponding random number generators (see, e.g., [86, 173, 197]). For step III we can, in particular, form the mean estimate $\bar{\mathbf{a}}$ whose jth element $\bar{a}_j$ is the mean of the jth column of A and the sample covariance matrix

$$\bar{V}_{\mathbf{a}} = \frac{1}{N-1} \bar{A}^{\mathrm{T}} \bar{A} \tag{8.1}$$

where $\bar{A}$ is the mean-centred matrix defined by

$$\bar{A}_{ij} = A_{ij} - \bar{a}_j.$$

The difference

$$\beta_j = a_j^* - \bar{a}_j$$

between the mean estimate and the true value of the jth parameter is a measure of the bias of the estimator. The standard deviations s_j defined by

$$s_j^2 = \frac{1}{N-1} \sum_{q=1}^{N} (A_{j,q} - \bar{a}_j)^2$$

of the columns of A, that is, the square roots of the diagonal elements of $\bar{V}_{\mathbf{a}}$, are a measure of the variability of the estimator. A combined measure of estimator performance is given by the *mean squared error* (MSE)

$$\mathrm{MSE}_j = \frac{1}{N-1} \sum_{q=1}^{N} (A_{j,q} - a_j^*)^2, \tag{8.2}$$

or by the *root mean squared error*

$$\mathrm{RMSE}_j = (\mathrm{MSE}_j)^{1/2}, \tag{8.3}$$

for the jth parameter. We can also evaluate the model at $\bar{\mathbf{a}}$ or at any $\mathbf{a}_q$ and compare it with the model values at $\mathbf{a}^*$. The differences in these model values can be compared with the size of the perturbations Δ_q.

In Monte Carlo simulations, the functional model is only required to generate the exact data set $\mathbf{x}^*$. This means that the behaviour of approximate estimators can be validated against data generated using only function evaluations of a comprehensive model.

As stressed in [71], the sample covariance matrix $\bar{V}_{\mathbf{a}}$ derived from the matrix A of parameter estimates $\mathbf{a}_q$ represents the covariance matrix of the *actual*

distribution of the parameter estimates. No approximations or linearisations are involved in its calculation and it can be used to validate uncertainty estimates derived by other means.

The number of simulations performed is typically in the range 1000 to 100,000. However, a smaller number of simulations will often identify errors in implementations, for example. The Monte Carlo simulations can be repeated for a range of parameter values $\mathbf{a}^*$ and different measurement strategies to give a more complete picture of estimator behaviour and valuable information about the effect of measurement strategy on the parameter estimates (chapter 10).

8.2.1 Monte Carlo simulation for a least-squares estimator

See section 4.2.

To illustrate, we consider a (nonlinear) least-squares approximation problem (section 4.2). Suppose the functional and statistical model is of the form

$$y^* = \phi(\mathbf{x}, \mathbf{a}), \quad y = y^* + \epsilon, \quad \epsilon \in N(0, \sigma^2).$$

Given a data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, least-squares estimator $\mathcal{A}$ calculates (an estimate of) the solution of

$$\min_{\mathbf{a}} \frac{1}{2} \sum_{i=1}^{m} (y_i - \phi(\mathbf{x}_i, \mathbf{a}))^2.$$

The uncertainty (covariance) matrix associated with the fitted parameters is estimated according to

$$V_{\mathbf{a}} = \sigma^2 (J^{\mathrm{T}} J)^{-1}, \tag{8.4}$$

where J is the Jacobian matrix defined by

$$J_{ij} = \frac{\partial}{\partial a_j} (y_i - \phi(\mathbf{x}_i, \mathbf{a})), \tag{8.5}$$

evaluated at the solution.

This estimator is appropriate for the model (and in fact is a maximum likelihood estimator). However, if the function ϕ is highly nonlinear the estimator could be biased.[1] Secondly, the estimates of the uncertainty matrix associated with the fitted parameters are based on a linearisation about the solution and could give misleading results. The aims of the Monte Carlo simulation are to estimate the bias in the parameter estimates and the validity of the uncertainty estimates.

Given a choice of $\mathbf{a}^*$ and values of the covariates $\mathbf{x}_i$, $i = 1, \ldots, m$, the Monte Carlo simulation in this case is summarised by:

I Calculate $y_i^* = \phi(\mathbf{x}_i, \mathbf{a}^*)$ and set $\mathbf{z}^* = \{(\mathbf{x}_i, y_i^*)\}_{i=1}^m$.

II For $q = 1, 2, \ldots, N$,

[1] Although maximum likelihood estimators are asymptotically unbiased (section 4.10), for a fixed and modest number of data points they can be biased.

II.1.q Generate $y_{i,q} = y_i^* + \epsilon_{i,q}$, where $\epsilon_{i,q}$ is determined using a random number generator for the normal distribution and set $\mathbf{z}_q = \{(\mathbf{x}_i, y_{i,q})\}_{i=1}^m$.

II.2.q Determine the least-squares estimate $\mathbf{a}_q = \mathcal{A}(\mathbf{z}_q)$ by solving

$$\min_{\mathbf{a}_q} \sum_{i=1}^{m} (y_{i,q} - \phi(\mathbf{x}_i, \mathbf{a}_q))^2,$$

and store $\mathbf{a}_q$ as the qth row of an $N \times n$ matrix A.

III Compare the statistics associated with A with the choice of $\mathbf{a}^*$.

Let $\bar{\mathbf{a}}$ and $\bar{V}_{\mathbf{a}}$ be defined as above (equation (8.1)). The bias of the estimator can be measured by the difference between the mean parameter estimates and the true values

$$\boldsymbol{\beta} = \bar{\mathbf{a}} - \mathbf{a}^*,$$

and differences $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_m)^{\mathrm{T}}$ between the model predictions at $\bar{\mathbf{a}}$ and at $\mathbf{a}^*$ where

$$\delta_i = \phi(\mathbf{x}_i, \bar{\mathbf{a}}) - \phi(\mathbf{x}_i, \mathbf{a}^*).$$

The former can be compared directly with accuracy requirements on $\mathbf{a}$ while the latter can be compared with the size of the perturbations (simulating random effects associated with the measurements) in terms of the standard deviation σ.

The sample covariance matrix $\bar{V}_{\mathbf{a}}$ can be used to validate the estimate $V_{\mathbf{a}}$ defined in (8.4).

8.3 Null space benchmarking

The Monte Carlo simulation technique is a useful tool in validating estimator performance or in comparing one estimator with another. In this section we consider a more direct method of comparing one estimator with another, particularly in the case where we wish to compare a simpler, approximate estimator with a comprehensive, more complicated one. The main advantage of the approach is that an implementation of the comprehensive estimator is not required, although some analysis of its behaviour is generally required.

Suppose there are two estimators $\mathcal{A}$ and $\mathcal{B}$ associated with a measurement model with $\mathcal{A}$ representing an optimal data analysis we would ideally like to use (e.g., a maximum likelihood estimator, section 4.10) and $\mathcal{B}$ an approximate estimator that is already implemented. We wish to know whether or not $\mathcal{B}$ is fit for purpose or if it will be necessary to implement $\mathcal{A}$.

The main steps are as follows.

I Given $\mathbf{a}$, determine a data set $\mathbf{z}$ (appropriate for the statistical model) for which $\mathbf{a} = \mathcal{A}(\mathbf{z})$, i.e., $\mathbf{a}$ is the solution supplied by estimator $\mathcal{A}$ for data set $\mathbf{z}$. Often, the uncertainty matrix $V_{\mathbf{a}}$ associated with the parameters estimates $\mathbf{a}$ can also be calculated.

II Apply estimator $\mathcal{B}$ to $\mathbf{z}$ to determine estimates $\mathbf{b} = \mathcal{B}(\mathbf{z})$.

III Compare $\mathbf{b}$ with $\mathbf{a}$.

(IV) Compare the covariance matrix $V_{\mathbf{b}}$ of $\mathbf{b}$ with $V_{\mathbf{a}}$, if they are available.

The method can also be implemented in a Monte Carlo setting in which a number of data sets $\mathbf{z}_q$ are generated for which $\mathcal{A}(\mathbf{z}_q) = \mathbf{a}$.

The key to the approach is step I, determining the data set $\mathbf{z}$. We now describe ways to solve the problem for three important applications.

8.3.1 Null space benchmarking for least-squares problems

A common simplification in model fitting is to replace a nonlinear least-squares problem with a linear least-squares problem. In this section we show how to generate null space data for the nonlinear problem.

Suppose a nonlinear least-squares estimation process requires the solution of

$$\min_{\mathbf{a}} \frac{1}{2} \sum_{i=1}^{m} (y_i - \phi(\mathbf{x}_i, \mathbf{a}))^2. \tag{8.6}$$

The (first order) optimality conditions for $\mathbf{a}^*$ to be a local solution of (8.6) is that

$$J^{\mathrm{T}}\mathbf{f} = \mathbf{0}$$

where $J = J(\mathbf{a}^*)$ is the Jacobian matrix of partial derivatives defined by (8.5) evaluated at $\mathbf{a}^*$ and $\mathbf{f}(\mathbf{a}^*) = (f_1, \ldots, f_m)^{\mathrm{T}}$ with $f_i = y_i - \phi(\mathbf{x}_i, \mathbf{a}^*)$. This equation states that the vector of residual errors $\mathbf{f}$ lies in the *null space* of the matrix J^{T}.

The QR decomposition (section 3.7, [119]) can be used to generate an orthogonal basis of the null space of any $m \times n$, $m \geq n$, matrix C. Let

$$C = Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \tag{8.7}$$

where Q is $m \times m$ orthogonal and R_1 is $n \times n$ upper triangular. Let $Q_2 = \begin{bmatrix} \mathbf{q}_{n+1} & \mathbf{q}_{n+2} & \cdots & \mathbf{q}_m \end{bmatrix}$ be the submatrix of Q consisting of the last $m - n$ columns. If $\mathbf{z}$ is such that $C^{\mathrm{T}}\mathbf{z} = \mathbf{0}$ then $\mathbf{z}$ can be written uniquely as a linear combination of the columns of Q_2. That is, there exists a unique $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{m-n})^{\mathrm{T}}$ such that

$$\mathbf{z} = Q_2\boldsymbol{\nu} = \nu_1\mathbf{q}_{n+1} + \nu_2\mathbf{q}_{n+2} + \cdots + \nu_{m-n}\mathbf{q}_m, \tag{8.8}$$

and vice versa: any such linear combination $\mathbf{z}$ will be such that $C^{\mathrm{T}}\mathbf{z} = \mathbf{0}$. Since Q_2 is orthogonal the Euclidean norm of $\mathbf{z}$ in (8.8) is that of $\boldsymbol{\nu}$, i.e., $\|\mathbf{z}\| = \|\boldsymbol{\nu}\|$.

Given a choice of $\mathbf{a}^*$, σ and values of the covariates $\mathbf{x}_i$, $i = 1, \ldots, m$, the null space method for generating reference data and results is summarised by:

I Calculate $y_i^* = \phi(\mathbf{x}_i, \mathbf{a}^*)$ and set $\mathbf{z}^* = \{(\mathbf{x}_i, y_i^*)\}_{i=1}^m$.

II Calculate the $m \times n$ Jacobian matrix J^*:

$$J^*_{ij} = \frac{\partial}{\partial a_j}(y^*_i - \phi(\mathbf{x}_i, \mathbf{a}^*)),$$

and an orthogonal basis for the null space $Q_2 = [\mathbf{q}_{n+1} \dots \mathbf{q}_m]$ of ${J^*}^{\mathrm{T}}$. Generate $\boldsymbol{\nu} = (\nu_1, \dots, \nu_{m-n})$ sampled from a normal distribution and normalise them so that

$$\|\boldsymbol{\nu}\|/(m-n)^{1/2} = \sigma.$$

III Set

$$\left.\begin{array}{rcl} \boldsymbol{\delta} & = & (\delta_1, \dots, \delta_m)^{\mathrm{T}} = \sum_{k=1}^{m-n} \nu_k \mathbf{q}_{n+k}, \\ y_i & = & y^*_i + \delta_i, \\ \mathbf{z}_{\boldsymbol{\delta}} & = & \{(\mathbf{x}_i, y_i)\}_{i=1}^m. \end{array}\right\} \tag{8.9}$$

Then if σ is small enough, $\mathbf{a}^*$ are the least-squares best-fit model parameters to $\mathbf{z}_{\boldsymbol{\delta}}$ and $\boldsymbol{\delta}$ is the vector of residuals

$$\delta_i = y_i - \phi(\mathbf{x}_i, \mathbf{a}^*)$$

and satisfies $\|\boldsymbol{\delta}\|/(m-n)^{1/2} = \sigma$. Furthermore,

$$V^*_{\mathbf{a}} = \sigma^2({J^*}^{\mathrm{T}} J^*)^{-1},$$

is an estimate of the covariance matrix for the least-squares estimator.

8.3.2 Null space benchmarking for generalised distance regression

The ordinary linear and nonlinear least-squares estimators are appropriate if only one measured variable is subject to significant random effects. However, in many metrological situations, more than one of the measured variables is subject to random effects and it is important to take this into account in determining parameter estimates that are free from significant bias. A typical situation for which this is appropriate is where the response $y = \phi(x, \mathbf{a})$ is modelled as a function of the variable x and parameters $\mathbf{a}$ and both y and x are measured subject to random effects, giving rise to a model of the form

$$x_i = x^*_i + \delta_i, \quad y_i = \phi(x_i, \mathbf{a}) + \epsilon_i, \qquad \delta_i \in N(0, \sigma_i^2), \quad \epsilon_i \in N(0, \rho_i^2).$$

The maximum likelihood estimate of the parameters is found by solving

$$\min_{\mathbf{a}, \{x^*_i\}} \sum_{i=1}^{m} \left\{ \alpha_i^2 (x_i - x^*_i)^2 + \beta_i^2 (y_i - \phi(x^*_i, \mathbf{a}))^2 \right\}, \tag{8.10}$$

with $\alpha_i = 1/\sigma_i$, $\beta_i = 1/\rho_i$. Note that this problem involves $\mathbf{a}$ and the *footpoint parameters* $\{x^*_i\}$ that specify the points on the curve closest to the data points in the appropriate metric. In section 4.3.2, these types of problems are posed as generalised distance regression (GDR) problems and effective methods for

their solution described. However, these problems are more complex than their standard counterparts. In particular, even if ϕ is linear in the parameters $\mathbf{a}$ the optimisation problem (8.10) requires nonlinear iterative techniques for its solution. With this in view, given a problem of this type it may be worth investigating whether or not a generalised distance regression approach is required. Here we show how the null space method can be used to benchmark ordinary least-squares estimators against a GDR estimator.

The GDR problem (8.10) is a nonlinear least-squares problem for which the null space data generation approach described above can be applied. Let

$$\begin{aligned} \phi_i^* &= \phi(x_i^*, \mathbf{a}^*), \\ \dot{\phi}_i &= \frac{\partial \phi_i}{\partial x}(x_i^*, \mathbf{a}^*), \\ s_i &= \left(\frac{\dot{\phi}_i^2}{\alpha_i^2} + \frac{1}{\beta_i^2} \right)^{1/2}, \end{aligned}$$

$$d_i = \frac{1}{s_i}(-(x_i - x_i^*)\dot{\phi}_i + (y_i - \phi_i^*)), \tag{8.11}$$

and J^* the Jacobian matrix defined by

$$J_{ij}^* = -\frac{1}{s_i}\frac{\partial \phi_i}{\partial a_j}(x_i^*, \mathbf{a}^*).$$

If $\mathbf{a}^*$ and $\{x_i^*\}$ solve (8.10) for data $X = \{(x_i, y_i)\}_{i=1}^m$ then necessarily (i) there exists $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_m)^{\mathrm{T}}$ such that

$$x_i = x_i^* - \delta_i \frac{\dot{\phi}_i}{\alpha_i^2}, \quad y_i = \phi_i^* + \delta_i \frac{1}{\beta_i^2}, \tag{8.12}$$

and (ii)

$$J^{*\mathrm{T}}\mathbf{d} = \mathbf{0},$$

i.e.,

$$\sum_{i=1}^{m} \frac{1}{s_i}\frac{\partial \phi_i}{\partial a_j} d_i = 0, \quad j = 1, \ldots, n. \tag{8.13}$$

Substituting (8.12) in (8.11) we obtain

$$d_i = \delta_i s_i$$

so that the conditions (8.13) on $\mathbf{d}$ become

$$A^{\mathrm{T}}\boldsymbol{\delta} = \mathbf{0},$$

where $A_{ij} = \frac{\partial \phi_i}{\partial a_j}$. The perturbations δ_i can be scaled so that the estimate of the standard deviation of the residuals d_i is a pre-assigned value σ.

Taking into account this analysis, null space data can be generated according to the following scheme:

Given $\mathbf{a}^*$, σ, $\{x_i^*\}$, $\{\alpha_i\}$ and $\{\beta_i\}$,

I Set $y_i^* = \phi(x_i^*, \mathbf{a}^*)$ and $\mathbf{z}^* = \{(x_i^*, y_i^*)\}$.

II For each i, calculate

$$\dot{\phi}_i = \frac{\partial \phi}{\partial x}(x_i^*, \mathbf{a}^*), \quad s_i = \left(\frac{\dot{\phi}_i^2}{\alpha_i^2} + \frac{1}{\beta_i^2} \right)^{1/2}.$$

III Calculate A given by

$$A_{ij} = \frac{\partial \phi_i}{\partial a_j}.$$

IV Determine $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_m)^{\mathrm{T}}$ such that $A^{\mathrm{T}}\boldsymbol{\delta} = 0$ (using the QR factorisation of A, for example). For each i, set

$$p_i = -\delta_i \frac{\dot{\phi}_i}{\alpha_i^2}, \quad q_i = \delta_i \frac{1}{\beta_i^2},$$

and calculate $S = \left\{ \sum_{i=1}^m (\alpha_i^2 p_i^2 + \beta_i^2 q_i^2) \right\}^{1/2}$ and $K = (m-n)^{1/2}\sigma/S$.

V For each i, set

$$x_i = x_i^* + K p_i, \quad y_i = y_i^* + K q_i.$$

Then $\mathbf{a}^*$ and $\{x_i^*\}$ solves (8.10) for dataset $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m$ with

$$\frac{1}{m-n} \sum_{i=1}^m \left\{ \alpha_i^2 (x_i - x_i^*)^2 + \beta_i^2 (y_i - \phi(x_i^*, \mathbf{a}))^2 \right\} = \sigma^2.$$

Furthermore, the estimate $V_{\mathbf{a}}$ of the uncertainty matrix associated with the fitted parameters $\mathbf{a}^*$ for *the GDR estimator* is

$$V_{\mathbf{a}} = \sigma^2 (J^{\mathrm{T}} J)^{-1}, \tag{8.14}$$

where the Jacobian matrix J can be calculated from $J_{ij} = A_{ij}/s_i$. This null space approach allows us to generate data sets for which the GDR solution estimates are known along with their uncertainties without having to implement the estimator. All that is required are the evaluation ϕ, $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial a_j}$. We can use this information to benchmark the performance of an approximate estimator both in terms of bias and variability against the GDR estimator.

8.3.3 Null space benchmarking for generalised Gauss-Markov problems

We wish to generate test data for the problem of generalised Gauss-Markov regression with curves (section 4.4). We assume that the uncertainty matrix V is full rank. A first order condition for $\mathbf{y}^*$ and $\mathbf{x}^*$ to define a local minimum is that

$$J^{\mathrm{T}} V^{-1} \begin{bmatrix} \mathbf{x} - \mathbf{x}^* \\ \mathbf{z} - \mathbf{y}^* \end{bmatrix} = \mathbf{0},$$

where, as before, $y_i^* = \phi(y_i^*, \mathbf{a})$ and J is the Jacobian matrix defined in (4.27). Therefore, if

$$J^{\mathrm{T}}\boldsymbol{\delta} = \mathbf{0},$$

and

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^* \\ \mathbf{y}^* \end{bmatrix} + V\boldsymbol{\delta},$$

then these optimality conditions are satisfied. Below we describe a data generation scheme based on the analysis above that uses the generalised QR factorisation to improve the efficiency and numerical stability.

Suppose $2m \times 2m$ uncertainty matrix $V = LL^{\mathrm{T}}$ has been specified. The simulation data is generated according to the following scheme.

I Fix parameters $\mathbf{a}^\sharp = (a_1^\sharp, \ldots, a_n^\sharp)^{\mathrm{T}}$ and abscissae $\mathbf{x}^\sharp = (x_1^\sharp, \ldots, x_m^\sharp)^{\mathrm{T}}$.

II Generate $\mathbf{y}^\sharp$ so that $y_i^\sharp = \phi(x_i^\sharp, \mathbf{a}^\sharp)$, $i = 1, \ldots, m$.

III Evaluate $2m \times (m+n)$ Jacobian matrix J for parameters $\mathbf{x}^\sharp$ and $\mathbf{a}^\sharp$ and form the generalised QR factorisation for the pair $[J,\ L]$:

$$J = Q\begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix}, \quad Q^{\mathrm{T}}L = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} U.$$

IV Evaluate the $(m+n) \times (m+n)$ matrix

$$V^\sharp = KK^{\mathrm{T}}, \quad R_1 K = T_{11}.$$

V Generate at random a $m-n$ vector $\boldsymbol{\zeta}$ and normalise it so that $\boldsymbol{\zeta}^{\mathrm{T}}\boldsymbol{\zeta} = m-n$. Set

$$\boldsymbol{\epsilon} = Q\begin{bmatrix} T_{12} \\ T_{22} \end{bmatrix}\boldsymbol{\zeta}, \quad \boldsymbol{\eta} = U^{\mathrm{T}}\begin{bmatrix} \mathbf{0} \\ \boldsymbol{\zeta} \end{bmatrix}.$$

VI Set

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^\sharp \\ \mathbf{y}^\sharp \end{bmatrix} + \boldsymbol{\epsilon}.$$

Then $\mathbf{a}^\sharp$ is the generalised Gauss-Markov estimate associated with the data $\mathbf{x}$ and $\mathbf{y}$ and $V^\sharp$ is the uncertainty matrix associated with the fitted parameters $\mathbf{x}^\sharp$ and $\mathbf{a}^\sharp$. The lower right $n \times n$ submatrix of $V^\sharp$ is the uncertainty matrix $V_{\mathbf{a}}^\sharp$ associated with the parameters $\mathbf{a}$.

Steps V and VI can be repeated any number of times if required.

8.3.4 Validation criteria

One aspect relating to estimator validation is the criteria by which we can accept or reject a particular method of analysis (see also [10]). Suppose data $\mathbf{z}$ is collected and two methods of analysis $\mathcal{A}_1$ and $\mathcal{A}_2$ produce estimates of the model parameters $\mathbf{a}_1 = \mathcal{A}_1(\mathbf{z})$ and $\mathbf{a}_2 = \mathcal{A}_2(\mathbf{z})$. There are two general

approaches to determining how close the two estimates are to each other. The first is to look at some norm of $\mathbf{a}_1 - \mathbf{a}_2$ and compare this with the uncertainty associated with $\mathbf{a}$, either as estimated from the analysis or as specified by the requirements. The second looks at minimal perturbations of the data Δ_1, Δ_2 such that $\mathbf{a}_1 = \mathcal{A}_2(\mathbf{z} + \Delta_1)$, $\mathbf{a}_2 = \mathcal{A}_1(\mathbf{z} + \Delta_2)$. The size of these perturbations can then be compared with the measurement uncertainty.

Example: linear least squares

Suppose $\mathbf{a}$ is the solution of the linear least problem

$$\min_{\mathbf{a}} \|\mathbf{y} - C\mathbf{a}\|, \tag{8.15}$$

and $u(a_j)$ the corresponding standard uncertainties associated with the solution parameters. Let $\mathbf{b}$ be the solution produced by an other estimator. The closeness of $\mathbf{b}$ to $\mathbf{a}$ can be measured by the norm of their difference:

$$\|\mathbf{b} - \mathbf{a}\|_2, \quad \text{or} \quad \|\mathbf{b} - \mathbf{a}\|_\infty = \max_j |b_j - a_j|,$$

for example. The measures can also take into account the standard uncertainty in the estimates by considering quantities such as $|a_j - b_j|/u(a_j)$ or

$$(\mathbf{b} - \mathbf{a})^{\mathrm{T}} V_{\mathbf{a}}^{-1} (\mathbf{b} - \mathbf{a}).$$

A measure based on a perturbation of the data can be derived as follows. The solution $\mathbf{a}$ of (8.15) must satisfy

$$C^{\mathrm{T}}(\mathbf{y} - C\mathbf{a}) = \mathbf{0}.$$

(This equation is just a restatement of the normal equations (4.1).) We ask what perturbation Δ of $\mathbf{y}$ is required to make $\mathbf{b}$ a solution of the linear least-squares problem for data $\mathbf{y} + \Delta$ and observation matrix C. Let

$$\mathbf{f}_{\mathbf{a}} = \mathbf{y} - C\mathbf{a}, \quad \mathbf{f}_{\mathbf{b}} = \mathbf{y} - C\mathbf{b},$$

be the vectors of residuals associated with each solution and $\Delta = \mathbf{f}_{\mathbf{a}} - \mathbf{f}_{\mathbf{b}} = C\mathbf{b} - C\mathbf{a}$ be the difference in these residuals. Then

$$C^{\mathrm{T}}(\mathbf{y} + \Delta - C\mathbf{b}) = C^{\mathrm{T}}(\mathbf{y} + C\mathbf{b} - C\mathbf{a} - C\mathbf{b}) = C^{\mathrm{T}}(\mathbf{y} - C\mathbf{a}) = 0.$$

It can be shown that Δ represents a minimal perturbation of $\mathbf{y}$. The size of the perturbations Δ can be related directly to the uncertainties associated with $\mathbf{y}$. If these perturbations are relatively small, then the two approaches can be regarded as having comparable performance (for the type of data considered).

8.4 Estimator analysis

In sections 8.2 and 8.3 we have described quantitative methods for validating estimators. The main aim has been to validate the performance of an estimator given a statement of the functional and statistical model. In this section, we look at a more qualitative approach in which we start from the definition of an estimator and then consider for which class of model it is (optimally) appropriate. We can then assess if it is valid for a particular estimation problem on the basis of how well matched the model is to the specified class.

8.4.1 Analysis of least-squares methods

Least-squares methods are the most common estimators implemented and are appropriate for many practical model fitting problems. For linear models the *Gauss-Markov Theorem* (section 4.1.9) can be used to justify their use. We recall that the theorem states that for models of the form

$$\mathbf{y} = C\mathbf{a} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \in \boldsymbol{E}, \quad E(\boldsymbol{E}) = \mathbf{0}, \quad V(\boldsymbol{E}) = \sigma^2 I,$$

where C is an $m \times n$ full rank matrix, $m \geq n$, the linear least-squares estimator

$$\mathbf{a} = (C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}\mathbf{y}$$

is unbiased and has a smaller covariance matrix than that of any other linear estimator. From this point of view, least-squares estimation is optimal for these models. Note that there is no assumption that the random effects are normally or even symmetrically distributed, only that they are uncorrelated and have equal variance. This generality supports the use of least squares methods. Assumptions about normality are usually only invoked when it is required to provide coverage limits associated with the fitted parameters. The theorem does not apply to nonlinear models. However, if the model is only mildly nonlinear then it can be expected that nonlinear estimation techniques will also have favourable properties. The theorem does not apply to models of the form $y = \phi(\mathbf{x}, \mathbf{a})$ if one or more of the variables x_k are subject to significant random effects.

Whenever a least-squares estimation method is proposed, it is a good idea to examine the extent to which the prior conditions of the theorem concerning linearity, correlation and variance hold. As described in section 4.1.6, least-squares methods can be adapted to deal with unequal variances, known correlation, etc., and it is important to make sure the least-squares method is tuned as far as possible to the model. Often a least-squares method is implemented without sufficient consideration.

Validation responses

- Monte Carlo simulations to examine the bias and variation of the solution estimates on data sets generated according to the statistical model.
- Monte Carlo simulations to compare the predicted variation of parameter estimates with the actual variation on data sets generated according to the statistical model.
- Apply the estimator to data sets for which the estimates provided by an optimal estimator are known.
- Compare the actual variation of parameter estimates on data sets generated according to the statistical model with the predicted variation for an optimal estimator.
- Compare the actual statistical model with the statistical model for which the estimator is known to perform well.

Chapter 9

Validation of model solutions

The validation of estimators considered in the previous chapter was concerned with the process of how a model solution is determined. In this chapter we are concerned with the question of validating specific model solutions to data, regardless of how these solutions were arrived at. There are two aspects to this question, firstly, how well does the model represent the data and secondly how well does the data specify the model. This second aspect is considered further in chapter 10.

9.1 Goodness of fit and residuals

In solving a system of equations $C\mathbf{a} = \mathbf{y}$ where C is a square $n \times n$ matrix, a solution $\mathbf{a}$ can be validated by calculating how well it satisfies the equations. If the matrix C is full rank then (in exact arithmetic) the residual vector $\mathbf{r} = \mathbf{y} - C\mathbf{a}$ should be identically zero. For over-determined systems in which there are more equations than parameters, not all the equations can be satisfied exactly in general and the residual vector will be nonzero. A measure of the goodness of fit is derived from a norm of $\mathbf{r}$, e.g., the Euclidean norm

$$\|\mathbf{r}\|_2 = \left(\sum_i r_i^2 \right)^{1/2},$$

or the Chebyshev norm

$$\|\mathbf{r}\|_\infty = \max_i |r_i|.$$

The central question in determining whether a specific model solution is a valid fit to data relates to what is the expected behaviour of the residual vector: the more sophisticated the model, the more sophisticated the validation criteria.

9.1.1 Signal to noise ratio

A minimal model associated with a set of data specifies only the likely signal to noise ratio, for example, in a statement of the form "the data is likely to be accurate to 1 part in 10^4". Suppose $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ represents the measurements of a response variable y_i corresponding to values $\mathbf{x}_i$ for the covariates. A model of the form

$$y = \phi(\mathbf{x}, \mathbf{a})$$

is fitted to the data and the residuals $r_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$ calculated. The ratio

$$R^2 = \frac{\left(\sum_i y_i^2\right) - \left(\sum_i r_i^2\right)}{\sum_i y_i^2}$$

is a measure of how much of the variation of the response y is accounted for by the model ϕ. A value of 1 indicates that all of the variation is explained by the model; value of 0 indicates that none has been accounted for. Related measures are discussed in [40].

9.1.2 Statistical model for the data

A model fit can be validated against a statistical model for the data. For example, suppose the model for data $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ is

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad \boldsymbol{\epsilon} \in D(\mathbf{b}),$$

where $D(\mathbf{b})$ is a multivariate distribution associated with the random effects. Given estimates of the fitted parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$, we can calculate the residuals $r_i = y_i - \phi(\mathbf{x}_i, \mathbf{a})$. If the model is correct the vector $\mathbf{r}$ can be regarded as representing an observation drawn from $D(\mathbf{b})$. If it is plausible that the observed $\mathbf{r}$ is a sample from $D(\mathbf{b})$ then we have no reason to doubt the model. If, on the other hand, it is very unlikely that $\mathbf{r}$ represents a sample $D(\mathbf{b})$ then the model solution is called into question.

An important case is where $\boldsymbol{\epsilon} \in N(0, \sigma^2 I)$ or is at least approximately normally distributed. For this case, if

$$\hat{\sigma}^2 = \frac{1}{m-n} \sum_{i=1}^{m} r_i,$$

the model predicts that

$$\frac{\hat{\sigma}^2}{\sigma^2} \in \chi^2_{m-n},$$

that is, the sum of squares of the residuals is related to the χ^2 distribution with $m-n$ degrees of freedom. We refer to $\hat{\sigma}$ as the *root mean square (RMS) residual*. We recall that the χ^2_p distribution with p degrees of freedom has an expected value of p and a variance of $2p$. Distributions for 2, 5 and 10 degrees of freedom are plotted in figure 9.1. For small values of p, the distribution is significantly asymmetric. For large values, the distribution is approximated by $N(p, 2p)$, i.e., the normal distribution with mean p and variance $2p$. For large numbers of

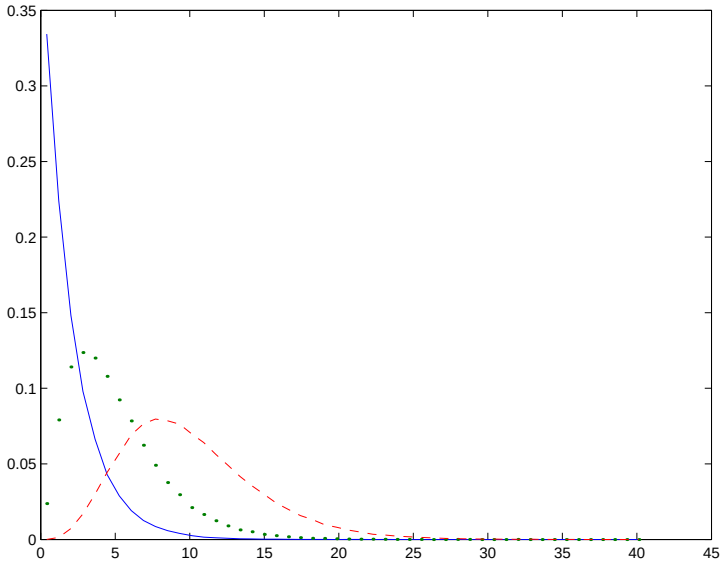

Figure 9.1: χ^2 distributions with 2 (solid), 5 (dotted) and 10 (dashed) degrees of freedom.

degrees of freedom $p = m-n$, say $p \geq 50$, the difference $(\hat{\sigma}^2 - \sigma^2)/\sigma^2$ is expected to be smaller in magnitude than $2(2/p)^{1/2}$ at approximately the 95% confidence level. For small numbers of degrees of freedom it is better to compare this ratio with the 2.5% and 97.5% points of the appropriate χ^2-distribution. Adjustment procedures for modifying the input uncertainties in light of the observed RMS residual are discussed in [66].

The randomness of the residuals can be tested most simply by visual inspection. For small numbers of residuals, it is usually quite difficult to show that there is a nonrandom behaviour. For larger data sets, systematic behaviour is often quite obvious.

9.1.3 Residuals and model selection

The RMS residual can be used to select a model from a range of model solutions, for example, in choosing the degree of a polynomial fit. To illustrate this, data (x_i, y_i) has been generated according to the model

$$y^* = x + 0.01x^2 + 0.01x^3 + 0.02x^4, \quad y = y^* + \epsilon, \quad \epsilon \in N(0, \sigma^2), \tag{9.1}$$

for x in the interval $[0,\ 1]$ and three values of $\sigma = 0.01,\ 0.001$ and 0.0001. For each value of σ, polynomials of degree 0 to 9 were fitted to the data and $\hat{\sigma}_n$ recorded for each degree n.

Figure 9.2 is the log plot of $\hat{\sigma}_n$ against degree n for each of the three values of σ.

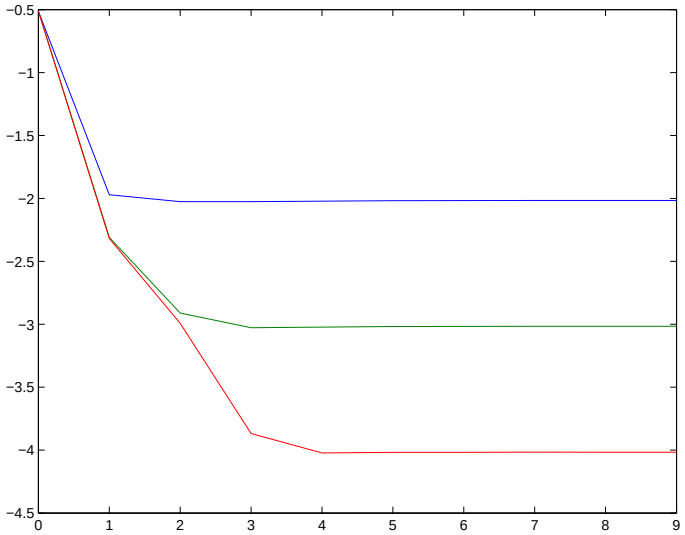

Figure 9.2: Graph of $\log \hat{\sigma}_n$ of the RMS residual against order of polynomial fit for data generated according to the model (9.1) with $\sigma = 0.01, 0.001$ and 0.0001.

For $\sigma = 0.01$ (upper curve), the RMS plot shows that increasing the degree from 2 to 3 reduces the RMS by a small amount but increasing the degree further gives no improvement. Similarly for $\sigma = 0.001$, there is an improvement from degree 3 to 4 but none beyond. On this basis we would chose degrees 2, 3 and 4 for $\sigma = 0.01, 0.001$ and 0.0001, respectively. In each case the saturation level is $\log_{10} \sigma$. This agrees with the plots of the residuals associated with the degree 2 (upper graph) and 3 (lower graph) fits shown in figure 9.3. The residuals associated with the degree 2 fit show systematic cubic behaviour while those for degree 3 appear random. A similar picture is shown in figure 9.4 for the case $\sigma = 0.0001$ in which the residuals associated with the degree 3 fit show a quartic (W-shaped) systematic behaviour while the degree 4 residuals appear random.

9.1.4 Validation of model outputs

So far in this chapter we have only considered the fit of a model solution to data and its validation in terms of the goodness of fit against the statistical model for the data; we have as yet paid no attention to where the model came from or to any other characteristics of the model. However, an estimation process will generally produce (and is usually required to produce) estimates of the uncertainty associated with the fitted parameters and related statistical information. If this extra information has been derived in a valid way, i.e., the model and the solution process are consistent with each other, then it provides additional model outputs that can be checked against actual behaviour.

We will examine these issues in more detail for the important case of the least-

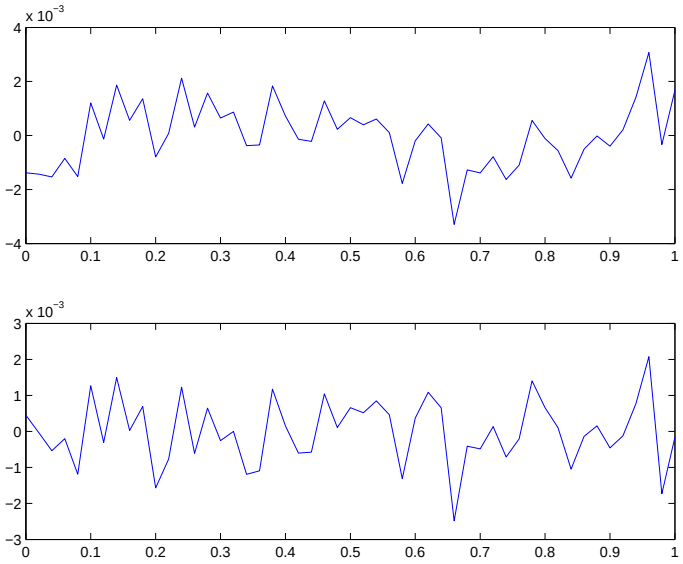

Figure 9.3: Graphs of residuals for degree 2 (upper graph) and degree 3 (lower graph) polynomial fits to data generated with $\sigma = 0.001$.

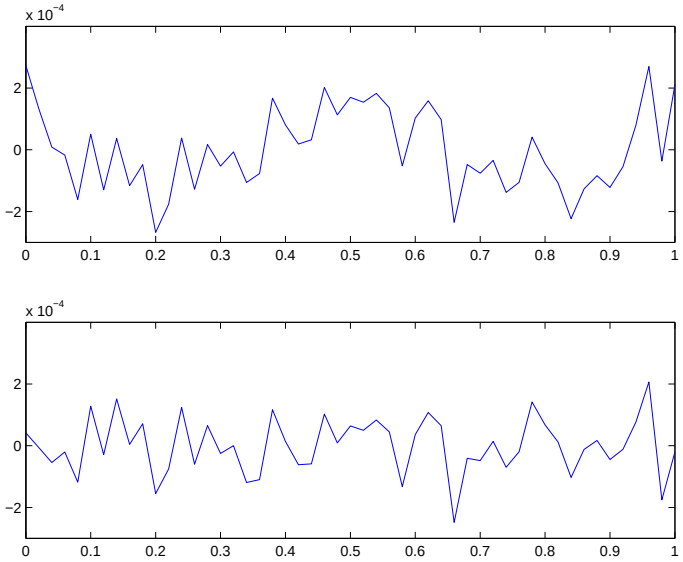

Figure 9.4: Graphs of residuals for degree 3 (upper graph) and degree 4 (lower graph) polynomial fits to data generated with $\sigma = 0.0001$.

squares fit of linear models of the form

$$y = \phi(\mathbf{x}, \mathbf{a}) = \sum_{j=1}^{n} a_j \phi_j(\mathbf{x}_i),$$

and a measurement model of the form

$$y_i = \phi(\mathbf{x}_i, \mathbf{a}) + \epsilon_i, \quad \boldsymbol{\epsilon} \in N(\mathbf{0}, \sigma^2 I).$$

Given data $\{(\mathbf{x}_i, y_i)\}_1^m$, $m > n$, let C be the observation matrix of basis functions ϕ_j evaluated at $\mathbf{x}_i$, i.e.,

$$C_{ij} = \phi_j(\mathbf{x}_i).$$

It is assumed that C has full column rank. The least-squares model solution is determined by solving the over-determined system of equations $C\mathbf{a} = \mathbf{y}$ in the least-squares sense (section 4.1.4).

If the functional and statistical model is valid then all the model predictions are valid. Conversely, if any one of these outputs is inconsistent with experimental data then there must be some aspect of the model that is invalid, putting into question the validity of all other model predictions. Each of the model predictions, described in section 4.1.4, can be used to examine aspects of the model validity.

The residuals and the RMS residual $\hat{\sigma}$. Their role in model validation has already been discussed above. If there is only a small number of degrees of freedom associated with the system of equations then the expected variation associated with $\hat{\sigma}$ is relatively large and any statistics depending on its value are also subject to relatively large variation. This is because there is only limited information with which to validate the model. If only a small amount of data is available then more emphasis should be placed in validating the model inputs (assumptions) and its internal consistency.

If an experiment is repeated a number of times providing systems of equations $C_k\mathbf{a}_k = \mathbf{y}_k$, where C_k is $m_k \times n$, $m_k > n$, residual vectors $\mathbf{r}_k = \mathbf{y}_k - C_k\mathbf{a}_k$ and estimates $\hat{\sigma}_k$, $k = 1, \ldots, K$, then a combined estimate of the RMS residual $\hat{\sigma}$ is given by

$$\hat{\sigma}^2 = \frac{1}{m - Kn} \sum_k \|\mathbf{r}_k\|^2 = \frac{1}{m - Kn} \sum_k (m_k - n)\hat{\sigma}_k^2.$$

where $m = \sum_k m_k$ is the total number of equations. This value can be subsequently used in the uncertainty evaluation associated with the individual model fits.

The value of $\hat{\sigma}$ is also a useful measure of the quality of the data and a value much larger than expected can indicate invalid input data. For this purpose, the maximum absolute residual $r_{max} = \max_i |r_i|$ should also be computed.

Uncertainty matrix and standard uncertainties associated with the fitted parameters $\mathbf{a}$. The statistical information associated with the fitted parameters is a key component of the model predictions. We know that, due to the stochastic nature of experimentation, if we repeat an experiment we will obtain different values for the fitted parameters. The uncertainty (covariance) matrix indicates

the likely spread of parameter estimates. So, for example, if an experiment is performed twice producing estimates $\mathbf{a}_k = (a_{k,1}, \ldots, a_{k,n})^{\mathrm{T}}$, covariance matrices $V_{\mathbf{a}_k}$ and standard uncertainties $u_k(a_j)$, $k = 1, 2$, then, assuming a valid model, the covariance matrix of the difference $\mathbf{a}_1 - \mathbf{a}_2$ is $V_{\mathbf{a}_1} + V_{\mathbf{a}_2}$ and the standard uncertainties of individual parameter differences are

$$u(a_{1,j} - a_{2,j}) = \left(u_1^2(a_j) + u_2^2(a_j)\right)^{1/2}.$$

The actual differences can be compared with these standard uncertainties (taking into account the degrees of freedom). For a large number of degrees of freedom, a difference in parameter estimates more than, say, three times the standard uncertainty will indicate some aspect of the model could be invalid.

The uncertainty matrix also provides direct information about the effectiveness of the experimental design (see chapter 10). At a very minimum, the computation of and check on the standard uncertainties associated with the fitted parameters is required to verify (or otherwise) that the information in the data is sufficient to determine reliable estimates of the model parameters.

Covariance matrix and standard uncertainties associated with the model predictions $\hat{\mathbf{y}}$. The standard uncertainties $u(\hat{y}_i)$ associated with the model values $\hat{y}_i = \phi(\mathbf{x}_i, \mathbf{a})$ indicate how well the model is defined at $\mathbf{x}_i$. Often, the uncertainty associated with the model predictions is the key output of the modelling process. We wish to understand a physical system. We build a model and gather a set of measurements $\{(\mathbf{x}_i, y_i)\}$ to characterise the model. From this information, we hope to predict the behaviour of the system at any set of values of the independent variables $\mathbf{x}$. The standard uncertainties associated with the model predictions indicate how well the model has been characterised and provide a measure of the quality of the predictions.

For example, suppose we wish to characterise the response of a system $y = \phi(x, \mathbf{a})$ for x in the range $[0,\ 1]$. Measurements are made at values $(0.1, 0.2, \ldots, 0.9)^{\mathrm{T}}$ of the covariate $\mathbf{x}$. Measurements $\mathbf{y} = (y_1, \ldots, y_9)^{\mathrm{T}}$ are gathered subject to measurement uncertainty with variance σ^2. The standard uncertainty $u(w_q)$ of the model predictions $w_q = \phi(u_i, \mathbf{a})$ are then calculated at values $\mathbf{u} = (0.00, 0.01, 0.02, \ldots, 1.00)^{\mathrm{T}}$. If these standard uncertainties are sufficiently small relative to σ, then the system is adequately characterised.

Figure 9.5 (on page 146) graphs the residuals $r_i = y_i - \phi(x_i, \mathbf{a})$ associated with the fit of a quartic polynomial to data generated according to model (9.1) along with values $\pm 2u(w_q)$ and the band $\pm 2\sigma$. The solid curves indicate that in the range $0.1 \leq z \leq 0.9$ the uncertainty in the model fit $u(w)$ is less than the uncertainty in the measurements but outside this range the uncertainty associated with the model fit increases (quartically). The figure also graphs the model uncertainty band for a quadratic fit to the data and shows that these uncertainties are significantly larger than the measurement uncertainty indicating that a quadratic model is an inadequate representation of the system.

The uncertainty in the model predictions can be used to differentiate between two competing models. Figure 9.6 shows data generated according to the model

$$y_i = Ae^{-Bx_i}\sin(Cx_i + D) + \epsilon_i, \quad \boldsymbol{\epsilon} \in N(\mathbf{0}, \sigma^2 I),$$

with $\sigma = 0.001$. The data is fitted with two models, the physical, harmonic oscillator model used to generate the data and an empirical, polynomial model.

Figure 9.7 graphs the log of the RMS residual $\hat{\sigma}_n$ against order n (= degree + 1) of the polynomial. On the basis of this a polynomial of degree 24 was chosen to represent the data.

Figure 9.8 shows the residuals, the model prediction uncertainty band, i.e., $\pm 2u(\hat{y})$ and the $\pm 2\sigma$ band for the physical model fit. The figure shows that the model is a good representation of the data and is well-characterised over the range, including the interval in the middle where there is no data.

Figure 9.9 presents the same information for the polynomial fit. While providing a good representation of the data, the polynomial model is poorly characterised in the middle interval where the data is missing.

Figure 9.10 illustrates why this is so. It shows two degree 24 polynomials which both fit the data well (visually) but differ significantly in the middle interval.

Validation responses.

- Goodness of fit in terms of the size of the residuals.
- Comparison of the residuals with the statistical model for the random effects.
- Plot of the residuals to check for random/systematic behaviour.
- Plot of the RMS residual for a number of model fits to select an appropriate model (e.g., the polynomial of appropriate degree).
- Calculate and check the uncertainty matrix associated with the fitted parameters against requirements and/or expected behaviour.
- Calculate and check the uncertainty associated with the model predictions against requirements and/or expected behaviour.

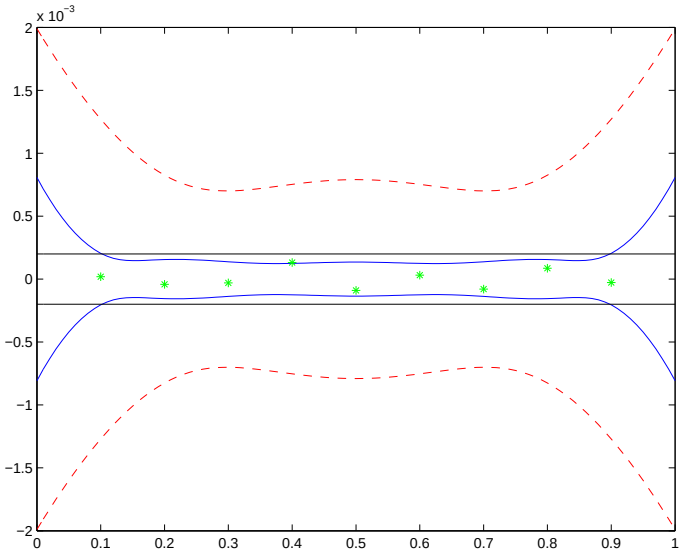

Figure 9.5: Uncertainty band for the model predictions associated with a polynomial of degree 4 (solid curves) and degree 2 (dashed curves) to data generated according to the model (9.1) compared with the uncertainty in the measured data. The residuals '*' associated with the degree 4 fit are also plotted.

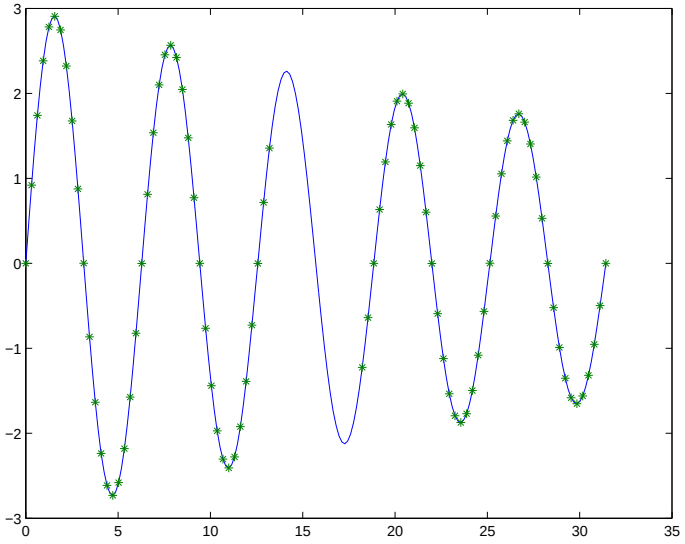

Figure 9.6: Data generated according to a harmonic oscillation model.

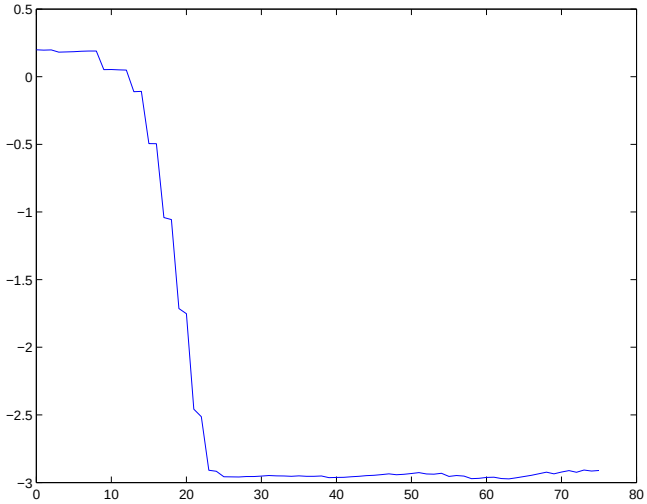

Figure 9.7: Graph of the log of the RMS residual $\hat{\sigma}_n$ associated with a polynomial fit against order $n(=$ degree $+ 1)$ to data generated according to a harmonic oscillator model (figure 9.6).

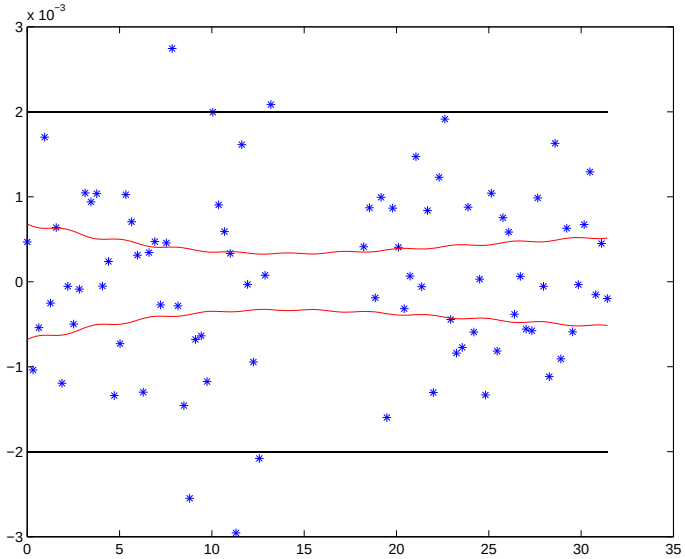

Figure 9.8: Residuals and uncertainty band associated with the fit of a physical model to data generated according to harmonic oscillator model (figure 9.6).

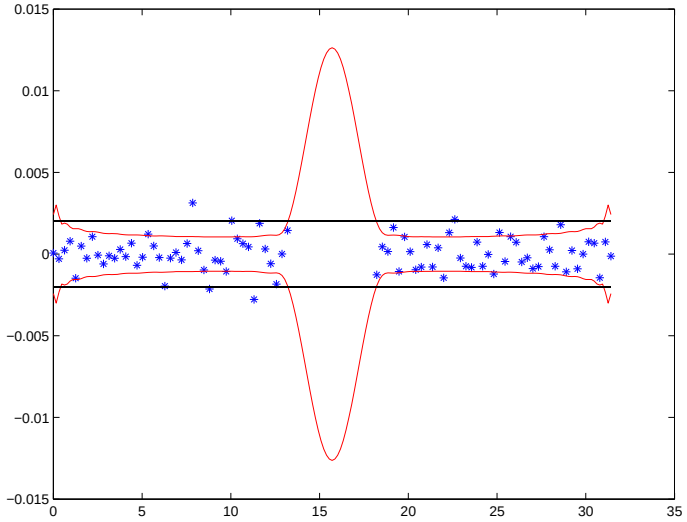

Figure 9.9: Residuals and uncertainty band associated with the fit of a polynomial model of degree 24 to data generated according to harmonic oscillator model (figure 9.6).

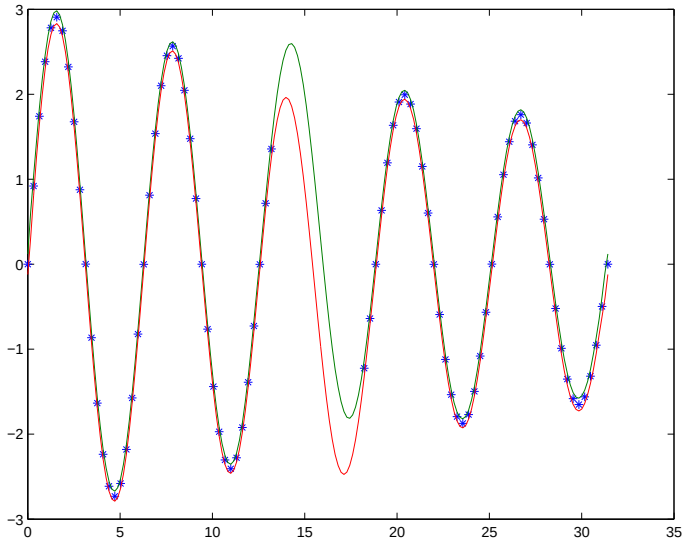

Figure 9.10: Two degree 24 polynomial fits to data generated according to harmonic oscillator model (figure 9.6).

Chapter 10

Validation of experimental design and measurement strategy

Once the functional and statistical models and estimation method have been validated, it is possible to apply the estimator to examine issues concerning experimental design and measurement strategy. In the sections below we consider (i) system identifiability – the extent to which all the parameters can be identified from the data and (ii) system effectiveness – the extent to which parameters of the system can be determined to within uncertainty requirements.

10.1 System identifiability

A system is identifiable using a given measurement strategy if all the system parameters $\mathbf{a}$ can be determined from the resulting measurement data. Usually, the identifiability of a system corresponds to a matrix associated with the model solution being full rank.

The rank of a matrix can be examined from its singular value decomposition (SVD), section 3.7.1. Suppose

$$C = USV^{\mathrm{T}}, \tag{10.1}$$

where U is an $m \times n$ orthogonal matrix, S an $n \times n$ diagonal matrix with diagonal entries $s_1 \geq s_2 \geq \ldots \geq s_n$, and V an $n \times n$ orthogonal matrix.

If C is of column rank $p < n$ then its singular values $s_{p+1} = \ldots = s_n = 0$ and for each $\mathbf{v}_k$, $k = p+1, \ldots, n$, $C\mathbf{v}_k = \mathbf{0}$. Conversely, if $C\mathbf{b} = \mathbf{0}$, $\mathbf{b}$ can be written uniquely as a linear combination

$$\mathbf{b} = \alpha_{p+1}\mathbf{v}_{p+1} + \ldots + \alpha_n\mathbf{v}_n.$$

Thus, $\mathbf{v}_k$, $k = p+1, \ldots, n$, provide an orthogonal set of basis vectors for the linear space of vectors $\{\mathbf{b} : C\mathbf{b} = \mathbf{0}\}$. These vectors can be used to analyse in what way a system fails to be identifiable.

Suppose C is the observation matrix in a linear least-squares system of equations and $\mathbf{y}_0 = C\mathbf{a}_0$. Since $C\mathbf{v}_k = \mathbf{0}$, $k = p+1, \ldots, n$ then $C(\mathbf{a}_0 + \mathbf{v}_k) = \mathbf{y}_0$ also. In other words, the vectors $\mathbf{v}_k$ characterise the set of linear combinations of the parameters that are not determined from the data.

For more general least-squares estimation problems of the form

$$\min_{\mathbf{a}} \sum_i f^2(\mathbf{x}_i, \mathbf{a}),$$

system identifiability can be examined using the following approach. Given software to calculate $\mathbf{f}$ and the associated Jacobian matrix $J_{ij} = \frac{\partial f_i}{\partial a_j}$ and nominal values $\mathbf{a}_0$ of the parameters:

1 Generate a dense sample of data points $\{\mathbf{x}_i\}$ satisfying the model equations exactly so that $f(\mathbf{x}_i, \mathbf{a}_0) = 0$.

2 Calculate the associated Jacobian matrix J.

3 Determine the SVD of $J = USV^{\mathrm{T}}$ and singular values $s_1, \ldots, s_n$.

4 If J is full rank ($s_n > 0$) then the system is identifiable.

5 If J is not full rank, so that $s_{p+1} = \ldots = s_n = 0$, examine the corresponding columns $\mathbf{v}_{p+1}, \ldots, \mathbf{v}_n$ of V to determine which combinations of parameters are not determined from the data.

Example: incorrect parameterisation of a straight line.

A line in two dimensions is specified uniquely by two distinct points (x_0, y_0) and (u_0, v_0) lying on the line. Assuming that $x_0 \neq u_0$, let $\mathbf{a} = (x_0, y_0, u_0, v_0)^{\mathrm{T}}$ and

$$\phi(x, \mathbf{a}) = y_0 + \frac{v_0 - y_0}{u_0 - x_0}(x - x_0),$$

then the set of points on the line is described by $\{(x, y) : y = \phi(x, \mathbf{a})\}$. However the mapping

$$L : \mathbf{a} \mapsto \{(x, y) : y = \phi(x, \mathbf{a})\}$$

is not a proper parameterisation (we know that only two parameters are required). For a data set $X = \{(x_i, y_i)\}_{i=1}^m$ define

$$f_i(x_i, y_i, \mathbf{a}) = y_i - \phi(x_i, \mathbf{a})$$

and J the $m \times 4$ Jacobian matrix defined by $J_{ij} = \frac{\partial f_i}{\partial a_j}$. Below are example calculations for the case $\mathbf{a} = (1, 1, 5, 5)$ specifying the line $y = x$ and data $X = \{(1,1), (2,2), (3,3), (4,4), (5,5)\}$. In this case the Jacobian matrix J is

```
J =
    1.0000   -1.0000         0         0
    0.7500   -0.7500    0.2500   -0.2500
    0.5000   -0.5000    0.5000   -0.5000
    0.2500   -0.2500    0.7500   -0.7500
         0         0    1.0000   -1.0000
```

and has singular value decomposition $J = USV^{\mathrm{T}}$ with

```
U =
   -0.4472    0.6325    0.5410    0.2405
   -0.4472    0.3162   -0.2608   -0.6388
   -0.4472    0.0000   -0.6107    0.6441
   -0.4472   -0.3162   -0.1601   -0.3339
   -0.4472   -0.6325    0.4906    0.0881


S =
    2.2361         0         0         0
         0    1.5811         0         0
         0         0    0.0000         0
         0         0         0    0.0000


V =
   -0.5000    0.5000    0.7071         0
    0.5000   -0.5000    0.7071    0.0000
   -0.5000   -0.5000   -0.0000   -0.7071
    0.5000    0.5000    0.0000   -0.7071
```

The singular values are the diagonal elements of the matrix S and it is seen that the last two are zero (in floating point arithmetic they are of the order of 10^{-16} or smaller). The rightmost two singular vectors $\mathbf{v}_3$ and $\mathbf{v}_4$ of V indicate the two modes of behaviour of the parameters $\mathbf{a}$ that are not specified by the data. The vector $\mathbf{v}_3$ corresponds to (x_0, y_0) moving along the line $y = x$ and $\mathbf{v}_4$ corresponds to moving (u_0, v_0). These two degrees of freedom can be removed by fixing x_0 and u_0 at distinct values leaving a well defined parameterisation of the line in terms of the values of the y-coordinate at x_0 and u_0.

Example: fitting a sphere to data on a circle.

The least-squares best-fit sphere defined by centre co-ordinates $(a_1, a_2, a_3)^{\mathrm{T}}$ and radius a_4 to data $X = \{\mathbf{x}_i = (x_i, y_i, z_i)^{\mathrm{T}}\}_1^m$ is found by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} d^2(\mathbf{x}_i, \mathbf{a})$$

where

$$d(\mathbf{x}_i, \mathbf{a}) = \{(x_i - a_1)^2 + (y_i - a_2)^2 + (z_i - a_3)^2\}^{1/2} - a_4,$$

see section 5.10.

Below are the example calculations for the case $\mathbf{a} = (0, 0, 100, 100\sqrt{2})^{\mathrm{T}}$ and data set

```
  100.0000         0         0
   70.7107   70.7107         0
    0.0000  100.0000         0
  -70.7107   70.7107         0
 -100.0000    0.0000         0
  -70.7107  -70.7107         0
   -0.0000 -100.0000         0
   70.7107  -70.7107         0
```

representing eight points uniformly spaced around the circle $x^2 + y^2 = 100^2$ in the $z = 0$ plane. For this data the Jacobian matrix is

```
J =
   -0.5774         0    0.8165   -1.0000
   -0.4082   -0.4082    0.8165   -1.0000
   -0.0000   -0.5774    0.8165   -1.0000
    0.4082   -0.4082    0.8165   -1.0000
    0.5774   -0.0000    0.8165   -1.0000
    0.4082    0.4082    0.8165   -1.0000
    0.0000    0.5774    0.8165   -1.0000
   -0.4082    0.4082    0.8165   -1.0000
```

with singular values s_j and fourth right singular vector $\mathbf{v}_4$

```
    s            v4
3.6515            0
1.1547       0.0000
1.1547      -0.7746
0.0000      -0.6325
```

The vector $\mathbf{v}_4$ shows that by decreasing the z-coordinate of the sphere by 0.7746ϵ then we can ensure the sphere surface passes through the data points if we simultaneously decrease the radius by 0.6325ϵ. The system becomes fully identifiable if, for example, we add information about the radius from measurements of sphere diameter. Physically, this corresponds to the well-defined solution of a ball of radius $100\sqrt{2}$ sitting on a cylindrical hole of radius 100.

If we perform the same analysis but with $\mathbf{a} = (0, 0, 0, 100)^{\mathrm{T}}$, then we obtain singular values and fourth right singular vector

```
    s            v4
2.8284            0
2.0000            0
2.0000           -1
     0            0
```

The situation for this case is qualitatively different from the previous one, since adding information about the radius does not lead to system identifiability. Even if we know the sphere radius exactly, the z-component of the sphere centre is not determined from the data (from first order information).

10.2 System effectiveness

Given an identifiable system, i.e., one for which the Jacobian matrix is full rank, it is possible to use the Jacobian evaluation software in numerical simulations of an experiment to determine the likely uncertainties that will be obtained for a given measurement strategy. In this situation, exact measurement data $\mathbf{x}_i^*$ are generated according to a specified measurement strategy and from this, the Jacobian matrix J. Given an estimate of the standard deviation σ associated with the likely random effects, the uncertainty matrix associated with the estimates of the model parameters is given by

$$V_{\mathbf{a}} = \sigma^2 (J^{\mathrm{T}} J)^{-1},$$

section 4.2.3. By changing the measurement strategy and monitoring the effect on the variances of the parameters of interest it is often possible to improve the experimental efficiency. Importantly, this can be achieved by using the same data analysis modules required to determine estimates of the parameters and their uncertainties from actual measurement data. In other words, with very little additional effort the model solving tools can be used to improve experimental strategy.

Example: sphere fit to data on an equatorial band.

Following on from the example in section 10.1, suppose we wish to fit a sphere with nominal parameters $\mathbf{a} = (0, 0, 0, 100)^{\mathrm{T}}$ to data points lying in an equatorial band at height between zero and 20 above the equator. We are limited to choosing a total 30 data points uniformly distributed on circles at heights $h_1 = 0$, $h_2 = 10$ and $h_3 = 20$. What is the best distribution of points to determine a) the radius a_4 or b) the z-coordinate a_3 of the centre? Four measurement strategies M_k, $k = 1, 2, 3, 4$, are proposed specified by the number of points on each circle:

hk	M1	M2	M3	M4
0	10	15	28	2
10	10	0	0	0
20	10	15	2	28

Assuming that the standard deviation for the random effects is $\sigma = 0.01$, the standard uncertainties $u(a_j) = (\mathrm{var}(a_j))^{1/2}$ for each of the four sphere parameters a_j for the four measurement strategies M_k are

	M1	M2	M3	M4	
a1	0.0026	0.0026	0.0025	0.0025	
a2	0.0026	0.0026	0.0027	0.0027	
a3	0.0224	0.0183	0.0366	0.0366	z-coordinate
a4	0.0029	0.0026	0.0019	0.0071	radius

There results show that measurement strategy M_2, placing equal numbers of data points on the $z = 0$ and $z = 20$ circles and none on the $z = 10$ circle, is best to determine the z-coordinate a_3 of the centre while strategy M_3, placing nearly all the data points on the $z = 0$ circle (i.e., the equator) is best to determine the radius. While strategy M_1 seems appropriate, M_2 out performs

it in terms of determining both radius and z-coordinate. For either requirement, M_4 is a poor strategy.

Chapter 11

Case Studies

11.1 Univariate linear regression: study 1

11.1.1 Description

In this case study we are concerned with fitting a linear model

$$y = a_1 + a_2 x$$

to data points $\{(x_i, y_i)\}_1^m$. The case study uses this simplest of models to examine:

- the effectiveness of different estimators on data generated according to different statistical models; see section 3.3.

11.1.2 Statistical models

We consider the following statistical models.

M1 Standard experiment: the measurements of the response y are subject to uncorrelated Gaussian random effects:

$$y_i = a_1 + a_2 x_i + \epsilon_i, \quad \epsilon_i \in N(0, \sigma^2).$$

M2 The measurements of the response y are subject to uncorrelated Gaussian random effects with standard deviation proportional to x:

$$y_i = a_1 + a_2 x_i + \epsilon_i, \quad \epsilon_i \in N(0, \sigma^2 x_i^2).$$

M3 The measurements of the response y are subject to correlated Gaussian random effects drawn from a multinormal distribution with covariance matrix V.

M4 The measurements of both x and y are subject to Gaussian random effects:

$$x_i = x_i^* + \delta_i, \quad y_i = a_1 + a_2 x_i + \epsilon_i, \qquad \delta_i,\ \epsilon_i \in N(0, \sigma^2).$$

11.1.3 Estimators

We consider the following estimators:

E1 Linear least-squares estimator which solves

$$\min_{a_1, a_2} \sum_{i=1}^{m} (y_i - a_1 - a_2 x_i)^2.$$

This estimator is optimal for model M1.

E2 Weighted linear least-squares estimator which solves

$$\min_{a_1, a_2} \sum_{i=1}^{m} w_i^2 (y_i - a_1 - a_2 x_i)^2,$$

with $w_i = 1/x_i$. This estimator is optimal for model M2.

E3 Gauss Markov estimator which solves

$$\min_{\mathbf{a}} \mathbf{f}^{\mathrm{T}} V^{-1} \mathbf{f}$$

where $f_i = y_i - a_1 - a_2 x_i$. This estimator is optimal for model M3.

E4 Othogonal regression estimator which solves

$$\min_{\mathbf{a}, \{x_i^*\}} \sum_{i=1}^{m} (y_i - a_1 - a_2 x_i^*)^2 + (x_i - x_i^*)^2.$$

This is equivalent to the nonlinear least-squares estimator

$$\min_{\mathbf{a}} f_i^2(\mathbf{a})$$

with

$$f_i = \frac{y_i - a_1 - a_2 x_i}{(1 + a_2^2)^{1/2}}, \tag{11.1}$$

i.e., f_i is the orthogonal distance from (x_i, y_i) to the line $y = a_1 + a_2 x$. This estimator is optimal for model M4.

E5 Approximately weighted linear least-squares estimator

$$\min_{a_1, a_2} \sum_{i=1}^{m} w_i^2 (y_i - a_1 - a_2 x_i)^2,$$

with weights w_i randomly assigned in the range $[1, 2]$.

E6 Estimator which fits the line to the first and last data points only.

All estimators except E4 (and E6) are implemented by solving a linear least-squares problem involving the observation matrix

$$C = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix},$$

as described in section 4.1.

There are two approaches to finding the ODR estimator E4. The first is to treat it simply as a nonlinear least-squares problem. We note that, if f_i is given by (11.1),

$$\begin{aligned} \frac{\partial f_i}{\partial a_1} &= -\frac{1}{(1+a_2^2)^{1/2}}, \\ \frac{\partial f_i}{\partial a_2} &= -\frac{x_i + a_2(y_i - a_1)}{(1+a_2^2)^{3/2}}. \end{aligned}$$

Initial estimates for the parameters can be determined using estimator E1 for example. The second approach uses the singular value decomposition (SVD) described in section 4.3.4. The ODR estimate passes through the centroid $(\bar{x}, \bar{y})^{\mathrm{T}}$, where

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i, \qquad \bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i,$$

of the data (as does the E1 estimate) and has slope b_2/b_1 where $\mathbf{b} = (b_1, b_2)^{\mathrm{T}}$ is the right singular vector associated with the smallest singular value of the centred data matrix

$$\bar{\mathbf{z}} = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_m - \bar{x} & y_m - \bar{y} \end{bmatrix},$$

see [96].

11.1.4 Monte Carlo data generation

As an example, we consider lines with intercept $a_1 = 0$ and various slopes and 10 data points with nominal x-coordinates $x = 1, 2, \ldots, 10$. For each line, we generate exact data lying on the line $y_i = a_1 + a_2 x_i$ and N replicate data sets

$$\mathbf{z}_q = \{(x_{q,i}, y_{q,i})\}_i = \{(x_i, y_i) + (\delta_{q,i}, \epsilon_{q,i})\}_i, \quad q = 1, 2, \ldots,$$

where $\delta_{q,i}$ and $\epsilon_{q,i}$ are generated according to the appropriate statistical model.

11.1.5 Estimator assessment

We apply each estimator to each data set to determine best-fit parameters $\mathbf{a}_{k,q} = \mathcal{A}_k(\mathbf{z}_q)$ and then look at the mean value $\bar{a}_{j,k}$ and standard deviation $s_{j,k}$ of the

parameter estimates, $k = 1, \ldots, 6$, $j = 1, 2$. In the tables that follow, relative measures of performance are presented. For each parameter a_j, $j = 1, 2$, the minimum standard deviation $s_{j,min} = \min_k s_{k,j}$ is recorded. For each estimator and parameter, the quantity

$$\frac{\bar{a}_{j,k} - a_j^*}{s_{j,min}}, \quad j = 1, 2,$$

is a measure of the bias relative to the standard deviation, and

$$s_{j,k}/s_{j,min}$$

a measure of the efficiency relative to the estimator with the smallest standard deviation for the jth parameter. Tables 11.1–11.4 record these measures for 10,000 Monte Carlo simulations of data sets generated for the line $y = x$ according to each of the four models M1 – M4.

We note the following:

- For each of the first three models, only estimator E4, the ODR estimator shows significant bias. This is to be expected as, apart from E4, all the estimators can be derived from each other by a reweighting and reweighting has no effect on bias.
- For the first three models, the bias in estimator E4 arises from the fact that the measure of approximation error in (11.1) is the same as E1 except for the term $(1 + a_2^2)^{1/2}$ in the denominator. This approximation error is made smaller if a_2 is increased, so that relative to E1 estimator E4 is biased towards larger slopes. Similarly for model M4, all estimators apart from E4 and E6 are biased towards smaller slopes relative to E4. These biases are increased as the slope of the underlying line is increased. Tables 11.5 and 11.6 show the results for data generated under model M1 for the lines $y = 2x$ and $y = 10x$.
- For models M1, M2, and M3, the optimal estimators E1, E2, and E3 respectively, show the smallest standard deviations, i.e., they are most efficient. These results demonstrate the value of using the correct weighting strategy or taking into account correlation in the data.
- For all models the variations (as measured by $s_{j,k}$) in E1 and E4 are very close to each other, as to be expected. For model M4, E1 has a slightly smaller variation even although E4 is optimal. We can emphasise that a good estimator has to have good properties with respect to bias *and* efficiency and the minimum variance estimator may be entirely unsuitable due to bias.
- Only estimator E6 is relatively unbiased for all models. However, it is inefficient and not consistent in the sense that taking more data does not improve its performance.

Estimator E6 may be seen as an unrealistic type of estimator in that its deficiencies can be anticipated and it is therefore unlikely to be implemented. In

fact, this type of estimator is often implemented in situations in which a small subset of the data is used to determine values for a number of the parameters and these parameters are subsequently treated as constants.

11.1.6 Bibliography

The performance of algorithms for univariate regression in metrology is considered in [38, 211], for example.

M1	E1	E2	E3	E4	E5	E6
$a_1 = 0.0000$	0.0134	0.0123	0.0126	-0.1822	0.0188	0.0108
$s_{1,min} = 0.3436$	1.0000	1.5100	1.2400	1.0100	1.0900	1.6300
$a_2 = 1.0000$	-0.0209	-0.0192	-0.0053	0.1998	-0.0259	-0.0058
$s_{2,min} = 0.0554$	1.0000	1.9100	1.4600	1.0100	1.1100	1.4200

Table 11.1: Estimates of the bias and efficiency of six estimators for data generated according to model M1.

M2	E1	E2	E3	E4	E5	E6
$a_1 = 0.0000$	0.0385	0.0163	0.0285	-0.3800	0.0560	0.0053
$s_{1,min} = 0.0603$	2.3800	1.0000	4.0400	2.4000	2.7600	1.3100
$a_2 = 1.0000$	-0.0326	-0.0195	-0.0018	0.1616	-0.0386	0.0008
$s_{2,min} = 0.0236$	1.5700	1.0000	3.0700	1.5700	1.6500	2.3700

Table 11.2: Estimates of the bias and efficiency of six estimators for data generated according to model M2.

M3	E1	E2	E3	E4	E5	E6
$a_1 = 0.0000$	0.0047	-0.0059	0.0035	-0.1977	0.0157	-0.0092
$s_{2,min} = 0.2487$	1.2000	2.3500	1.0000	1.2100	1.1400	2.5000
$a_2 = 1.0000$	-0.0232	0.0024	-0.0033	0.3470	-0.0420	0.0094
$s_{2,min} = 0.0247$	1.5900	4.7500	1.0000	1.6100	1.7100	2.5200

Table 11.3: Estimates of the bias and efficiency of six estimators for data generated according to model M3.

M4	E1	E2	E3	E4	E5	E6
$a_1 = 0.0000$	0.2511	0.0507	0.2156	-0.0260	0.3370	-0.0681
$s_{1,min} = 0.4747$	1.0000	1.5100	1.2600	1.0300	1.0900	1.7000
$a_2 = 1.0000$	-0.2858	-0.3498	-0.0991	0.0288	-0.3893	0.0679
$s_{2,min} = 0.0760$	1.0000	1.8600	1.4700	1.0300	1.1100	1.4800

Table 11.4: Estimates of the bias and efficiency of six estimators for data generated according to model M4.

M4	E1	E2	E3	E4	E5	E6
$a_1 = 0.0000$	0.0134	0.0123	0.0126	**-0.2994**	0.0188	0.0108
$s_{1,min} = 0.6872$	1.0000	1.5100	1.2400	1.0100	1.0900	1.6300
$a_2 = \mathbf{2.0000}$	-0.0209	-0.0192	-0.0053	**0.3321**	-0.0259	-0.0058
$s_{2,min} = 0.1107$	1.0000	1.9100	1.4600	1.0100	1.1100	1.4200

Table 11.5: Estimates of the bias and efficiency of six estimators for data generated for the line $y = 2x$ according to model M1.

M4	E1	E2	E3	E4	E5	E6
$a_1 = 0.0000$	0.0134	0.0123	0.0126	**-0.3730**	0.0188	0.0108
$s_{1,min} = 3.4359$	1.0000	1.5100	1.2400	1.0000	1.0900	1.6300
$a_2 = \mathbf{10.0000}$	-0.0209	-0.0192	-0.0053	**0.4152**	-0.0259	-0.0058
$s_{2,min} = 0.5536$	1.0000	1.9100	1.4600	1.0000	1.1100	1.4200

Table 11.6: Estimates of the bias and efficiency of six estimators for data generated for the line $y = 10x$ according to model M1.

11.2 Univariate linear regression: study 2

This case study involves the univariate linear model in which a response y is modelled as a linear function of a single variable x:

$$y = a_1 + a_2 x.$$

It is a continuation of the case study in section 11.1 which considered the behaviour of different fitting methods for different statistical models.

11.2.1 Statistical model for correlated random effects

In this case study there is structural correlation associated with the random effects. An example application is in modelling the linear response of a balance to the application of a mass y where the mass is made up from a number of component masses $\{m_k\}_{k=1}^K$ of same nominal mass along with a mass m_0 present in all measurements. We assume that the component masses m_k have been calibrated using the same equipment so that the estimate m_k is related to the "true" mass m_k^* through

$$m_k = m_k^* + \mu_k, \quad \mu_k \in N(0, \sigma_M^2).$$

Furthermore the "constant" mass has also been measured so that

$$m_0 = m_0^* + \mu_0, \quad \mu_0 \in N(0, \sigma_0^2).$$

If, in the ith measurement, the mass y_i is assembled from masses $\{m_{i_1}, \ldots, m_{i_q}\}$ and the response x_i is accurately measured, then an appropriate model is

$$\begin{aligned} y_i^* &= a_1 + a_2 x_i, \quad y_i^* = m_0^* + \sum_{k=1}^{q} m_{i_k}^*, \\ y_i &= y_i^* + \epsilon_i + \mu_0 + \sum_{k=1}^{q} \mu_{i_k}, \\ \epsilon_i &\in N(0, \sigma^2), \quad \mu_0 \in N(0, \sigma_0^2), \quad \mu_k \in N(0, \sigma_M^2),\ k = 1, \ldots, K, \end{aligned} \tag{11.2}$$

where μ_0 accounts for uncertainty associated with the constant mass, μ_{i_k} that in the component masses and ϵ_i accounts for a random disturbance for that measurement (due to random air buoyancy effects, for example). The random effects associated with the y_i are correlated with each other due to their common dependence on the component masses and m_0.

Since (11.2.1) gives the explicit dependence of y_i on the measurement of $\{m_k\}$ and m_0 the correlation can be determined from the model. For example, suppose in an experiment the masses m_k are added one at a time until all K masses are loaded and the response x accurately measured. Subsequently, the masses are removed one at a time, so that two measurements are recorded for each set of masses $m_1 + \cdots + m_q$. Let V_M be the $(K+1) \times (K+1)$ diagonal covariance

matrix with σ_M^2 in the first K diagonal elements and σ_0^2 in the $(K+1)$th:

$$V_M = \begin{bmatrix} \sigma_M^2 & & & & \\ & \sigma_M^2 & & & \\ & & \ddots & & \\ & & & \sigma_M^2 & \\ & & & & \sigma_0^2 \end{bmatrix},$$

and G the $2K \times (K+1)$ matrix with

$$G_{ij} = \frac{\partial y_i}{\partial m_j}, \quad 1 \le j \le K, \quad G_{ij} = \frac{\partial y_i}{\partial m_0}, \quad j = K+1,$$

so that G has the form

$$G = \begin{bmatrix} 1 & & & & 1 \\ 1 & 1 & & & 1 \\ \vdots & \vdots & & & \vdots \\ 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & & & \vdots \\ 1 & 1 & & & 1 \\ 1 & & & & 1 \end{bmatrix}.$$

Then the covariance matrix for $\mathbf{y} = (y_1, \ldots, y_{2K})^{\mathrm{T}}$ is

$$V_{\mathbf{y}} = G V_M G^{\mathrm{T}} + \sigma^2 I, \tag{11.3}$$

where I is the $2K \times 2K$ identity matrix. The first term is the covariance that arises from the measurements of the masses m_k and m_0 while the second is the random, uncorrelated component.

For this model, the Gauss-Markov least-squares estimator (GMLS) that determines estimates by the solving

$$\min_{\mathbf{a}} \ \mathbf{f}^{\mathrm{T}} V_{\mathbf{y}}^{-1} \mathbf{f}, \tag{11.4}$$

where $\mathbf{f} = (f_1, \ldots, f_{2K})^{\mathrm{T}}$ with $f_i = y_i - a_1 - a_2 x_i$, is optimal (section 4.1.6). If C is the observation matrix with $(1, x_i)$ in the ith row, the covariance matrix $\tilde{V}_{\mathbf{a}}$ of the GMLS estimates is

$$\tilde{V}_{\mathbf{a}} = (C^{\mathrm{T}} V_{\mathbf{y}}^{-1} C)^{-1}. \tag{11.5}$$

However, we can also apply the simpler linear least-squares estimator (LLS) that solves

$$C\mathbf{a} = \mathbf{y}$$

in the least-squares sense, so that mathematically,

$$\mathbf{a} = C^{\dagger}\mathbf{y}, \quad C^{\dagger} = (C^{\mathrm{T}} C)^{-1} C^{\mathrm{T}}.$$

As before, an estimate of the covariance matrix of the fitted parameters is given by

$$V_{\mathbf{a}} = \hat{\sigma}(C^{\mathrm{T}}C)^{-1},$$

where $\hat{\sigma}$ is an estimate of the standard deviation of the residuals.

We ask:

V1 Is the LLS estimator appropriate for this model?

V2 Are the estimates of uncertainties valid?

The covariance matrix for the LLS estimates for this model is

$$V^*_{\mathbf{a}} = C^{\dagger}V_{\mathbf{y}}(C^{\dagger})^{\mathrm{T}}. \tag{11.6}$$

By comparing this matrix with $\tilde{V}_{\mathbf{a}}$, we can assess the LLS estimator relative to the optimal GMLS estimator. We can also use Monte Carlo simulation to check the performance of the LLS estimator and examine the validity of the evaluation $V_{\mathbf{a}}$. Given values of σ_M, σ_0, σ, $\{x_i\}_{i=1}^m$ and the model parameters $\mathbf{a}$, we generate data sets $\mathbf{z}_q = \{(x_i, y_{i,q})\}_{i=1}^m$, $q = 1, \ldots, N$, according to the model:

I Generate exact data

$$y^*_i = a_1 + a_2 x_i.$$

II$_q$ For each q,

II.1$_q$ Generate

$$\mu_{0,q} \in N(0, \sigma_0^2), \quad \mu_{k,q} \in N(0, \sigma_M^2), \quad k = 1, \ldots, K.$$

II.2$_q$ If the ith mass is assembled from masses $\{m_{i_1}, \ldots, m_{i_q}\}$, set

$$y_{i,q} = y^*_i + \epsilon_{i,q} + \mu_{0,q} + \sum_{k=1}^{q} \mu_{i_k,q}, \quad \epsilon_{i,q} \in N(0, \sigma^2).$$

For each data set $\mathbf{z}_q$ the LLS estimates $\mathbf{a}_q$ are recorded along with the estimate

$$\hat{\sigma}_q = \frac{1}{m-n}\|\mathbf{r}_q\|$$

of the standard deviation of residuals calculated for the qth residual $\mathbf{r}_q = \mathbf{y}_q - C\mathbf{a}_q$. As before, an estimate of the average covariance matrix is given by

$$V_{\mathbf{a}} = \bar{\sigma}^2(C^{\mathrm{T}}C)^{-1}, \quad \bar{\sigma} = \left(\frac{1}{N}\sum_{q=1}^{N}\hat{\sigma}_q^2\right)^{1/2}. \tag{11.7}$$

We have performed 5000 Monte Carlo simulations for the data set

$$\mathbf{x}^* = (1, 2, \ldots, 8, 9, 9, 8, \ldots, 2, 1)^{\mathrm{T}}$$

and $\mathbf{a}^* = (1, 1)^{\mathrm{T}}$ and various values of σ_M, σ_0 and σ. For each set of experiments, we calculate:

	$u(\tilde{a}_j)$	$u^*(a_j)$	$\bar{u}(a_j)$	$u(a_j)$
E1	$\sigma_M = 0.001$, $\sigma_0 = 0.001$, $\sigma = 0.01$			
a_1	5.4130e-003	5.4152e-003	5.1437e-003	5.4295e-003
a_2	9.8478e-004	9.8489e-004	9.1406e-004	9.8536e-004
E2	$\sigma_M = 0.01$, $\sigma_0 = 0.001$, $\sigma = 0.001$			
a_1	1.0684e-002	1.3956e-002	4.1260e-003	1.3906e-002
a_2	3.5377e-003	3.6980e-003	7.3321e-004	3.6612e-003
E3	$\sigma_M = 0.01$, $\sigma_0 = 0.001$, $\sigma = 0.001$			
a_1	1.0081e-002	1.0109e-002	6.5555e-004	1.0135e-002
a_2	3.7093e-004	3.8079e-004	1.1650e-004	3.7730e-004

Table 11.7: Estimates of the standard uncertainties of the fitted parameters to data generated according to the statistical model (11.2.1).

$u(\tilde{a}_j)$ the standard uncertainty associated with the GMLS estimates calculated according to (11.5).

$u^*(a_j)$ the standard uncertainty associated with the LLS parameter estimates calculated according to (11.6).

$\bar{u}(a_j)$ the sample standard deviation associated with the LLS estimates $\mathbf{a}_q$.

$u(a_j)$ the standard uncertainty associated with the LLS parameter estimates calculated from (11.7).

Results of these calculations are presented in table 11.7. We can also analyse these results in terms of the estimates of the standard uncertainties $u(\hat{y})$ of the model predictions $\hat{y}$ at any point x graphed in figures 11.1–11.3.

From the table and figures, we note

- For the case $\sigma_M = 0.001$, $\sigma_0 = 0.001$ and $\sigma = 0.01$ in which the independent random component ϵ_i dominates (E1), the behaviour of the LLS estimator is very similar to that of the GMLS estimator. The statistics based on the approximate model derived from (11.7) underestimates slightly the uncertainty in the model predictions. The pairs of residuals for each ordinate value x_i show a random behaviour.

- For the case $\sigma_M = 0.01$, $\sigma_0 = 0.001$ and $\sigma = 0.001$ in which the component μ_k dominates (E2), the LLS estimator performs less well than the GMLS estimator, but not drastically so. However, the uncertainties based on the approximate statistical model completely underestimate the uncertainties in the model predictions. The pairs of residuals exhibit a strong correlation.

- For the case $\sigma_M = 0.001$, $\sigma_0 = 0.01$ and $\sigma = 0.001$ in which the component μ_0 associated with the constant mass dominates (E3), the LLS and GMLS estimator have very similar behaviour but the uncertainties based on the approximate statistical model completely underestimate the uncertainties in the model predictions. The pairs of residuals show a random behaviour.

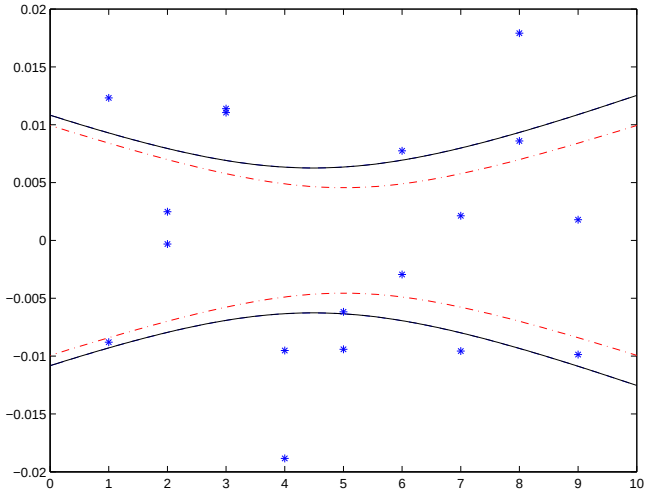

Figure 11.1: Estimates of uncertainty in the model predictions for the statistical model (11.2.1) with $\sigma_M = 0.001$, $\sigma_0 = 0.001$ and $\sigma = 0.01$ (E1). The three pairs of graphs present $\pm 2u(\hat{y})$ as a function of x determined from three estimates of the covariance matrix: (i) $\tilde{V}_{\mathbf{a}}$, the covariance matrix of the fitted parameters for the GMLS estimator (Eq. (11.5), solid curve), (ii) $V^*_{\mathbf{a}}$, the correct estimate of the covariance matrix of the LLS estimates (Eq. (11.6), dashed curve essentially coincident with the solid curve) and (iii) $V_{\mathbf{a}}$, the covariance matrix for the LLS estimates based on approximate statistical model (Eq. (11.7), dot-dashed curve). The residuals for an example fit are also graphed ('*').

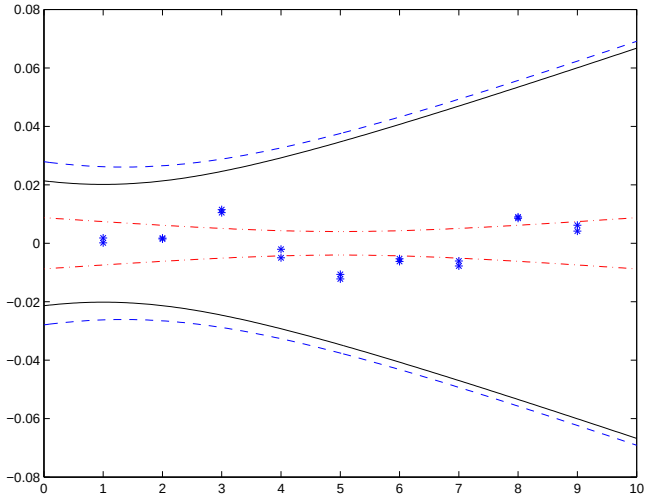

Figure 11.2: As figure 11.1 but with $\sigma_M = 0.01$, $\sigma_0 = 0.001$ and $\sigma = 0.001$ (E2).

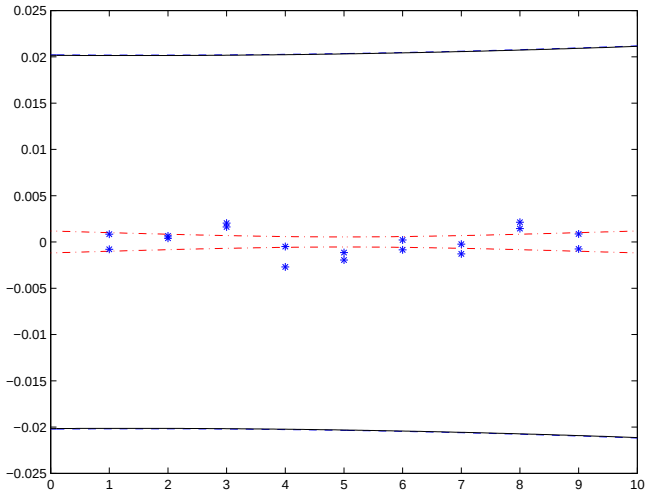

Figure 11.3: As figure 11.1 but with $\sigma_M = 0.001$, $\sigma_0 = 0.01$ and $\sigma = 0.001$ (E3).

This example illustrates an important point. The GMLS estimator is optimal but the LLS estimator performs quite well for the three types of experiment: the variation in the LLS estimates are not drastically worse than that of the GMLS estimates. However, the estimates of uncertainties associated with the fitted parameters based on the approximate statistical model can be completely misleading. The uncertainties for the approximate statistical model cannot be made valid simply by introducing a fudge factor to increase the estimate $\hat{\sigma}$ of the standard uncertainty of the residuals. The estimate of the model predictions for experiment E2 graphed in figure 11.2 has the wrong shape compared with the actual uncertainties. Changing the value of $\hat{\sigma}$ will only change the scale, not the underlying shape. In the case of the third set of experiments (E3), the residuals shown in figure 11.3 are consistent with the approximate statistical model and the fit of the model to data is a good one. From an analysis of the consistency of the model with the measured data, we can be misled into believing that the model outputs, including the uncertainty estimates, are valid when in fact they are not.

11.3 Fitting a quadratic to data

In this case study, we examine the problem of fitting a quadratic model

$$y = a_1 + a_2 x + a_3 x^2$$

to data $\{(x_i, y_i)\}_{i=1}^m$ where both x_i and y_i are subject to uncertainty. The complete model is of the form

$$x_i = x_i^* + \delta_i, \quad y_i = a_1 + a_2 x_i + a_3 x_i^2 \epsilon_i, \qquad \delta_i \in N(0, \rho^2), \quad \epsilon_i \in N(0, \gamma^2). \tag{11.8}$$

The preferred estimator for this type of model is the generalised distance regression estimator (GDR, section 4.3) whose estimates are given by the solution of

$$\min_{\mathbf{a}, \mathbf{x}^*} \sum_{i=1}^m \left\{ \alpha^2 (x_i - x_i^*)^2 + \beta^2 (y_i - \phi(x_i^*, \mathbf{a}))^2 \right\} \tag{11.9}$$

where $\phi(x, \mathbf{a}) = a_1 + a_2 x + a_3 x^2$, $\alpha = 1/\rho$ and $\beta = 1/\gamma$. However, we want to know if the linear least-squares estimates (LLS) determined by the solution of the simpler problem

$$\min_{\mathbf{a}} \sum_{i=1}^m (y_i - \phi(x_i, \mathbf{a}))^2$$

are also suitable.

If C is the $m \times 3$ matrix with ith row given by $(1, x_i, x_i^2)$ then the LLS estimate of the parameters is defined mathematically by

$$\mathbf{a} = C^\dagger \mathbf{y}, \quad C^\dagger = (C^{\mathrm{T}} C)^{-1} C^{\mathrm{T}}. \tag{11.10}$$

As before, an estimate of the covariance matrix of the fitted parameters is given by

$$V_{\mathbf{a}} = \hat{\sigma} (C^{\mathrm{T}} C)^{-1}, \tag{11.11}$$

where $\hat{\sigma}$ is an estimate of the standard deviation of the residuals:

$$\hat{\sigma} = \|\mathbf{r}\| / (m-3)^{1/2}, \quad \mathbf{r} = (r_1, \ldots, r_m)^{\mathrm{T}}, \quad r_i = y_i - \phi(x_i, \mathbf{a}). \tag{11.12}$$

We ask:

V1 Is the LLS estimator appropriate for the model (11.8)?

V2 Are the associated estimates of uncertainties (11.11) valid?

We can use Monte Carlo simulation and null space benchmarking described in sections 8.2 and 8.3 to answer these questions. Given values of ρ, γ, $\{x_i^*\}_{i=1}^m$ and the model parameters $\mathbf{a}^*$, we generate data sets $\mathbf{z}_q = \{(x_{i,q}, y_{i,q})\}_{i=1}^m$, $q = 1, \ldots, N$, according to the model:

$$\begin{aligned} y_i^* &= a_1^* + a_2^* x_i^* + a_3^* (x_i^*)^2, \\ x_{i,q} &= x_i^* + \delta_{i,q}, \quad \delta_{i,q} \in N(0, \rho^2), \\ y_{i,q} &= y_i^* + \epsilon_{i,q}, \quad \epsilon_{i,q} \in N(0, \gamma^2). \end{aligned}$$

For each data set $\mathbf{z}_q$ the observation matrix C_q with ith row equal to $(1, x_{i,q}, x_{i,q}^2)$ is calculated and the LLS estimates $\mathbf{a}_q$ found by solving

$$C_q \mathbf{a}_q = \mathbf{y}_q.$$

The estimate $\hat{\sigma}_q$ of the standard deviation of residuals calculated according to (11.12) for the qth set of residuals $\mathbf{r}_q = \mathbf{y}_q - C_q \mathbf{a}_q$ is also recorded. For small data sets (number of data points $m < 50$, say), the estimates $\hat{\sigma}_q$ will have relatively large variation. Consequently, the estimates of the covariance matrix of the fitted parameters will also vary. An estimate of the average covariance matrix is given by

$$V_{\mathbf{a}} = \bar{\sigma}^2 (C^{\mathrm{T}} C)^{-1}, \quad \bar{\sigma} = \left(\frac{1}{N} \sum_{q=1}^{N} \hat{\sigma}_q^2 \right)^{1/2}. \tag{11.13}$$

From simulations we can compare the actual variation in the LLS estimates with the variation predicted using the approximate statistical model. In this way we can answer question V2. To answer question V1, we would like to compare the variation in the LLS estimates with those for the GDR estimator. We can do this using null space benchmarking.

In section 8.3.2, we showed how to generate data according to models like (11.8) for which the GDR solution estimates are known. For the model (11.8), the data generation scheme is as follows. Given $\mathbf{a}^* = (a_1^*, a_2^*, a_3^*)^{\mathrm{T}}$, α, β, σ and $\{x_i^*\}$,

I Set $y_i^* = \phi(x_i^*, \mathbf{a}^*) = a_1^* + a_2^* x_i^* + a_3^* (x_i^*)^2$ and $\mathbf{z}^* = \{(x_i^*, y_i^*)\}$.

II For each i, calculate

$$\dot{\phi}_i = \frac{\partial \phi}{\partial x}(x_i^*, \mathbf{a}^*) = a_2^* + 2 a_3^* x_i^*, \quad s_i = \left(\frac{\dot{\phi}_i^2}{\alpha^2} + \frac{1}{\beta^2} \right)^{1/2}.$$

III Calculate A given by

$$A_{ij} = \frac{\partial \phi_i}{\partial a_j},$$

so that the ith row of A is $(1, x_i^*, (x_i^*)^2)$.

IV Determine $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_m)^{\mathrm{T}}$ such that $A^{\mathrm{T}} \boldsymbol{\delta} = 0$ (using the QR factorisation of A, for example). For each i, set

$$p_i = -\delta_i \frac{\dot{\phi}_i}{\alpha^2}, \quad q_i = \delta_i \frac{1}{\beta^2}.$$

and calculate $S = \left\{ \sum_{i=1}^{m} (\alpha^2 p_i^2 + \beta^2 q_i^2) \right\}^{1/2}$ and $K = (m-n)^{1/2} \sigma / S$.

V For each i, set

$$x_i = x_i^* + K p_i, \quad y_i = y_i^* + K q_i.$$

Then $\mathbf{a}^*$ and $\{x_i^*\}$ solves (11.9) for data set $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m$ with

$$\frac{1}{m-n}\sum_{i=1}^{m}\left\{\alpha^2(x_i - x_i^*)^2 + \beta^2(y_i - \phi(x_i^*, \mathbf{a}^*))^2\right\} = \sigma^2,$$

and the estimate $\tilde{V}_{\mathbf{a}}$ of the covariance matrix of the fitted parameters $\mathbf{a}^*$ for the GDR estimator is

$$\tilde{V}_{\mathbf{a}} = \sigma^2(J^{\mathrm{T}}J)^{-1}, \tag{11.14}$$

where the Jacobian matrix J can be calculated from $J_{ij} = A_{ij}/s_i$.

We have performed 5000 Monte Carlo simulations for the data sets generated for the quadratic defined by $\mathbf{a}^* = (0, 0, 5)^{\mathrm{T}}$ and exact data $\{(x_i^*, y_i^*)\}$:

x*	y*	delta	epsilon
-1.0000	5.0000	0.0050	0.0005
-0.8000	3.2000	-0.0156	-0.0020
-0.6000	1.8000	0.0100	0.0017
-0.4000	0.8000	0.0004	0.0001
-0.2000	0.2000	-0.0018	-0.0009
0	0	0	0.0015
0.2000	0.2000	0.0010	-0.0005
0.4000	0.8000	-0.0014	0.0004
0.6000	1.8000	0.0028	-0.0005
0.8000	3.2000	0.0141	-0.0018
1.0000	5.0000	-0.0143	0.0014

for different values of γ and ρ. Also listed above are example null space perturbations δ_i and ϵ_i generated for the case $\rho = \gamma = 0.01$.

For each set of experiments, we calculate

$u(\tilde{a}_j)$ the standard uncertainty of the GDR parameter estimates calculated according to (11.14) for data generated using the null space method.

$\bar{u}(a_j)$ the sample standard deviation of the LLS estimates a_j.

$u(a_j)$ the standard uncertainty of the LLS parameter estimates calculated from (11.13).

Results of these calculations are presented in table 11.8.

We can also analyse these results in terms of the estimates of the standard uncertainties $u(\hat{y})$ of the model predictions $\hat{y}$ at any point x graphed in figures 11.4–11.6.

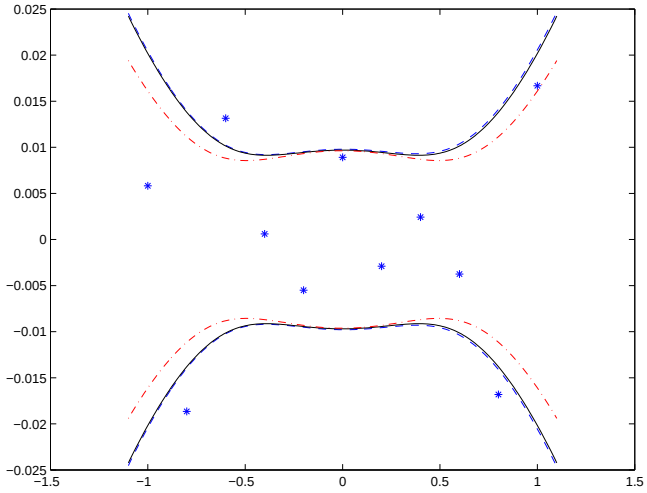

Figure 11.4: Estimates of uncertainty in the model predictions for the statistical model (11.8) with $\rho = 0.001$ and $\gamma = 0.01$ (E1). The three pairs of graphs present $\pm 2u(\hat{y})$ as a function of x determined from three estimates of the covariance matrix: (i) $\tilde{V}_{\mathbf{a}}$, the covariance matrix of the fitted parameters for the GDR estimator (Eq. (11.14), solid curve), (ii) $\bar{V}_{\mathbf{a}}$, the estimate of the covariance matrix of the LLS estimates determined from the Monte Carlo simulations (dashed curve, essentially coincident with the solid curve) and (iii) $V_{\mathbf{a}}$, the covariance matrix for the LLS estimates based on an approximate statistical model (Eq. (11.13), dot-dashed curve). The residuals for an example LLS fit are also graphed ('*').

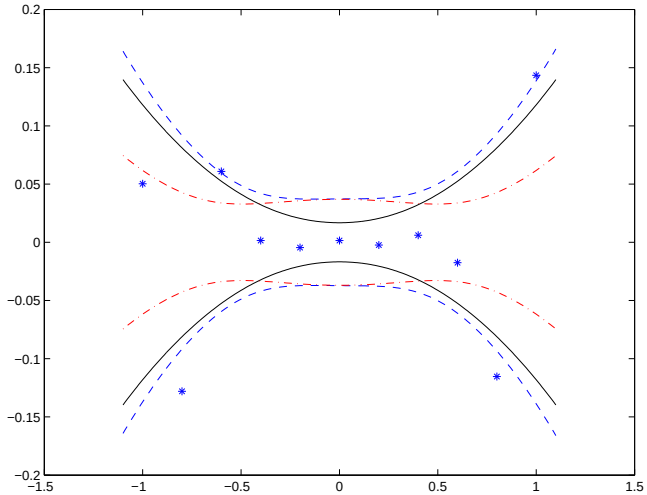

Figure 11.5: As figure 11.4 but for $\rho = 0.01$ and $\gamma = 0.01$ (E2).

	$u(\tilde{a}_j)$	$\bar{u}(a_j)$	$u(a_j)$
E1	$\rho = 0.001$, $\gamma = 0.01$		
a_1	0.0048	0.0049	0.0052
a_2	0.0061	0.0063	0.0055
a_3	0.0106	0.0107	0.0098
E2	$\rho = 0.01$, $\gamma = 0.01$		
a_1	0.0084	0.0186	0.0266
a_2	0.0327	0.0404	0.0279
a_3	0.0515	0.0668	0.0499
E3	$\rho = 0.01$, $\gamma = 0.001$		
a_1	0.0010	0.0181	0.0263
a_2	0.0316	0.0401	0.0275
a_3	0.0477	0.0664	0.0492

Table 11.8: Estimates of the standard uncertainties of the fitted parameters to data generated according to the statistical model (11.8): $u(\tilde{a}_j)$ is the standard uncertainty of the GMLS parameter estimates calculated according to (11.14), $\bar{u}(a_j)$ is the actual variation in the LLS estimates and $u(a_j)$ is the predicted variation calculated assuming an approximate statistical model as in (11.13)

.

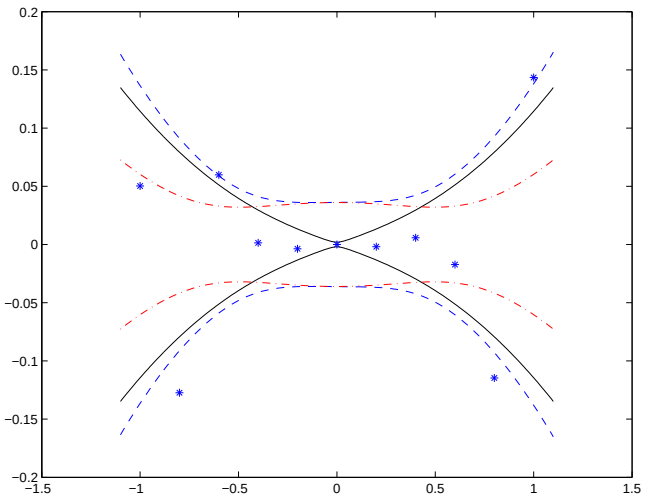

Figure 11.6: As figure 11.4 but for $\rho = 0.01$ and $\gamma = 0.001$ (E3).

From the table and figures, we note:

- For the case $\rho = 0.001$ and $\gamma = 0.01$ in which the uncertainty associated with y dominates (E1), the behaviour of the LLS estimator is very similar to that of the GDR estimator. The statistics based on the approximate model derived from (11.13) underestimates slightly the uncertainty in the model predictions.
- For the case $\rho = 0.01$ and $\gamma = 0.01$ in which the uncertainties associated with x and y are equal (E2), the LLS estimator performs less well than the GDR estimator, particularly in terms of the uncertainty of the model predictions near $x = 0$. However, the uncertainties based on the approximate statistical model significantly underestimates the uncertainty in the model predictions away from $x = 0$.
- The case $\rho = 0.01$ and $\gamma = 0.001$ in which the uncertainty associated with x dominates (E3) is similar to that for (E2) only with larger differences in behaviour.

This analysis is similar to the analysis carried out for the case of univariate linear regression in section 11.2. In both cases, we are interested in determining how well an approximate estimator performs against an estimator that is known to be optimal. In the univariate regression case, we were able to derive correct estimates for the actual variation of the approximate estimator for the statistical model due to the fact that the estimates provided by the approximate method were a linear function of the measurement data. For the case considered in this section, the estimates defined in (11.10) are a nonlinear function of the data since the matrix $C^{\dagger}$ also depends on the measured x variables. This means that it not so straightforward to provide valid estimates for uncertainty associated with the LLS estimator. However, Monte Carlo simulations provide a simple method of determining them.

11.4 Generalised maximum likelihood estimation and multiple random effects

See section 4.11 and [31].

In many measuring instruments, the variance of the random effects associated with the measurements has a dependence on the response value. As an example, suppose the model is

$$\boldsymbol{\eta} = C\boldsymbol{\alpha}, \quad Y_i = \eta_i + E_i, \quad E_i \sim N(0, (\sigma_1 + \sigma_2\eta_i)^2), \tag{11.15}$$

with the random variables E_i independently distributed and that $\mathbf{y}$ is a set of observations of $\boldsymbol{Y}$. The likelihood of observing y_i given $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma} = (\sigma_1, \sigma_2)^{\mathrm{T}}$ is

$$p(y_i|\boldsymbol{\alpha}, \boldsymbol{\sigma}) = \left(\frac{\phi_i}{2\pi}\right)^{1/2} \exp\left[-\frac{\phi_i}{2}(y_i - \mathbf{c}_i^{\mathrm{T}}\boldsymbol{\alpha})^2\right],$$

where $\phi_i = \phi_i(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = 1/(\sigma_1 + \sigma_2\eta_i)^2$.

The log likelihood $L(\boldsymbol{\alpha}, \boldsymbol{\sigma}|\mathbf{y}) = \log p(\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\sigma})$ is given by

$$\begin{aligned} -L(\boldsymbol{\alpha}, \boldsymbol{\sigma}|\mathbf{y}) &= -\sum_{i=1}^{m} \log p(y_i|\boldsymbol{\alpha}, \boldsymbol{\sigma}), \\ &= \frac{1}{2}\left\{ -\sum_{i=1}^{m} \log \phi_i \; + \; \sum_{i=1}^{m} \phi_i (y_i - \mathbf{c}_i^{\mathrm{T}}\boldsymbol{\alpha})^2 \right\}. \end{aligned}$$

For a prior distribution, we set

$$-\log p(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = u^2(\log\sigma_1 - \log\sigma_{1,0})^2 + v^2(\sigma_2 - \sigma_{2,0})^2 \quad + \quad \text{Const.},$$

where u and v are weights that reflect our confidence in the prior estimates $\sigma_{k,0}$, $k = 1, 2$. This distribution reflects some prior information about $\boldsymbol{\sigma}$ but none about $\boldsymbol{\alpha}$ since with $\boldsymbol{\sigma}$ fixed, $p(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ is constant. The use of a log normal prior distribution is intended to reflect our belief that the estimate $\boldsymbol{\sigma}_0$ is equally likely to be an under- or overestimate by a multiplicative factor. As defined, $p(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ is an improper distribution as its integral over $\boldsymbol{\alpha}$ is infinite. We could instead choose a prior which was zero outside some region $\Omega \subset \mathbb{R}^n$ of sufficiently large but finite volume. However, since our approximation to the posterior distribution is based only local information, both priors would lead to the same parameter estimates and uncertainties (so long as the region Ω contained the solution estimate of $\boldsymbol{\alpha}$).

Estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$ are found by minimising

$$F(\boldsymbol{\alpha}, \boldsymbol{\sigma}|\mathbf{y}) = -L(\boldsymbol{\alpha}, \boldsymbol{\sigma}|\mathbf{y}) + u^2(\log\sigma_1 - \log\sigma_{1,0})^2 + v^2(\sigma_2 - \sigma_{2,0})^2,$$

with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$. If H is the Hessian matrix at the solution $(\mathbf{a}, \mathbf{s})$ and $V = H^{-1}$ its inverse, then the standard uncertainties associated with the estimates of the fitted parameters are the square roots of the diagonal elements of V.

To illustrate the GMLE approach we have generated data according to the model (11.15) for a quadratic response $\eta = \alpha_1 + \alpha_2\xi + \alpha_3\xi^2$ to data generated with $\boldsymbol{\alpha} = (0.0, 1.0, 2.0)^{\mathrm{T}}$ and firstly with $\boldsymbol{\sigma} = (0.10, 0.02)^{\mathrm{T}}$: see figure 11.7. We have set prior estimates $\sigma_{k,0} = 0.05$, $k = 1, 2$, and weights 1) $u = v = 0.0001$ and 2) $u = v = 10000.0$, corresponding to weakly and strongly weighted prior information, respectively. Table 11.9 gives the resulting estimates $\mathbf{a}$ and $\mathbf{s}$ along with their associated uncertainties $\mathbf{u}$. Table 11.10 gives corresponding results for data generated with $\boldsymbol{\sigma} = (0.02, 0.10)^{\mathrm{T}}$, figure 11.8, with all other parameters as above. The tables show that for the weakly weighted prior information, the posterior estimates of $\boldsymbol{\sigma}$ are reasonable while for the strongly weighted, the posterior estimates are close to the prior values, as to be expected.

To obtain an indication of the validity of the uncertainty estimates $\mathbf{u}$, we have repeated these numerical simulations N times, recording the estimates $\mathbf{a}_q$, $\mathbf{s}_q$ and $\mathbf{u}_q$, $q = 1, \ldots, N$, and then calculating the sample means $\bar{\mathbf{a}}$, $\bar{\mathbf{s}}$ and $\bar{\mathbf{u}}$ and sample standard deviations $s(\mathbf{a})$ and $s(\mathbf{s})$. At the same time we compare the behaviour of the GMLE algorithm with a weighted least-squares algorithm (WLS) which we now describe.

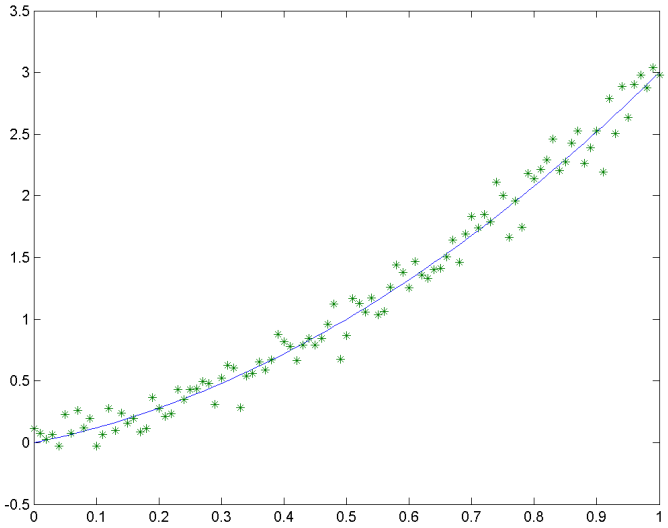

Figure 11.7: Data generated for a quadratic response and model (11.15) with $\boldsymbol{\alpha} = (0.0, 1.0, 2.0)^{\mathrm{T}}$ and $\boldsymbol{\sigma} = (0.10, 0.02)^{\mathrm{T}}$.

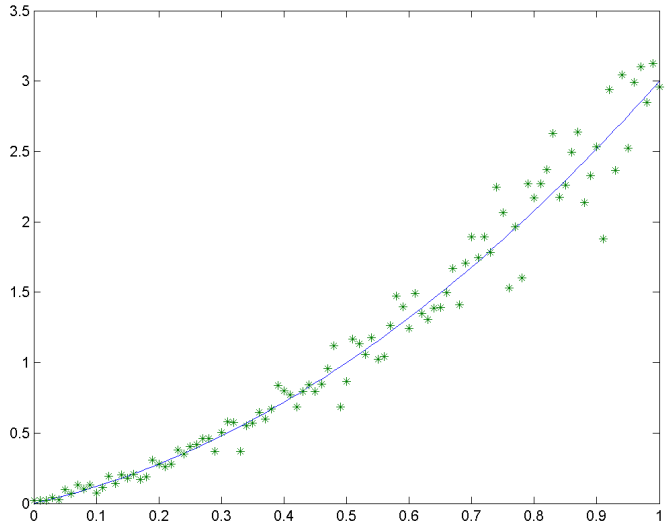

Figure 11.8: Data generated for a quadratic response and model (11.15) with $\boldsymbol{\alpha} = (0.0, 1.0, 2.0)^{\mathrm{T}}$ and $\boldsymbol{\sigma} = (0.02, 0.10)^{\mathrm{T}}$.

		$u, v = 0.0001$		$u, v = 10000.0$	
		a, **s**	**u**	**a**, **s**	**u**
α_1	0.00	0.0338	0.028	0.0471	0.018
α_2	1.00	0.8400	0.145	0.7673	0.115
α_3	2.00	2.1643	0.153	2.2375	0.138
σ_1	0.10	0.0874	0.012	0.0502	0.001
σ_2	0.02	0.0247	0.010	0.0547	0.006

Table 11.9: Estimates **a** and **s** of $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$ determined by the GMLE algorithm and associated uncertainties **u** for data generated with $\boldsymbol{\alpha} = (0.0, 1.0, 2.0)^{\mathrm{T}}$, $\boldsymbol{\sigma} = (0.10, 0.02)^{\mathrm{T}}$, prior estimates $\sigma_{k,0} = 0.05$, $k = 1, 2$, and weights 1) $u = v = 0.0001$ and 2) $u = v = 10000.0$.

		$u, v = 0.0001$		$u, v = 10000.0$	
		a, **s**	**u**	**a**, **s**	**u**
α_1	0.00	0.0142	0.008	0.0082	0.018
α_2	1.00	0.8899	0.078	0.9182	0.120
α_3	2.00	2.1537	0.119	2.1311	0.147
σ_1	0.02	0.0145	0.004	0.0499	0.001
σ_2	0.10	0.0997	0.010	0.0638	0.006

Table 11.10: Estimates **a** and **s** of $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$ — but for data generated with $\sigma_1 = 0.02$, $\sigma_2 = 0.10$.

	α_1	α_2	α_3	σ_1	σ_2
$\boldsymbol{\alpha}$, $\boldsymbol{\sigma}$	0.0000	1.0000	2.0000	0.1000	0.0200
$\bar{\mathbf{a}}$, $\bar{\mathbf{s}}$	0.0001	0.9990	2.0011	0.0977	0.0198
$s(\mathbf{a})$, $s(\mathbf{s})$	0.0311	0.1572	0.1618	0.0131	0.0107
$\bar{\mathbf{u}}$	0.0303	0.1541	0.1589	0.0128	0.0104
$\bar{\mathbf{a}}_{WLS}$	-0.0187	1.0085	1.9990		
$s(\mathbf{a}_{WLS})$	0.0341	0.1794	0.1871		

Table 11.11: Results of 5000 Monte Carlo trials comparing GMLE and WLS algorithms on datasets generated with $\sigma_1 = 0.10$ and $\sigma_2 = 0.02$.

	α_1	α_2	α_3	σ_1	σ_2
$\boldsymbol{\alpha}$, $\boldsymbol{\sigma}$	0.0000	1.0000	2.0000	0.0200	0.1000
$\bar{\mathbf{a}}$, $\bar{\mathbf{s}}$	0.0000	1.0005	1.9990	0.0185	0.0998
$s(\mathbf{a})$, $s(\mathbf{s})$	0.0092	0.0882	0.1307	0.0051	0.0113
$\bar{\mathbf{u}}$	0.0086	0.0849	0.1259	0.0047	0.0109
$\bar{\mathbf{a}}_{WLS}$	-0.0007	0.9942	1.9566		
$s(\mathbf{a}_{WLS})$	0.0120	0.1142	0.1609		

Table 11.12: Results of 5000 Monte Carlo trials comparing GMLE and WLS algorithms on datasets generated with $\sigma_1 = 0.02$ and $\sigma_2 = 0.10$.

Given a model of the form

$$\boldsymbol{\eta} = C\boldsymbol{\alpha}, \quad Y_i = \eta_i + E_i, \quad E_i \sim N(0, \sigma_i^2), \tag{11.16}$$

with σ_i known, the appropriate least-squares estimate $\mathbf{a}$ of $\boldsymbol{\alpha}$ is found by solving the weighted linear least squares problem

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{m} w_i^2 (y_i - \mathbf{c}_i^{\mathrm{T}} \boldsymbol{\alpha})^2, \tag{11.17}$$

with $w_i = 1/\sigma_i$. The difficulty with applying (11.16) to the problem formulated by (11.15) is that the standard deviations $\sigma_1 + \sigma_2 \eta_i$ depend on the unknowns $\boldsymbol{\alpha}$ through $\boldsymbol{\eta}$. However, we can use the observed y_i as an estimate of η_i and solve (11.17)

with $\mathbf{c}_i = (1, x_i, x_i^2)^{\mathrm{T}}$ and $w_i = 1/(\sigma_{1,0} + \sigma_{2,0} y_i)$ to provide a solution estimate $\mathbf{a}_{WLS}$.

For the N Monte Carlo trials we record estimates $\mathbf{a}_{WLS,q}$, sample mean $\bar{\mathbf{a}}_{WLS}$ and sample standard deviation $s(\mathbf{a}_{WLS})$. Tables 11.11 and 11.12 give the results for $N = 5000$ Monte Carlo trials for data generated with $\boldsymbol{\alpha} = (0.0, 1.0, 2.0)^{\mathrm{T}}$, $\sigma_{k,0} = 0.05$, $k = 1, 2$, $u = v = 0.0001$, and $\boldsymbol{\sigma} = (0.10, 0.02)^{\mathrm{T}}$ and $\boldsymbol{\sigma} = (0.02, 0.10)^{\mathrm{T}}$, respectively. The tables show i) the GMLE algorithm produces good estimates of both $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$, ii) the estimated uncertainties $\bar{\mathbf{u}}$ are in line with the sample standard deviations $s(\mathbf{a})$ and $s(\mathbf{s})$ and iii) on average, the GMLE algorithm produces better estimates of the parameters $\boldsymbol{\alpha}$ than the WLS algorithm.

For both types of dataset illustrated by figures 11.7 and 11.8, the data has provided sufficient information from which to provide point estimates of the parameters $\boldsymbol{\sigma}$. If we consider instead data as in figure 11.9, the fact that the responses η_i are approximately constant means that there is little information from which to determine both σ_1 and σ_2. Increasing σ_1 has the same effect as increasing σ_2, for example. For this dataset, the results corresponding to table 11.9 are presented in table 11.13. For the case of the weakly weighted prior information, the estimate $\mathbf{s}$ of $\boldsymbol{\sigma}$ differs significantly from the values used to generate the data but are consistent with the data. The correlation matrix associated with the estimates $\mathbf{s}$ of $\boldsymbol{\sigma}$ is

$$\begin{bmatrix} 1.0000 & -0.9874 \\ -0.9874 & 1.0000 \end{bmatrix}$$

showing that σ_1 is negatively correlated with σ_2.

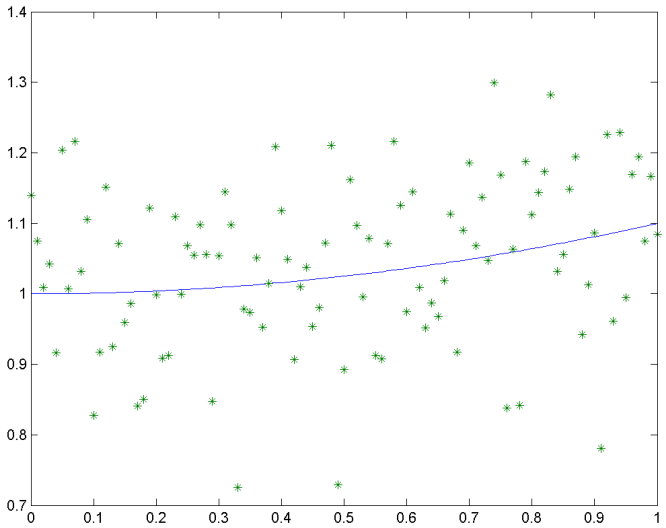

Figure 11.9: Data generated for a quadratic response and model (11.15) with $\boldsymbol{\alpha} = (1.0, 0.0, 0.1)^{\mathrm{T}}$ and $\boldsymbol{\sigma} = (0.10, 0.02)^{\mathrm{T}}$.

		$u, v = 0.0001$		$u, v = 10000.0$	
		a, **s**	**u**	**a**, **s**	**u**
α_1	1.00	1.0319	0.032	1.0330	0.031
α_2	0.00	-0.1340	0.149	-0.1362	0.146
α_3	0.10	0.2304	0.147	2.2319	0.143
σ_1	0.10	0.0011	0.050	0.0500	0.001
σ_2	0.02	0.1078	0.050	0.0571	0.005

Table 11.13: Estimates **a** and **s** of $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}$ determined by the GMLE algorithm and associated uncertainties **u** for data generated with $\boldsymbol{\alpha} = (1.0, 0.0, 0.1)^{\mathrm{T}}$, $\boldsymbol{\sigma} = (0.10, 0.02)^{\mathrm{T}}$, prior estimates $\sigma_{k,0} = 0.05$, $k = 1, 2$, and weights 1) $u = v = 0.0001$ and 2) $u = v = 10000.0$.

11.5 Fitting a Gaussian peak to data

This case study concerns fitting a Gaussian peak to data.

11.5.1 Functional and statistical model

A Gaussian peak can be defined by

$$y = A \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}, \tag{11.18}$$

in terms of three parameters, the peak height A, mean μ and standard deviation σ or, more generally, in terms of quadratic coefficients $\mathbf{a} = (a_1, a_2, a_3)^{\mathrm{T}}$, as

$$y = e^{a_1 + a_2 x + a_3 x^2}. \tag{11.19}$$

(These two models are not equivalent since (11.19) can represent curves corresponding to a negative σ^2.) It is assumed that measurements of the variable x are free from random effects but that measurements of y are subject to independent random effects drawn from a normal distribution with standard deviation σ:

$$y_i^* = e^{a_1 + a_2 x_i + a_3 x_i^2}, \quad y_i = y_i^* + \epsilon_i, \quad \epsilon_i \in N(0, \sigma^2).$$

11.5.2 Estimators

We consider three methods of determining the best-fit parameters to data points $\{(x_i, y_i)\}_1^m$.

The first is the nonlinear least-squares estimator (NLLS) that estimates $\mathbf{a}_N$ by solving the nonlinear least squares problem

$$\min_{\mathbf{a}} \sum_{i=1}^{m} (y_i - e^{a_1 + a_2 x_i + a_3 x_i^2})^2.$$

From maximum likelihood principles (section 3.4), we expect this estimator to perform well. This type of nonlinear optimisation problem can be solved iteratively by (variants of the) Gauss Newton algorithm (section 4.2.2). An estimate of the covariance matrix $V_{\mathbf{a}}^N$ of the fitted parameters is derived from the Jacobian matrix J evaluated at the solution. If J is the $m \times 3$ matrix

$$J_{ij} = \frac{\partial f_i}{\partial a_j}, \qquad f_i(\mathbf{a}) = y_i - e^{a_1 + a_2 x_i + a_3 x_i^2}, \quad i = 1, \ldots, m,$$

then

$$V_{\mathbf{a}}^N = \hat{\sigma}_N^2 (J^{\mathrm{T}} J)^{-1}, \tag{11.20}$$

where

$$\hat{\sigma}_N = \|\mathbf{f}\| / (m-3)^{1/2} = \left(\frac{\sum_i f_i^2}{m-3} \right)^{1/2}.$$

The second estimator arises from a transformation of (11.19). We note that if we take natural logarithms of both sides of this equation, we obtain

$$\log y = a_1 + a_2 x + a_3 x^2, \tag{11.21}$$

an expression linear in the parameters $\mathbf{a}$. From this transformation, we define the linear least-squares estimator (LLS) that determines $\mathbf{a}_L$ from the solution of

$$\min_{\mathbf{a}} \sum_{i=1}^{m} (\log y_i - a_1 - a_2 x_i - a_3 x_i^2)^2.$$

If C is the $m \times 3$ observation matrix with $(1, x_i, x_i^2)$ in the ith row and $\mathbf{z}$ is the m-vector with $\log y_i$ in the ith element, then $\mathbf{a}_L$ solves the matrix equation

$$C\mathbf{a}_L = \mathbf{z}$$

in the least-squares sense (section 4.1.2). An estimate of the covariance matrix of the fitted parameters is given by

$$V_{\mathbf{a}}^L = \hat{\sigma}_L^2 (C^{\mathrm{T}} C)^{-1}, \tag{11.22}$$

where

$$\hat{\sigma}_L = \|\mathbf{r}_L\|/(m-3)^{1/2}, \quad \mathbf{r}_L = \mathbf{z} - C\mathbf{a}_L.$$

The third estimator is the weighted linear least-squares estimator (WLLS) which defines $\mathbf{a}_W$ as the solution of

$$\min_{\mathbf{a}} \sum_{i=1}^{m} w_i^2 (\log y_i - a_1 - a_2 x_i - a_3 x_i^2)^2,$$

where $w_i \geq 0$ are specified weights (section 4.1). If $\tilde{\mathbf{z}}$ and $\tilde{C}$ are the weighted versions of $\mathbf{z}$ and C:

$$\tilde{z}_i = w_i z_i, \qquad \tilde{C}_{ij} = w_i C_{ij},$$

then $\mathbf{a}_W$ and $V_{\mathbf{a}_W}$ are estimated from

$$\tilde{C}\mathbf{a}_W = \tilde{\mathbf{z}}, \qquad V_{\mathbf{a}_W} = \hat{\sigma}_W^2 (\tilde{C}^T \tilde{C})^{-1}, \tag{11.23}$$

where, in this case,

$$\hat{\sigma}_W = \|\mathbf{r}_W\|/(m-3)^{1/2}, \quad \mathbf{r}_W = \tilde{\mathbf{z}} - \tilde{C}\mathbf{a}_W.$$

While, from maximum likelihood principles, the NLLS estimator is preferred, the LLS estimator is the easiest one to implement. We ask two questions

V1 Does the LLS estimator provide adequate solutions?

V2 Are the uncertainty estimates provided by (11.22) valid?

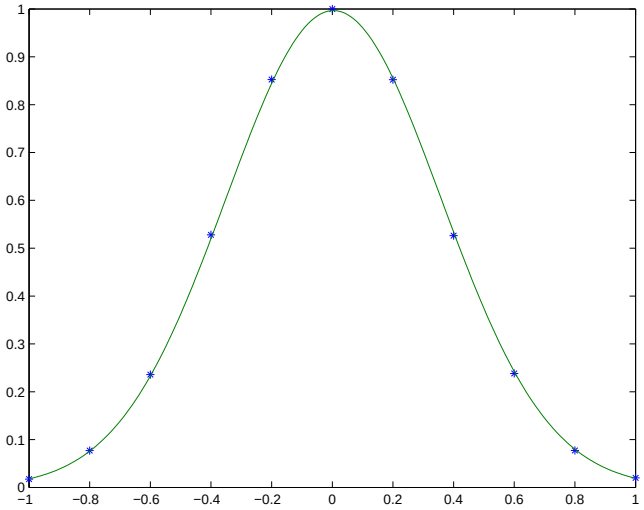

Figure 11.10: LLS fit of a Gaussian peak to simulated data.

11.5.3 Evaluation of LLS estimator using Monte Carlo simulations

Given typical values of the parameters $\mathbf{a}^*$ and a set of values $\mathbf{x} = (x_1, \ldots, x_m)^{\mathrm{T}}$ for the independent variable, we generate data according to the model:

$$y_i^* = e^{a_1^* + a_2^* x_i + a_3^* x_i^2}, \quad y_i = y_i^* + \epsilon_i, \quad \epsilon_i \in N(0, \sigma^2),$$

and then apply the estimator to the data. Below is an example data set

x	y*	y
-1.0000	0.0183	0.0175
-0.8000	0.0773	0.0770
-0.6000	0.2369	0.2359
-0.4000	0.5273	0.5279
-0.2000	0.8521	0.8524
0	1.0000	0.9999
0.2000	0.8521	0.8522
0.4000	0.5273	0.5263
0.6000	0.2369	0.2383
0.8000	0.0773	0.0772
1.0000	0.0183	0.0200

generated with $\mathbf{a}^* = (0, 0, -4)^{\mathrm{T}}$, i.e., for the curve $y = e^{-4x^2}$, with $\sigma = 0.001$. The solution estimates provided by LLS are $\mathbf{a} = (-0.0034, 0.0324, -3.9837)^{\mathrm{T}}$. From these estimates alone, it is not easy to say whether these are adequate estimates or not. Figure 11.10 shows the fitted curve and the data points $\{(x_i, y_i)\}$ — the fit looks reasonable. However, in figure 11.11 we plot the normalised residuals $\mathbf{r}/\sigma$ where

$$r_i = y_i - e^{a_1 + a_2 x_i + a_3 x_i^2}. \tag{11.24}$$

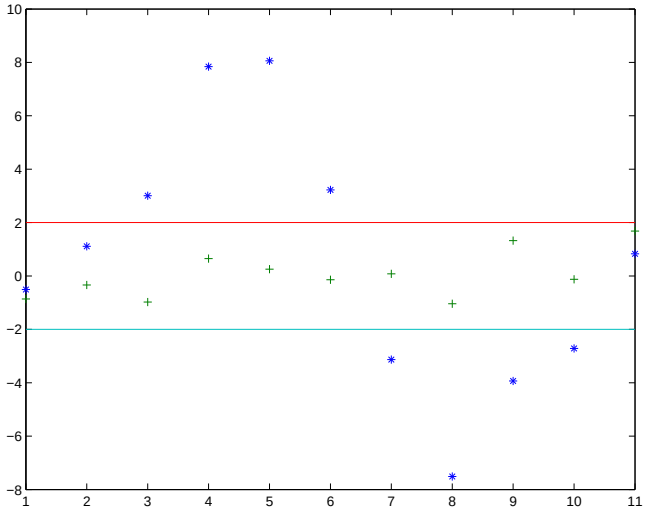

Figure 11.11: Residuals r_i '*' for the LLS fit of a Gaussian peak to simulated data compared with the perturbations ϵ_i '+'.

From this figure, we see that the residuals are much larger than we would expect and have a non-random behaviour. Instead of most lying in the band ± 2, 8 of the 11 are outside it with three residuals greater that 6σ in absolute value. From this view, we would say that the solution produced by the LLS estimator is poor. If we repeat the same experiment in Monte Carlo simulations, we can see if this behaviour is typical. For each of 5000 trials, we generate data sets $\mathbf{z}_q$, determine the best-fit parameters $\mathbf{a}_q$ and residual vector $\mathbf{r}_q = (r_{1,q}, \ldots, r_{m,q})^{\mathrm{T}}$ calculated as in (11.24). For each $\mathbf{r}_q$ we determine an estimate

$$\hat{\sigma}_q = \|\mathbf{r}_q\|/(m-3)^{1/2}.$$

Figure 11.12 is a histogram showing the spread of the normalised estimates $\hat{\sigma}_q/\sigma$ of the standard deviation of the residuals. This figure shows that the LLS consistently produces fits with residuals significantly larger than those expected from the value of σ used to generate the perturbations ϵ_i in the data. The answer to question V1 would appear to be 'No'.

The Monte Carlo simulations can also give an answer to question V2. For the data set $\mathbf{z}$, the covariance matrix $V_{\mathbf{a}}$ of the fitted parameters calculated according to (11.22) is

```
 1.2968e-004 -1.9486e-021 -1.8214e-004
-1.9486e-021  1.4207e-004 -2.2979e-021
-1.8214e-004 -2.2979e-021  4.5534e-004
```

while that $\bar{V}_{\mathbf{a}}$ derived from the estimates $\{\mathbf{a}_q\}_1^{5000}$ according to (8.1) is

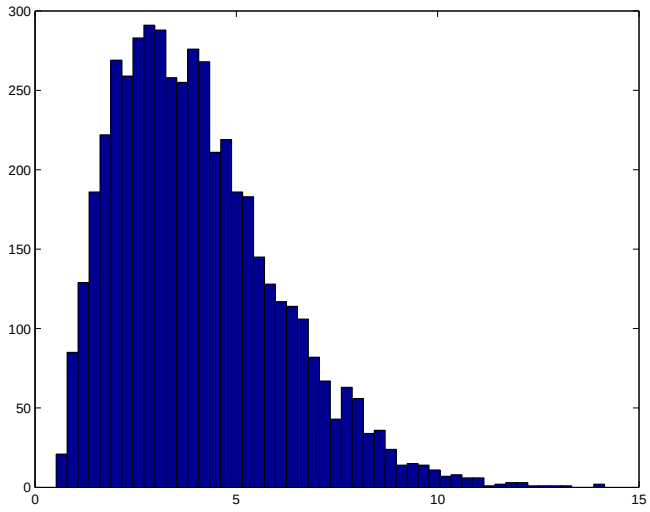

Figure 11.12: Spread of the normalised estimates $\hat{\sigma}_q/\sigma$ of the standard deviation of the residuals for the LLS fit of a Gaussian to data over 5000 Monte Carlo simulations.

```
 4.4171e-005  7.5385e-007 -2.2509e-004
 7.5385e-007  3.2608e-004 -3.0955e-006
-2.2509e-004 -3.0955e-006  1.1879e-003
```

The two sets of standard uncertainties $u(a_j)$ and $\bar{u}(a_j)$) of the fitted parameters (i.e., the square roots of the diagonal elements of the corresponding covariance matrix) are given in table 11.14. These results show that the uncertainty estimates derived from (11.22) are not reliable.

	$u(a_j)$	$\bar{u}(a_j)$
a_1	0.0114	0.0066
a_2	0.0119	0.0181
a_3	0.0213	0.0345

Table 11.14: Estimates of the standard uncertainties of the fitted parameters for the LLS estimator computed from (i) equation (11.22) (second column), (ii) 5000 Monte Carlo simulations (third column).

11.5.4 Null space benchmarking for the LLS estimator

The Monte Carlo simulations have shown that the LLS estimator provides poor estimates of the parameters and their uncertainties. However, we do not know how well this estimator compares with the preferred NLLS estimator. In this

section we use null space benchmarking (section 8.3) to answer the following questions:

V3 How much do LLS parameters estimates differ from NLLS estimates?

V4 How variable are LLS parameter estimates compared with NLLS estimates?

Given a choice of $\mathbf{a}^*$, σ and values of the independent variables x_i, $i = 1, \ldots, m$, the null space method for generating reference data for the NLLS estimator is summarised by:

I Calculate $y_i^* = e^{a_1^* + a_2^* x_i + a_3^* x_i^2}$ and set $\mathbf{z}^* = \{(x_i, y_i^*)\}$.

II Calculate the $m \times 3$ Jacobian matrix J^*:

$$J_{ij}^* = \frac{\partial}{\partial a_j}(y_i^* - e^{a_1^* + a_2^* x_i + a_3^* x_i^2}),$$

and an orthogonal basis for the null space $Q_2 = [\mathbf{q}_4 \ldots \mathbf{q}_m]$ of ${J^*}^{\mathrm{T}}$. Compute random multipliers $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{m-3})$ satisfying

$$\|\boldsymbol{\nu}\|/(m-3)^{1/2} = \sigma.$$

III Set

$$\left.\begin{array}{rcl} \boldsymbol{\delta} & = & (\delta_1, \ldots, \delta_m)^{\mathrm{T}} = \sum_{k=1}^{m-3} \nu_k \mathbf{q}_{3+k}, \\ y_i & = & y_i^* + \delta_i, \\ \mathbf{z}_{\boldsymbol{\delta}} & = & \{(x_i, y_i)\}_{i=1}^m. \end{array}\right\} \tag{11.25}$$

Then if σ is small enough, $\mathbf{a}^*$ are the NLLS best-fit model parameters to $\mathbf{z}_{\boldsymbol{\delta}}$ and $\boldsymbol{\delta}$ is the vector of residuals

$$\delta_i = y_i - e^{a_1^* + a_2^* x_i + a_3^* x_i^2}$$

and satisfies $\|\boldsymbol{\delta}\|/(m-3)^{1/2} = \sigma$. Furthermore,

$$V_{\mathbf{a}}^* = \sigma^2 ({J^*}^{\mathrm{T}} J^*)^{-1}, \tag{11.26}$$

is an estimate of the covariance matrix for the NLLS estimator.

In fact, the data set listed above was generated using the null space approach so that $\mathbf{a}^* = (0, 0, -4)^{\mathrm{T}}$ is the NLLS solution. The Jacobian matrix J^* for the exact data $\{(x_i, y_i^*)\}$ is

```
-0.0183    0.0183   -0.0183
-0.0773    0.0618   -0.0495
-0.2369    0.1422   -0.0853
-0.5273    0.2109   -0.0844
-0.8521    0.1704   -0.0341
-1.0000         0         0
-0.8521   -0.1704   -0.0341
-0.5273   -0.2109   -0.0844
-0.2369   -0.1422   -0.0853
-0.0773   -0.0618   -0.0495
-0.0183   -0.0183   -0.0183
```

	$u(a_j)$	$\bar{u}(a_j)$	$u^*(a_j)$	$\lvert a_j - a_j^*\rvert/u_j^*$
a_1	0.0114	0.0066	0.0007	4.8709
a_2	0.0119	0.0181	0.0023	14.3439
a_3	0.0213	0.0345	0.0064	2.5445

Table 11.15: Estimates of the standard uncertainties of the fitted parameters for the LLS estimator computed from (i) equation (11.22) (second column), (ii) 5000 Monte Carlo simulations (third column) along with estimates of the standard uncertainties for the NLSS estimator (fourth column). The fifth column shows the difference between the LLS estimate and NLLS estimate relative to the standard uncertainties $u_j^* = u^*(a_j)$.

from which we can calculate the covariance matrix $V_{\mathbf{a}}^*$

```
 4.7894e-007  5.6509e-023 -2.5569e-006
 5.6509e-023  5.1072e-006 -4.1411e-018
-2.5569e-006 -4.1411e-018  4.0917e-005
```

We can now compare the variation of the LLS estimates with those expected using the NLLS estimator. Table 11.15 shows the two sets of estimates $u(a_j)$ and $\bar{u}(a_j)$ of the standard uncertainties of the LLS fitted parameters (as in table 11.14) along with those, $u_j^* = u^*(a_j)$ for the NLLS derived from $V_{\mathbf{a}}^*$. It is seen that actual variation $\bar{u}(a_j)$ is almost an order of magnitude greater than that predicted for the NLLS estimates. The table also shows the difference between the LLS estimate and NLLS estimate relative to the standard uncertainties $u_j^* = u^*(a_j)$. On the basis of these results, we can say that the NLLS estimator will perform much better than the LLS estimator.

11.5.5 Analysis of the LLS estimator

From the large difference between the behaviour of the LLS and NLLS estimator indicated from the null space benchmarking, it is clear that the transformation (11.21) is producing a significant change in the model. In this section, we show from an analysis of the LLS estimator why this is so. At the same time, the analysis will indicate how to define a weighted least-squares estimator that can be expected to perform better.

A linear least-squares estimator can be expected to perform satisfactorily if the random effects associated with the data are uncorrelated and have equal variance (from the Gauss-Markov theorem, section 4.1.9). We ask, therefore,

V5 To what extent are the random effects associated with the model equations

$$\log y_i = \log(y_i^* + \epsilon_i), \qquad y_i^* = a_1 + a_2 x_i + a_3 x_i^2, \quad \epsilon_i \in N(0, \sigma^2),$$

correlated and of equal variance?

If the ϵ_i are independent of each other, then so are $\log y_i$. However, if $\epsilon_i \in N(0, \sigma^2)$, then the variance of $z_i = \log y_i$ is estimated by

$$\left(\frac{\partial z_i}{\partial y}\right)^2 \sigma^2 = \frac{\sigma^2}{y_i^2}. \tag{11.27}$$

Hence, the variances of $\log y_i$ are approximately constant only if the $\{y_i\}$ are approximately equal.

For the data set $\mathbf{z}$, the y_i's are not constant:

x	y	1/y	1/y^2
-1.0000e+000	1.7454e-002	5.7293e+001	3.2825e+003
-8.0000e-001	7.6968e-002	1.2992e+001	1.6880e+002
-6.0000e-001	2.3595e-001	4.2382e+000	1.7962e+001
-4.0000e-001	5.2794e-001	1.8941e+000	3.5878e+000
-2.0000e-001	8.5240e-001	1.1732e+000	1.3763e+000
0	9.9986e-001	1.0001e+000	1.0003e+000
2.0000e-001	8.5222e-001	1.1734e+000	1.3769e+000
4.0000e-001	5.2625e-001	1.9002e+000	3.6109e+000
6.0000e-001	2.3825e-001	4.1973e+000	1.7617e+001
8.0000e-001	7.7181e-002	1.2957e+001	1.6787e+002
1.0000e+000	1.9995e-002	5.0011e+001	2.5011e+003

Equation (11.27) indicates that with weights $w_i = y_i$, the conditions of the Gauss-Markov theorem will hold for the weighted least-squares estimator

$$\min_{\mathbf{a}} \sum_{i=1}^{m} y_i^2 (\log y_i - a_1 - a_2 x_i - a_3 x_i^2)^2, \tag{11.28}$$

and we would expect this estimator to perform better than the unweighted version.

11.5.6 Evaluation of WLLS estimator

We can repeat the validation process, this time for the WLLS estimator (11.28). Table 11.16 gives the results for 5000 Monte Carlo simulations in the same format as table 11.15. The results show (i), there is good agreement of the WLLS estimate with the NLLS estimate (column 5), (ii) the estimates $u(a_j)$ of the standard uncertainties derived from (11.23) are in good agreement with estimates $\bar{u}(a_j)$ determined in the Monte Carlo simulations and (iii) the variation in the WLLS estimates is close to the predicted variation $u^*(a_j)$ in the NLLS estimates. From this we can conclude that WLLS estimator is fit for purpose for this type of data. If the WLLS estimator is a good approximation to the NLLS estimator, then it follows that the LLS estimator will behave similarly to the weighted nonlinear least-squares estimator

$$\min_{\mathbf{a}} \sum_i \frac{1}{y_i^2} (y_i - e^{a_1 + a_2 x_i + a_3 x_i^2}).$$

This shows that the LLS estimator gives the largest weights to data points with the smallest y-values. From the values of $1/y_i$, it is seen that the first and last

	$u(a_j)$	$\bar{u}(a_j)$	$u^*(a_j)$	$\lvert a_j - a_j^*\rvert/u_j^*$
a_1	0.0007	0.0007	0.0007	0.0188
a_2	0.0023	0.0022	0.0023	-0.0093
a_3	0.0065	0.0064	0.0064	-0.0423

Table 11.16: Estimates of the standard uncertainties of the fitted parameters for the WLLS estimator computed from (i) equation (11.23) (second column), (ii) 5000 Monte Carlo simulations (third column) along with estimates of the standard uncertainties for the NLLS estimator (fourth column). The fifth column shows the difference between the WLLS estimate and NLLS estimate relative to the standard uncertainties $u_j^* = u^*(a_j)$.

data point are given a weight more that 50 times the weight attributed to middle data points. The LLS estimator is inefficient for this type of data because most of the data points are essentially ignored.

11.5.7 Evaluation of nonlinear effects using Monte Carlo simulations

Linearisation has so far entered into the calculations at two steps, firstly in the calculation of the covariance matrix for the NLLS estimator and secondly in the calculation of the variance of $\log y_i$. We therefore ask

V6 Is the estimate of the variance of $\log y_i$ determined in (11.27) valid?

V7 Are the estimates of $V_{\mathbf{a}}$ calculated in (11.20) valid?

To answer V6, we employ the following strategy.

I Given σ and a range of y_i, calculate $z_i = \log y_i$ and an estimate $u(z_i) = \sigma/y_i$ of the standard uncertainty of z_i.

II For each i and for $q = 1, \ldots, N$ calculate

$$y_{i,q} = y_i + \epsilon_{i,q}, \quad z_{i,q} = \log y_{i,q},$$

where $\epsilon_{i,q}$ is generated from a distribution with variance σ^2.

III For each i calculate the standard deviation $\bar{u}_i$ of $\{z_{i,q}\}_1^N$.

Table 11.17 summarises the results of $N = 10000$ Monte Carlo simulations with $\sigma = 0.001$ and perturbations drawn from rectangular (uniform) and normal distributions. For values of y far from zero relative to σ, there is good agreement between the predicted and actual variations. Differences become significant when y approaches σ in value.

The validity of the estimate $V_{\mathbf{a}}$ for the NLLS estimates can also be examined from Monte Carlo simulations. Below are the estimates $V_{\mathbf{a}}$ calculated from (11.20):

y_i	$z_i = \log y_i$	$u(z_i)$	$\bar{u}_n(z_i)$	$\bar{u}_r(z_i)$
0.0050	-5.2983	0.2000	0.2124	0.2061
0.0100	-4.6052	0.1000	0.1016	0.1009
0.0500	-2.9957	0.0200	0.0201	0.0200
0.1000	-2.3026	0.0100	0.0100	0.0100
0.5000	-0.6931	0.0020	0.0020	0.0020
1.0000	0	0.0010	0.0010	0.0010

Table 11.17: Comparison of the predicted standard deviation $\bar{u}$ of $\log y_i$ with actual standard deviations u_n and u_r generated by 10,000 Monte Carlo simulations with perturbations drawn from normal and rectangular distributions with standard deviation $\sigma = 0.001$.

```
 4.7894e-007  5.6509e-023 -2.5569e-006
 5.6509e-023  5.1072e-006 -4.1411e-018
-2.5569e-006 -4.1411e-018  4.0917e-005
```

and $\bar{V}_{\mathbf{a}}$ calculated in 5000 Monte Carlo simulations:

```
 4.7215e-007  1.8884e-008 -2.5406e-006
 1.8884e-008  5.0213e-006 -1.2392e-007
-2.5406e-006 -1.2392e-007  4.0844e-005
```

The two sets of standard uncertainties $u(a_j)$ and $\bar{u}(a_j)$ of the fitted parameters (i.e., the square roots of the diagonal elements of the corresponding covariance matrix) are given in table 11.18. The results show good agreement.

	$u(a_j)$	$\bar{u}(a_j)$
a_1	0.0007	0.0007
a_2	0.0023	0.0022
a_3	0.0064	0.0064

Table 11.18: Estimates of the standard uncertainties of the fitted parameters for the NLLS estimator computed from (i) equation (11.20) (second column), (ii) 5000 Monte Carlo simulations (third column).

11.5.8 Valid uncertainty estimates for the LLS estimator

While we have found that (11.22) does not produce valid estimates of the uncertainty in the fitted parameters for the LLS estimator, we can derive an alternative estimate that is valid. The solution of the linear least-squares problem

$$C\mathbf{a} = \mathbf{z}$$

is defined by

$$\mathbf{a} = C^{\dagger}\mathbf{z}, \quad C^{\dagger} = (C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}},$$

	$u(a_j)$	$\bar{u}(a_j)$
a_1	0.0064	0.0066
a_2	0.0176	0.0181
a_3	0.0334	0.0345

Table 11.19: Estimates of the standard uncertainties of the fitted parameters for the LLS estimator computed from (i) equation (11.29) (second column), (ii) 5000 Monte Carlo simulations (third column).

and the covariance matrix of the solution parameters is estimated by

$$V_{\mathbf{a}} = C^{\dagger} V_{\mathbf{z}} (C^{\dagger})^{\mathrm{T}} \tag{11.29}$$

where $V_{\mathbf{z}}$ is the diagonal matrix with σ^2/y_i^2 in the ith diagonal element. Table 11.19 compares the standard uncertainties $u(a_j)$ calculated from (11.29) and those $\bar{u}(a_j)$) calculated from the Monte Carlo simulations. The results show good agreement (certainly compared with those in table 11.14).

11.6 Estimation of the effective area of a pressure balance

An important step in the calibration of pressure balances is the estimation of the effective area of the piston-cylinder assembly. The effective area is a function of pressure and is usually modelled as a linear function involving two parameters. The calibration process therefore involves fitting a straight line to measurement data, at first sight, a straightforward process. However, in a typical cross-floating experiment in which a balance under test is hydrostatically compared with a reference balance, there is a large number of factors that need to be taken into account, including the uncertainties associated with (a) the calibration of the reference balance, and (b) the masses generating the applied loads. The fact that the same mass units are used in different combinations to produce different loads means that there is correlation amongst the random effects associated with the mass measurements. Furthermore, there are additional loads (or equivalents) that need to be measured or estimated from the data. In this case study we analyse a number of algorithms for determining estimates of the effective area parameters of a pressure balance from measurement data recorded during a calibration experiment.

11.6.1 Models

At temperature t with applied mass m (corrected for air buoyancy), a pressure balance generates a pressure p given implicitly by

$$p = \frac{(m+c)g}{A(p,\mathbf{a})(1+\phi(t))}, \tag{11.30}$$

where ϕ is a known function of t, $|\phi(t)| \ll 1$, that accounts for a temperature correction, c is a measured constant obtained from a precise characterisation of the balance, $A(p, \mathbf{a})$ describes the effective area of the balance in terms of the pressure p and calibration parameters $\mathbf{a}$, and g is gravitational acceleration. In practice, $A(p, \mathbf{a}) = a_1 + a_2 p$ with $a_1 = A_0$, $a_2 = A_0\lambda$, and $a_2 p$ small compared to a_1. Given a set of measurements p_i, m_i and t_i of pressure, applied load and temperature, and knowledge of c, calibrating a pressure balance means finding values for the parameters $\mathbf{a}$ that best-fit the model equations (11.30).

In a cross-float experiment [82], the pressure balance to be calibrated is connected to a reference balance whose effective area parameters $\mathbf{b}$ have previously been measured, and the applied loads on the two balances are adjusted so that both are in pressure and flow equilibrium. Suppose the reference balance generates a pressure given by (11.30) with m, c, $\mathbf{a}$ and $\phi(t)$ replaced by M, C, $\mathbf{b}$ and $\Phi((T))$, respectively. Then, when the balances are in equilibrium, we have

$$\frac{(m+c)g}{A(p,\mathbf{a})(1+\phi(t))} = \frac{(M+C)g}{A(p,\mathbf{b})(1+\Phi(T))}, \tag{11.31}$$

where the pressure p is estimated from the calibration of the reference balance.

11.6.2 Estimators

We consider three related estimators to determine the best-fit parameters $\mathbf{a}$ to measurement data [105].

The weighted linear least-squares estimator (WLLS) determines estimates of the parameters $\mathbf{a}$ by solving

$$\min_{\mathbf{a}} \sum_i w_i^2 f_i^2$$

where

$$f_i = \frac{(m_i + c)g}{1+\phi(t_i)} - a_1 p_i - a_2 p_i^2$$

a linear function of $\mathbf{a} = (a_1, a_2)^{\mathrm{T}}$. The P-estimator (PLLS) determines parameter estimates by solving

$$\min_{\mathbf{a}} \sum_i e_i^2$$

where

$$e_i = \left[\frac{m_i + c}{M_i + C}\right]\left[\frac{1+\Phi(T_i)}{1+\phi(t_i}\right] - \frac{a_1 + a_2 p_i}{b_1 + b_2 p_i}.$$

The ΔP-method (DLLS) determines parameter estimates by solving

$$\min_{\mathbf{a}} \sum_{i>1} d_i^2$$

where

$$d_i = \frac{(m_i - m_1)g}{[p_i - p_1 + (\phi(t_i) - \phi(t_1)p_1][1+\phi(t_i)]} - a_1 - a_2(p_i + p_1). \tag{11.32}$$

We ask the general validation question

V1 Under what circumstances are these estimators expected to perform well?

11.6.3 Analysis of estimator behaviour

All three estimators are linear least-squares estimators and so we can be guided by the Gauss-Markov theorem (section 4.1.9). Let

$$y_i = y(m_i, t_i) = \frac{(m_i + c)g}{1 + \phi(t_i)}.$$

The WLLS can be expect to perform well if (i) the pressures p_i are known accurately, (ii) the random effects associated with y_i are independent and (iii) the weights w_i are chosen to be inversely proportional to the standard deviation of y_i. Condition (ii) implies that both c and g are known accurately and the uncertainties in m_i and t_i are independent from those of m_j and t_j, $i \neq j$, for otherwise the uncertainties in y_i would be correlated. If we model the measurements of m_i and t_i as

$$m_i = m_i^* + \mu_i, \qquad t_i = t_i^* + \tau_i, \quad \mu_i \in N(0, \nu_i^2), \quad \tau_i \in N(0, \xi_i^2),$$

then the variance of y_i is estimated by

$$\sigma_i^2 = \left(\frac{\partial y}{\partial m}\right)^2 \nu_i^2 + \left(\frac{\partial y}{\partial t}\right)^2 \xi_i^2$$

where the partial derivatives (sensitivity coefficients) are evaluated at (m_i, t_i). If the weights w_i are set to $w_i = 1/\sigma_i$, then condition (iii) is satisfied.

If the pressures p_i are known accurately, then $e_i \approx f_i/p_i$ and so the PLLS estimator can be expect to work well under the same conditions as WLLS with the additional condition that (iv) the standard uncertainty associated with y_i are approximately proportional to p_i. Since, p_i is approximately proportional to m_i, this last condition will hold if the uncertainty associated with y_i is dominated by uncertainty associated with m_i and that the uncertainty associated with m_i is proportional to m_i.

The DLLS estimator is less obviously related to WLLS. The motivation for its definition is that the constant c is eliminated from the calculation and so c does not have to be estimated. We note that

$$a_1 + a_2(p_i + p_1) = \{a_1 p_i + a_2 p_i^2 - a_1 p_1 - a_2 p_1\} \times \frac{1}{p_i - p_1},$$

and the first term of the right-hand side of (11.32) is an approximation to

$$(y_i - y_1) \times \frac{1}{p_i - p_1}.$$

From this, we can argue that this estimator will perform well if the measurement of y_1 is known accurately and the standard deviation of the random effects of y_i, $i > 1$, is proportional to $p_i - p_1$. The results of Monte Carlo simulation verifying this analysis are presented in [105]. We note that by choosing different weighting strategies the WLLS estimator can approximate the behaviour of both PLLS and DLLS.

11.7 Circle fitting

11.7.1 Description

In this case study we are concerned with fitting a circle to data $\{(x_i, y_i)\}_1^m$. The case study uses this simple model to examine:

- parameterisation of the model space; see section 2.3,
- bias in nonlinear models,
- the effectiveness of different estimators on data generated according to different uncertainty models; see section 3.3.

11.7.2 Metrological area

Circle fitting is important in dimensional metrology where the assessment of roundness of an artefact is one of the fundamental tasks. It is also important in the use of automatic network analysers in electrical metrology.

11.7.3 Space of models

The space of models is the set of circles. We consider three parameterisations associated with the equations

$$\begin{aligned} \mathcal{C}_1 : &\quad (x-a_1)^2 + (y-a_2)^2 - a_3^2 = 0, \\ \mathcal{C}_2 : &\quad x^2 + y^2 + a_1 x + a_2 y + a_3 = 0, \\ \mathcal{C}_3 : &\quad a_1(x^2+y^2) + a_2 x + y + a_3 = 0. \end{aligned}$$

The parameterisations $\mathcal{C}_1$ and $\mathcal{C}_2$ are equivalent to each other in that they can represent exactly the same set of circles but not to $\mathcal{C}_3$. The parameterisation $\mathcal{C}_3$ can be used to model arcs of circles approximately parallel to the $x-$axis in a stable way. Parameterisations $\mathcal{C}_2$ and $\mathcal{C}_3$ are two of a number of parameterisations derived from the following general equation for a circle

$$l(x,y) = A(x^2+y^2) + Bx + Cy + D = 0. \tag{11.33}$$

(This equation cannot be used to parameterise the circle as replacing the coefficients with any non-zero multiple of them still defines the same circle. The parameterisations $\mathcal{C}_2$ and $\mathcal{C}_3$ are derived by setting $A = 1$ and $C = 1$, respectively. Other resolving constraints on A through D can be considered such as $A^2 + B^2 + C^2 + D^2 = 1$.)

We note that if (x_0, y_0) and r_0 are the circle centre coordinates and radius and r the distance from (x, y) to the circle centre, then

$$\begin{aligned} x_0 &= \frac{-B}{2A}, \quad y_0 = \frac{-C}{2A}, \\ r_0 &= \frac{V}{2|A|}, \quad r = \frac{U}{2|A|}, \end{aligned}$$

where in the above

$$\begin{aligned} U &= [(2Ax+B)^2 + (2Ay+C)^2]^{1/2}, \\ V &= [B^2 + C^2 - 4AD]^{1/2}. \end{aligned} \tag{11.34}$$

The signed distance d of (x, y) to the circle can be written as

$$\begin{aligned} d &= r - r_0, \\ &= 2l/(U+V), \\ &= \frac{2[A(x^2+y^2) + Bx + Cy + D]}{\sqrt{[(2Ax+B)^2 + (2Ay+C)^2]} + \sqrt{[B^2 + C^2 - 4AD]}}. \end{aligned} \tag{11.35}$$

The derivatives of d with respect to A, B, C and D are given by

$$\begin{aligned} \frac{\partial d}{\partial A} &= \frac{1}{UV}[(x^2+y^2)V - d(dV - 2D)], \\ \frac{\partial d}{\partial B} &= \frac{1}{UV}[xV - dB], \\ \frac{\partial d}{\partial C} &= \frac{1}{UV}[yV - dC], \end{aligned} \tag{11.36}$$

$$\frac{\partial d}{\partial C} = \frac{1}{V}. \tag{11.37}$$

11.7.4 Statistical models

We consider two types of statistical model.

M1 The measurements of x and y are together subject to uncorrelated Gaussian radial random effects: if the true point on the circle is

$$(x_i^*, y_i^*) = (a_1 + a_3 \cos t_i, a_2 + a_3 \sin t_i),$$

then

$$(x_i, y_i) = (a_1 + (a_3 + \epsilon_i)\cos t_i, a_1 + (a_3 + \epsilon_i)\sin t_i), \quad \epsilon_i \in N(0, \sigma^2).$$

M2 The measurements of x and y are each subject to uncorrelated Gaussian random effects: if the true point on the circle is (x_i^*, y_i^*), then

$$(x_i, y_i) = (x_i^* + \delta_i, y_i^* + \epsilon_i), \quad \delta_i, \epsilon_i \in N(0, \sigma^2).$$

11.7.5 Estimators

We consider three least-squares estimators of the form

$$\min_{\mathbf{a}} \sum_{i=1}^{m} f_i^2(\mathbf{a}).$$

E1 Least-squares ODR estimator with $f_i = d_i(\mathbf{a})$ where $d_i(\mathbf{a})$ is the orthogonal distance from the data point $\mathbf{x}_i$ to the circle specified by the parameters $\mathbf{a}$. For example, if parameterisation $\mathcal{C}_1$ is used, then

$$\begin{aligned} d_i &= r_i - a_3, \quad \text{where} \\ r_i &= ((x_i - a_1)^2 + (y_i - a_2)^2)^{1/2}. \end{aligned}$$

This estimator is optimal for model M1 and has favourable properties for model M2.

E2 Least-squares estimator associated with parameterisation $\mathcal{C}_2$ with

$$f_i(\mathbf{a}) = (x_i^2 + y_i^2) + a_1 x_i + a_2 y_i + a_3.$$

Note that if (x_0, y_0) and r_0 are the circle centre coordinates and radius and r_i is the distance from $\mathbf{x}_i$ to the circle centre, then

$$\begin{aligned} a_1 &= -2x_0, \\ a_2 &= -2y_0, \\ a_3 &= x_0^2 + y_0^2 - r_0^2, \quad \text{and} \\ f_i &= r_i^2 - r_0^2. \end{aligned}$$

E3 Least-squares estimator associated with parameterisation $\mathcal{C}_3$ with

$$f_i(\mathbf{a}) = a_1(x_i^2 + y_i^2) + a_2 x_i + y_i + a_3. \tag{11.38}$$

Again, f_i is linear in these parameters.

11.7.6 Estimator algorithms

Estimators E2 and E3 can be posed as linear least-squares problems

$$\min_{\mathbf{a}} \|\mathbf{z}_k - C_k \mathbf{a}\|,$$

where, for estimator E2,

$$C_2 = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & & \vdots \\ x_m & y_m & 1 \end{bmatrix}, \quad \mathbf{z}_2 = \begin{bmatrix} -x_1^2 - y_1^2 \\ \vdots \\ -x_m^2 - y_m^2 \end{bmatrix},$$

and, for estimator E3,

$$C_3 = \begin{bmatrix} x_1^2 + y_1^2 & x_1 & 1 \\ \vdots & & \vdots \\ x_m^2 + y_m^2 & x_m & 1 \end{bmatrix}, \quad \mathbf{z}_3 = \begin{bmatrix} -y_1 \\ \vdots \\ -y_m \end{bmatrix}.$$

Estimates produced by E1 are found by solving the nonlinear least-squares problem

$$\min_{\mathbf{a}} \sum_{i=1}^{m} d_i^2(\mathbf{a}),$$

using the Gauss-Newton algorithm (section 4.2), for example. To employ this algorithm we are required to evaluate d_i and its derivatives $\frac{\partial d_i}{\partial a_j}$. If parameterisation $\mathcal{C}_1$ is used, then

$$d_i = r_i - a_3, \quad r_i = \sqrt{[(x_i - a_1)^2 + (y_i - a_2)^2]}$$

and

$$\begin{aligned} \frac{\partial d_i}{\partial a_1} &= -(x_i - a_1)/r_i, \\ \frac{\partial d_i}{\partial a_2} &= -(y_i - a_2)/r_i, \\ \frac{\partial d_i}{\partial a_3} &= -1. \end{aligned}$$

If parameterisation $\mathcal{C}_3$ is used then d_i and its derivatives can be evaluated as in (11.35) and (11.36) with $(A, B, C, D) = (a_1, a_2, 1, a_3)$.

11.7.7 Monte Carlo data generation

In this case study, we generate two types of data:

D1 Data points $\{(x_i, y_i)\}$ nominally lying uniformly around a circle centred at $(0, 0)$ with radius 1. Figure 11.13 shows an example data set.

D2 Data points $\{(x_i, y_i)\}$ lying on the arc $y = y_0 - (1 + y_0^2 - x^2)^{1/2}$, $|x| < 1$, of a circle centred at $(0, y_0)$ and passing through $(\pm 1, 0)$. Figure 11.14 shows an example data set.

For each circle, we generate data points (x_i, y_i) lying on the circle and N replicated data sets

$$\mathbf{z}_q = \{(x_{q,i}, y_{q,i})\}_i = \{(x_i, y_i) + (\delta_{q,i}, \epsilon_{q,i})\}_i$$

where $\delta_{q,i}$ and $\epsilon_{q,i}$ are generated according to the appropriate statistical model.

11.7.8 Estimator assessment

We apply each estimator to each data set to determine best-fit parameters $\mathbf{a}_{k,q} = \mathcal{A}_k(\mathbf{z}_q)$ and then look at the mean value $\bar{a}_{j,k}$ and standard deviation $s_{k,j}$ of the parameter estimates, $k = 1, 2, 3$, $j = 1, 2, 3$.

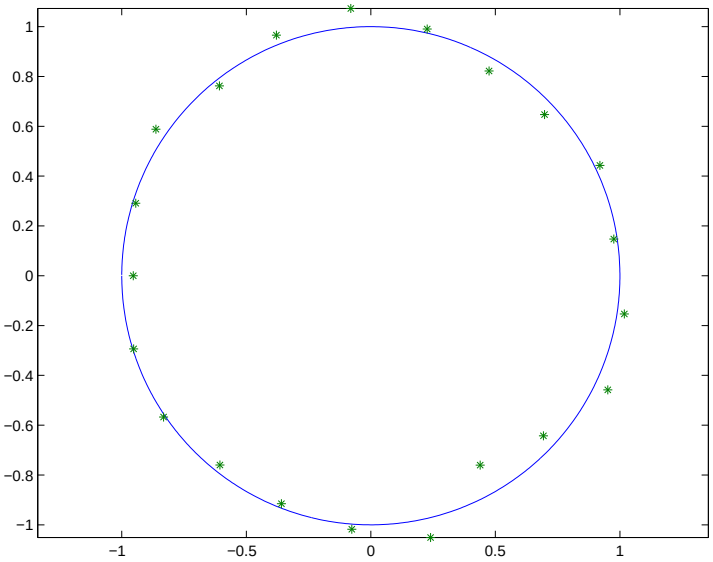

Figure 11.13: Data uniformly distributed around a circle of radius 1.

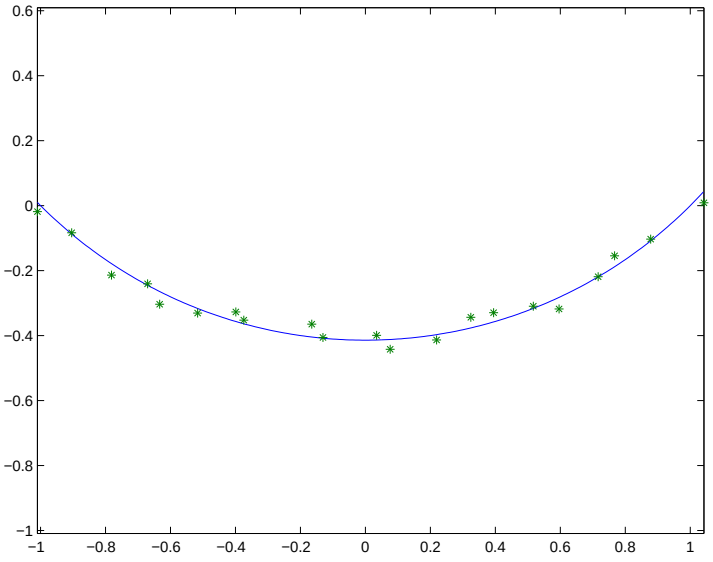

Figure 11.14: Data generated on an arc of a circle passing through $(\pm 1, 0)$.

11.7.9 Experiment 1: data uniformly distributed around a circle

In this experiment, we consider the behaviour of estimators E1 and E2 for data uniformly distributed around a circle (D1) according to models M1 and M2. For this type of data, we expect E1 to be unbiased for M1 but produce biased estimates of the radius for M2. We expect E2 to provide biased estimates of the radius for both models. The reasons for these expectations are as follows. Let (x, y) be a point on the unit circle centred at the origin and consider perturbations according to model M1:

$$x_i = x(1 + \epsilon_i), \quad y_i = y(1 + \epsilon_i), \quad \epsilon_i \in N(0, \sigma^2).$$

Then

$$r_i - 1 = (1 + \epsilon_i)(x^2 + y^2)^{1/2} - 1 = \epsilon_i,$$

so that the expected estimate of the radius produced by E1 is 1 (since the expected value of ϵ_i is 0). On the other hand,

$$r_i^2 - 1 = (1 + \epsilon_i)^2 - 1 = 2\epsilon_i + \epsilon_i^2.$$

Since the expected value of ϵ_i^2 is greater than zero, E2 will produce estimates of the radius that are greater than the true radius for this type of data.

For data generated according to model M2, we have

$$x_i = x + \delta_i, \quad y_i = y + \epsilon_i, \quad \delta_i, \epsilon_i \in N(0, \sigma^2),$$

and

$$r_i - 1 = (1 + 2\delta_i x + 2\epsilon_i y + \delta_i^2 + \epsilon_i^2)^{1/2} - 1 \approx \frac{1}{2}[2\delta_i x + 2\epsilon_i y + \delta_i^2 + \epsilon_i^2]. \quad (11.39)$$

In this case, the presence of the term $\delta_i^2 + \epsilon_i^2$ means that the expected value of the radius produced by E1 is greater than 1. Similarly,

$$r_i^2 - 1 = 2\delta_i x + 2\epsilon_i y + \delta_i^2 + \epsilon_i^2, \quad (11.40)$$

so that its expected value of the radius produced by E2 is also greater than 1.

Tables 11.20 and 11.21 show the results of applying both estimators (E1 *and* E2) to data generated according to models M1 and M2, respectively. Each data set $\mathbf{z}_q = \mathbf{z} + \Delta_q$ represents the perturbation of a fixed data set $\mathbf{z} = \{(x_i, y_i)\}_1^{21}$ of 21 data points uniformly distributed around the circle centred at $(x_0, y_0) = (0, 0)$ with radius $r_0 = 1$. Three sizes of perturbations were used: $\sigma = 0.001,\ 0.01,\ 0.1$. For each trial, 5000 data sets were generated.

We apply each estimator to each data set to determine best-fit parameters $\mathbf{a}_q = \mathcal{A}(\mathbf{z}_q)$. For each parameter a_j, we calculate the mean value $\bar{a}_j$ of the parameter estimates, the difference $b_j = \bar{a}_j - a_j^*$ between the mean value and the true value a_j^*, the standard deviation s_j of the parameter estimates $\{a_{j,q}\}$, and the ratio b_j/s_j, $j = 1, 2, 3$.

We note the following:

M1	$b_j = \bar{a}_j - a_j^*$	$s_j = \mathrm{std}_q\{a_{j,q}\}$	b_j/s_j
E1	$\sigma = 0.001$		
x_0	6.4094e-007	3.0872e-004	2.0761e-003
y_0	-1.1645e-006	3.0962e-004	-3.7610e-003
r_0	-3.1300e-006	2.1746e-004	-1.4393e-002
E2	$\sigma = 0.001$		
x_0	6.5031e-007	3.0874e-004	2.1064e-003
y_0	-1.1587e-006	3.0962e-004	-3.7424e-003
r_0	-2.6998e-006	2.1746e-004	-1.2415e-002
E1	$\sigma = 0.01$		
x_0	6.4380e-006	3.0875e-003	2.0852e-003
y_0	-1.1667e-005	3.0966e-003	-3.7677e-003
r_0	-2.6998e-005	2.1746e-003	-1.2415e-002
E2	$\sigma = 0.01$		
x_0	7.3802e-006	3.0890e-003	2.3892e-003
y_0	-1.1089e-005	3.0962e-003	-3.5815e-003
r_0	1.6022e-005	2.1745e-003	7.3679e-003
E1	$\sigma = 0.1$		
x_0	6.9282e-005	3.1156e-002	2.2237e-003
y_0	-1.2031e-004	3.1251e-002	-3.8497e-003
r_0	1.7384e-004	2.1741e-002	7.9961e-003
E2	$\sigma = 0.1$		
x_0	1.6639e-004	3.1508e-002	5.2809e-003
y_0	-6.4161e-005	3.1415e-002	-2.0424e-003
r_0	4.4870e-003	2.1691e-002	2.0686e-001

Table 11.20: Estimates of the bias and efficiency of estimators E1 and E2 for data generated according to model M1.

M2	$b_j = \bar{a}_j - a_j^*$	$s_j = \mathrm{std}_q\{a_{j,q}\}$	b_j/s_j
E1		$\sigma = 0.001$	
x_0	-3.5046e-006	3.0800e-004	-1.1379e-002
y_0	-4.7664e-006	3.0712e-004	-1.5519e-002
r_0	-2.8705e-006	2.1901e-004	-1.3107e-002
E2		$\sigma = 0.001$	
x_0	-3.4935e-006	3.0800e-004	-1.1343e-002
y_0	-4.7681e-006	3.0713e-004	-1.5524e-002
r_0	-2.4441e-006	2.1901e-004	-1.1160e-002
E1		$\sigma = 0.01$	
x_0	-3.4986e-005	3.0794e-003	-1.1361e-002
y_0	-4.8097e-005	3.0705e-003	-1.5664e-002
r_0	2.0558e-005	2.1902e-003	9.3866e-003
E2		$\sigma = 0.01$	
x_0	-3.3886e-005	3.0795e-003	-1.1004e-002
y_0	-4.8260e-005	3.0715e-003	-1.5712e-002
r_0	6.3196e-005	2.1902e-003	2.8854e-002
E1		$\sigma = 0.1$	
x_0	-3.3628e-004	3.0957e-002	-1.0863e-002
y_0	-5.1712e-004	3.0828e-002	-1.6774e-002
r_0	5.1554e-003	2.1867e-002	2.3576e-001
E2		$\sigma = 0.1$	
x_0	-2.4281e-004	3.1167e-002	-7.7906e-003
y_0	-5.3501e-004	3.1140e-002	-1.7181e-002
r_0	9.3823e-003	2.1828e-002	4.2982e-001

Table 11.21: Estimates of the bias and efficiency of estimators E1 and E2 for data generated according to model M2.

- For all experiments, the estimates of the centre coordinates (x_0, y_0) is unbiased. The ratio of the bias to the standard deviation is of the order of 0.01.
- For model M1 (table 11.20), estimator E2 shows a significant bias in the estimate of the radius in the case $\sigma = 0.1$. No such bias is detected for estimator E1.
- For model M2 (table 11.21), both estimators show significant bias in the estimate of the radius in the case $\sigma = 0.1$ (with the bias for E2 approximately double that for E1; see equations (11.39), (11.40)).

Although the numerical experiments show that the bias does exist, it can be argued that the bias is only apparent for noisy data ($\sigma = 0.1$) and that the bias is only of the order of σ^2. However, if the data is noisy, we often take more data points so that the extra redundancy reduces the uncertainty in the parameter estimates. Table 11.22 shows the results of repeating the experiment for model M2 with $\sigma = 0.1$ but with each data set having 201 data points instead of 21. This results show that the standard deviations are reduced by a factor of $3 \approx (201/21)^{1/2}$ but the bias remains the same, so that the bias as a percentage of the standard deviation increases by a factor of 3.

Taking this further, we have fitted a circle using estimator E1 to 20,001 data points generated according to M2 with $\sigma = 0.1$ and estimated the covariance matrix $V_{\mathbf{a}}$ of the fitted parameters according to

$$V_{\mathbf{a}} = \hat{\sigma}^2 (J^{\mathrm{T}} J)^{-1}, \qquad \hat{\sigma} = \|\mathbf{f}\|/(m-n)^{1/2},$$

where J is the Jacobian matrix and $\mathbf{f}$ the vector of function values at the solution (section 4.2). We obtain estimates

$$\mathbf{a} = (4.7204e-004, -6.9506e-004, 1.0047e+000)^{\mathrm{T}},$$

$\hat{\sigma} = 1.0006e-001$ and

$$V_{\mathbf{a}} = \begin{bmatrix} 1.0004e-006 & -5.3485e-010 & -2.1129e-011 \\ -5.3485e-010 & 1.0019e-006 & 3.9490e-010 \\ -2.1129e-011 & 3.9490e-010 & 5.0056e-007 \end{bmatrix}.$$

The standard uncertainties $\mathbf{u} = (u_1, u_2, u_3)^{\mathrm{T}}$ of the fitted parameters are the square roots of the diagonal elements of $V_{\mathbf{a}}$ so that

$$\mathbf{u} = (1.0002e-003, 1.0009e-003, 7.0750e-004)^{\mathrm{T}}.$$

Table 11.23 shows the confidence intervals of the form $[a_j - ku_j, a_j + ku_j]$, $k = 2$, based on these estimates. It shows that intervals for the circle centre coordinates (x_0, y_0) contain the true values but that for the radius r_0 does not. In fact, k has to be expanded to nearly 7 for the true value to be included.

The estimates of the covariance matrix and the standard uncertainties are valid and are confirmed by Monte Carlo simulations.

M2	$b_j = \bar{a}_j - a_j^*$	$s_j = \mathrm{std}_q\{a_{j,q}\}$	b_j/s_j
E1	$\sigma = 0.1$		
x_0	-2.0745e-004	9.9786e-003	-2.0790e-002
y_0	1.1825e-004	1.0189e-002	1.1605e-002
r_0	4.8934e-003	7.0501e-003	6.9409e-001
E2	$\sigma = 0.1$		
x_0	-1.5901e-004	1.0153e-002	-1.5661e-002
y_0	1.0889e-004	1.0244e-002	1.0630e-002
r_0	9.7519e-003	7.0383e-003	1.3855e+000

Table 11.22: Estimates of the bias and efficiency of estimators E1 and E2 for data generated according to model M2. Each data set has 201 data points.

M2	$a_j - 2u_j$	$a_j + 2u_j$
x_0	-1.5283e-003	2.4724e-003
y_0	-2.6969e-003	1.3068e-003
r_0	1.0033e+000	1.0061e+000

Table 11.23: Confidence intervals ($k = 2$) for the circle parameters for the best-fit circle to 20,001 data points using estimator E1 for data generated according to model M2.

This example illustrates the fact that for biased estimators, an analysis of the uncertainty of the fitted parameters based only on the covariance matrix of the fitted parameters can be misleading. For the case of circle fitting, the problem is simple enough to allow for a complete analysis. For more complicated models, we are unlikely to be able to quantify the extent of the bias in an estimator from analysis alone. In these situations, Monte Carlo simulations provide a practical method of analysing the behaviour of an estimator.

11.7.10 Experiment 2: data uniformly distributed on an arc of a circle

In this experiment, we consider the behaviour of all three estimators for data generated on an arc of a circle (D2) according to model M2 (figure 11.14). For this type of data we expect E1 and E3 to be relatively unbiased. This is because i) for this case models M1 and M2 produce almost the same type of data and E1 is unbiased for M1 and ii), E3 produces almost the same estimates as E1 since f_i in (11.38) is approximately the distance to circle. (If $y_0 \gg 1$, the arc of the circle approximates the line $y = 0$ and in the equation (11.33) for this circle $A, B, D \approx 0$. If follows from (11.35) that $f_i \approx d_i$.) However, we expect estimator E2 to produce biased estimates of the y_0-co-ordinate and the circle radius since

$$f_i = r_i^2 - r_0^2 = (r_i - r_0)(r_i + r_0) = d_i(r_i + r_0).$$

Since for $y_0 \gg 1$, decreasing y_0 and r_0 leaves $r_i - r_0$ relatively unchanged but reduces $r_i + r_0$, the estimates are biased to producing smaller radius.

M2	$b_j = \bar{a}_j - a_j^*$	$s_j = \mathrm{std}_q\{a_{j,q}\}$	b_j/s_j
E1	$\sigma = 0.0001$		
x_0	5.8940e-005	3.5956e-003	1.6392e-002
y_0	4.9218e-002	1.3473e+000	3.6529e-002
r_0	4.9217e-002	1.3473e+000	3.6530e-002
E2	$\sigma = 0.0001$		
x_0	5.8823e-005	3.5842e-003	1.6412e-002
y_0	-2.7113e-001	1.3388e+000	-2.0251e-001
r_0	-2.7112e-001	1.3388e+000	-2.0251e-001
E3	$\sigma = 0.0001$		
x_0	5.8940e-005	3.5956e-003	1.6392e-002
y_0	4.9239e-002	1.3473e+000	3.6545e-002
r_0	4.9238e-002	1.3473e+000	3.6546e-002
E1	$\sigma = 0.001$		
x_0	5.4955e-004	3.7073e-002	1.4824e-002
y_0	2.2453e+000	1.4641e+001	1.5336e-001
r_0	2.2453e+000	1.4641e+001	1.5336e-001
E2	$\sigma = 0.001$		
x_0	4.5515e-004	2.7664e-002	1.6453e-002
y_0	-2.3631e+001	8.4045e+000	-2.8117e+000
r_0	-2.3630e+001	8.4043e+000	-2.8117e+000
E3	$\sigma = 0.001$		
x_0	5.4956e-004	3.7074e-002	1.4823e-002
y_0	2.2475e+000	1.4641e+001	1.5350e-001
r_0	2.2475e+000	1.4641e+001	1.5351e-001

Table 11.24: Estimates of the bias and efficiency of estimators E1, E2 and E3 for data generated according to model M2. Each data set $\mathbf{z}_q$ had 21 data points.

Table 11.24 shows the results of applying estimators E1, E2 and E3 to data generated according to model M2. Each data set $\mathbf{z}_q = \mathbf{z} + \Delta_q$ represents the perturbation of a fixed data set $\mathbf{z} = \{(x_i, y_i)\}_1^{21}$ of 21 data points uniformly distributed on the arc from $(-1, 0)$ to $(1, 0)$ of the circle centred at $(x_0, y_0) = (0, 100)$. Two sizes of perturbations were used: $\sigma = 0.0001,\ 0.001$. For each trial, 5000 data sets were generated. Table 11.25 shows the results of repeating the calculation for $\sigma = 0.0001$ but for data sets with 2001 data points.

We apply each estimator to each data set to determine best-fit parameters $\mathbf{a}_q = \mathcal{A}(\mathbf{z}_q)$. As above, for each parameter a_j, we calculate the mean value $\bar{a}_j$, the difference $b_j = \bar{a}_j - a_j^*$, the standard deviation s_j, and the ratio b_j/s_j, $j = 1, 2, 3$.

We note the following:

- For all experiments, the estimates of the parameters produced by estimators E1 and E3 are relatively unbiased. E1 and E3 show substantially the same behaviour.
- Estimator E2 gives biased estimates of the y-co-ordinate of the circle centre and radius r_0 as expected. The bias relative to the standard deviation is

M2	$b_j = \bar{a}_j - a_j^*$	$s_j = \mathrm{std}_q\{a_{j,q}\}$	b_j/s_j
E1	$\sigma = 0.0001$		
x_0	-4.3570e-006	3.8809e-004	-1.1227e-002
y_0	8.1324e-004	1.4732e-001	5.5202e-003
r_0	8.1331e-004	1.4732e-001	5.5208e-003
E2	$\sigma = 0.0001$		
x_0	-4.3372e-006	3.8635e-004	-1.1226e-002
y_0	-4.4556e-001	1.4611e-001	-3.0495e+000
r_0	-4.4555e-001	1.4611e-001	-3.0495e+000
E3	$\sigma = 0.0001$		
x_0	-4.3570e-006	3.8809e-004	-1.1227e-002
y_0	8.4313e-004	1.4732e-001	5.7231e-003
r_0	8.4320e-004	1.4732e-001	5.7237e-003

Table 11.25: Estimates of the bias and efficiency of estimators E1, E2 and E3 for data generated according to model M2. Each data set $\mathbf{z}_q$ had 2001 data points.

large for $\sigma = 0.001$ or if there are a large number of data points.

While in the previous set of experiments, the bias in the estimators appeared only for noisy data, the bias associated with E2 appears for reasonably accurate data. Figure 11.15 gives a graphical example of the difference in the fits produced by E1 and E2.

11.7.11 Comparison of parameterisations for arc data

Using parameterisation $\mathcal{C}_3$ (and related parameterisations) is a much more stable approach for dealing with arc data than $\mathcal{C}_1$ or $\mathcal{C}_2$. We can see this by examining the condition of the Jacobian matrices generated using the parameterisations (section 3.7). For parameterisation $\mathcal{C}_1$, the ith row of the associated Jacobian matrix J_1 is $[\ c_i\ s_i\ 1\ ]$ where

$$c_i = \cos t_i = \frac{x_i - a_1}{r_i}, \quad s_i = \sin t_i = \frac{y_i - a_2}{r_i}.$$

For the data generated on an arc (D2), $s_i \approx 1$ so that angle between the second and third columns of J_q is small and gets smaller as a_2 gets larger.

For parameterisation $\mathcal{C}_3$, and this type of data the ith row of the associated Jacobian matrix J_3 is approximated by $[\ x_i^2\ x_i\ 1\ ]$ where $-1 \leq x_i \leq 1$ and is well conditioned. For the data sets here, the condition of J_1 is of the order of 10^5, while the condition of J_3 is less than 10. As the radius increases, the condition of J_1 worsens to the extent that the optimisation algorithm is likely to break down. There is no problem using parameterisation $\mathcal{C}_3$. On the other hand, parameterisation $\mathcal{C}_3$ is unsuitable for data representing a complete circle.

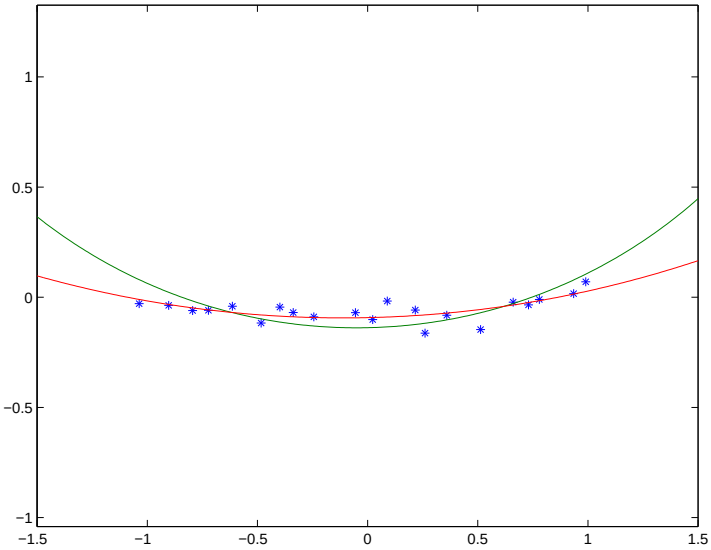

Figure 11.15: Fits produced by estimators E1 (larger radius) and E2 (smaller radius) to arc data.

This example illustrates the more general point that models can have different parameterisations with different characteristics, some best for one type of data, others suitable for a different type. Often a model is deemed inappropriate because the wrong parameterisation is used when in fact the model is perfectly satisfactory if an appropriate parameterisation is employed. Unfortunately, determining a good parameterisation of a model for a given experimental situation is often not straightforward.

11.7.12 Bibliography

The use of different estimators and parameterisations for circles and spheres is discussed in [7, 77, 97]. The numerical properties of different circle fitting algorithms are also considered in [65]. For more general issues in model parameterisation, see [103].

11.8 Circle fitting and roundness assessment

This case study is a continuation of the circle fitting case study in section 11.7. We are concerned with fitting a circle to data points $\{\mathbf{x}_i = (x_i, y_i)\}_{i=1}^m$. We assume that the measurements of x and y are each subject to uncorrelated normally distributed random effects: if the true point on the circle is (x_i^*, y_i^*),

then

$$(x_i, y_i) = (x_i^* + \delta_i, y_i^* + \epsilon_i), \quad \delta_i, \epsilon_i \in N(0, \sigma^2). \tag{11.41}$$

We consider two estimators. The first is the nonlinear least-squares estimator (NLLS) that solves

$$\min_{\mathbf{a}} \sum_{i=1}^{m} d_i^2(\mathbf{a}) \tag{11.42}$$

where $d_i(\mathbf{a})$ is the orthogonal distance from the data point $\mathbf{x}_i = (x_i, y_i)$ to the circle specified by the parameters $\mathbf{a}$. For example, if the circle is specified by centre co-ordinates (a_1, a_2) and radius a_3, then

$$\begin{aligned} d_i &= r_i - a_3, \quad \text{where} \\ r_i &= ((x_i - a_1)^2 + (y_i - a_2)^2)^{1/2}. \end{aligned} \tag{11.43}$$

NLLS estimates can be determined using the Gauss-Newton algorithm (section 4.2). If J is the Jacobian matrix defined by

$$J_{ij} = \frac{\partial d_i}{\partial a_j}$$

evaluated at the solution, then an estimate of the covariance matrix of the fitted parameter is given by

$$\hat{V}_{\mathbf{a}}^N = \hat{\sigma}_N^2 (J^{\mathrm{T}} J)^{-1}, \tag{11.44}$$

where

$$\hat{\sigma}_N = \|\mathbf{d}\|/(m-3)^{1/2}.$$

We also consider

$$V_{\mathbf{a}}^N = \sigma^2 (J^{\mathrm{T}} J)^{-1} \tag{11.45}$$

calculated with the prior value of the standard deviation.

The second is the linear least-squares estimator (LLS)

$$\min_{\mathbf{a}} \sum_{i=1}^{m} (y_i + a_1(x_i^2 + y_i^2) + a_2 x_i + a_3)^2. \tag{11.46}$$

This estimator is derived from the implicit equation

$$y + a_1(x^2 + y^2) + a_2 x + a_3 = 0 \tag{11.47}$$

that can describe an arc of a circle approximately parallel with the x-axis. We can convert from these parameters to the circle centre co-ordinates (x_0, y_0) and radius r_0 using the equations

$$x_0 = -\frac{a_2}{2a_1}, \quad y_0 = -\frac{1}{2a_1}, \quad r_0 = \frac{(1 + a_2^2 - 4a_1 a_3)^{1/2}}{2|a_1|}. \tag{11.48}$$

If C is the $m \times 3$ observation matrix with ith row equal to $(x_i^2 + y_i^2, x_i, 1)$ and $\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$, then LLS estimates are found by solving the linear least-squares problem

$$C\mathbf{a} = -\mathbf{y}.$$

An estimate of the covariance matrix of fitted parameters is given by

$$V_{\mathbf{a}}^L = \hat{\sigma}_L^2 (C^{\mathrm{T}} C)^{-1}, \tag{11.49}$$

where

$$\hat{\sigma}_L = \|\mathbf{r}_L\|/(m-3)^{1/2}, \quad \mathbf{r}_L = \mathbf{y} + C\mathbf{a}.$$

We also can calculate

$$V_{\mathbf{a}}^L = \sigma^2 (C^{\mathrm{T}} C)^{-1}. \tag{11.50}$$

We are interested in how these estimators behave for two types of data, the first with data points approximately uniformly spaced around the circle as in figure 11.13, the second with data points on a small arc of the circle (figure 11.14). It was shown in section 11.7 that the NLLS provided appropriate estimates of the circle parameters for both types of data and the LLS provided appropriate estimates only for arc data. Here we ask the question:

V1 Are the estimates of the covariance matrices given in (11.45) and (11.50) valid?

We use Monte Carlo simulation to test these estimates.

In the first set of experiments we apply NLLS to data sets of 11 points uniformly distributed around a circle centred at the origin $(0, 0)$ with radius 1. The data sets $\mathbf{z}_q = \{(x_{i,q}, y_{i,q})\}_{i=1}^m$ are generated from exact data (x_i^*, y_i^*) lying on the circle, and then perturbed according to the model

$$(x_{i,q}, y_{i,q}) = (x_i^* + \delta_{i,q}, y_i^* + \epsilon_{i,q}), \quad \delta_{i,q},\ \epsilon_{i,q} \in N(0, \sigma^2).$$

a_j	u	$\hat{u}$	$\bar{u}$
E1	$\sigma = 0.001$		
a_1	4.2640e-004	4.2656e-004	4.3465e-004
a_2	4.2640e-004	4.2656e-004	4.2743e-004
a_3	3.0151e-004	3.0162e-004	3.0263e-004
E2	$\sigma = 0.01$		
a_1	4.2640e-003	4.2653e-003	4.3477e-003
a_2	4.2640e-003	4.2653e-003	4.2735e-003
a_3	3.0151e-003	3.0160e-003	3.0256e-003
E3	$\sigma = 0.05$		
a_1	2.1320e-002	2.1306e-002	2.1797e-002
a_2	2.1320e-002	2.1306e-002	2.1383e-002
a_3	1.5076e-002	1.5066e-002	1.5106e-002

Table 11.26: Three estimates of the standard uncertainty of the fitted parameters for the NLLS estimator for full circle data.

We consider three estimates of the standard uncertainties of the fitted parameters:

$u(a_j)$ calculated from (11.45) using the value of σ used to generate the data.

$\hat{u}(a_j)$ calculated as in (11.44) but with $\hat{\sigma}_N$ replaced by

$$\bar{\sigma}_N = \left(\frac{1}{N} \sum_{q=1}^{N} \hat{\sigma}_{N,q}^2 \right)^{1/2},$$

i.e., the root-mean-square value of the estimates of the standard deviation of the residuals $\hat{\sigma}_{N,q}$ calculated for each data set $\mathbf{z}_q$,

$\bar{u}(a_j)$ the standard deviation of the parameters estimates $\mathbf{a}_q$ determined in the Monte Carlo simulations.

Table 11.26 gives these three estimates for $\sigma = 0.001, 0.01$ and 0.05. For each value, the estimates u and $\hat{u}$ are consistent with the actual variation $\bar{u}$.

In the second set of experiments we apply both the NLLS and LLS estimators to data generated from exact data $\{(x_i^*, y_i^*)\}$ lying on the arc of a circle of radius 10 between the points $(\pm 1, 0)$:

x*	y*
-1.0000	0
-0.8000	-0.0181
-0.6000	-0.0321
-0.4000	-0.0421
-0.2000	-0.0481
0	-0.0501
0.2000	-0.0481
0.4000	-0.0421
0.6000	-0.0321
0.8000	-0.0181
1.0000	0

As for the NLLS estimator, we consider three estimates of the standard uncertainty of fitted parameters for the LLS estimator:

$u_L(a_j)$ calculated from (11.50) using the value of σ used to generate the data.

$\hat{u}_L(a_j)$ calculated as in (11.49) but with $\hat{\sigma}_L$ replaced by

$$\bar{\sigma}_L = \left(\frac{1}{N} \sum_{q=1}^{N} \hat{\sigma}_{L,q}^2 \right)^{1/2},$$

i.e., the root-mean-square value of the estimates of the standard deviation of the residuals $\hat{\sigma}_{L,q}$ calculated for each data set $\mathbf{z}_q$,

$\bar{u}_L(a_j)$ the standard deviation of the parameters estimates $\mathbf{a}_q$ determined in the Monte Carlo simulations.

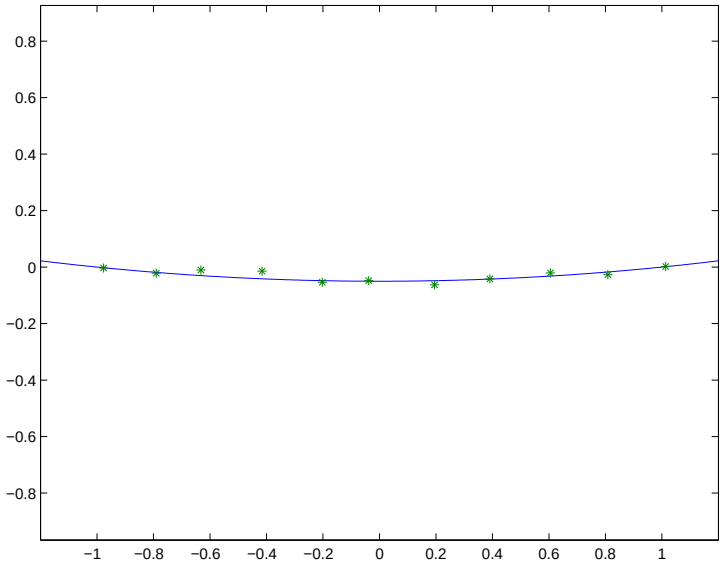

Figure 11.16: Example data generated on an arc of a circle passing through $(\pm 1, 0)$ with $\sigma = 0.015$.

a_j	u_L	$\hat{u}_L$	$\bar{u}_L$
E3	$\sigma = 0.001$		
a_1	8.5784e-004	8.6436e-004	8.5467e-004
a_2	4.7690e-004	4.8052e-004	4.6947e-004
a_3	4.5740e-004	4.6087e-004	4.5450e-004
E3	$\sigma = 0.01$		
a_1	8.7793e-003	8.8464e-003	8.5501e-003
a_2	4.7840e-003	4.8206e-003	4.6937e-003
a_3	4.6197e-003	4.6551e-003	4.5456e-003
E3	$\sigma = 0.015$		
a_1	1.3339e-002	1.3441e-002	1.2827e-002
a_2	7.1883e-003	7.2434e-003	7.0398e-003
a_3	6.9695e-003	7.0230e-003	6.8186e-003

Table 11.27: Three estimates the standard uncertainty of the fitted parameters for the LLS estimator for arc data.

a_j	u	$\bar{u}^L$	$\hat{u}$	$\bar{u}$
	$\sigma = 0.001$			
a_1	4.7673e-003	4.6706e-003	4.7794e-003	4.6706e-003
a_2	1.7027e-001	1.6938e-001	1.7070e-001	1.6938e-001
a_3	1.6993e-001	1.6904e-001	1.7036e-001	1.6904e-001
	$\sigma = 0.01$			
a_1	4.7673e-002	4.8813e-002	4.7793e-002	4.8746e-002
a_2	1.7027e+000	1.9342e+000	1.7070e+000	1.9295e+000
a_3	1.6993e+000	1.9310e+000	1.7036e+000	1.9263e+000
	$\sigma = 0.0125$			
a_1	5.9591e-002	6.3093e-002	5.9739e-002	6.2950e-002
a_2	2.1284e+000	2.6832e+000	2.1337e+000	2.6722e+000
a_3	2.1241e+000	2.6793e+000	2.1294e+000	2.6684e+000
	$\sigma = 0.015$			
a_1	7.1510e-002	8.1248e-002	7.1712e-002	8.0912e-002
a_2	2.5540e+000	4.1562e+000	2.5613e+000	4.1189e+000
a_3	2.5489e+000	4.1522e+000	2.5561e+000	4.1149e+000

Table 11.28: Four estimates the standard uncertainty of the fitted parameters (circle centre co-ordinates and radius) for circle fits to arc data. $\bar{u}^L$ is the estimate derived from variation in the parameters for the circles determined by the LLS estimator and $\bar{u}$ is that corresponding to NLLS.

Table 11.27 gives the estimates of the standard uncertainties of the circle parameters $\mathbf{a}$ (for the parameterisation of equation (11.47)) for the LLS estimator for $\sigma = 0.001, 0.01$ and 0.015. The nominal values of the parameters are $\mathbf{a}^* = (-0.0503, 0.0000, 0.0503)^{\mathrm{T}}$ (to four decimal places). A value of $\sigma = 0.015$ represents noisy data as can be seen in figure 11.16. The table shows that the estimates u_L and $\hat{u}_L$ are consistent with $\bar{u}_L$, the standard deviation of the Monte Carlo estimates.

Table 11.28 gives the four estimates of the standard uncertainties of the circle parameters $\mathbf{a} = (x_0, y_0, r_0)^{\mathrm{T}}$, centre co-ordinates and radius for the cases $\sigma = 0.001, 0.01, 0.0125$ and 0.015. The estimates are u, $\hat{u}$ and $\bar{u}$ defined above along with

$\bar{u}^L$: the standard deviation of the circle centre co-ordinates and radius derived from the LLS parameter estimates using (11.48).

From the results in the table we note:

- For all values of σ the estimates $\bar{u}^L$ and $\bar{u}$ are consistent with each other. This is because for this type of data the estimators produce virtually the same circle fits.
- For accurate data, $\sigma = 0.001$, there is good agreement with the predicted estimates u and $\hat{u}$ with the estimates $\bar{u}^L$ and $\bar{u}$ derived from the actual variation.

- As the value of σ becomes larger, the agreement becomes progressively worse: the estimates u and $\hat{u}$ underestimate the actual variation.

To summarise:

- We have two methods of fitting a circle to data, LLS and NLLS.
- For the type of data lying on an arc of a circle considered above, the two methods provide give essentially the same fitted circles but defined by different sets of parameters.
- The standard uncertainty estimates for the parameters determined by LLS are valid but those provided by NLLS are not.

The reason that the two methods of uncertainty estimation have different behaviour is the parameters $\mathbf{a}$ determined by the LLS estimator depend approximately linearly on the data while those $\mathbf{a} = (x_0, y_0, r_0)^{\mathrm{T}}$ determined by NLLS do not. The calculation of $V_{\mathbf{a}}^N$ in (11.44) assumes that the Jacobian matrix is approximately constant in the neighbourhood of the solution and this is not the case. In fact, we can see this from the equation for the $y-$co-ordinate of the circle centre in terms of the LLS parameters:

$$y_0 = -\frac{1}{2a_1}.$$

For simulations with $\sigma = 0.01$, we have $a_1 = -0.0503$ and, from table 11.27, $\bar{u}(a_1) = 0.0085$. Using these values, we can estimate the standard uncertainty associated with y_0 from

$$u(y_0) = \left|\frac{\partial y_0}{\partial a_1} u(a_1)\right| = 1.6798.$$

This calculation assumes that the function is sufficiently linear at the value of a_0 and corresponds to the calculation of the standard uncertainty associated with y_0 from (11.45). However, we can use a simple Monte Carlo simulation to estimate the uncertainty by generating

$$a_{1,q} = a_1 + \epsilon_q, \quad \epsilon_q \in N(0, 0.0085^2)$$

and calculating the standard deviation of the results. For a Monte Carlo trial with 5000 simulations, we obtain an estimate of $\bar{u}(y_0) = 1.9108$. This shows that the estimate based on linearisation is an underestimate. The situation is illustrated by figure 11.17 which graphs $y_0 = -1/(2a)$ and its tangent line at $(-0.0503, 9.9404)$. The figure shows how a normally distributed set of a-values (displayed on the lower horizontal axis) is mapped to a set of y-co-ordinate values (displayed on the right-hand vertical axis) and a linearised set of estimates of y-co-ordinates projected from the tangent line (displayed on the left-hand vertical axis). The rapid change in curvature of the graph $y_0 = -1/(2a)$ relative to the standard deviation of the a-values accounts for the discrepancy for the two sets of estimates.

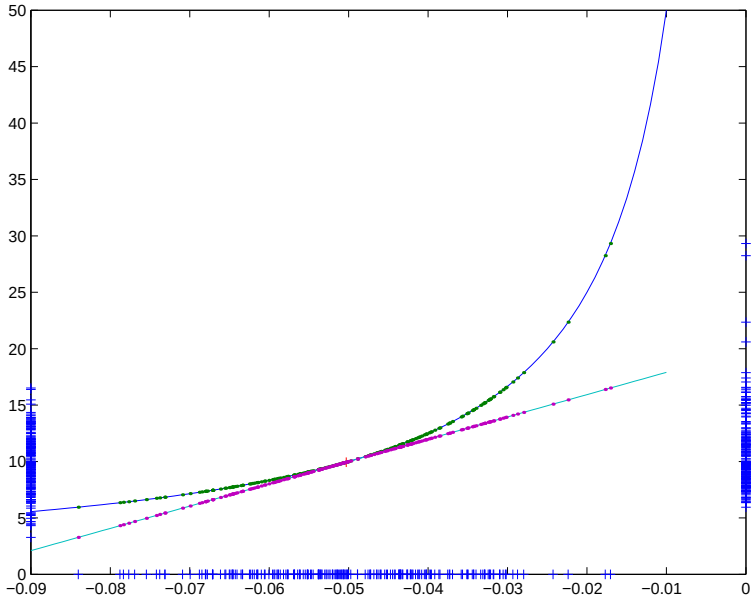

Figure 11.17: Uncertainty estimates for the y-co-ordinate of a circle fitted to data covering an arc of a circle.

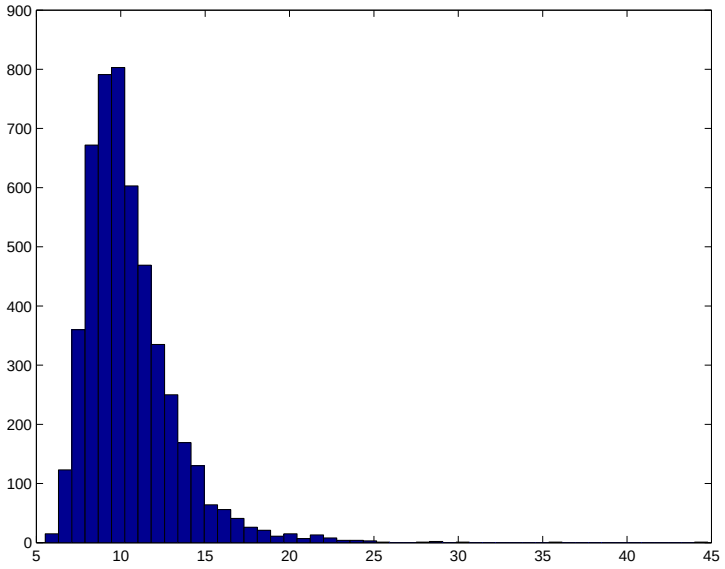

Figure 11.18: Approximate (un-normalised) distribution of radius values derived from 5000 Monte Carlo simulations for the case $\sigma = 0.0125$.

The nonlinearity associated with the estimation of the radius means that for this type of data, the standard uncertainties provided by (11.45) can be unreliable. However, even if the correct covariances are calculated (from Monte Carlo simulations for example) the estimates of standard uncertainty alone do not provide a complete picture of the distribution of the radius values due to its asymmetry. For example, figure 11.18 shows the approximate (un-normalised) distribution of radius values derived from 5000 Monte Carlo simulations for the case $\sigma = 0.0125$.

11.9 Roundness assessment in the presence of form errors

Form error is a measure of the departure of a workpiece from its nominal, ideal geometry. It is a quantity that is estimated from measurements and, in common with other measured quantities, valid ways of providing such estimates and evaluating their associated uncertainties are required. This case study is concerned with the calculation of the radius and form error and their uncertainties from measurements.

We first consider exactly the same model (11.41) and apply the NLLS estimator (11.42) to determine estimates $\mathbf{a}$ and associated estimate of the covariance matrix of the fitted parameters given by

$$\hat{V}_{\mathbf{a}} = \hat{\sigma}^2 (J^{\mathrm{T}} J)^{-1}, \tag{11.51}$$

where $\hat{\sigma}$ is the standard deviation of the residuals $\mathbf{d} = (d_1, \ldots, d_m)^{\mathrm{T}}$ at the solution, calculated according to

$$\hat{\sigma} = \|\mathbf{d}\|/(m-3)^{1/2}.$$

We also consider the estimate

$$V_{\mathbf{a}} = \sigma^2 (J^{\mathrm{T}} J)^{-1}. \tag{11.52}$$

Suppose a circular artefact of nominal radius $a_3 = 100$ is measured by experimenter A using a co-ordinate measuring machine (CMM). From previous experiments, it is known that random effects associated with the CMM are approximately normal distributed with standard deviation $\sigma = 0.001$.

Gathering data points uniformly space around the artefact, the experiment calculates the best-fit circle parameters, residuals $\mathbf{d}$ and estimate $\hat{\sigma}$ of the standard deviation of the residuals and standard uncertainties $u(a_j)$ using (11.52) and $\hat{u}(a_j)$ using (11.51) of the fitted parameters. In section 11.8, it was shown that these uncertainty estimates are reliable for this type of data and model.

The expected value of $\hat{\sigma}$ is σ but the actual estimate is significantly larger. In order to obtain more information, the measurements are repeated from which mean estimates $\bar{\mathbf{a}}$ of the parameters $\mathbf{a}$ and estimates $\bar{u}(a_j)$ of their uncertainties can be determined. There are now three estimates of the uncertainty of the

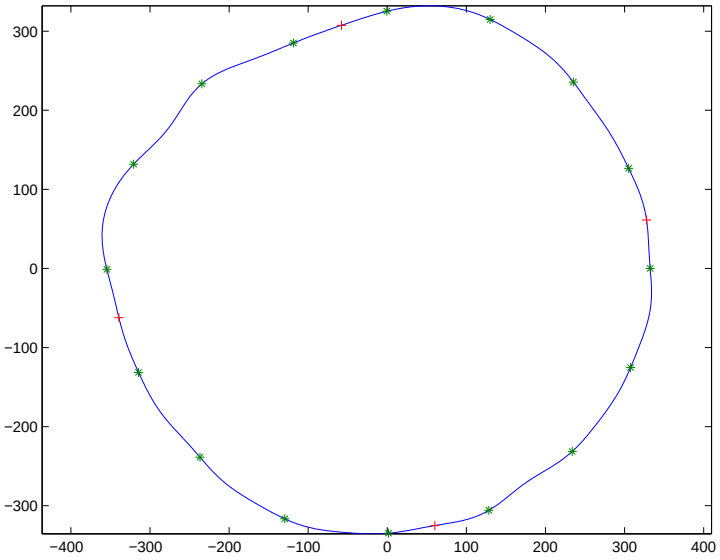

Figure 11.19: Circle form and measurement data for experimenters A ('*') and B ('+').

radius of the artefact. Which, if any, are valid? In order to resolve this question, experimenter B is asked to repeat the measurements, calculating the uncertainties using the three approaches. These uncertainty estimates for A and B are shown in table 11.29. While for experimenter A the repeat experiments seem to confirm the estimates u derived from (11.52) based on a prior estimate of the uncertainty of the CMM measurement uncertainty σ, for experimenter B the repeat experiments indicate that both estimates $\hat{u}$ and u underestimate the actual variation.

The six sets of estimates produce a confusing picture. The explanation of why they are so different comes from two sources a) form error and b) measurement strategy. Figure 11.19 shows the actual shape of the circular artefact (with the form error greatly magnified) and example measurement strategies for experimenters A (data points marked '*') and B (marked '+').

	$\hat{u}$	u	$\bar{u}$
A	8.6474e-004	2.5000e-004	2.2082e-004
B	3.2264e-004	5.0000e-004	1.7325e-003

Table 11.29: Two sets of estimates of the uncertainty in the radius of artefact: $\hat{u}$ calculated from (11.51), u calculated from (11.52) and $\bar{u}$ calculated from 5 repeat experiments.

The model (11.41) assumes that the departures from circularity in the data are due purely to random, independent random effects associated with the CMM measurements and that the artefact is perfectly circular.

In fact, the artefact has a profile, expressed in polar co-ordinates, of the form

$$r(\theta, \mathbf{b}) = r_0 + s(\theta, \mathbf{b})$$

where $s(\theta, \mathbf{b})$ is a periodic function depending on parameters $\mathbf{b}$. A better model is therefore

$$\begin{aligned} x_i^* &= a_1 + r(\theta_i^*, \mathbf{b}) \cos\theta_i^*, \\ y_i^* &= a_2 + r(\theta_i^*, \mathbf{b}) \sin\theta_i^*, \\ x_i &= x_i^* + \epsilon_i, \quad y_i = y_i^* + \delta_i, \quad \epsilon_i,\ \delta_i \in N(0, \sigma^2), \end{aligned}$$

where the angles θ_i^* specify where the artefact is measured. When experimenter A first measures the artefact, the systematic errors associated with the departure from circularity lead to much larger residuals and hence a larger estimate of $\hat{\sigma}$ and standard uncertainties than those expected from the prior estimate of σ. When A remeasures the artefact, the artefact is kept in its original position on the CMM and the same measurement strategy is executed so that the measured data is gathered from the same locations on the artefact. This means that differences between A's repeat measurements are accounted for by the random effects associated with the CMM. For this reason, the estimates $\bar{u}$ are consistent with u.

When B measures the artefact, only four measurement points are chosen on the artefact instead of the 16 chosen by A. For B's first measurement, the systematic errors associated with the form of the circle happen to have a small impact on the residual errors and the estimate $\hat{u}$ is smaller than anticipated by u. When B repeats the measurement, the artefact is replaced at random orientations so that new sets of four points are measured on the artefact, leading a much larger range of estimates for the radius and hence a larger $\bar{u}$.

11.9.1 Taking into account form error

The inconsistent information in table 11.29 arises from differences in measurement strategy and experimental design in the presence of systematic effects that have not been included in the model. This situation illustrates a general problem in metrology: how to account for small systematic errors in uncertainty estimation. In this section we look more closely at the problems of determining estimates of radius and form error of a nominally circular artefact.

The first issue is what do we mean by form error. A mathematical definition of roundness, i.e., form error for a circle, is as follows. Let $P = \{\mathbf{x} = (x, y)\}$ be the set of points lying on the profile of a nominally circular artefact. Given a circle $C(\mathbf{a})$ defined in terms of centre co-ordinates (a_1, a_2) and radius a_3, let $e(\mathbf{a}) = \max_{\mathbf{x}\in P} d(\mathbf{x}, \mathbf{a})$ be the maximum distance from a point in P to the circle, measured orthogonally to the circle as in (11.43). Then the form error (peak to valley) is $2e^*$ where $e^* = e(\mathbf{a}^*)$ is the value for e for the circle specified by $\mathbf{a}^*$ for

which $e(\mathbf{a})$ is minimised. In other words, the form error is determined in terms of the best-fit circle that minimises the maximum deviation.

Note that in order to determine an artefact's roundness an accurate knowledge of the complete profile is required. In general, only a finite number of points gathered subject to random effects is available. The estimate of form error derived from a finite set of accurate data is likely to be an underestimate of the form error. On the other hand, if the uncertainty associated with the measurements are comparable to the likely form error, it is possible that a significant component of the form error determined from the data is due solely to measurement error, potentially producing an overestimate.

In many circumstances, it is expected that the uncertainties due to the probing strategy, as a consequence of form error, constitute the major contribution to measurement uncertainty, as the example above shows. However, the relationship between probing strategy and the reliability of the resulting estimates is often complex. This leads to two fundamental problems: (a) how to assess the uncertainties associated with a given choice of probing strategy, and (b) given a measurement task, how to determine a probing strategy that is fit for purpose.

A solution to these problems can be found if we have a valid model of the likely form and random effects associated with the measurements.

Example: circularity of a lobed circle.

Consider the problem of determining the roundness of a lobed circle. This problem has practical importance since many manufacturing processes introduce lobing to surfaces of revolution. A lobed circle (with q lobes) is described in polar coordinates by

$$r(\theta) = r_0 + A\cos(q\theta + \theta_0);$$

and the roundness of such a lobed circle is $2A$. If we sample the circle at angles $\{\theta_i\}_{i=1}^{m}$, then the estimate of circularity obtained from the points is bounded above by

$$2a = \max_i A\cos(q\theta_i + \theta_0) - \min_i A\cos(q\theta_i + \theta_0).$$

We wish to choose θ_i so that this bound is as close to $2A$ as possible. If there are m points uniformly spaced around the circle, it is not difficult to show that if m and q have no common factor, then

$$\begin{aligned} A \geq \quad a \quad &\geq A\cos\frac{\pi}{m}, \quad m \quad \text{even}, \\ A\cos^2\frac{\pi}{2m} \geq \quad a \quad &\geq A\cos\frac{\pi}{2m}, \quad m \quad \text{odd}. \end{aligned}$$

We note that for m odd, $2a$ underestimates the circularity by at least a factor of $\cos\frac{\pi}{2m}$ and we can therefore take as our estimate of A

$$\hat{a} = a/\cos\frac{\pi}{2m} \geq A\cos\frac{\pi}{2m}.$$

Table 11.30 shows the value of $\cos\frac{\pi}{2m}$ for m small and prime. Five points will detect 95% of the lobing if the number of lobes q is not a multiple of 5, while

m	$\cos\frac{\pi}{2m}$	m	$\cos\frac{\pi}{2m}$
5	0.9511	19	0.9966
7	0.9749	23	0.9977
11	0.9898	29	0.9985
13	0.9927	31	0.9987
17	0.9957	37	0.9991

Table 11.30: Values of $\cos\frac{\pi}{2m}$ for small primes.

eleven points will detect 99% of the lobing if q is not multiple of 11. The use of an even number of points for detecting lobing is not recommended. Figure 11.20 shows two distributions of six points on a three lobed circle; one set marked $*$ detects 100% of the lobing while the other set marked o fails to detect any lobing. By contrast the seven points shown in figure 11.21 detects 97.5% of the lobing, slightly above the theoretical minimum given in table 11.30.

The analysis in the example above exploits a known model for the expected form error, namely lobing. In many situations, it is possible to derive a model for the likely form error either from an assessment of the manufacturing process, past history or from the detailed analysis of a number of typical workpieces.

Figure 11.22 shows the profile of a circular artefact in terms of its departure for circularity at 2000 uniformly spaced data points. We wish to determine the roundness of artefacts with similar profile characteristics from measurements using a CMM.

V1 How do we assign uncertainties to these roundness estimates?

V2 How do we determine a measurement strategy sufficient to provide accurate roundness estimates?

From the measurements, we define a virtual feature of the form

$$x^* = a_1 + r(\theta, \mathbf{b})\cos\theta, \quad y^* = a_2 + r(\theta, \mathbf{b})\sin\theta, \tag{11.53}$$

where $r(\theta, \mathbf{b}) = a_3 + s(\theta, \mathbf{b})$ is a best-fit periodic function to the profile. In this situation it is appropriate represent the profile as a Fourier series of the form

$$s_n(\theta, \mathbf{b}) = b_1 + \sum_{j=1}^{n}[b_{2j}\sin j\theta + b_{2j+1}\cos j\theta]$$

parameterised by the $2n+1$ coefficients $\mathbf{b} = (b_1, \ldots, b_{2n+1})^{\mathrm{T}}$. Fourier series are a class of empirical functions that are well suited to modelling periodic empirical functions. The Fast Fourier Transform algorithm (FFT) enables the least-squares approximation of data by Fourier series to be implemented efficiently and in a numerically stable way. In order to determine a suitable fit it is necessary to select the number n of harmonics.

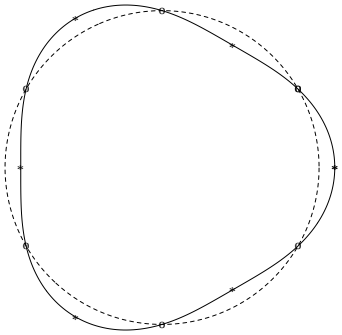

Figure 11.20: Two sets of six points uniformly spaced on a three-lobed circle. The points marked $*$ detect 100% of the lobing while the points marked $\circ$ fail to detect any form deviation.

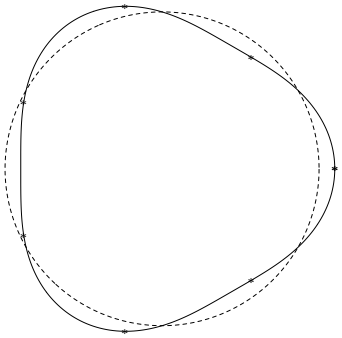

Figure 11.21: Seven points uniformly spaced on a three-lobed circle detect at least 97% of the lobing.

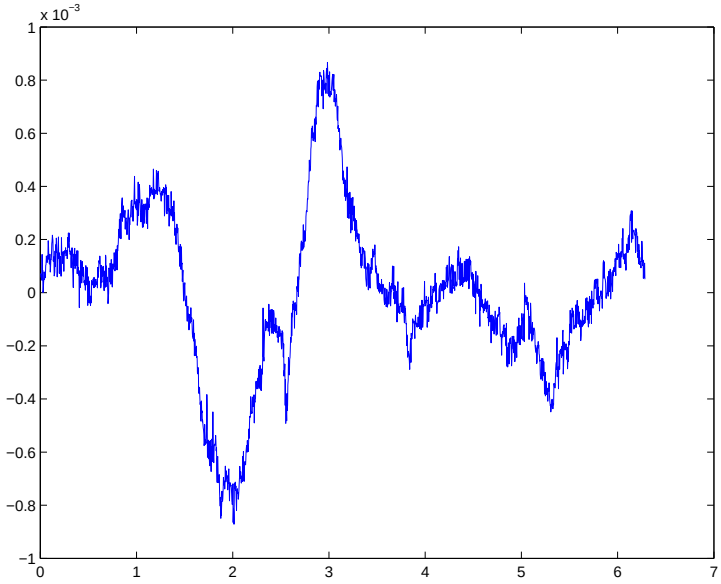

Figure 11.22: Profile of a circular artefact determined from 2000 measured points.

V3 What is a good choice of number n of harmonics for a Fourier series representation of the data represented in figure 11.22?

This question can be answered using the strategy outlined in section 9.1:

- For $n = 1, 2, \ldots$, fit a Fourier series $s_n = s_n(\theta, \mathbf{b}_n)$ of n harmonics to the data (θ_i, r_i) and evaluate residuals $\mathbf{f}_n = (f_1, \ldots, f_m)^{\mathrm{T}}$ where

$$f_i = r_i - s_n(\theta_i, \mathbf{b}_n).$$

- For each n, determine the estimate

$$\hat{\sigma}_n = \|\mathbf{f}_n\|/(m - 2n - 1)^{1/2}$$

of the standard deviation of residuals.

- Plot $\hat{\sigma}_n$ and choose n where at a point where the decrease in $\hat{\sigma}_n$ is judged to have levelled off.

Figure 11.23 shows the values of $\hat{\sigma}_n$ up to 50 harmonics. From this figure, 18 harmonics is judged suitable. Figure 11.24 shows the fits of Fourier series with 5 and 18 harmonics to the data in figures 11.22. Figure 11.25 shows the residuals f_i corresponding to a fit to the profile by a Fourier series with 18 harmonics. These residuals look largely random, certainly in comparison with those associated with a circle fit in figure 11.22.

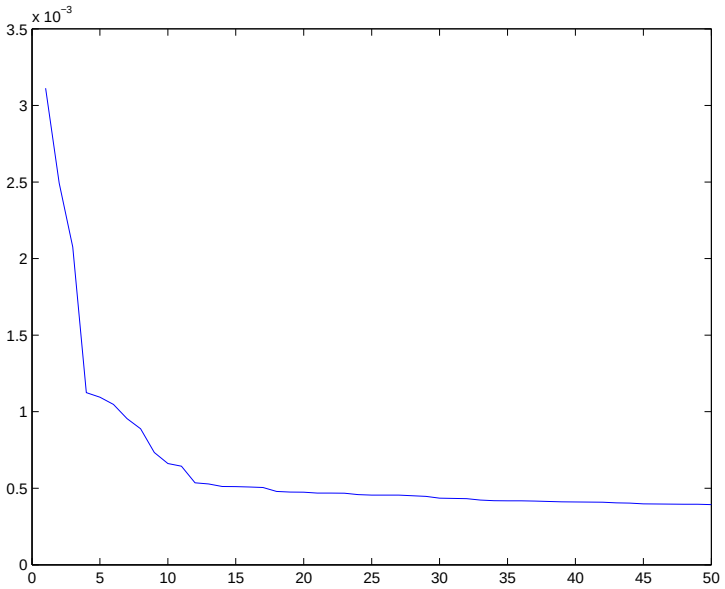

Figure 11.23: Estimates of the RMS residuals for fits of Fourier series up to 50 harmonics to profile data in figure 11.22.

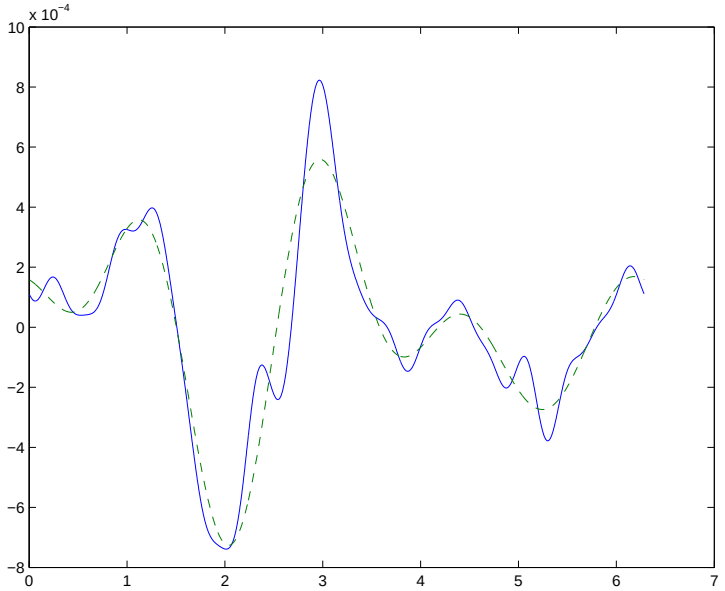

Figure 11.24: Fourier series with 5 (dashed) and 18 (solid) harmonics fitted to the profile of a circular artefact.

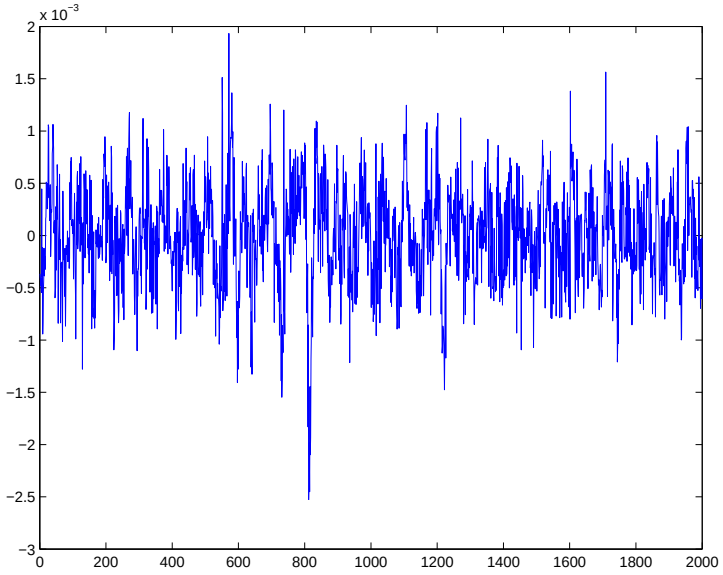

Figure 11.25: Residuals associated with the fit of a Fourier series of degree 18 to the profile of a circular artefact.

With $r(\theta, \mathbf{b}) = a_3 + s_{18}(\theta, \mathbf{b}_{18})$ the virtual feature (11.53) is defined and can be used to generate data sets simulating measurements of the artefact according to a proposed measurement strategy. For example, suppose a measurement strategy of eight data points uniformly distribution around the circle is to be tested. Data sets can be generated according to

$$\begin{aligned} x_{i,q} &= a_1 + r(\theta_{i,q}, \mathbf{b}) \cos, \theta_{i,q} + \delta_{i,q}, \quad \delta_{i,q} \in N(0, \sigma^2), \\ y_{i,q} &= a_2 + r(\theta_{i,q}, \mathbf{b}) \sin \theta_{i,q} + \epsilon_{i,q}, \quad \epsilon_{i,q} \in N(0, \sigma^2), \end{aligned}$$

with

$$\theta_{i,q} = i\pi/4 + \phi_q, \qquad 0 \leq \phi_q \leq \pi/4, \quad i = 1, \ldots, 8,$$

where ϕ_q is a randomly chosen phase angle specifying where on the circle the measurements are made and $\delta_{i,q}$ and $\epsilon_{i,q}$ represent random effects.

Circle fitting software is then applied to these data sets to calculate the circle parameters $\mathbf{a}_q$, residuals $\mathbf{d}_q = (d_{1,q}, \ldots, d_{8,q})^{\mathrm{T}}$ and estimate

$$e_q = \max_i d_{i,q} - \min_i d_{i,q}$$

of the maximum form error for the qth data set. From a number of repeat Monte Carlo simulations we can determine the mean estimate $\bar{e}$ of the form error and its standard deviation $\bar{u}(e)$. Table 11.31 presents these estimates for measurement strategies with different numbers m and values $\sigma = 10^{-7}$ representing accurate

	$\sigma = 1e-7$		$\sigma = 0.0005$	
m	$\bar{e}$	$\bar{u}$	$\bar{e}$	$\bar{u}$
4	0.0023	0.0010	0.0023	0.0011
5	0.0037	0.0016	0.0038	0.0010
14	0.0084	0.0011	0.0085	0.0012
19	0.0091	0.0008	0.0091	0.0010
25	0.0095	0.0005	0.0096	0.0007
31	0.0096	0.0003	0.0098	0.0007
38	0.0098	0.0002	0.0100	0.0006
47	0.0098	0.0001	0.0101	0.0006
57	0.0099	0.0001	0.0104	0.0006
68	0.0099	0.0001	0.0104	0.0005

Table 11.31: Mean estimates of the maximum form error $\bar{e}$ and their standard deviations $\bar{u}$ for different numbers of data points m and noise levels σ.

measurement data and $\sigma = 0.0005$ for data generated for a circle with form error $e^* = 0.01$. The results show that for the case $\sigma = 1.0^{-7}$, the estimates of the form error e approach the true value of 0.01 and the standard deviation approaches zero as the number of points m increases. For the case $\sigma = 0.0005$, the estimate e overshoots the true value while the standard deviation levels off at approximately 0.0005 as m increases. This data in represented graphically in figures 11.26 and 11.27.

These results were generated from just one example profile: shown in figure 11.22. The experiments can be repeated for other profiles either based on measurements of workpieces or synthesised from empirical models. For example, we can generate profiles P_q represented by Fourier series with up to twenty harmonics with Fourier coefficients $\mathbf{b}_q$ selected at random and normalised to represent form error $e = 0.01$. The measurement strategies can then be tested on these range of profiles. Figure 11.28 presents the same information as figure 11.26 calculated for 1000 randomly generated profiles.

From this type of simulation, it is possible to

- Quantify the bias in the estimate of form error due to under-sampling.
- Quantify the bias in the estimate of form error due to random measurement error.
- Provide valid estimates of the uncertainty in form error estimates.
- Select a measurement strategy that provides adequate estimates of form error.

Simulation is the only practical way of arriving at this information.

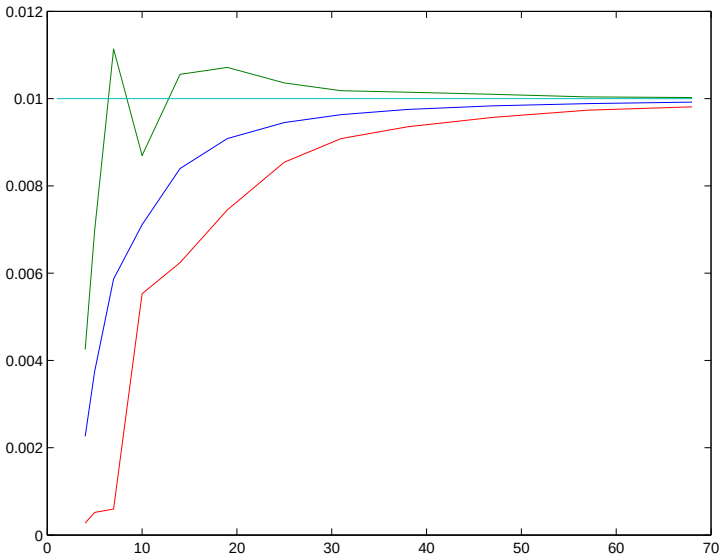

Figure 11.26: Estimates of the maximum for error $\bar{e}$ and $\bar{e} \pm 2\bar{u}$ where $\bar{u}$ is the standard deviation of the estimates determined from 1000 Monte Carlo simulations for accurate measurement data, $\sigma = 10^{-7}$.

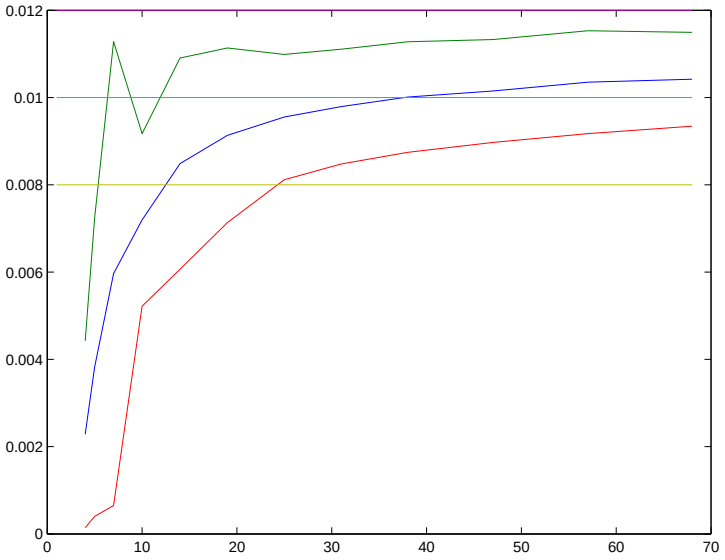

Figure 11.27: Estimates of the maximum for error $\bar{e}$ and $\bar{e} \pm 2\bar{u}$ where $\bar{u}$ is the standard deviation of the estimates determined from 1000 Monte Carlo simulations for measurement data with $\sigma = 0.0005$.

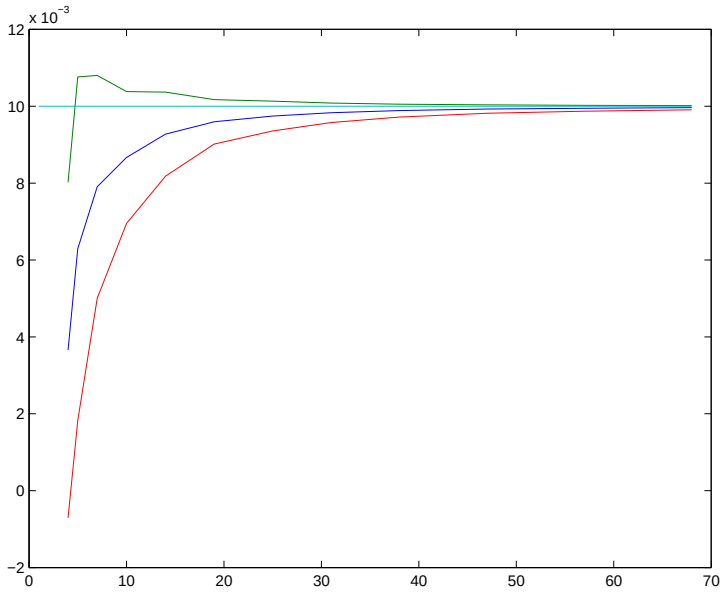

Figure 11.28: Estimates of the maximum for error $\bar{e}$ and $\bar{e} \pm 2\bar{u}$ where $\bar{u}$ is the standard deviation of the estimates determined from 1000 Monte Carlo simulations for accurate measurement data, $\sigma = 10^{-7}$. Each simulation involved a profile represented by a Fourier series with 20 harmonics generated at random.

While this case study has been concerned with the effect of form error on roundness assessment, the underlying features are quite generic. Many experiments in metrology are influenced by small systematic effects that are incompletely characterised. For example, properties of materials are not known exactly and materials are not perfectly homogeneous. The departure from homogeneity can be modelled using empirical models and the effect of inhomogeneity examined using simulations, both in terms of parameter estimates and estimates of their uncertainties.

11.10 Polynomial, spline and RBF surface fits to interferometric data

We illustrate the use of polynomials, splines and RBFs in approximating data arising from the interferometric measurement of an optical surface. Figure 11.29 (on page 225) shows an example set of measurements of heights z_i in nanometres over a regular 146×146 grid.

We have fitted this data using the following models and algorithms:

A Gaussian RBF with centres on a regular 10×10 grid,

B thin plate spline RBF with the same centres as A,

C 105 orthogonal polynomials of up to total degree 13 generated using the algorithm of Huhtanen and Larsen [138],

D tensor product polynomial with basis functions $\gamma_{kl} = T_k(x)T_l(y)$, $0 \leq k, l \leq 9$, where T_k is a Chebyshev basis function of degree k, (i.e., 100 parameters in all),

E as D but with $0 \leq k, l \leq 39$ (i.e., 1600 parameters),

F a tensor product bicubic spline with 6 interior knots along each axis (i.e., 100 parameters in all), and

G as F with with 36 interior knots on each axis (i.e., 1600 parameters).

Figure 11.30 plots the fitted surface determined using algorithm A to the data in figure 11.29 while figure 11.31 plots the associated residuals. Figures 11.32–11.37 graph the residuals associated with algorithms B – G. Figures 11.38 and 11.39 shows the fitted surfaces associated with algorithms E and G.

11.10.1 Assessment of model fits

Quality of fit

All methods generally produce a good fit. Visually the fitted surfaces seem to model the data very well. Only the fit associated with the orthogonal polynomial generated using the HL algorithm shows unwanted edge effects in the residuals (figure 11.33). The RMS residual for all fits ranges from approximately 1 nm for algorithms E and G involving 1600 parameters to 3 nm for algorithm B.

Computational efficiency

These experiments involve over 21,000 data points and approximately 100 to 1600 parameters and represent computational problems quite large by comparison with many approximation problems in metrology. The tensor product

approaches for a regular grid data are approximately n times faster than the full matrix approach associated with RBF approximation where n is the number of parameters.

To give an idea of the computational requirements of the various algorithms, using Matlab 6.5 and a Dell Optiplex GX240, Intel Pentium 4, 1.7 GHz PC, the time taken for algorithm A was i) 8.5 s to determine the matrix D of distances $d_{ij} = \|\mathbf{x}_i - \boldsymbol{\lambda}_j\|$, ii) 0.9 s to apply ρ to D, iii) 4.1 s to solve the linear least-squares system, making a total of 13.5 s. By comparison algorithm D took 0.02 s. For algorithm C, the time taken to generate the orthogonal basis was 9.8 s. The time taken to calculate the matrix D is comparatively slow since it involves two iterative loops. In Fortran, for example, this step would be relatively much quicker. Similar remarks apply to the implementation of the HL algorithm in Matlab.

Condition of the observation matrices

For the tensor polynomial and splines approaches, the condition numbers of the matrices generated were all less than 10. For the Gaussian RBF (algorithm A) the condition number was 4.2×10^6, while that for the thin plate spline (algorithm B) was 1.1×10^4. For the HL algorithm (C), the maximum absolute value of the off-diagonal elements of the computed matrix $Q^{\mathrm{T}}Q$ was 4.0×10^{-14}.

The data set in figure 11.29 lay on a regular grid and we were able to exploit this in fitting tensor product polynomial and spline surfaces. The RBF approach applies equally to scattered data. We have taken a random subset of the data in figure 11.29 and fitted Gaussian and thin plate spline RBFs to the data. Figure 11.40 plots the $xy-$coordinates of the data along with the RBF centres; figure 11.41 plots the coordinate data. Figure 11.42 plots the residuals associated with the fits of a Gaussian and thin plate spline RBF and bivariate polynomial of total degree 13 generated using the HL algorithm.

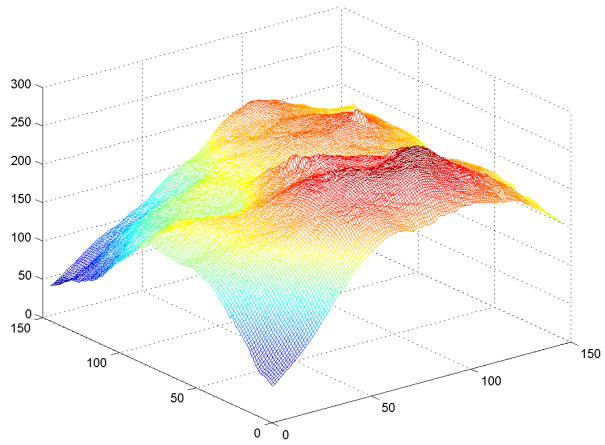

Figure 11.29: Measurements of an optical surface using interferometry. The units associated with the vertical axis are nanometres.

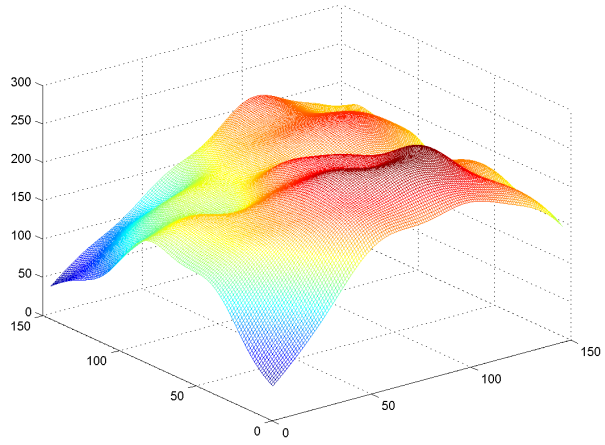

Figure 11.30: Gaussian RBF fitted to interferometric data (figure 11.29).

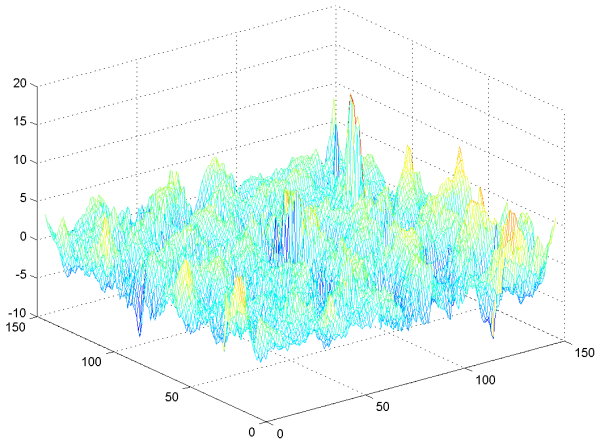

Figure 11.31: Residuals associated with the fit of a Gaussian RBF (figure 11.30) to interferometric data (figure 11.29).

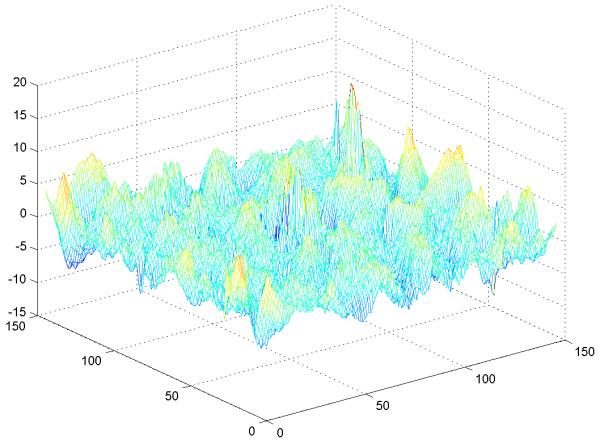

Figure 11.32: Residuals associated with the fit of a thin plate spline RBF to interferometric data (figure 11.29).

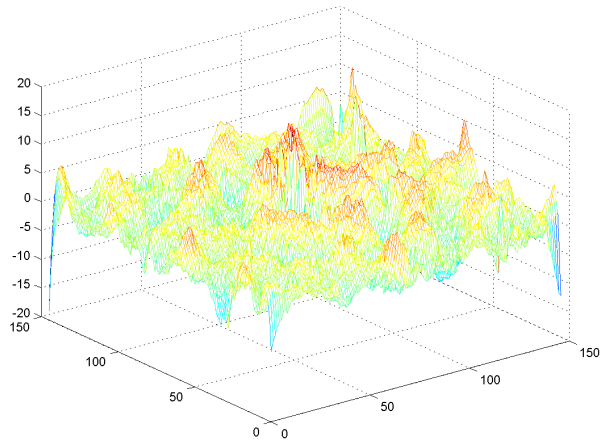

Figure 11.33: Residuals associated with the fit of a discrete orthogonal bivariate polynomial of total degree 13 to interferometric data (figure 11.29).

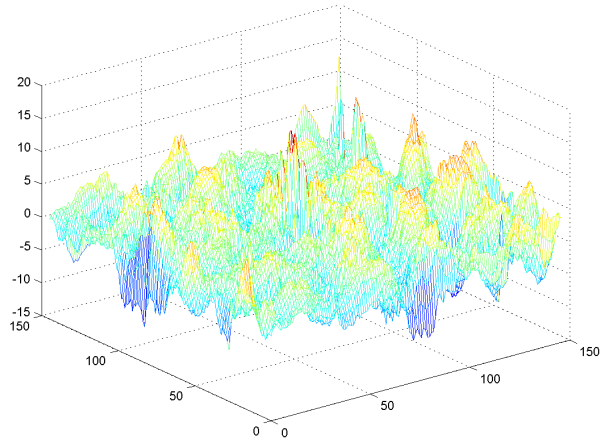

Figure 11.34: Residuals associated with the fit of an order 10 tensor product Chebyshev bivariate polynomial to interferometric data (figure 11.29).

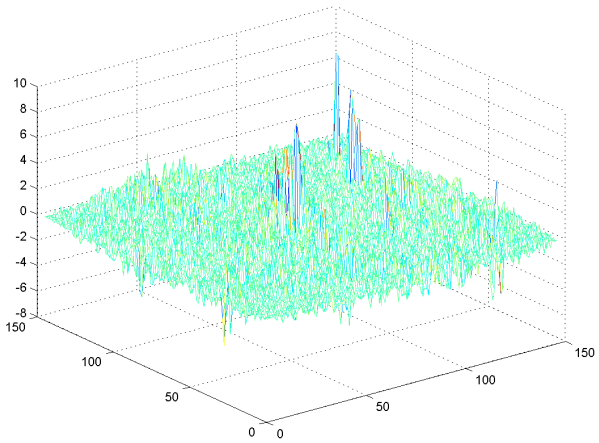

Figure 11.35: Residuals associated with the fit of an order 40 tensor product Chebyshev bivariate polynomial to interferometric data (figure 11.29).

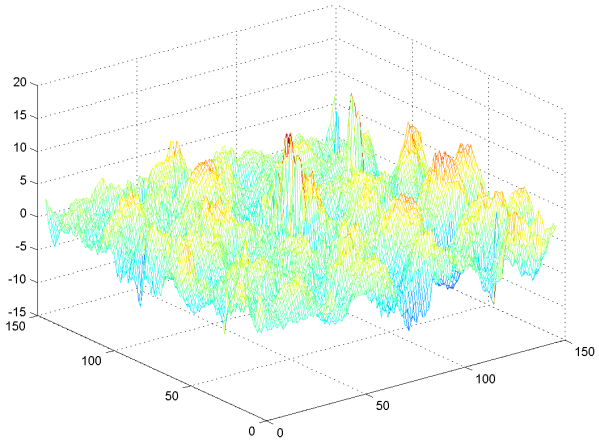

Figure 11.36: Residuals associated with the fit of a tensor product bicubic spline with 6 interior knots along each axis to interferometric data (figure 11.29).

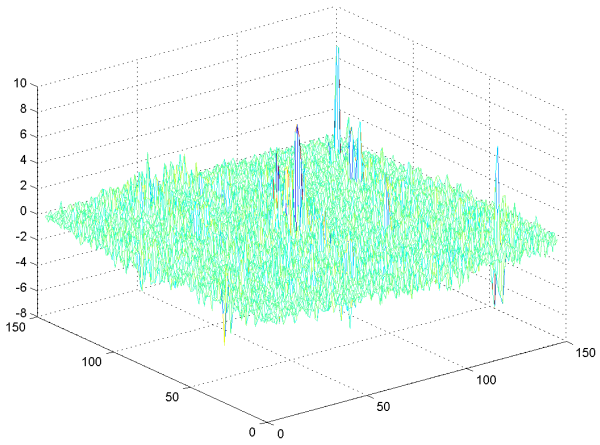

Figure 11.37: Residuals associated with the fit of a tensor product bicubic spline with 36 interior knots along each axis to interferometric data (figure 11.29).

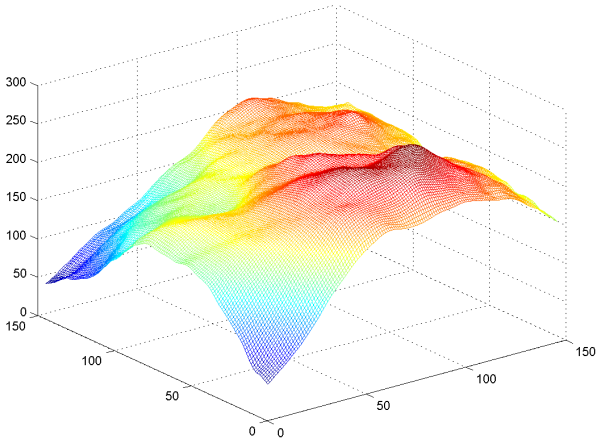

Figure 11.38: An order 40 tensor product Chebyshev bivariate polynomial fitted to interferometric data (figure 11.29).

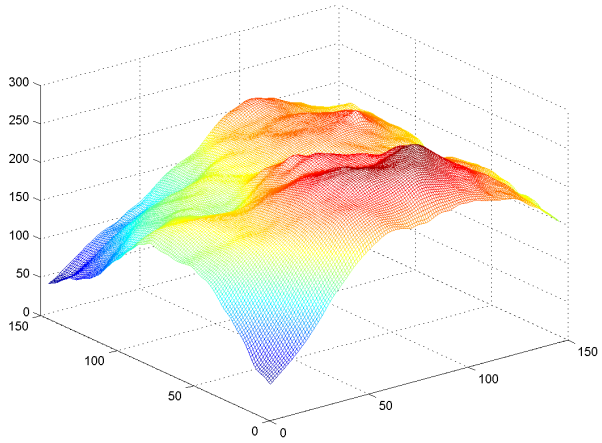

Figure 11.39: A fit of a tensor product bicubic spline with 36 interior knots along each axis to interferometric data (figure 11.29).

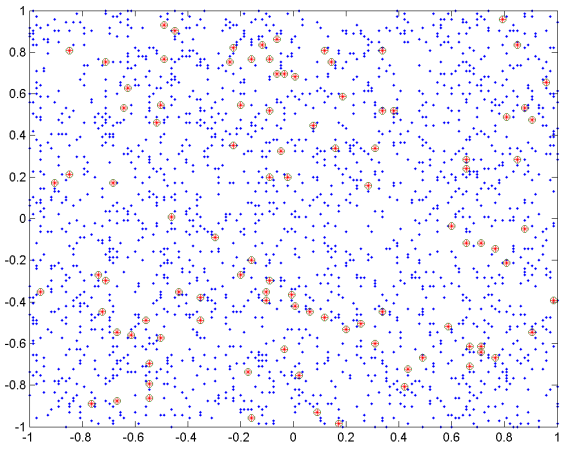

Figure 11.40: xy-coordinates of a subset of the interferometric data (figure 11.29). The RBF centres are marked with an 'o'.

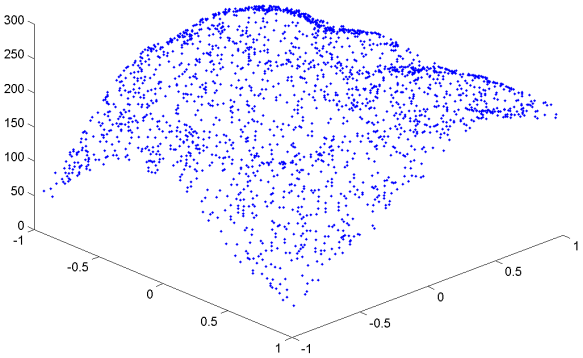

Figure 11.41: A randomly selected subset of the interferometric data (figure 11.29).

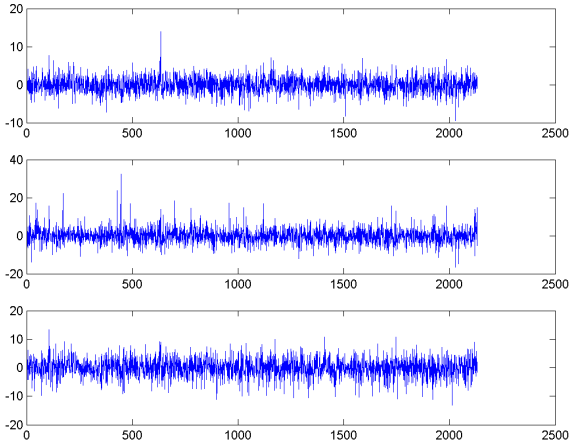

Figure 11.42: Residuals associated with the fit of a Gaussian RBF (top), thin plate spline RBF (middle) and bivariate polynomial of total degree 13 (bottom) to interferometric data (figure 11.41).

Chapter 12

Best practice in discrete modelling and experimental data analysis: a summary

We summarise the main issues that need to be addressed in discrete modelling and in metrological data analysis.

Functional model consists of:

- Problem variables representing all the quantities that are known or measured.
- Problem parameters representing the quantities that have to be determined from the measurement experiment. The problem parameters describe the possible behaviour of the system.
- The functional relationship between the variables and parameters.

Statistical model for the measurements consists of:

- The uncertainty structure describing which variables are known accurately and which are subject to significant random effects.
- The description of how the random effects are expected to behave, usually in terms means, variances (standard deviations) or probability density functions.

Estimator. An estimator is a method of extracting estimates of the problem parameters from the measurement data. Good estimators are unbiased, efficient and consistent.

The behaviour of an estimator can be analysed from maximum likelihood principles or using Monte Carlo simulations.

Estimator algorithm. An estimator requires the solution of a computational problem. An algorithm describes how this can be achieved.

Good algorithms determine an estimate of the solution that is close to the true solution of the computational problem and is efficient in terms of computational speed and memory requirements.

Problem conditioning and numerical stability. The effectiveness of an algorithm will depend on the conditioning the computational problem. For well conditioned problems, a small change in the data corresponds to a small change in the solution parameters, and conversely.

The conditioning of a problem depends on the parameterisation of the model. Often, the key to being able to determine accurate solution parameters is in finding the appropriate parameterisation.

A numerically stable algorithm is one that introduces no unnecessary ill-conditioning in the problem.

Software implementation and reuse. Calculations with a model should be split up into model key functions such as calculating function values and partial derivatives.

Optimisation software in the form of key solver functions can be used in implementing estimators that work with a wide range of model key functions.

For some models, special purpose solvers that exploit special features in the model are useful or necessary.

Many calculations required in discrete modelling can be performed using standard library/public domain software.

EUROMETROS. The Metrology Software environment developed under the Software Support for Metrology Programme aims to bridge the gap between library software and the metrologists needs, promoting and developing re-usable software performing the main calculations required by metrologists.

The following summarises the validation techniques that can be applied to validate each of these components.

Function model validation:

- Conduct design review by an expert in the metrology field to examine the choice of model variables and check assumptions.
- Perform numerical simulations to compare the behaviour of comprehensive models with simpler models.
- Conduct design review by an expert in the metrology area to check the modelling of the underlying physics.
- Conduct design review by a modelling expert to check the mathematical derivation of equations.
- Perform numerical simulations to check the effect of approximations, simplifications, linearisations, etc., on the model values (relative to the likely measurement error).
- Evaluate the model at variable/parameter values for which the physical response is known accurately.

- Perform numerical simulations to check the qualitative behaviour of the model against expected behaviour.
- Conduct design review by a metrology and/or modelling expert to check an empirical model is appropriate for the expected type of behaviour.
- Perform numerical simulations to check the qualitative behaviour of the model against expected behaviour.

Statistical model validation:

- Conduct design review by an expert in the metrology field to examine the statistical model for the measurement data and check assumptions.
- Conduct design review by modelling expert to check the statistical models for derived quantities.
- Perform numerical simulation to check the effect of approximations, simplifications, linearisations, etc., associated with the statistical model.
- Perform Monte Carlo simulations to check the variation in derived quantities against the predicted variation.

Estimator validation:

- Perform Monte Carlo simulations to examine the bias and variation of the solution estimates on datasets generated according to the statistical model.
- Perform Monte Carlo simulations to compare the predicted variation of parameter estimates with the actual variation on datasets generated according to the statistical model.
- Apply the estimator to datasets for which the estimates provided by an optimal estimator are known.
- Compare the actual variation of parameter estimates on datasets generated according to the statistical model with the predicted variation for an optimal estimator.
- Compare the actual statistical model with the statistical model for which the estimator is known to perform well.

Validation of the model solution:

- Examine the goodness of fit in terms of the size of the residual errors.
- Compare of the size of the residual errors with the statistical model for the measurement errors.
- Plot the residual errors to check for random/systematic behaviour.
- Plot the root-mean-square residual error for a number of model fits to select an appropriate model (e.g., the polynomial of appropriate degree).

- Calculate and check the covariance matrix for the fitted parameters against requirements and/or expected behaviour.
- Calculate and check the uncertainty associated with the model predictions against requirements and/or expected behaviour.

Validation of the experimental design and measurement strategy:

- Examine the singular values of the matrix defining the solution for exact data.

  If the matrix is full rank, the measurements are sufficient to determine all the parameters. If the matrix is rank deficient the right singular vectors contain information on what degrees of freedom are left unresolved.
- Calculate the covariance matrix of the fitted parameters for data generated according to the proposed experimental design. Check that the uncertainties in the fitted parameters are sufficient to meet the required uncertainty targets.

Bibliography

[1] S. J. Ahn, E. Westkämper, and Rauh. W. Orthogonal distance fitting of parametric curves and surfaces. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 122–129. University of Huddersfield, 2002. 4.3.5

[2] AMCTM, `http://www.amctm.org`. *Advanced Mathematical and Computation Tools in Metrology*. 2

[3] I. J. Anderson, M. G. Cox, A. B. Forbes, J. C. Mason, and D. A. Turner. An efficient and robust algorithm for solving the footpoint problem. In M. Daehlen, T. Lyche, and L. L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces II*, pages 9–16, Nashville, TN, 1998. Vanderbilt University Press. 4.3.5

[4] I. J. Anderson, M. G. Cox, and J. C. Mason. Tensor-product spline interpolation to data on or near a family of lines. *Numerical Algorithms*, 5:193–204, 1993. 5.5.1

[5] G. T. Anthony, H. M. Anthony, B. Bittner, B. P. Butler, M. G. Cox, R. Drieschner, R. Elligsen, A. B. Forbes, H. Groß, S. A. Hannaby, P. M. Harris, and J. Kok. Chebyshev best-fit geometric elements. Technical Report DITC 221/93, National Physical Laboratory, Teddington, 1993. 4.5.3, 4.9.1, 5.10.2

[6] G. T. Anthony, H. M. Anthony, B. Bittner, B. P. Butler, M. G. Cox, R. Drieschner, R. Elligsen, A. B. Forbes, H. Groß, S. A. Hannaby, P. M. Harris, and J. Kok. Reference software for finding Chebyshev best-fit geometric elements. *Precision Engineering*, 19:28 – 36, 1996. 4.5.3, 4.9.1, 5.10.2

[7] G. T. Anthony, H. M. Anthony, M. G. Cox, and A. B. Forbes. The parametrization of fundamental geometric form. Technical Report EUR 13517 EN, Commission of the European Communities (BCR Information), Luxembourg, 1991. 5.10.2, 11.7.12

[8] G. T. Anthony and M. G. Cox. The National Physical Laboratory's Data Approximation Subroutine Library. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation*, pages 669 – 687, Oxford, 1987. Clarendon Press. 1, 4.1.10, 5.1.3, 5.1.4, 5.2.4

[9] R. M. Barker. Software Support for Metrology Good Practice Guide No. 5: Guide to EUROMETROS: a manual for users, contributors and testers. Technical report, National Physical Laboratory, Teddington, 2004. `http://www.npl.co.uk/ssfm/download/bpg.html#ssfmgpg5`. 2, 4.3.5, 5.1.4, 5.2.4, 5.10.2

[10] R. M. Barker, M. G. Cox, P. M. Harris, and I. M. Smith. Testing algorithms in Standards and METROS. Technical Report CMSC 18/03, National Physical Laboratory, March 2003. `http://www.npl.co.uk/ssfm/download/#cmsc18_03`. 2, 8, 8.3.4

[11] R. M. Barker and A. B. Forbes. Software Support for Metrology Best Practice Guide No. 10: Discrete Model Validation. Technical report, National Physical Laboratory, Teddington, March 2001. `http://www.npl.co.uk/ssfm/download/bpg.html#ssfmbpg10`. 1.2

[12] V. A. Barker, L. S. Blackford, J. L. Dongarra, J. Du Croz, S. Hammarling, M. Marinova, J. Wasniewski, and P. Yalamov. *The LAPACK95 User's Guide*. SIAM, Philadelphia, 2001. 1.3.3

[13] I. Barrodale and C. Phillips. Algorithm 495: Solution of an overdetermined system of linear equations in the Chebyshev norm. *ACM Transactions of Mathematical Software*, pages 264 – 270, 1975. 4.5.3

[14] I. Barrodale and F. D. K. Roberts. An efficient algorithm for discrete l_1 linear approximation with linear constraints. *SIAM Journal of Numerical Analysis*, 15:603 – 611, 1978. 4.6.3

[15] I. Barrodale and F. D. K. Roberts. Solution of the constrained ℓ_1 linear approximation problem. *ACM Trans. Math. Soft.*, 6(2):231 –235, 1980. 4.6.3

[16] R. Bartels and A. R Conn. A program for linearly constrained discrete ℓ_1 problems. *ACM Trans. Math. Soft.*, 6(4):609–614, 1980. 4.6.3

[17] R. Bartels, A. R. Conn, and J. W. Sinclair. Minimization techniques for piecewise differentiable functions: The ℓ_1 solution to an overdetermined linear system. *SIAM Journal of Numerical Analysis*, 15:224–241, 1978. 4.6.3

[18] R. Bartels and G. H. Golub. Chebyshev solution to an overdetermined linear system. *Comm. ACM*, 11(6):428–430, 1968. 4.5.3

[19] M. Bartholomew-Biggs, B. P. Butler, and A. B. Forbes. Optimisation algorithms for generalised regression on metrology. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology IV*, pages 21–31, Singapore, 2000. World Scientific. 1.3.2, 4.3.5

[20] M. C. Bartholomew-Biggs, S. Brown, B. Christianson, and L. Dixon. Automatic differentiation of algorithms. *J. Comp. App. Math.*, 124:171–190, 2000. 4.2.8

[21] N. Bellomo and L. Preziosi. Mathematical problems in metrology: modelling and solution methods. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools for Metrology,*, pages 23–36. World Scientific, 1994. 1.3.2

[22] W. Bich. The ISO guide to the expression of uncertainty in measurement: A bridge between statistics and metrology. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, III*, pages 1–11, Singapore, 1997. World Scientific. 1.3.2

[23] W. Bich and P. Tavella. Calibrations by comparison in metrology: a survey. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools in Metrology,*, pages 155–166, Singapore, 1994. World Scientific. 1.3.2

[24] BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, and OIML. *Guide to the Expression of Uncertainty in Measurement*. Geneva, Switzerland, 1995. ISBN 92-67-10188-9, Second Edition. 2.4.4

[25] C. M. Bishop. *Neural networks and pattern recognition*. Oxford Univeristy Press, 1995. 5.9.1

[26] C. M. Bishop, editor. *Neural networks and Machine Learning*. Springer, 1998. 1997 NATO Advanced Study Institute. 5.9.1

[27] A. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996. 4.1.6, 4.1.10

[28] P. T. Boggs, R. H. Byrd, and R. B. Schnabel. A stable and efficient algorithm for nonlinear orthogonal distance regression. *SIAM Journal of Scientific and Statistical Computing*, 8(6):1052–1078, 1987. 4.3.5

[29] P. T. Boggs, J. R. Donaldson, R. H. Byrd, and R. B. Schnabel. ODRPACK: software for weighted orthogonal distance regression. *ACM Trans. Math. Soft.*, 15(4):348–364, 1989. 4.3.5

[30] R. Boudjemaa, M. G. Cox, A. B. Forbes, and P. M. Harris. Automatic differentiation and its applications to metrology. Technical Report CMSC 26/03, National Physical Laboratory, June 2003. `http://www.npl.co.uk/ssfm/download/#cmsc26_03`. 4.2.7, 4.2.8

[31] R. Boudjemaa and A. B. Forbes. Parameter estimation methods for data fusion. Technical Report CMSC 38/04, National Physical Laboratory, February 2004. `http://www.npl.co.uk/ssfm/download/#cmsc38_04`. 7.1.1, 11.4

[32] G. E. P. Box and G. M. Jenkins. *Time Series Analysis*. Holden-Day, San Francisco, 1976. 7.2

[33] G. E. P. Box and G. C. Tiao. *Bayesian inference in statistical analysis*. Wiley, New York, Wiley Classics Library Edition 1992 edition, 1973. 5, 4.11.1

[34] R. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, New York, 3rd edition, 1999. 5.3.4

[35] E. O. Brigham. *The Fast Fourier Transform and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1988. 5.3.4

[36] D. S. Broomhead and D. Lowe. Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988. 5.9.1

[37] B. P. Butler. A framework for model validation and software testing in regression. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, III*, pages 158–164, Singapore, 1997. World Scientific. 1.3.2

[38] B. P. Butler, M. G. Cox, S. Ellison, and W. A. Hardcastle, editors. *Statistical Software Qualification*. Royal Society of Chemistry, Cambridge, 1996. 11.1.6

[39] B. P. Butler, M. G. Cox, and A. B. Forbes. The reconstruction of workpiece surfaces from probe coordinate data. In R. B. Fisher, editor, *Design and Application of Curves and Surfaces*, pages 99–116. Oxford University Press, 1994. IMA Conference Series. 4.3.5, 5.11

[40] B. P. Butler, M. G. Cox, A. B. Forbes, P. M. Harris, and G. J. Lord. Model validation in the context of metrology: a survey. Technical Report CISE 19/99, National Physical Laboratory, Teddington, 1999. `http://www.npl.co.uk/ssfm/download/#cise19_99`. 9.1.1

[41] B. P. Butler, A. B. Forbes, and P. M. Harris. Algorithms for geometric tolerance assessment. Technical Report DITC 228/94, National Physical Laboratory, Teddington, 1994. 4.5.3, 4.9.1, 5.10.2

[42] B. P. Butler, A. B. Forbes, and P. M. Harris. Geometric tolerance assessment problems. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools in Metrology*, pages 95–104, Singapore, 1994. World Scientific. 4.5.3, 5.10.2

[43] P. Ciarlini. Bootstrap algorithms and applications. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, III*, pages 12–23, Singapore, 1997. World Scientific. 1.3.2, 3.9

[44] P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors. *Advanced Mathematical and Computational Tools in Metrology, V*, Singapore, 2001. World Scientific. 1.3.2

[45] P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors. *Advanced Mathematical Tools in Metrology*, Singapore, 1994. World Scientific. 1.3.2

[46] P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors. *Advanced Mathematical Tools in Metrology, II*, Singapore, 1996. World Scientific. 1.3.2

[47] P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors. *Advanced Mathematical Tools in Metrology, III*, Singapore, 1997. World Scientific. 1.3.2

[48] P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors. *Advanced Mathematical and Computational Tools in Metrology, IV*, Singapore, 2000. World Scientific. 1.3.2

[49] C. W. Clenshaw. A note on the summation of Chebyshev series. *Math. Tab. Wash.*, 9:118–120, 1955. 5.1.3

[50] C. W. Clenshaw. A comparison of"best" polynomial approximations with truncated Chebyshev series expansions. *SIAM J. Num. Anal.*, 1:26–37, 1964. 5.1.4

[51] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *LANCELOT: a Fortran package for large-scale nonlinear optimization, release A.* Springer-Verlag, Berlin, 1992. 3.6.2

[52] J. W. Cooley and O. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Math. Comput.*, 19:297–301, 1965. 5.3.4

[53] M. G. Cox. The numerical evaluation of B-splines. *Journal of the Institute of Mathematics and its Applications*, 8:36–52, 1972. 5.2.4

[54] M. G. Cox. Cubic-spline fitting with convexity and concavity constraints. Technical Report NAC 23, National Physical Laboratory, Teddington, UK, 1973. 5.1.3

[55] M. G. Cox. The numerical evaluation of a spline from its B-spline representation. *Journal of the Institute of Mathematics and its Applications*, 21:135–143, 1978. 5.2.3, 5.2.4

[56] M. G. Cox. The least squares solution of overdetermined linear equations having band or augmented band structure. *IMA J. Numer. Anal.*, 1:3 – 22, 1981. 4.1.2, 5.2.4

[57] M. G. Cox. Practical spline approximation. In P. R. Turner, editor, *Lecture Notes in Mathematics 965: Topics in Numerical Analysis*, pages 79–112, Berlin, 1982. Springer-Verlag. 5.2.4

[58] M. G. Cox. Linear algebra support modules for approximation and other software. In J. C. Mason and M. G. Cox, editors, *Scientific Software Systems*, pages 21–29, London, 1990. Chapman & Hall. 4.1.2, 4.3.2

[59] M. G. Cox. A classification of mathematical software for metrology. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools for Metrology*, pages 239–246. World Scientific, 1994. 1.3.2

[60] M. G. Cox. Survey of numerical methods and metrology applications: discrete processes. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools for Metrology*, pages 1–22. World Scientific, 1994. 1.3.2, 5.1.3

[61] M. G. Cox. Constructing and solving mathematical models of measurement. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology II*, pages 7–21, Singapore, 1996. World Scientific. 1.3.2, 5.1.3

[62] M. G. Cox. Graded reference data sets and performance profiles for testing software used in metrogy. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, III*, pages 43–55, Singapore, 1997. World Scientific. 1.3.2

[63] M. G. Cox. A discussion of approaches for determining a reference value in the analysis of key-comparison data. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, IV*, pages 45–65, Singapore, 2000. World Scientific. 4.8

[64] M. G. Cox, M. P. Dainton, A. B. Forbes, P. M. Harris, H. Schwenke, B. R. L. Siebert, and W. Woeger. Use of Monte Carlo simulation for uncertainty evaluation in metrology. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology V*, pages 94–106, Singapore, 2001. World Scientific. 1.3.2, 3.9

[65] M. G. Cox and A. B. Forbes. Strategies for testing form assessment software. Technical Report DITC 211/92, National Physical Laboratory, Teddington, December 1992. 5.10.2, 11.7.12

[66] M. G. Cox, A. B. Forbes, J. Flowers, and P. M. Harris. Least squares adjustment in the presence of discrepant data. In *Advanced Mathematical and Computational Tools in Metrology VI*, 2004. To appear. 9.1.2

[67] M. G. Cox, A. B. Forbes, P. M. Fossati, P. M. Harris, and I. M. Smith. Techniques for the efficient solution of large scale calibration problems. Technical Report CMSC 25/03, National Physical Laboratory, Teddington, May 2003. `http://www.npl.co.uk/ssfm/download/#cmsc25_03`. 4.1.2, 4.2.2, 4.3.2, 4.3.5, 5.9.1

[68] M. G. Cox, A. B. Forbes, and P. M. Harris. Software Support for Metrology Best Practice Guide No. 11: Numerical analysis for algorithm design in metrology. Technical report, National Physical Laboratory, Teddington, 2004. `http://www.npl.co.uk/ssfm/download/bpg.html#ssfmbpg11`. 3.8, 5.1.3, 8.1

[69] M. G. Cox, A. B. Forbes, P. M. Harris, and G. N. Peggs. Determining CMM behaviour from measurements of standard artefacts. Technical Report CISE 15/98, National Physical Laboratory, Teddington, March 1998. 5.5.1

[70] M. G. Cox, A. B. Forbes, P. M. Harris, and I. M. Smith. Classification and solution of regression problems for calibration. Technical Report CMSC 24/03, National Physical Laboratory, May 2003. `http://www.npl.co.uk/ssfm/download/#cmsc24_03`. 4.1.6, 4.1.6, 4.3.5, 4.4.1

[71] M. G. Cox and P. M. Harris. Software Support for Metrology Best Practice Guide No. 6: Uncertainty evaluation. Technical report, National Physical Laboratory, Teddington, 2004. `http://www.npl.co.uk/ssfm/download/bpg.html#ssfmbpg6`. 2.4, 2.4.4, 3.9, 7.2, 8.2, 8.2

[72] M. G. Cox and P. M. Harris. Statistical error modelling. NPL report CMSC 45/04, National Physical Laboratory, Teddington, 2004. `http://www.npl.co.uk/ssfm/download/#cmsc45_04`. 2.4, 2.4.6

[73] M. G. Cox, P. M. Harris, and P. D. Kenward. Fixed- and free-knot least-squares univariate data approximation by polynomial splines. NPL report CMSC 13/02, National Physical Laboratory, Teddington, 2002. `http://www.npl.co.uk/ssfm/download/#cmsc13_02`. 5.2.4, 5.5.4

[74] M. G. Cox, P. M. Harris, and P. D. Kenward. Fixed- and free-knot univariate least-squares data approximation by polynomial splines. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 330–345. University of Huddersfield, 2002. 5.2.4

[75] M. G. Cox, P. M. Harris, and Annarita Lazzari. The applicability of non-parametric methods of statistical analysis to metrology. NPL report CMSC 46/04, National Physical Laboratory, Teddington, 2004. `http://www.npl.co.uk/ssfm/download/#cmsc46_04`. 3.9

[76] M. G. Cox and J. G. Hayes. Curve fitting: a guide and suite of algorithms for the non-specialist user. Technical Report NAC 26, National Physical Laboratory, Teddington, UK, 1973. 5.1.3

[77] M. G. Cox and H. M. Jones. A nonlinear least squares data fitting problem arising in microwave measurement. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation II*, London, 1990. Chapman & Hall. 11.7.12

[78] M. G. Cox and J. C. Mason, editors. *Algorithms for Approximation III*, Basel, November 1993. J. C. Baltzer AG. Special issue of *Numerical Algorithms*, volume **5**, nos. 1-4. 1.3.2

[79] M. G. Cox and E. Pardo-Igúzquiza. The total median and its uncertainty. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology V*, pages 106–117, Singapore, 2001. World Scientific. 4.8

[80] R. T. Cox. Probability, frequency, and reasonable expectation. *Amer. J. Phys.*, 4:1–13, 1946. 2

[81] A. Crampton and J. C. Mason. Surface approximation of curved data using separable radial basis functions. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology V*, pages 298–306, Singapore, 2001. World Scientific. 1.3.2

[82] R. S. Dadson, S. L. Lewis, and G. N. Peggs. *The Pressure Balance*. HMSO, London, 1982. 11.6.1

[83] G. B. Dantzig. *Linear programming and extensions*. Princeton University Press, Princeton, N.J., 1963. 4.5.2, 4.6.2

[84] H. F. Davis. *Fourier Series and Orthogonal Functions*. Dover, New York, 1963. 5.3.4

[85] C. de Boor. On calculating with B-splines. *J. Approx. Theory*, 6:50–62, 1972. 5.2.4

[86] J. J. Dongarra and E. Grosse. Distribution of mathematical software via electronic mail. *Communications of the ACM.*, pages 403–407, 1987. `http://www.netlib.org`. 1.3.3, 5.1.4, 5.2.4, 8.2

[87] J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart. *LINPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, 1979. 1, 3.7.1, 4.1.10

[88] J. Du Croz. Relevant general-purpose mathematical and statistical software. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, II*, pages 22–28, Singapore, 1996. World Scientific. 1.3.2

[89] C. F. Dunkl and Y. Xu. *Orthogonal polynomials of several variables*. Cambride University Press, 2001. 5.7.1, 5.7.2

[90] B. Effron. *The Jackknife, the Bootstrap and Other Resampling Plans*. SIAM, Philadelphia, 1982. 3.9

[91] EUROMET, `http://www.euromet.org`. *A European collaboration in measurement standards*. 2

[92] *EUROMETROS — EUROMET Repository of Software*. `http://www.eurometros.org`. 4.3.5, 5.1.4, 5.2.4, 5.10.2

[93] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 1992. 5.11.1

[94] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, Chichester, second edition, 1987. 1.3.1, 4.2.2, 4.2.8, 4.5.3

[95] A. B. Forbes. Fitting an ellipse to data. Technical Report DITC 95/87, National Physical Laboratory, Teddington, 1987. 5.10.2

[96] A. B. Forbes. Least-squares best-fit geometric elements. Technical Report DITC 140/89, National Physical Laboratory, Teddington, 1989. 4.3.5, 5.10.2, 11.1.3

[97] A. B. Forbes. Robust circle and sphere fitting by least squares. Technical Report DITC 153/89, National Physical Laboratory, Teddington, 1989. 5.10.2, 11.7.12

[98] A. B. Forbes. Least squares best fit geometric elements. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation II*, pages 311–319, London, 1990. Chapman & Hall. 4.3.2, 4.3.5, 5.10.2

[99] A. B. Forbes. Geometric tolerance assessment. Technical Report DITC 210/92, National Physical Laboratory, Teddington, October 1992. 4.5.3, 4.9.1

[100] A. B. Forbes. Generalised regression problems in metrology. *Numerical Algorithms*, 5:523–533, 1993. 4.3.1, 4.3.5

[101] A. B. Forbes. Mathematical software for metrology – meeting the metrologist's needs. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools in Metrology*, pages 247–254, Singapore, 1994. World Scientific. 1.3.2

[102] A. B. Forbes. Validation of assessment software in dimensional metrology. Technical Report DITC 225/94, National Physical Laboratory, February 1994. 5.10.2

[103] A. B Forbes. Model parametrization. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools for Metrology*, pages 29–47, Singapore, 1996. World Scientific. 1.3.2, 5.11, 11.7.12

[104] A. B. Forbes. Efficient algorithms for structured self-calibration problems. In J. Levesley, I. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 146–153. University of Huddersfield, 2002. 4.3.2, 4.3.5

[105] A. B. Forbes and P. M. Harris. Estimation algorithms in the calculation of the effective area of pressure balances. *Metrologia*, 36(6):689–692, 1999. 11.6.2, 11.6.3

[106] A. B. Forbes and P. M. Harris. Simulated instruments and uncertainty estimation. Technical Report CMSC 01/00, National Physical Laboratory, 2000. `http://www.npl.co.uk/ssfm/download/#cmsc01_00`. 7.1.1

[107] A. B. Forbes, P. M. Harris, and I. M. Smith. Generalised Gauss-Markov Regression. In J. Levesley, I. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 270–277. University of Huddersfield, 2002. 4.3.5

[108] A. B. Forbes, P. M. Harris, and I. M. Smith. Correctness of free form surface fitting software. In D. G. Ford, editor, *Laser Metrology and Machine Performance VI*, pages 263–272, Southampton, 2003. WIT Press. 4.3.5, 5.10.2

[109] G. E. Forsythe. Generation and use of orthogonal polynomials for data fitting with a digital computer. *SIAM Journal*, 5:74–88, 1957. 5.1.3, 5.1.4, 5.4.2

[110] G. E. Forsythe, M. A. Malcolm, and C. B. Moler. *Computer Methods for Mathematical Computation*. Prentice-Hall, Englewood Cliffs, 1977. 1.3.1

[111] L. Fox and I. B. Parker. *Chebyshev polynomials in numerical analysis*. Oxford University Press, 1968. 5.1.4

[112] M. Frigo and S. G. Johnson. FFTW: An adaptive software architecture for the FFT. In *Proc. 1998 IEEE Intl. Conf. Acoustics Speech and Signal Processing*, volume 3, pages 1381–1384. IEEE, 1998. 5.3.4

[113] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):845–848, 1989. 5.9.1

[114] W. Gander, G. H. Golub, and R. Strebel. Least squares fitting of circles and ellipses. *BIT*, 34, 1994. 5.10.2

[115] B. S. Garbow, K. E. Hillstrom, and J. J. Moré. User's guide for MINPACK-1. Technical Report ANL-80-74, Argonne National Laboratory, Argonne, IL, 1980. 1, 4.1.10, 4.2.2, 4.2.8

[116] W. M. Gentleman. An error analysis of Goertzel's (Watt's) method for computing Fourier coefficients. *Comput. J.*, 12:160–165, 1969. 5.1.3

[117] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, 1981. 1.3.1, 3.6.2, 4.2.8, 4.5.3, 5.9.1

[118] G. H. Golub. The singular value decomposition with applications. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, II*, pages 48–55, Singapore, 1996. World Scientific. 1.3.2

[119] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, third edition, 1996. 1.3.1, 4.1.2, 4.1.2, 8, 4.1.10, 4.3.5, 8.3.1

[120] A. Greenbaum. *Iterative methods for solving linear systems*. SIAM, Philadelphia, 1997. 5.9.1

[121] A. Griewank and G. F. Corliss, editors. *Automatic Differentiation of Algorithms: Theory, Implementation and Applications*, Philadelphia, 1991. Society for Industrial and Applied Mathematics. 4.2.8

[122] S. Hammarling. The numerical solution of the general Gauss-Markov linear model. Technical Report TR2/85, Numerical Algorithms Group, Oxford, 1985. 4.1.6

[123] D. C. Handscombe and J. C. Mason. *Chebyshev Polynomials*. Chapman&Hall/CRC Press, London, 2003. 5.1.4

[124] P. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM J. Sci. Stat. Comp.*, 34(4):561–580, 1992. 5.9.1

[125] P. Hansen. Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. *Num. Alg.*, 6:1–35, 1994. 5.9.1

[126] R. J. Hanson and K. H. Haskell. Algorithm 587: two algorithms for linearly constrained least squares problems. *ACM Trans. Math. Soft.*, 8(3):323–333, 1982. 4.1.10

[127] P. M. Harris. The use of splines in the modelling of a photodiode response. Technical Report DITC 88/87, National Physical Laboratory, Teddington, UK, 1987. 5.5.1

[128] *Harwell Subroutine Library: a Catalogue of Subroutines*. Harwell Laboratory, England. 1

[129] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan, New York, second edition, 1999. 5.9.1, 5.9.1

[130] H.-P. Helfrich and D. Zwick. A trust region method for implicit orthogonal distance regression. *Numerical Algorithms*, 5:535 – 544, 1993. 4.3.5

[131] H.-P. Helfrich and D. Zwick. Trust region algorithms for the nonlinear distance problem. *Num. Alg.*, 9:171 – 179, 1995. 4.3.5

[132] H.-P. Helfrich and D. Zwick. A trust region method for parametric curve and surface fitting. *Journal of Computational and Applied Mathematics*, 73:119 – 134, 1996. 4.3.5

[133] H.-P. Helfrich and D. Zwick. ℓ_1 and ℓ_∞ fitting of geometric elements. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 162–169. University of Huddersfield, 2002. 4.3.5, 4.5.3

[134] N. J. Higham. *Accuracy and numerical stability of algorithms*. SIAM, Philadelphia, PA, second edition, 2002. 8.1

[135] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. 5.9.1

[136] P. J. Huber. Robust estimation of a location parameter. *Ann. Math. Stat.*, 35:73–101, 1964. 4.8

[137] P. J. Huber. *Robust Statistics*. Wiley, New York, 1980. 4.8

[138] M. Huhtanen and R. M Larsen. On generating discrete orthogonal bivariate polynomials. *BIT*, 42:393–407, 2002. 5.7.1, 5.7.2, 11.10

[139] ISO. ISO 3534 statistics – vocabulary and symbols – part 1: probability and general statistical terms. Technical report, International Standards Organisation, Geneva, 1993. 2.4

[140] D. P. Jenkinson, J. C. Mason, A. Crampton, M. G. Cox, A. B. Forbes, and R. Boudjemaa. Parameterized approximation estimators for mixed noise distributions. In *International Conference on Advanced Mathematical and Computational Tools in Metrology VI, Torino, September, 2003*, 2003. Invited paper. 4.7.4

[141] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley and Sons, eighth edition, 1999. 5.3.4

[142] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, 1974. 1.3.1, 4.1.10

[143] D. Lei, I. J. Anderson, and M. G. Cox. An improve algorithm for approximating data in the ℓ_1 norm. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools for Metrology V*, pages 247–250, Singapore, 2001. World Scientific. 4.6.3

[144] D. Lei, I. J. Anderson, and M. G. Cox. A robust algorithm for least absolute deviation curve fitting. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 470–477. University of Huddersfield, 2002. 4.6.3

[145] J. Levesley, I. J. Anderson, and J. C. Mason, editors. *Algorithms for Approximation IV*. University of Huddersfield, 2002. 1.3.2

[146] G. L. Lord, E. Pardo-Igúzquiza, and I. M. Smith. A practical guide to wavelets for metrology. Technical Report NPL Report CMSC 02/00, National Physical Laboratory, Teddington, June 2000. `http://www.npl.co.uk/ssfm/download/#cmsc02_00`. 5.6.1

[147] T. Lyche and K. Mørken. A discrete approach to knot removal and degree reduction for splines. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation*, pages 67–82, Oxford, 1987. Clarenden Press. 5.2.4

[148] Z. A. Maany. Building numerical libraries using Fortran 90/95. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology IV*, pages 143–156. World Scientific, 2000. 1.3.2, 1.3.3

[149] P. Maas. Wavelet methods in signal processing. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, III*, pages 43–55, Singapore, 1997. World Scientific. 1.3.2

[150] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, London, 1979. 1.3.1, 3.3, 4.1.9

[151] J. C. Mason and M. G. Cox, editors. *Algorithms for Approximation*, Oxford, 1987. Clarenden Press. 1.3.2

[152] J. C. Mason and M. G. Cox, editors. *Algorithms for Approximation II*, London, 1990. Chapman & Hall. 1.3.2

[153] J. C. Mason and D. A. Turner. Applications of support vector machine regression in metrology and data fusion. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computation Tools in Metrology V*, Singapore, 2001. World Scientific. 1.3.2

[154] MathSoft, Inc., Cambridge, MA. *MathCad 2000*. `http://www.mathsoft.com`. 1.3.3

[155] MathSoft, Inc, Seattle, WA. *S-PLUS 2000 Guide to Statistics, Volumes 1 and 2*, 1999. `http://www.mathsoft.com`. 1.3.3

[156] MathWorks, Inc., Natick, Mass. *Using Matlab*, 2002. `http://www.mathworks.com`. 1.3.3, 4.1.10, 4.5.3, 5.2.4

[157] M. Metcalf and J. Reid. *Fortran 90/95 Explained*. Oxford University Press, 1996. 1.3.3

[158] H. S. Migon and D. Gamerman. *Statistical inference: an integrated approach*. Arnold, London, 1999. 4.11.1, 4.11.2

[159] M. L. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969. 5.9.1

[160] J. J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In G. A. Watson, editor, *Lecture Notes in Mathematics 630*, pages 105–116, Berlin, 1977. Springer-Verlag. 4.2.2

[161] J. J. Moré and S. J. Wright. *Optimization Software Guide*. SIAM, Philadelphia, 1993. 3.6.2, 4.2.8, 4.5.3, 4.8

[162] N. Morrison. *Introduction to Fourier Analysis*. Wiley, New York, 1994. 5.3.4

[163] W. Murray and M. L Overton. A projected Lagrangian algorithm for nonlinear minimax optimization. *SIAM Journal for Scientific and Statistical Computing*, 1(3):345–370, 1980. 4.9.1

[164] W. Murray and M. L. Overton. A projected Lagrangian algorithm for nonlinear ℓ_1 optimization. *SIAM Journal for Scientific and Statistical Computing*, 2:207–224, 1981. 4.9, 4.9.1

[165] J. C. Nash. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation, Second Edition*. Adam Hilger, Bristol & American Institute of Physics, New York, 1990. 1.3.1

[166] National Instruments, Corp., Austin, TX. *LabVIEW*. `http://www.ni.com/`. 1.3.3

[167] National Physical Laboratory, `http://www.npl.co.uk/ssfm/`. *Software Support for Metrology Programme*. 1.3.4

[168] J. A. Nelder and R. Mead. A simplex method for function minimization. *Comp. J.*, 7:308–313, 1965. 4.5.2

[169] G. L. Nemhhauser, A. H. G. Rinnooy Kan, and M. J. Todd, editors. *Handbooks in Operations Research and Management Science, Volume 1: Optimization*, Amsterdam, 1989. North-Holland. 1.3.1

[170] NIST, `http://gams.nist.gov`. *GAMS: guide to available mathematical software*. 1.3.3

[171] NIST/SEMATECH, `http://www.itl.nist.gov/div898/handbook/`. *e-Handbook of Statistical Methods*. 1.3.3

[172] *NPLFit — Software for fitting polynomials and polynomial splines to experimental data*. `http://www.eurometros.org/packages/#nplfitlib`. 5.1.4, 5.2.4

[173] The Numerical Algorithms Group Limited, Wilkinson House, Jordan Hill Road, Oxford, OX2 8DR. *The NAG Fortran Library, Mark 20, Introductory Guide*, 2002. `http://www.nag.co.uk/`. 1, 4.1.10, 4.2.8, 4.5.3, 4.9.1, 5.1.4, 5.2.4, 8.2

[174] The Numerical Algorithms Group Limited, Wilkinson House, Jordan Hill Road, Oxford, OX2 8DR. *The NAG Fortran 90 Library*, 2004. `http://www.nag.co.uk/`. 1.3.3

[175] M. J. L. Orr. Introduction to radial basis function networks. Technical report, Centre for Cognitive Science, University of Edinburgh, April 1996. 5.9.1

[176] M. J. L. Orr. Recent advances in radial basis function networks. Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh, June 1999. 5.9.1

[177] M. R. Osborne and G. A. Watson. An algorithm for minimax approximation in the nonlinear case. *Computer Journal*, 12:63–68, 1969. 4.9, 4.9.1

[178] M. R. Osborne and G. A. Watson. On an algorithm for non-linear l_1 approximation. *Computer Journal*, 14:184–188, 1971. 4.9, 4.9.1

[179] C. C. Paige. Fast numerically stable computations for generalized least squares problems. *SIAM J. Numer. Anal.*, 16:165–171, 1979. 4.1.6

[180] C. C. Paige and M. A. Saunders. LSQR: and algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1), 1982. 4.1.2, 4.1.10

[181] F. Pavese et al., editors. *Advanced Mathematical and Computational Tools in Metrology, VI*, Singapore, 2004(?). World Scientific. Turin, 8-12th September , 2003. 1.3.2

[182] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, NY, 2nd edition, 1996. 5.11.1

[183] M. J. D. Powell. *Approximation Theory and Methods*. Cambridge University Press, Cambridge, 1981. 1.3.1, 4.5.1, 4.5.3, 4.6.3, 5.1.4

[184] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes: the Art of Scientific Computing*. Cambridge University Press, Cambridge, 1989. `http://www.nr.com`. 1.3.1

[185] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran 90*. Cambridge University Press, Cambridge, 1996. 1.3.3

[186] D. Rayner and R. M. Barker. METROS – a website for algorithms for metrology and associated guidance. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology V*, pages 298–306, Singapore, 2001. World Scientific. 1.3.2

[187] J. A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, Belmont, CA, second edition, 1995. 1.3.1, 4.10.3, 7.2

[188] J. R. Rice and R. J. Hanson. References and keywords for Collected Algorithms from ACM. *ACM Trans. Math. Softw.*, 10(4):359–360, December 1984. 1.3.3

[189] C. Ross, I. J. Anderson, J. C. Mason, and D. A. Turner. Approximating coordinate data that has outliers. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology IV*, pages 210–219. World Scientific, 2000. 4.8

[190] SIAM, Philadelphia. *The LAPACK User's Guide*, third edition, 1999. 1, 3.7.1, 4.1.6, 4.1.6, 4.1.10

[191] B. R. L. Siebert. Discussion of methods for the assessment of uncertainties in Monte Carlo particle transport calculation. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, IV*, pages 220–229, Singapore, 2000. World Scientific. 3.9

[192] D. S. Sivia. *Data analysis: a Bayesian tutorial*. Clarendon Press, Oxford, 1996. 4.11.1

[193] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler. *Matrix Eigensystems Routines - EISPACK Guide*. Springer-Verlag, New York, 1977. Lecture Notes in Computer Science, Vol. 51. 1

[194] D. Sourlier and W. Gander. A new method and software tool for the exact solution of complex dimensional measurement problems. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, II*, pages 224–237, Singapore, 1996. World Scientific. 5.10.2

[195] Mark Spink. *NURBS toolbox for Matlab*. `http://www.aria.uklinux.net/nurbs.php3`, 2000. 5.11.1

[196] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Rev.*, 40:110–112, 1998. 4.2.8

[197] StatLib, Statistics Department, Carnegie-Mellon University, `http://lib.stat.cmu.edu/`. *StatLib — Data, Software and News from the Statistics Community*. 1.3.3, 8.2

[198] G. Szego. *Orthogonal Polynomials*. American Mathematical Society, New York, 1959. 5.1.3, 5.1.4

[199] A. N. Tikhonov and V. Y. Arsenin. *Solutions to Ill–Posed Problems*. Winston and Sons, Washington D. C., 1977. 5.9.1

[200] S van Huffel, editor. *Recent Advances in Total Least Squares and Errors-in-Variables Techniques*, Philadelphia, 1997. SIAM. 4.3.5

[201] S. van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. SIAM, Philadelphia, 1991. 4.3.5

[202] D. Vecchia and J. D. Splett. Outlier-resistant methods for estimation and model fitting. In P. Ciarlini, M. G. Cox, R. Monaco, and F. Pavese, editors, *Advanced Mathematical Tools in Metrology*, pages 143–154. World Scientific, 1994. 4.8

[203] Visual Numerics, Inc., 12657 Alcosta Boulevard, Suite 450, San Ramon, CA 94583 USA. *IMSL Fortran numerical library, version 5.0*. `http://www.vni.com/`. 1, 4.1.10, 4.2.8, 4.5.3, 4.9.1, 5.1.4, 5.2.4

[204] G. A. Watson. *Approximation Theory and Numerical Methods*. John Wiley & Sons, Chichester, 1980. 1.3.1, 4.5.1, 4.5.3, 4.6.1, 4.6.3, 4.9, 4.9.1, 5.1.4

[205] G. A. Watson. Some robust methods for fitting parametrically defined curves or surfaces to measured data. In P. Ciarlini, A. B. Forbes, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology IV*, pages 256–272. World Scientific, 2000. 4.3.5, 4.8

[206] B. A. Wichmann, G. Parkin, and R. M. Barker. Software Support for Metrology Best Practice Guide No. 1: Validation of software in measurement systems. Technical report, National Physical Laboratory, Teddington, 2004. `http://www.npl.co.uk/ssfm/download/bpg.html#ssfmbpg1`. 6.2

[207] J. H. Wilkinson and C. Reinsch. *Handbook of Automatic Computation Volume II: Linear Algebra*. Springer-Verlag, Berlin, 1971. 1.3.1, 4.1.10

[208] S. Wolfram. *The Mathematica Book*. Cambridge University Press, Cambridge, third edition, 2000. 1.3.3

[209] Wolfram Research, Inc., 100 Trade Center Drive, Champaign, IL 61820-7237, USA. *Mathematica — the way the world calculates*. `http://www.wolfram.com/mathematica/`. 1.3.3

[210] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *Trans. Math. Soft*, 23(4), 1997. 3.6.2, 5.9.1

[211] D. Zwick. Algorithnms for orthogonal fitting of lines and planes: a survey. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical Tools in Metrology, II*, pages 272–283, Singapore, 1996. World Scientific. 5.10.2, 11.1.6

Index