

Methodologies and Software for PEST-Based Model Predictive Uncertainty Analysis

by

John Doherty
Watermark Numerical Computing
September 2010

Support for the writing of this document was provided by South Florida Water Management District.

Table of Contents

1. Introduction.....	1
General.....	1
This Document.....	3
Case Studies.....	5
2. What will happen if...?.....	13
Introduction.....	13
Making a Decision.....	13
Environmental Management.....	14
Reducing Model Augmentations to Uncertainty.....	20
The Scientific Method.....	23
Summary.....	23
3. Models, Simulation and Uncertainty.....	25
Expert Knowledge.....	25
What a Model can Provide.....	25
What an Uncalibrated Model can Provide.....	26
Linear Analysis.....	28
Exercises.....	30
4. Getting Information out of Data.....	39
History-matching.....	39
Bayes Equation.....	39
Calibration.....	40
The Null Space.....	42
Regularisation.....	46
Structural Noise.....	54
Exercises.....	56
5. How Wrong Can a Prediction Be? Linear Analysis.....	76
Error and Uncertainty.....	76
The Predictive Error Term.....	78
Linear Parameter and Predictive Uncertainty Analysis.....	79
Linear Parameter and Predictive Error Analysis.....	81
Over-Determined Parameter Estimation.....	83
Derived Quantities.....	84
Exercises.....	88
6. How Wrong can a Prediction Be? Nonlinear Analysis.....	102
Error and Uncertainty.....	102
Constraints.....	103
Well-Posed Inverse Problems.....	105
Ill-Posed Inverse Problems.....	107
Exercises.....	111
7. Hypothesis-Testing and Pareto Methods.....	125
Where are we at?.....	125
Where do we go from here?.....	126
The Scientific Method.....	127
The Role of Model Calibration.....	128
Pareto Concepts - Model Calibration.....	131
Pareto Concepts - Model Prediction.....	133
Pareto Methods - Some Final Words.....	135
Exercises.....	135

8. Conclusions.....	146
9. References.....	147
Appendix 1. PEST Utilities.....	149
Appendix 2. PEST Groundwater Data Utilities	153

1. Introduction

General

It is my hope that this document is more exciting than its title. It has been written with this intent, though in the eyes of many it will no doubt fail in this regard. After all, it is not a detective story, nor even a history of a journey of discovery.

Yet, in some respects it is both of these.

Part of the aim of this document is to help modellers, and those who make decisions on the basis of modelling, to look at modelling through fresh eyes. It is written against a background of what the author sees as frequent and expensive misuse of what is fundamentally a very useful technology. Sadly, the costs of this misuse can be high. These costs include human resources that are wasted on expensive yet fruitless modelling exercises that do not provide the insights into the future that was promised as justification for their construction. However even greater costs are incurred by making decisions on the basis of model outcomes that are thought to provide robust insights into the future but that, in fact, provide no such thing.

Let us briefly look at the context in which most environmental modelling takes place. A decision of some importance must be made. The outcome of this decision is that someone may invest a great deal of money in a venture that has the potential to cause damage to someone else's investment, with this damage being propagated through environmental pathways. Or damage may be inflicted on the environment itself. Alternatively, someone may be denied the opportunity to make such an investment because it is judged that the consequences of this investment will be damage inflicted on other parties, or on the environment. The costs of modelling are normally small compared with those associated with the projects whose implementation and design are based on modelling outcomes. This is where the real cost of less-than-optimal modelling is incurred. And that cost may be very large indeed. In recognition of this, the topic of model predictive uncertainty analysis is gaining increasing interest at a rapid rate.

The subject of model uncertainty has never been absent from the modelling literature. However it is also fair to say that it has never been a "headline topic". It has always maintained the interest of some, but has never received the cult status that some other topics seem to have received from time to time. Even a casual inspection of the academic literature reveals large differences in the means through which model predictive uncertainty is explored by different groups of researchers. In some cases there is considerable overlap between these approaches. In other cases different approaches appear to have very little in common. Nowhere does the divide appear to be greater than between approaches taken by groundwater and surface water modellers; even the vocabularies used by these groups are different. And just to add to the confusion, the mathematics of uncertainty often appears complicated. Where this is superimposed on the mathematics of simulation and published as a paper in an academic journal, that paper is sure to be given a wide berth by all except the most learned spectators.

Lately, however, much of the growth in interest of this topic appears to be coming from outside of the academic community. It appears to be coming from those who must make decisions, or who must convince others of the benefits of decisions that have already been taken. Inevitably, a model has formed part of the decision-making process; these days, that is almost inescapable. Almost inevitably, the decision will be challenged by individuals or

groups who see themselves as disadvantaged by the decision. In mounting the challenge, the focus of attack will almost certainly be the model. The model must therefore be “defensible”.

In the author’s opinion very few models are “defensible” - at least not in the way that they are traditionally defended. This is an outcome of the fact that in most cases there is no mathematical reason why environmental models can live up to the expectations that are placed upon them. Nevertheless a model will often survive the onslaughts of its detractors either because the instrument of attack is an equally indefensible model, or because no better alternative is available for scientifically-based decision-support. In the heat of battle the rhetoric of modelling rather than the science of modelling often decides the contest. This is partly due to the fact that there is no metric by which to independently judge the worth of a model, or the superior worth of one model over another. Where metrics are put forward by proponents of one side of an argument they are often spurious, with explicit reliance often being placed on whether one model is “better calibrated” than another model, and implicit reliance often being placed on the aesthetic appeal with which model outcomes are presented.

In recognition of this unsatisfactory state of affairs, battle-hardened decision-makers and managers are rightly turning their attention to the concept of model uncertainty. In part this, like all other human behaviour, has its roots in self interest; where models are built to support the making of expensive and controversial decisions, the state of defensibility of the primary mechanism for decision-support must be known. So too must the vulnerabilities of the models that are used to attack it. In part, however, it also arises from a growing feeling, born of witnessing too many occasions on which reality has recklessly gone its own way independently of model predictions, that managers are being “sold a lemon” when they agree to pay for an expensive model. There is a growing realization that models are not quite the predictive wonders that they are often made out to be.

So who is to blame for this situation? No-one in my opinion. Decisions-makers have wanted predictive certainty since decision-makers first existed. In the past they grasped the straws that were available to them at the time - oracles, signs, astrology, and other tools of the mystical trade. Today those straws are models. And modellers have been eager to please managers, or have evolved through natural selection to be eager to please managers, by presenting them with oracles; for career extinction awaits those who did not.

However the times, they are rapidly changing. As an industry, we are on the threshold of a paradigm shift in the way we build and use models. It is the growing recognition, born of painful experience, that model predictions may be seriously wrong that has led us to this point. This presents us with many dilemmas, the first of which is, of course, how to calculate the potential for wrongness that is associated with predictions made by a model. However as soon as this problem is addressed, an even greater dilemma awaits us. It is how to use models as a basis for decision-making when it is openly admitted that they cannot be construed as instruments for divining the future.

While it is the intention of this document to address the first of these issues rather than the second, neither of them can be addressed in isolation. Hence, instead of looking at the issue of uncertainty as yet another layer that must be superimposed on the existing conceptual edifice that has formed the basis for model deployment up until now, this document begins by briefly discussing the place that models occupy in the decision-making process. This forms the context in which they must be used. It is thus the context to which their performance must be tuned, and the context in which their imperfections must be recognized. This is also the context in which uncertainty must be embraced as an inevitable part of looking into the future, and as an inseparable aspect of any decision-making process.

This Document

This document is not meant to serve as a comprehensive review of work that has been carried out to date in the field of model predictive uncertainty analysis, in spite of the fact that reference is made to some of it. Nor does it purport to provide a comprehensive mathematical treatment of model predictive uncertainty analysis, despite the fact that some equations are presented. Instead, it intends to achieve the following.

- Provide an overview of impediments to our ability to predict the environmental future;
- Show that while we cannot be certain about the future, the magnitude of our uncertainty is, at to at least some extent, quantifiable;
- Show how uncertainty can be reduced through appropriate use of simulation software in partnership with software that facilitates the flow of information and ideas from environmental data and user expertise to these simulators;
- Illustrate that, while there is a theoretical lower limit to the uncertainty associated with predictions of different types at a specific study site, that limit may be difficult to attain because of model imperfections and practical computing requirements;
- Identify strategies that can be used to approach the theoretical lower limit of uncertainty for a particular prediction at a particular site.

Practical demonstrations of concepts discussed herein will be provided. Many of the following chapters conclude with worked examples based on two simple models used in conjunction with programs of the PEST suite of software. Files for these exercises are supplied. Hence, in addition to the above roles, this document also provides a tutorial on use of PEST-suite programs in analysing model predictive uncertainty.

It is important to point out, however, that an over-riding consideration in writing this document has been that it be easy to read by both experts and non-experts alike. As stated above, while some equations are presented, the use of mathematics is kept to a minimum. Where equations are presented, it is not essential that they be understood - only that the principles that underlie them be understood.

Similar considerations hold for the practical exercises. A reader of this document may ignore them altogether if he/she wishes, and nevertheless learn much about approaches to model predictive uncertainty analysis that are encapsulated in the PEST suite of software. Alternatively, a reader may wish to increase his/her understanding of PEST-suite software by simply reading the example descriptions and glancing at the files to which these descriptions pertain. Or a reader may wish to follow all of the instructions provided through the examples, thereby acquiring maximum knowledge of the workings of PEST and its ancillary support software in the uncertainty analysis context.

This document is organised as follows.

The remainder of this chapter provides a brief description of the two models on which practical examples used throughout this text are based.

Chapter 2 attempts to set the context for the discussions that follow by examining what is required of a model when it is used as a basis for environmental decision-making. It attempts to take a somewhat different view of modelling from that which often explicitly or implicitly accompanies model usage at the present time. In particular, the role of a model when used in conjunction with appropriate support software as an instrument through which scientific

hypotheses-testing can be implemented is emphasized. An incapacity to reject the hypothesis that a particular management strategy may have unwelcome consequences may constitute grounds for making the decision to implement an alternative management strategy.

Chapter 3 looks at what simulation of environmental processes as they operate at a particular study site can achieve. It makes the point that a model can never promise a prediction that is correct. However what it can aspire to do is guarantee that the correct prediction will lie within the interval of predictive uncertainty that it provides, and that this uncertainty approaches its theoretical lower limit given currently available information. Unfortunately, however, this lower limit may not be attainable because of the imperfect and simplistic nature of models as simulators of environmental processes. Model imperfections result in misdirection of information that is resident in historical system datasets. This, in turn, can lead to an increased penchant for model predictive error that must be accounted for when using a model to calculate predictive confidence intervals.

Chapter 4 explores the concept of model calibration. It is demonstrated that calibration does not endow a model with an ability to provide the “the right answer” when it is used to make a prediction of future environmental behaviour. It may, however, reduce the potential for error associated with one or a number of model predictions; alternatively it may not. In either case, calibration provides a pathway through which information can flow from data gathered at a study site to a simulator of environmental processes for that site. It is demonstrated that calibration also provides insights into model inadequacies, and the effects of these inadequacies on the model’s ability to reproduce past and future environmental behaviour. A distinction is made between model predictive uncertainty analysis and model predictive error analysis. While the two are often used interchangeably, the potential for predictive error is usually higher than the innate uncertainty of a prediction; uncertainty is an outcome of information inadequacy while potential for error includes both uncertainty and the effects on model predictions of its flawed capacity to simulate environmental processes.

Chapter 5 discusses linear analysis of model predictive uncertainty and error. It is shown that while linear analysis can only be approximate because the relationship between model outcomes and model parameters is in fact nonlinear, linear analysis can nevertheless provide some useful insights. In particular, a modeller can rapidly assess the extent to which data inadequacy and model imperfections detract from a model’s ability to predict future environmental behaviour. In addition to this, the worth of existing or yet-to-be acquired data can be assessed in terms of its ability (or otherwise) to reduce the uncertainty associated with specific model predictions. Such analysis can therefore provide a sound basis for investment in acquisition of further data at a particular study site.

Chapter 6 treats nonlinear predictive uncertainty analysis. While far more general than linear analysis as it does not require an assumption of linear model behaviour, this generality sometimes comes with a heavy computational cost. Nevertheless, through the use of methodologies such as the null space Monte Carlo scheme provided by PEST, the uncertainty associated with model predictions can be explored with reasonable levels of computational efficiency, even where parameter variability is subject to calibration constraints, and even where the number of parameters attributed to a model is made purposefully large in order to preclude underestimation of the extent of predictive variability.

Chapter 7 explores in detail one particular form of nonlinear uncertainty analysis that implements more-or-less directly the type of hypothesis-testing that forms the heart of the scientific method. Furthermore, it does this in a way that makes enough information available to the modeller for him/her to be capable of exercising necessarily subjective (though informed) expert judgement when assessing the likelihood or otherwise of a particular future

environmental occurrence. As such this methodology may find a useful role in contexts of collaborative decision-making that are becoming more and more widespread as attempts are made to reconcile the interests of different stakeholder groups when deciding on ways to manage the environment in a manner that maximises its benefit to all.

Chapter 8 provides a short conclusion.

Case Studies

General

Two cases are provided to illustrate use of PEST-suite software in analysis of predictive uncertainty. Most chapters of this document conclude with a description of how to run pertinent PEST utilities in conjunction with these cases. All required files are supplied so that the reader can run these programs him/herself, or simply inspect pertinent input and output files, if he/she wishes. Descriptions are necessarily brief; more comprehensive tutorials on use of PEST and its utility support software are available through PEST courses.

Both of the example cases discussed herein are characterised by models that run quickly. Hence the exercises provided in this document can be implemented with minimal computation effort. The first is a surface water modelling example, while the second is a groundwater modelling example. The first represents a well-posed inverse problem whereas the second represents an ill-posed inverse problem. As will be discussed, the well- or ill-posedness of the inverse problem that characterises model calibration has a profound effect on the ways in which predictive error/uncertainty arises, and on the manner in which it must be explored.

If a reader is not interested in gaining hands-on PEST experience based on the examples provided, pertinent sections of this document can remain unread with no deleterious impact on the flow of ideas that this document attempts to provide.

All files for the first of these examples have been placed in a folder named *sw* (for “surface water”), whereas all files for the second of these examples have been placed in a folder named *gw* (for “groundwater”). To save the trouble of installation, all executable programs required for the exercises are also provided in each of these folders.

Surface Water Model

General

This model is similar to that described by Gallagher and Doherty (2006) and Moore et al. (2010). Flow of water within a watershed is simulated using the Hydrologic Simulation Program FORTRAN (HSPF) model. This model is widely used in water quality studies; however in the present case only those aspects of its functionality that are related to the quantity, rather than quality, of water are employed. The HSPF model is fully described by Bicknell et al. (2001).

As is apparent from the HSPF input file *example.uci* the model runs for three years, this covering the period 1st January 1984 to 31st December 1986. However the first year of the simulation is used for warm-up while calibration is effected by matching modelled and observed flows over 1985. The model is then used to predict flows in 1986. The maximum flow occurring over a particular four day period during 1986 is treated as a prediction of interest whose uncertainty is analysed. Partial assessment of each uncertainty analysis methodology discussed herein is achieved through testing whether the maximum flow that was actually measured over that period lies within computed uncertainty limits.

Figure 1.1 shows observed flows at the pour point of the watershed over 1985 and 1986, together with rainfall over that same two-year period. An arrow signifies 20th August 1986. This is the day for which predicted flow will form the focus of uncertainty analysis. (Actually, in making the prediction, we will look at the maximum flow over a four day period which encompasses this day to ensure that we capture the small peak associated with this event, as the magnitude of the peak is the aspect of flow that is of greatest interest to us.) The recorded flow of $27.25 \text{ m}^3/\text{sec}$ on that day constitutes a minor peak preceding a much larger flow event. What makes prediction of the magnitude of this minor peak difficult is the fact that rainfall that gave rise to the peak followed a lengthy dry period. The magnitude of the consequential flow event is therefore heavily dependent on the status of landscape water storages - especially those that are close to the stream. Given the lumped nature of representation of these storages in HSPF (and other lumped-parameter regional rainfall-runoff models), this is thus a prediction that is particularly prone to the deleterious effects of model structural defects. As will be shown in later sections, notwithstanding its calibrated status, the model gets this prediction very wrong. This does not constitute grounds for condemnation of the model however, for it is wrong to expect that a model should get a prediction right. What does incriminate the model is that the correct prediction does not lie within the model's predictive uncertainty interval.

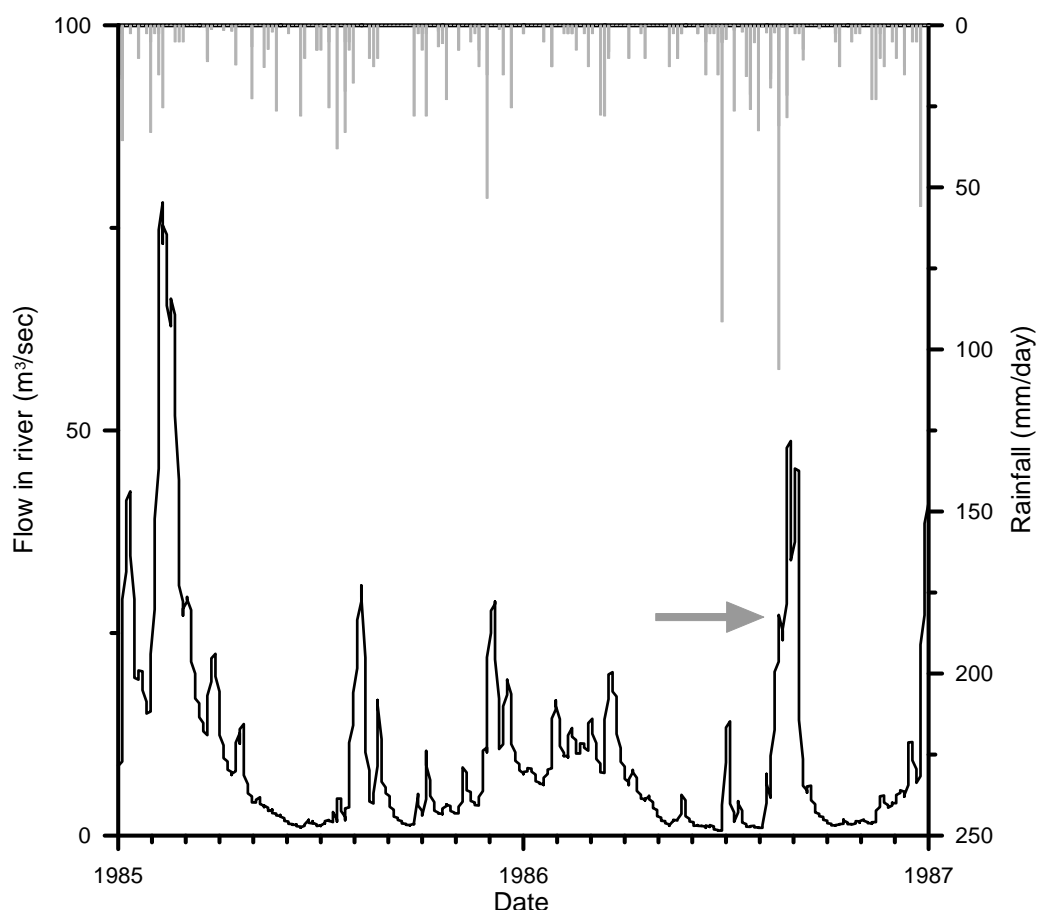


Figure 1.1 Rainfall and river flow over the years 1985 and 1986; the arrow depicts an event on 20th August 1986.

Model Parameters

A single PERLND (pervious land area) and a single RCHRES (reach reservoir unit) are used to simulate hydrologic processes within the watershed. Only parameters pertaining to the former are calibrated; these are listed in Table 1.1. The third column of this table lists bounds on parameter values imposed through the parameter estimation and predictive analysis processes described below; these were taken from USEPA (2000). Note that HSPF employs many parameters in addition to those listed in Table 1.1; however parameter estimation was limited to this group of parameters based on the importance of these parameters in streamflow calculation. With the inverse problem of model calibration restricted to the estimation of only these seven parameters, the resulting problem is well-posed.

Parameter Name	Parameter Function	Bounds
LZSN	Lower zone nominal storage	50 - 380 mm
UZSN	Upper zone nominal storage	1.3 – 51.0 mm
INFILT	Related to the infiltration capacity of the soil	0.25 – 12.7 mm/hr
BASETP	The fraction of potential ET that can be sought from baseflow	10^{-5} – 0.2
INTFW	Interflow inflow parameter	1.0 – 10
IRC	Interflow recession parameter	0.001 - 0.85 d ⁻¹
AGWRC	Groundwater recession parameter	0.001 - 0.999 d ⁻¹

Table 1.1. Estimated HSPF parameters. Parameter bounds are taken from USEPA (2000).

Extreme nonlinearity in the relationship between parameter values and model outputs is avoided by transforming the interflow and groundwater recession parameters listed in Table 1.1. Parameter estimation and uncertainty analysis is actually undertaken with respect to parameters IRCTRANS and AGWRCTRANS rather than IRC and AGWRC. Relationships between the native and transformed HSPF parameters are as follows:-

$$\text{IRCTRANS} = \text{IRC} / (1 - \text{IRC}) \quad (1.1)$$

$$\text{AGWRCTRANS} = \text{AGWRC} / (1 - \text{AGWRC}) \quad (1.2)$$

These transformed parameters approach infinity as the native parameters approach 1. This same transformation was employed by Doherty and Johnston (2003).

Model Files

An executable version of HSPF is supplied; it is named *hspf.exe*. A file named *hspfmmsg.wdm* must reside in the directory from which this is run. This “watershed data management” (i.e. *wdm*) file stores all messages that HSPF may need to write to the screen and to its output files as execution proceeds.

The input file for the current model is named *example.uci*. (*uci* stands for “users control input”.) An inspection of this file reveals the presence of “~” characters on certain lines. These direct *hspf.exe* to read these lines of data from a so-called “supplementary file”, in this case named *example.sup*. Rigid *uci* file protocols prevent the use of maximum precision in

recording numbers on these files. This presents a problem to finite-difference derivatives calculation where parameter preprocessing is undertaken (as is required in the present case to implement the transformations described by equations 1.1 and 1.2), as integrity of finite-difference derivatives calculation demands that full precision be retained when passing numbers between executables comprising the links in a model chain. No such restrictions are placed on numbers recorded on a supplementary input file however.

Run HSPF by typing the command:

```
hspf
```

at the screen prompt. Respond to its prompts in the manner shown below.

```
Enter the name of your input file: example.uci  
Enter name of HSPF supplementary input file: example.sup
```

The model writes nothing to the screen as it runs. However an inspection of file *example.ech* produced by HSPF when run on the basis of the *example.uci* input file (*ech* stands for “echo”) reveals that HSPF runs satisfactory. Model-computed time series are written in binary format to file *output.wdm*. Observed flows are recorded in file *observ.wdm*; we will read these later. HSPF obtains rainfall and other climatic quantities from another *wdm* file named *input.wdm*.

A Groundwater Model

The second example is a groundwater model. In contrast to the previous example, this is a synthetic case. It was specially designed to present a challenge to the process of predictive uncertainty analysis as the prediction whose uncertainty will be analyzed is one of relatively low (but nevertheless finite) likelihood; calculated uncertainty intervals must be wide enough to encapsulate this prediction. This case also differs from the above example in that the inverse problem through which the model is calibrated is an ill-posed one. Furthermore, the prediction whose uncertainty is analyzed is particularly sensitive to parameter components that lie within the calibration null space; hence calibration of the model imposes little constraint on this prediction.

The model is the same as that described by Moore and Doherty (2005; 2006), Tonkin et al. (2007), Tonkin and Doherty (2009) and Moore et al. (2010). Groundwater flow is simulated using the MODFLOW-2000 model (Harbaugh et al, 2000); movement of a particle within the flow field is simulated using the ADV2 process of MODFLOW-2000 (Anderman and Hill, 2001).

Figure 1.2 shows a rectangular model domain with dimension of 500m \times 800m. Superimposed on this is a finite difference grid comprised of 4000 cells, each 10m \times 10m. The hydraulic conductivity field which represents “reality” within the model domain was generated stochastically on the basis of a log exponential variogram with a sill of 0.2 and an effective range (three times the divisor of distance in the exponent of the variogram equation) of 600m. The average log hydraulic conductivity is 0.0, this corresponding to a native hydraulic conductivity value of 1.0m/day.

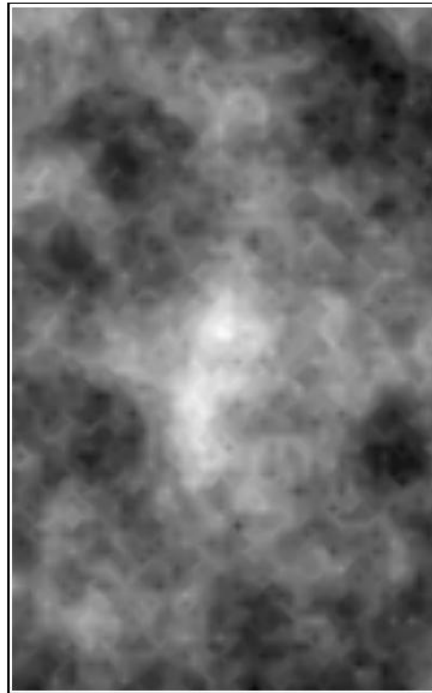


Figure 1.2 Distribution of log hydraulic conductivity within the rectangular model domain. Darker colours represent lower hydraulic conductivity. Hydraulic conductivity ranges between 0.133 and 18.1 m/day.

Water enters the system along the northern boundary of the model domain at a spatially uniform rate of $0.1\text{m}^3/\text{day}$ per metre of boundary. Fixed heads of 0.0 m comprise the southern boundary of the model domain. There is no recharge within the model domain itself. The simulated aquifer is confined and conditions are steady state. A single particle is released near the northern boundary of the model domain. Its trajectory, together with contours of model-calculated piezometric heads, are shown in Figure 1.3. The time taken by the particle to travel from its release point to the bottom fixed-head boundary is 3256 days. This comprises the prediction whose uncertainty is the focus of our attention.

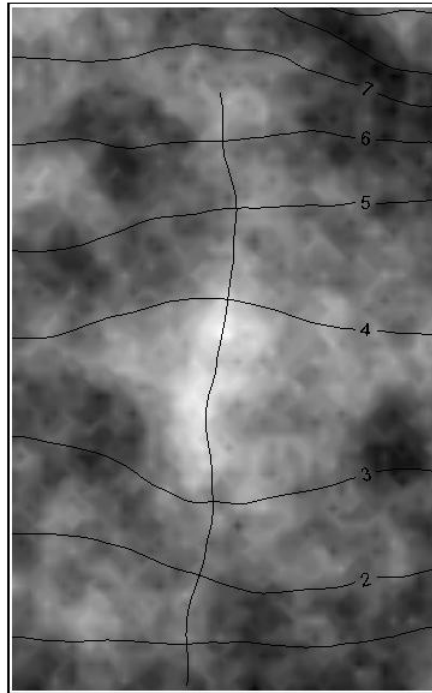


Figure 1.3. Piezometric heads (in metres) and particle trajectory.

Model Parameters

The model is calibrated by matching model-generated heads with “observed” heads at the sites of the 12 wells depicted in Figure 1.4. Independent Gaussian noise with a standard deviation of 0.1m was added to model-calculated heads to produce this calibration dataset. Pilot points are employed as a spatial parameterization device. Pilot point locations (of which there are 104) are also shown in Figure 1.4.

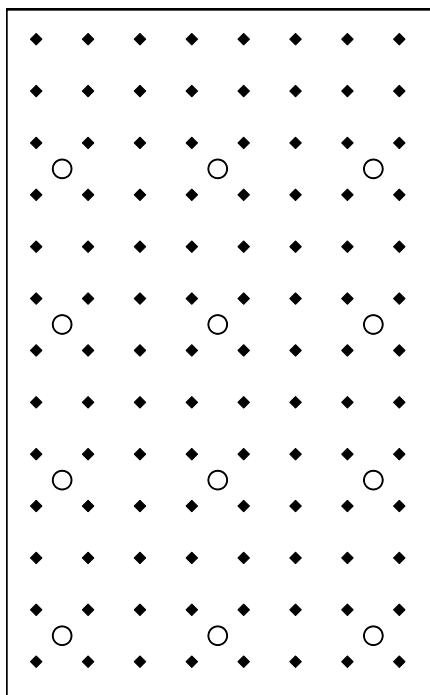


Figure 1.4. Locations of observation wells (circles) and pilot points (diamonds).

Model Files

The “name file” for the MODFLOW model is *rect_model.nam*. The names of other input files required by MODFLOW-2000 appear in this file. Although not used in the calibration process, the MODFLOW observation process must be activated to allow generation of the ADV2 outputs that constitute the prediction of interest.

A salient feature of the MODFLOW input dataset is that arrays holding hydraulic conductivity, porosity and cell activity (the former two are real arrays while the latter is an integer array) reside in external files; these are named *hk.ref*, *por.ref* and *ibound.inf* respectively. MODFLOW is directed to these files through OPEN/CLOSE statements residing at pertinent locations within its main package input files.

The hydraulic conductivity distribution featured in Figure 1.2 is stored in a file named *reality.ref*. Copy this to *hk.ref* using the command:

```
copy reality.ref hk.ref
```

Now run MODFLOW-2000 by typing:

```
mf2k
```

at the screen prompt. Inform it of the name file for the current model (i.e. *rect_model.nam*) when prompted for this. After the model has finished running, heads are recorded in binary format in file *heads.dat*. The particle path is recorded in ASCII format in file *path.dat*. While the time and location of particle emergence can be read from this file, it is more easily read if extracted from this file using a small utility program named FINALTIME whose executable file is named *finaltime.exe*. Run FINALTIME by typing its name at the screen prompt. Then respond to its prompts as follows:

```
Enter name of ADV output file: path.dat
Enter name for this program's output file: time.dat
```

Inspect *time.dat*. The time and location of particle emergence are recorded in this file.

It will be shown in later sections that if this model is calibrated against heads recorded in the 12 wells depicted in Figure 1.4, it over-predicts particle travel time by a factor of two notwithstanding a good fit being achieved with these heads. As stated above for the surface water model, this in itself does not justify condemnation of the model, or of the calibration process. In fact the hydraulic conductivity field of Figure 1.2 was especially selected to create this situation; in moving through the model domain, the particle exploits high conductivity pathways whose presence and disposition are quite compatible with the stochasticity of the hydraulic conductivity field but which are undetectable through the calibration process. This presents a challenge to the process of model predictive uncertainty analysis, for this process must encapsulate the correct value of the prediction even if the model makes this prediction with considerable error when using the hydraulic conductivity field that emerges from the calibration process.

2. What will happen if...?

Introduction

The purpose of this chapter is to set the scene for chapters that follow. It provides a brief description of the decision-making context in which many models operate. It is only after examining this context that optimal usage of models in this context can be pursued.

Before proceeding, however, it must be acknowledged that models are built for many reasons. Sometimes they are built for research purposes where their primary role is to allow a scientist to experience things that may otherwise be invisible, and to thereby gain a greater understanding of the interplay of the many different, and often competing, processes that determine the environmental future. The focus of the present document is not on this type of modelling. Rather the present focus is on models that underpin environmental management, and on which basis important management decisions must be made. This is not to say that complex process models have no place in decision-making; obviously, the better that a practitioner understands environmental processes as they are presumed to operate at his/her study site, the better is his/her ability to discern good management practice from bad management practice at that site. However in this document the focus is on models whose outcomes provide direct and quantitative inputs to a decision-making process. Such models are therefore built as a means for predicting the future behaviour of a specific system under existing, or yet-to-be-tested, management strategies.

Making a Decision

Making a decision would be easy if the repercussions of that decision were perfectly known. This, of course, entails the existence of some means of looking into the future - as it would unfold under existing management strategies and as it would exist under altered management strategies.

Decision-making, as it is implemented in the political and economic world, recognizes that the future cannot be perfectly known. However through careful analysis of all available data, combined with an understanding of the workings of a system, a course of action can often be chosen that maximizes the probability of some good thing happening, or minimizes the probability of some bad thing happen. These are really the same thing when it is considered that failure to maximize a good thing constitutes an opportunity cost that should ideally be minimized.

Unfortunately, environmental decision-making is often seen as a process that has more in common with engineering design than it does with notions of minimization of risk in the context of a system that is poorly understood. Engineering design is often based on the premise that perfect predictions of future system behaviour can in fact be made based on a complete mathematical characterisation of the system. System performance measures can be proposed; design is then targeted at satisfying those measures.

Even where a system is well understood however, good design often requires compromise - for example between cost and performance, or between this aspect of performance and that aspect of performance. Where compromises must be made, optimality must be defined, or at least explored. Of the continuum of designs that may allow a system to work to the satisfaction of all concerned, the one that is eventually chosen is that which achieves the highest level of satisfaction from as many points of view as possible, some of which may be conflicting, and many of which may include subjective considerations.

This illustrates an important point. Even when used in a laboratory setting where a great deal is known about system processes and about the properties of materials on which they operate, models are rarely used on their own in contexts of engineering design. Mostly they are employed in conjunction with other software that optimizes design, taking into account system knowledge as it is encapsulated in the model. It does this by reconciling conflicting design requirements in the most satisfactory way possible. The model thus forms part of a more complex software environment, part of which functions as a kind of “model supervisor” which employs the outcomes of many model runs to formulate designs that attempt to achieve optimality, in whatever way this is defined.

Environmental Management

While environmental management may sometimes be seen as closely related to engineering design, it probably has more in common with civic and economic management (see, for example, Orrell, 2007). This is because it is rarely, if ever, possible to construct a model that can predict the environmental future at a given site, either under management practices which prevail at present, or under those which may be proposed for the future. Reasons for this include the following.

- The complexity of environmental processes is virtually unbounded. Consider for example the plethora of chemical reactions that affect many contaminants, especially those of agricultural origin, as they make their way to and through underground and surface waterways.
- The properties of materials from which environmental systems are built are often poorly known.
- These properties may vary by several orders of magnitude over short distances (rock and soil permeabilities being a case in point).
- Important components of system geometry can often be inferred only vaguely. This applies particularly to the disposition of geological layering, and of fractures and shear zones that may intersect and offset this layering.
- While it may be possible to mathematically characterise environmental processes that are operative at a point or through a column, mathematical characterisation of these processes as they operate over larger areas that are the focus of management decisions is often impossible. Though considerable attention has been given to it in the academic literature, the problem of upscaling is far from solved; many questions, and much debate, still attends the manner in which this should be done. Issues include the following.
 - Should point-scale equations be applied to large areas (as they are in many so-called “physically-based” models)?
 - If so, how should hydraulic properties as they pertain to highly nonlinear processes be averaged over a large area?
 - If not, how should hydraulic properties be represented in modified equations?
 - In either case, what relevance (if any) do point measurements of system properties have to those used by a large-scale model?
- Measurements of system state from which system properties can be inferred through calibration are often scarce. For example wells in which groundwater head measurements are made often tap shallow layers rather than deep layers, and are often

concentrated in some parts of a model domain while being sparse in other parts. Rivers and streams are gauged at only a small number of locations. Water quality measurements in both contexts are often sporadic, and often exhibit a high degree of temporal variability.

- Historical land and water use is often only approximately known. Nevertheless such usage figures are often required by models during their calibration phase.
- Other important system inputs, both historical and present-day, are often only approximately known. This applies in particular to the spatial disposition of rainfall throughout a watershed
- Contaminant source strengths and locations, both industrial and agricultural, are extremely uncertain.
- Numerical problems often attend the simulation of complex environmental processes. In general, the greater the degree to which a numerical model attempts to be “physically based” - the greater will its run time be, and the greater will be the degree to which it may fall prey to numerical instability. Both of these reduce a model’s ability to be used in conjunction with software that facilitates calibration and uncertainty analysis on the one hand, and system design optimisation on the other hand.

The last of the above points deserves further consideration. In a typical engineering design scenario system properties are well known. In the environmental modelling context the areal distribution of system properties must be inferred from scarce point measurements of some of them, and/or back-calculated from historical measurements of system state. Thus solution of an ill-posed inverse problem is a fundamental aspect of environmental model usage. The extreme heterogeneity of most earth systems makes it very unlikely that exact values of system properties can be inferred throughout the system. Furthermore as the properties and processes represented within a model are often “lumped” or averaged analogues of their real-world counterparts, the salience of point measurements of system properties to parameters that represent these properties in a model is often questionable. Repercussions of this include the following.

- An environmental model must serve more than one purpose. While being used to make predictions on which environmental management must be based, it must also be used to extract as much information as possible from historical site data. If this is to be done with maximum efficacy, it requires software support. Hence, as stated above, a model must be capable of being deployed in partnership with other software that runs it repeatedly as part of a data extraction process.
- The complexity of real-world systems is such that even large calibration datasets cannot hope to provide unique estimates of all system properties at all places within a model domain. In general, estimates of averaged properties, or combinations of properties, will be available through solution of what may be a complex and ill-posed inverse problem.
- Even estimates of broad scale system properties will be degraded through:
 - lack of complete knowledge of current and historical system inputs;
 - inadequacies in a model’s ability to provide precise simulation of environmental processes; and

- the fact that measurements of system state on which basis system properties are inferred, are accompanied by measurement error.

From all of this it is obvious that it is not possible to construct a model that will provide accurate predictions of future system behaviour. Furthermore, the greater the extent to which a prediction is sensitive to non-inferable system properties, the lower will be the reliability of that prediction. In general, non-inferable properties will pertain to system detail, and/or to aspects of system behaviour that are rarely observed - or perhaps may never have been observed if contemplated changes to a system are likely to take it to places where it has never been before.

Unfortunately, these are the very types of prediction that are often of most interest from a management point of view. For example water quality, including the nature and disposition of surface or subsurface contaminants, is often dependent on process and property details that can only be vaguely measured or inferred. The response of a system to extreme events (for example the height of a flood peak) is, by definition, sensitive to aspects of the system that have rarely been observed in the past and whose measurement is subject to a high degree of uncertainty. The response of the system to sets of inputs that arise out of development, or to management plans that are put in place to ameliorate the deleterious effects of development, is dependent on aspects of system behaviour that may never have been experienced before.

It is obvious, therefore, that an environmental model cannot be employed as an engineering design tool through which the means to achieve some agreed-upon management outcome is optimised, with achievement of the outcome never in doubt and only the efficiency of its achievement being the chief design consideration. If an environmental model tells us anything, it tells us that the future cannot be exactly known. Use of a model in the decision-making context, and the very design of the decision-making context itself, must be based on recognition of this fact.

Risk

An environmental model cannot predict the future. However this does not render it useless. While it cannot be used to predict what *will* happen because of data and model inadequacies, it can often be used to discriminate between what *can* happen and what *cannot* happen. As the line between the two is most unlikely to be sharp, it may also be able to provide some indication of diminishing likelihood as the value of a prediction changes.

Many, if not most, environmental decisions are made in order to avoid an unwanted occurrence. Examples of unwanted occurrences include the following.

- Where groundwater cleanup is undertaken, a remediation strategy must be such as to ensure that the concentration of a contaminant at management boundaries will not exceed a certain threshold.
- Assurances must be provided to the public that the height of an imminent flood will be no greater than a level that is compatible with current evacuation boundaries.
- Water allocated to irrigators at the start of a water year must be such as to guarantee that no water deficit occurs during the water year.
- Alterations to land management practices must be such as to ensure that the concentrations of agricultural contaminants during periods of high or low flow (depending on the contaminant) do not rise above regulatory thresholds.

Obviously, the likelihood of an unwanted occurrence can be minimized by taking extreme measures to avoid it. However such measures may not be acceptable because the associated

costs may be too great. These must be balanced against the cost of an unwanted occurrence, together with the risk of its occurrence. Thus if the cost of the occurrence is high, only a low occurrence likelihood is tolerable. Conversely, if its cost is not too great a higher risk of its occurrence can be tolerated.

In a series of landmark papers (Freeze et al, 1990; Massmann et al, 1991; Sperling et al, 1992; Freeze et al, 1992), Freeze and his co-workers presented theory and examples in support of a suggested methodology for model-based decision analysis. At the heart of their methodology is an objective function defined as:

$$\Phi_j = \sum_{t=0}^T \frac{1}{(1+i)^t} [B_j(t) - C_j(t) - R_j(t)] \quad (2.1)$$

where:

- Φ_j = the objective function associated with alternative j in dollars;
- $B_j(t)$ = benefits of alternative j in year t in dollars;
- $C_j(t)$ = costs of alternative j in year t in dollars;
- $R_j(t)$ = risk of alternative j in year t in dollars;
- T = time horizon in years;
- i = discount rate as a decimal fraction.

Risk is defined through the equation:

$$R(t) = P_f(t) C_f(t) \gamma(C_f) \quad (2.2)$$

where:

- $P_f(t)$ = probability of failure in year t (decimal fraction);
- $C_f(t)$ = costs associated with failure in year t (dollars);
- $\gamma(C_f)$ = normalized utility function related to risk aversion (decimal fraction ≥ 1).

The optimal decision is that corresponding to the maximum of the above objective function. Other decision strategies are also discussed in these papers. For example the “maximin” decision criterion seeks to identify the least desirable consequence for each alternative under consideration and then selects the alternative that leads to the best of these least desirable consequences. Alternatively, the “minimax” regret strategy selects that alternative whose maximum regret is smallest, where “regret” is defined as the price that must be paid for selecting the non-optimal alternative given perfect knowledge of system properties.

The above authors point out that a model’s contribution to the objective function of equation (2.1) is expressed only through the risk term. This then defines the role of the model in the decision-making process.

Whether decision-making is based on formal analysis such as that presented in the above equation, or whether it is based on more subjective considerations, the role of an environmental model as a means of assessing the probability of an unwanted occurrence is the same. It is not reasonable to ask a model to predict the future, for it cannot do this. The fact that all predictions of future environmental behaviour can only be made with uncertainty must be embraced. If it is to provide a scientific foundation to the decision-making process, all that can be asked of a model is that it be used to assess probability and hence risk. Or to put it another way, a model must be used to explore whether, if a certain management action

is taken, an unwanted occurrence can be avoided, or that its occurrence is associated with a probability that is suitably low.

In some circumstances it may be possible to extend the role of models in the decision-making process a little beyond this. If a model exposes the possibility of a bad thing happening, then it may also be able to show *how* that bad thing can happen. This may promulgate the design of a suitable monitoring strategy for early detection of the untoward event. Implementation of that strategy may add to the cost of a certain management plan, while reducing the risk of that plan going awry. These terms can be incorporated formally or informally into the above decision equation.

Hypothesis-Testing

Two important roles played by environmental modelling when used in a decision-making framework emerge from the above discussion. The first is its role in assessing the probability of a future untoward occurrence, this being a probabilistic description of the confidence with which it can be said that a certain event will or will not happen. The second is its role in extracting information from existing site data, thereby reducing the range of possible predictive occurrences below that which would prevail in the absence of this data.

A model is able to provide these services because, ideally, it carries within it the entirety of our knowledge of a study site. This knowledge is encapsulated in the processes that it simulates, the boundary conditions that it implements, various aspects of system geometry that it features, and the range of parameter values that we allow it possess.

These aspects of model construction and usage can be related to the terms that appear in Bayes equation. Bayes equation can be written as:

$$P(\mathbf{k} | \mathbf{h}) \propto P(\mathbf{h} | \mathbf{k})P(\mathbf{k}) \quad (2.3)$$

where $P(\mathbf{k})$ describes the prior probability of model parameters, $P(\mathbf{h}|\mathbf{k})$ is a likelihood function calculated from the fit that a model provides to measurements of system state, and $P(\mathbf{k}|\mathbf{h})$ describes the posterior probability of model parameters. As the container for all of our expert knowledge, the model can be represented as the $P(\mathbf{k})$ term of Bayes equation. The model's role in extracting information from site data is represented by the $P(\mathbf{h}|\mathbf{k})$ likelihood term. Its role in providing risk analysis in support of the decision-making process is represented by the posterior probability term $P(\mathbf{k}|\mathbf{h})$. This represents the probability distribution of parameters after all information has been taken into account. The probability distribution of any prediction, and consequentially the risk associated with any untoward occurrence, is calculated from this term. This matter will be examined in greater detail in later chapters.

For the moment, however, we will look at the role that a model plays in the decision-making process in a slightly more qualitative way. As we shall see, however, the more qualitative nature of this description does not degrade its validity, as in most cases of model deployment, qualitative assessment of the likelihood or otherwise of a future event is all that is available to us anyway.

An environmental model can be considered to be a type of scientific instrument. As such, it can be used to test the hypothesis that the occurrence of a certain future event is consistent with all that is known about a system. "All that is known about a system" is encapsulated in the design of a model of that system, in the range of parameters we allow that model to possess, and in the constraints on those ranges that emerge from the necessity that the model reproduce historical system behaviour, all of this being in accordance with Bayes equation.

The hypothesis that a future event will occur given a certain management practice can be rejected if, at a certain confidence level, the occurrence of that event is inconsistent with some or all of this knowledge. Inconsistency is exhibited by an inability on the model's part to make this prediction unless it employs parameters that, in our view, are unrealistic (at a certain level of confidence), or that incur an unlikely (at a certain level of confidence) amount of misfit with historical observations of system state when the model is driven by historical stresses.

Using a model to test hypotheses is a conceptually straightforward undertaking. It is also an undertaking that is pivotal to its role in decision support. In practice however, it is an undertaking that is more easily said than done. Reasons for this include the following.

- The capacity of a model to encapsulate our knowledge often erodes its capacity to reduce posterior uncertainty through extracting information from historical datasets. Complex physically-based models are often employed because they provide appropriate receptacles for expert knowledge of processes and parameters, for they try to simulate, as well as possible, reality as we know it based on system properties that we can measure. However such complex models often have (extremely) long run times, and a high penchant for numerical instability. Both of these make their use with high-end parameter estimation software difficult or impossible.
- While complex models may allow us to represent the prior probability distribution of certain system properties with some degree of integrity, they make it almost impossible for us to represent other important aspects of prior uncertainty. For example, numerical grids are often designed specifically to accommodate the nuances of what we imagine to be the complex geology of three-dimensional systems, including the disposition of geological layers and faults, and the offsetting of layering and faults by other faults. While the locations of geological features that may be critical to the movement of underground water and contaminants may be only poorly known, making alterations to the disposition of these features to reflect their unknown status is often impossible without expensive and time-consuming redesign of the model grid or mesh. Practical difficulties can also be encountered in varying the geometric and hydraulic descriptors of certain boundary conditions (for example those related to rivers, streams and canals), and in introducing variability to factors affecting recharge processes (for example the location and timing of local ponding, temporal and spatial variations in diffuse vs. macropore recharge, etc).
- Parameters employed by a model cannot be represented with the same level of heterogeneity as that of hydraulic property variability in the real world. A lower limit is placed on model parameter variability by the model grid or element size in the case of discrete-element models. The algorithmic lumping of processes in many land use and surface water models places a lower limit on the spatial and temporal hydraulic property variability that can be represented in those kinds of models; at the same time it makes encapsulation of prior knowledge difficult because of the abstract nature of parameters employed by these kinds of models when compared with the measureable hydraulic properties that they purport to represent. Even where a physically-based model allows representation of heterogeneity at a cell-by-cell or element-by-element level, there are practical difficulties associated with this endeavour, especially where these parameters are subjected to adjustment through the model calibration process. Complex process models are often endowed with simplistic parameter fields to facilitate their calibration. However the use of a few parameters that represent average properties over many cells, or even a moderate number of parameters based on

devices such as pilot points, erodes the capacity of the model to represent fine-scale heterogeneity. Model outputs under both historical and predictive conditions may be compromised because of this. Avenues for representation of prior system knowledge acquired through point measurements of system properties are also partially blocked because of this.

- No model is perfect. All outputs of all models bear the imprints of model imperfections. When assessing parameter likelihood through history-matching, this must be taken into account as we will almost certainly need to tolerate a greater level of model-to-measurement misfit than that which would be suggested on the basis of measurement noise alone. But how much greater? This is something that can only be assessed subjectively. Alternatively, it may be possible to achieve a very good fit between model outputs and historical measurements of system state. But in achieving this fit, some parameters may need to assume unlikely values to compensate for model inadequacies. But if a model parameter has an abstract side to its nature for reasons discussed above, how is “unlikely” defined for such a parameter? Furthermore, is parameter compensation for model inadequacies always a bad thing? If compensatory behaviour on the part of one parameter enhances a model’s ability to replicate the past, may it not also enhance its ability to predict the future? By keeping parameters strictly “realistic” could we be shutting the door on information that would otherwise flow to a model and that would therefore make it a better predictor of (at least some aspects of) future system behaviour? Doherty and Welter (2010) show that the frustrating answer to this question is “sometimes yes and sometimes no, and it will often not be possible to discriminate between the two”.
- Even if every parameter employed by a model could be magically provided with its “correct” value (notwithstanding the abstract nature of many of them), its predictions would still be flawed because of the imperfect nature of a model’s capacity to simulate every nuance of future system behaviour. The extent to which a prediction of decision-making interest is flawed will depend on many things. Generally its proclivity for error will rise with the extent to which it depends on small-scale or extreme features of system behaviour, as these are the aspects of system behaviour that model imperfections are most likely to affect.

These considerations leave us in a position that is not altogether satisfying. Having dispensed with the illusion that a model can be used to predict the future, and replaced it with the assertion that a model can be used to assign confidence levels to various occurrences that are salient to management decisions, we must now conclude that the confidence with which we can assign a confidence level is frustratingly small. Or, to put it another way, the confidence with which we can assert that a future untoward event will not occur is lower than that which exists “in theory”, because we simply cannot process all of the data that are available to us using the cumbersome scientific instrument that is an environmental model. The widths of predictive confidence intervals on which decision-making is based must therefore be augmented to take into account the bluntness of our scientific instrument. However the magnitudes of these augmentations are almost impossible to calculate, so that their assessment will necessarily be somewhat subjective. Obviously they must be reduced as much as possible.

Reducing Model Augmentations to Uncertainty

Strictly speaking the term “model-added uncertainty” has no meaning. A better term may be “penchant for error”, or some similar phrase that would almost certainly include the word

“error”. This would reserve the word “uncertainty” for unambiguous use in describing the “purer” concepts that appear in Bayes equation that are characteristics of the system itself (as described by our expert knowledge), and of the information contained in measurements of that system. The fact that our prior knowledge must be housed in a flawed receptacle, and that information must be extracted from historical measurements of system state through a flawed vehicle that compromises this information, could then be described using other terminology. To some extent we will keep the terms “error” and “uncertainty” separate throughout this document, and will calculate them in different ways. However it must not be forgotten that the two become inseparably mixed in practice when we use a numerical model to assess whether, and at what confidence level, the occurrence of an untoward event can be rejected.

Meanwhile it must be remembered that, however it is characterized, the penchant for a model prediction to be wrong, even after steps have been taken to minimize its potential for predictive wrongness, is an outcome of two factors. The first is the inherent uncertainty of any prediction that it is required to make as it is described by Bayes equation. The second is the fact that information that is pertinent to the making of that prediction must be extracted from historical data using a flawed instrument, and that predictions of future system behaviour are made with the same flawed instrument.

The situation is made even more complex however, when it is recognized that a model’s flawed status may compromise some predictions but not others. Furthermore, as Doherty and Welter (2010) show, some model predictions may benefit from having some of a model’s flaws “calibrated out”, while the making of other predictions may suffer from such an exercise. Furthermore, the “prediction-tuning” process may be different for different predictions. It follows that a model should not be built and calibrated in isolation from the predictions that are required of it, and that environmental processes at the same site may need to be simulated in different ways to provide optimal bases for different management decisions.

Model-building can, in some ways, be seen as an optimisation process. At one end of a continuum of approaches to environmental modelling are complex, physically-based models that attempt to provide “realistic” simulation of all aspects of a system’s behaviour. This approach is seen by many as that which entails the highest level of scientific integrity because it purports to provide suitable receptacles for all aspects of expert knowledge. This line of argument leads to the conclusion that a single complex model of a study area can be employed to make a plethora of predictions of many different types with as high a level of integrity as current expert knowledge and hard data allows. Hence it can provide a universal basis for decision-making at a particular study site. In fact, as has been discussed, such a model may provide a worthy repository for expert knowledge (including the uncertainties associated therewith), and thus encapsulate the prior knowledge term of Bayes equation. As such it can provide a basis for analysis of the uncertainty of any system prediction based solely on expert knowledge (and lack thereof), while ignoring the likelihood term of Bayes equation. Presumably, such quantification of pre-calibration uncertainty would entail running the model countless times in order to undertake Monte-Carlo analysis based on variation of all of the myriad of parameters that such a complex model can employ. As such a model would probably have a high run time, parallelisation of this process would be essential.

Unfortunately, a complex model may be a far-from-optimal device for extracting information from an historical dataset because its inevitably high run time and penchant for numerical instability may render its use in conjunction with high-end inversion software impossible. Hence the second term of Bayes equation may be higher than it needs to be. Thus the

capacity for prior predictive uncertainty intervals to be reduced through history-matching may be reduced to almost zero, in spite of the fact that a wealth of information may reside in historical datasets. Predictive confidence intervals will therefore be wider than they need to be. The risk associated with untoward events may therefore be assessed as unnecessarily high, this leading to unnecessarily conservative management.

An alternative to use of a very complex model is use of a model that attempts to retain as much complexity as possible that is salient to the making of a prediction of interest, at the same time as it attempts to abandon non-salient complexity in the hope of decreasing model run time and increasing numerical stability. Unfortunately, such a model may not provide an optimal receptacle for expert knowledge. If prior uncertainty of model parameters is appropriately increased to accommodate their limited capacity to be informed by expert knowledge, this may then inflate predictive uncertainty. However if the historical dataset is rich in information - information that can now be extracted from it - reduction of the likelihood term of Bayes equation may more than compensate for increase of the prior information term. Caution must be exercised, of course, as some of the information that is extracted from the historical dataset may be misdirected to parameters that play compensatory roles for model defects. If a critical model prediction is sensitive to parameters that are misinformed in this manner, it may inherit this error. On the other hand if, as Doherty and Welter (2010) show, the prediction depends on the same parameter combinations as do model outputs used in the history-matching process, information that is resident in historical system measurements is probably directed to prediction-informative repositories; this will occur even though the parameters that constitute these repositories may not have as direct a relationship with the system properties after which they are named as a modeller would like, and hence cannot be as well informed by a modeller's expert knowledge as he/she would wish. However if information that is resident in expert knowledge is small compared with that residing in the historical dataset, then this may not matter because more will have been gained in terms of reduction of predictive uncertainty through reduction of the likelihood term than has been lost through raising of the prior probability term.

Of course if a model is too simple then fits between model outputs and field measurements will be poor and the likelihood term will be high. In addition to this, the likelihood term will be difficult or impossible to calculate because of the unknown stochastic characteristics of model-imperfection-induced misfit (often described by the term "structural noise"). Meanwhile the prior probability term will also be high because the abstract parameters employed by a simple model provide poor receptacles for expert knowledge.

It is apparent from the above considerations that design of a model to underpin environmental decision-making requires conceptual solution of an optimisation problem that has at its heart the interplay between the two terms constituting the right side of Bayes equation. The solution to this problem can only be context-specific. It must entail extraction of maximum prediction-specific information from both user expertise and site data. Furthermore, it is not a foregone conclusion that a strategy that minimizes uncertainty for one prediction constitutes an optimal strategy for minimizing the uncertainty of another prediction. In particular, where a prediction is similar in nature and location to historical measurements of system state, a modelling approach that maximizes transfer of information from the measurement dataset to model parameters (even if some of these parameters must assume surrogate roles in the information-extraction process) will probably be optimal. On the other hand, where a prediction is of a distinctly different kind from those comprising the historical site dataset, it may be sensitive to aspects of the system that are ill-informed by that dataset. In this case the uncertainty of that prediction may have a direct dependence on the prior probability term of Bayes equation. A model will therefore be maximally effective in reducing the uncertainty of

that prediction when it provides optimal receptacles for expert knowledge, and has a strong physical basis. It therefore follows that *the attempted use of a single model to make all predictions of all kinds within a study area will probably result in a failure to reduce the uncertainty of any one prediction to anything like its theoretical lower limit*. It may also make the analysis of posterior predictive uncertainty very difficult, this possibly resulting in underestimation of that uncertainty and therefore a failure to properly assess the probability of occurrence of unwanted events.

The Scientific Method

Ultimately, those who use models to provide as sound a basis for scientific environmental management as possible should aspire to implement the scientific method. After all, to what other goal should they aspire? As will be discussed in later chapters, at its most basic level implementation of the scientific method comprises the proposal of hypotheses together with subsequent attempts to reject them based on all information at hand. In the environmental management context a hypothesis comprises the conjecture that something bad will happen, this being a management outcome that it is desirable to avoid. The information on which it may, or may not, be possible to reject this hypothesis is composed of both expert knowledge and of information that resides in historical measurements of the state of the system that are available for the study area.

The history of scientific achievement is replete with stories of scientists who devised brilliant experiments to test their hypotheses and thereby provide hitherto unavailable insights into the natural world. Most of these experiments were targeted at the testing of individual hypotheses - not at the testing of *all* hypotheses. Their laboratory instruments were often “blunt” and cumbersome compared to the phenomenon being explored, particularly where these pertained to the nature of matter and of the atom. However through tuning these instruments to the problem at hand, and through focussing data acquisition and data processing so that it was maximally effective in falsifying, or failing to falsify, the particular hypothesis being tested, great advances in human knowledge were achieved.

In many respects, scientific inquiry into the nature of an environmental system is little different. Our best tools are often models. Despite the fact that their numerical inadequacies can make them blunt instruments, and despite the fact that data availability at a particular site may be scarce, it is nevertheless often possible to undertake incisive, prediction-specific inquiries of that data using a model - inquiries that may yield conclusions on which sound management can be based. This has a far greater chance of happening, however, if the design of the modelling instrument, and the manner in which it is used to extract information from available data, is optimized in relation to a specific management problem, and is thereby deployed with as much skill as the scientist can bring to bear.

Summary

The discussion of this chapter can be summarized as follows.

- Environmental decision-making, as does all decision-making, rests on an assessment of the risk associated with the happening of unwanted and costly events. It is the task of modelling to assess this risk.
- Informed model usage constitutes an implementation of Bayes equation. The end product of informed model usage can only be the definition of a posterior confidence interval associated with a prediction of interest. For maximum relevance to the decision-making process, this must be expressed as a level of confidence that an unwanted event will not occur.

-
- Complex models provide optimal receptacles for expert knowledge. However as scientific instruments they may be far from optimal as they may provide impediments to the flow of information from environmental datasets. Because of this, post-calibration predictive uncertainty may at best be higher than it needs to be, and at worst unquantifiable.
 - Simple models provide poor receptacles for expert knowledge. Furthermore history-matching based on these models may require that some of their parameters assume compensatory roles for model inadequacies. If a prediction resembles observations used in the history-matching process, reductions in predictive uncertainty may nevertheless be achieved through embracing these compensatory roles. If it does not, the uncertainty of a prediction may be increased through use of a simple model, at the same time as it is rendered virtually unquantifiable.
 - Model construction and deployment may therefore need to be prediction-specific. The idea that a single system simulator can provide the basis for all management decisions at a particular site or study area has no foundation in either common sense or theory.

3. Models, Simulation and Uncertainty

Expert Knowledge

The role of models in the decision-making process was discussed in the previous chapter. There it was pointed out that physically-based models that attempt to simulate as many details as possible of the behaviour of natural systems have advantages and disadvantages. Their main advantage is that they comprise a suitable repository for expert knowledge. However their main disadvantage is that detailed simulation of natural processes, and representation within a model of the heterogeneous nature of system properties on which natural processes depend, is a computationally demanding exercise. Use of such a model therefore makes it very difficult for a modeller to extract information from historical datasets, thereby devaluing the cost of such data. Even when used outside of the calibration context, design rigidities may be such as to make it difficult for a modeller to encapsulate his/her expert knowledge in a complex model to the extent that he/she would like.

At this stage it is worth pausing for a moment to ponder the question of what expert knowledge actually is. Expert knowledge is in fact a probabilistic form of knowledge. A hydrogeologist cannot say what the hydraulic properties of the subsurface are at every point within a (necessarily three-dimensional) groundwater flow domain. Nor can he/she know the disposition of rock boundaries throughout that domain, nor the variation in weathering depths, nor the changes in lithology along the strike of any sedimentary or structural feature, nor the paths of ancient meandering streams, nor the variations in fracture density throughout a model domain. Similarly, a surface water hydrologist cannot know infiltration properties pertaining to all soil types at all locations under all land use conditions throughout a study area at all times of the year. Nor can he/she know with certainty how land uses have changed over what may be a lengthy calibration period.

Conceptually, expert knowledge must be expressed stochastically (i.e. probabilistically), as it is rarely definitive. As has already been stated, probability distributions that arise from expert knowledge are in fact the prior probabilities that feature in Bayes equation. In theory, the use of a complex model allows us to expose this knowledge for what it is - a range of possibilities that hopefully encompass the true state of the system, whatever that may be.

What a Model can Provide

Given that expert knowledge is probabilistic in nature, it immediately follows that so too are predictions that are made by a model whose parameterization is based on expert knowledge alone. These predictions must therefore be expressed as probability distributions. The better is expert knowledge, the narrower will these probability distributions be.

Bayes equation shows that predictions made by models that have been “calibrated” must also be probabilistic in nature. Actually, the term “calibration” has no place in Bayes equation. To the extent that it has any meaning in the environmental modelling context at all, that meaning will be examined in the next chapter. Bayes equation shows that what history-matching can achieve is a narrowing of the uncertainty associated with some model parameters, for their propensity to vary is no longer limited by expert knowledge alone. Their propensity for variability is now also constrained by the necessity for the model to reproduce historical system behaviour as measured at certain points in space and time. Obviously, the less data that is available, and the greater the measurement noise that is associated with that data, the fewer will be the parameters that are constrained by this data, and the looser will be the

constraints on them that this data exerts. Greater amounts of data can lead to tighter constraints on some parameters - but not necessarily on all parameters - and not necessarily on the parameters to which a prediction of interest is most sensitive. Hence (as the synthetic groundwater model that forms one of the practical exercises associated with this document demonstrates), the process of model “calibration” may, or may not, lead to an enhanced ability on the part of the model to make predictions which are of most interest to us. This will depend entirely on the nature of the predictions we seek, and on the information content of existing site data.

These considerations lead us to the point where we can define what a modelling exercise can aspire to achieve.

- For a prediction of interest, a range of possible values which the prediction may take, all of which are compatible with all that is known of a system; collectively these define the uncertainty range of the prediction.
- A guarantee that the correct prediction lies within the uncertainty limits so defined (at a specified level of confidence).
- A modelling strategy which ensures that the range of predictive uncertainty calculated by a model is no wider than it needs to be, given the prevailing level of expert knowledge and the information content of site data. Thus the probability of occurrence of an untoward event will not be over-estimated.
- An uncertainty assessment strategy which ensures that calculated predictive uncertainty margins are no narrower than they should be through failure to account for all contributors to possible model predictive error, some of these arising from data inadequacy and some of them arising from model structural defects. Thus the risk associated with the occurrence of an untoward event will not be under-estimated.

These then define the aspirations of any modelling exercise. Unfortunately, for reasons already discussed, the meeting of these goals may not be straightforward. Furthermore compromise will always be required as attempts to reduce one aspect of uncertainty may lead to inflation of another aspect of uncertainty by an unknown amount.

What an Uncalibrated Model can Provide

Here, and in later sections of this document, the term “uncalibrated model” will refer to a model whose parameters are not constrained by history-matching. Hence it is a model that can represent only the prior probability term of Bayes equation. As such it is a model that is capable of defining the range of predictive probabilities that exist where no historical measurements of system state are available to constrain the uncertainty that arises from limitations in expert knowledge.

Probabilistic analysis is easy for an uncalibrated model. Conceptually it is most readily implemented using a Monte Carlo methodology. Using this methodology random parameter sets are generated on the basis of a prior parameter probability distribution which expresses expert knowledge (at the same time as it expresses expert ignorance due to its probabilistic nature). The model is run on the basis of each of these parameter sets in order to calculate the value of a prediction of interest. By collecting prediction values computed on the basis of all such parameter sets, an empirical probability density function can be built for the prediction. This probability density function can then be used to define the risk associated with the occurrence of prediction values that are considered to be unwelcome.

In practice, even the uncalibrated model can only approximately attain the goals to which modelling must aspire (assuming that justification for its uncalibrated status lies in the absence of any data to calibrate it against). Reasons for this include the following.

- Even the most complex model can only be an abstract reflection of reality, for not only are the parameters associated with environmental processes uncertain, but the equations that describe these processes are also often uncertain.
- Many environmental processes, and the system properties on which they rely, are difficult to characterise where they must be averaged over a model element or cell.
- Where a model domain is large (for example the domain of a regional ground or surface water model), the level of parameterization complexity that is required to characterize the degree of system property complexity that exists in the real world is far too large to handle in either probabilistic or deterministic analysis.
- Many important aspects of environmental uncertainty simply cannot be accommodated in uncertainty analysis based on numerical models. As has already been stated, this includes uncertainties in the disposition of geological layering, bedding and faulting.
- Even where it is numerically feasible to run a model many times based on different realisations of system properties, a suitable stochastic descriptor for variability of those properties may not be available. Instead simplistic (and often over-constraining) assumptions such as that of multi-normality and stationarity are used as a basis for random parameter set generation.
- It is simply not possible for numerical complexity to mimic the complexity of the real world. The numerical grids employed by groundwater models must be finite if these models are to have finite run times. Regional rainfall-runoff models must represent complex processes in lumped form despite the intricacies of surface water movement over the hundreds of different land use and soil types that prevail in large watersheds.

It is thus apparent that even the most “physically-based” uncalibrated model is compromised, as its construction and deployment requires medium to high levels of abstraction. Abstraction obviously comes at a cost. But against what must this cost be debited? Obviously it must be debited against what a model can promise. The latter is discussed in the previous subsection of this document. The cost is therefore paid through a decrease in quality of the predictive probability distributions that are the only scientifically based outcomes of environmental modelling.

How can model imperfections detract from model-calculated predictive probability distributions? There is no answer to this question that is universally applicable. However the following considerations are salient.

- In some cases they will create bias, thereby shifting a predictive probability distribution to one side. Certain unwanted environmental occurrences that, using a perfect model, may be assessed as having low but finite predictive likelihood, may be considered as impossible when their likelihood is assessed using a flawed model. On the other hand, at the other end of the shifted predictive probability distribution, events whose likelihood is in fact very low, may be considered to possess moderate to high likelihood of occurrence.
- Simplification and abstraction involves removal of detail. For predictions that are sensitive to detail, certain mechanisms that lead to low, but nevertheless finite,

predictive possibilities will become unavailable. Hence there is a risk that, for predictions of these types, model-calculated predictive probability distributions will be narrower than those which in fact prevail.

As both of the above phenomena may lead to under-estimation of risk certain steps can be taken to ameliorate a model's performance as a risk assessment tool. These steps are based on the assumption that it is better to over-estimate risk than to under-estimate it. They also presume that concluding that an unwanted event *cannot* occur when in fact it is entirely possible that it *can* occur must be avoided at all costs. These steps include the following.

- To the extent that predictive bias is introduced through the model construction process, the details of model construction should be such as to bias predictions toward pessimism rather than optimism.
- Some model parameters may need to be endowed with wider prior probability distributions than they would possess based on expert knowledge alone. This applies particularly to parameters whose more extreme values may provide surrogates for missing or defective simulated processes.
- A random “predictive noise” term may be added to model outcomes of interest in order to endow these noise-enhanced predictions with a wider range of variability than they would otherwise possess.
- Predictive probability distributions computed on the basis of model outcomes may be “stretched” in order to provide a suitable “engineering safety margin”.

All of the above strategies are necessarily heuristic and subjective. However subjectivity must be placed in its proper context. A great deal of expert knowledge is subjective, including definition of prior parameter probability distributions. The “art” of prediction-specific model abstraction and/or of definition of an appropriate engineering safety margin to employ when basing important decisions on model-computed predictive probability distributions is no less a form of expert knowledge, and no greater a form of expert judgement, than that which is required in all other phases of the model construction process.

Linear Analysis

In this subsection some concepts underpinning linear analysis, as it applies to an uncalibrated model, are introduced. In later sections this type of analysis will be expanded to accommodate the imposition of calibration constraints. As discussed in Chapter 1, mathematical presentations provided herein are brief. It is not necessary that the equations presented below be understood - only that the concepts behind them be understood.

Let the vector \mathbf{k} represent all parameters used by a model. Notionally, elements of \mathbf{k} can include any imperfectly known value that is used by a model, irrespective of whether the quantity to which this value is assigned is a system property, a boundary condition, or an aspect of the model's geometry.

Let the covariance matrix of \mathbf{k} be denoted as $C(\mathbf{k})$. If the vector \mathbf{k} has m elements, then $C(\mathbf{k})$ is an $m \times m$ matrix. As a covariance matrix, $C(\mathbf{k})$ provides a summary of the stochasticity of \mathbf{k} . Hence it expresses both the pre-calibration knowledge, and the pre-calibration ignorance of the modeller. The diagonal terms of $C(\mathbf{k})$ are perhaps the best expression of the modeller's ignorance. The i 'th diagonal term expresses the variance of the i 'th element of \mathbf{k} . Variance is the square of standard deviation. Hence the diagonal terms of $C(\mathbf{k})$ denote a modeller's inability to say exactly what the value of a particular system property is at a certain point within the model domain. However the fact that these diagonal elements are of finite

magnitude portrays a certain state of knowledge (or conversely, they portray boundaries to the modeller's ignorance).

Off-diagonal terms of $C(\mathbf{k})$ denote statistical interrelatedness between parameters of the same type, or even of different types. Zero-valued off-diagonal terms portray no statistical relationship at all. Thus if the element at row i and column j of $C(\mathbf{k})$ is zero, then k_i and k_j (these being the i 'th and j 'th elements of \mathbf{k}) are statistically independent. However if this term is non-zero, it signifies that if one of these parameters is higher than average, the other will tend to be either higher than average (if $C_{ij}(\mathbf{k})$ is positive) or lower than average (if $C_{ij}(\mathbf{k})$ is negative). This implies expert knowledge. Examples include the following.

- Subsurface hydraulic properties do not show completely random spatial variability. In most cases there is a tendency for some degree of spatial continuity in hydraulic properties to exist. Furthermore the length over which such statistical interrelatedness prevails may be longer in one direction than in others, this implying anisotropy of hydraulic properties.
- A soil with a high sand content is likely to allow greater infiltration of water than a soil with a high clay content. Its capacity to store water may also be greater, at the same time as its propensity to lose water through drainage may also be enhanced. All of these hydraulic characteristics of a particular soil may be represented by different model parameters; these parameters obviously possess a high degree of statistical correlation.

Let s (a scalar) denote a prediction. Let the $n \times 1$ vector \mathbf{y} denote the sensitivity of this prediction to all of the elements of \mathbf{k} . In a linear system, the following relationship then applies.

$$s - s_0 = \mathbf{y}^t(\mathbf{k} - \mathbf{k}_0) \quad (3.1)$$

where the superscript "t" designates the matrix transpose operation, and s_0 and \mathbf{k}_0 are reference values. For simplicity (and without loss of generality) these reference values will be omitted from future equations (implying that s and \mathbf{k} are defined as perturbations from these reference values), so that (3.1) becomes:

$$s = \mathbf{y}^t \mathbf{k} \quad (3.2)$$

We now introduce a basic matrix identity. Suppose that:

$$\mathbf{u} = \mathbf{A} \mathbf{v} \quad (3.3)$$

where \mathbf{u} and \mathbf{v} are random vectors (i.e. vectors whose elements are random numbers) and \mathbf{A} is a matrix. As \mathbf{v} is a random vector it possesses a covariance matrix. Let $C(\mathbf{v})$ denote the covariance matrix of \mathbf{v} . It is easily shown (see, for example, Koch, 1997) that:

$$C(\mathbf{u}) = \mathbf{A} C(\mathbf{v}) \mathbf{A}^t \quad (3.4)$$

where $C(\mathbf{u})$ is the covariance matrix of \mathbf{u} . If this is applied to equation (3.2), while bearing in mind that the covariance matrix of a scalar (which can be considered to be a 1×1 matrix) is the variance (square of standard deviation) of that scalar, the following equation results.

$$\sigma_s^2 = \mathbf{y}^t C(\mathbf{k}) \mathbf{y} \quad (3.5)$$

where σ_s^2 is the variance of the prediction s . Thus, for a linear model, a statistical characterization of model parameters leads immediately to statistical characterization of predictive uncertainty.

Exercises

General

As has already been stated, it is not the intention of this document to provide a comprehensive tutorial on PEST usage, and/or of usage of its utility support software. It is its intention, however, to provide some examples of the use of PEST and its utility support software as it pertains to model predictive uncertainty analysis.

All files referenced in the exercises provided in this and other chapters of this document are provided. Hence a user can reproduce the results that are reported herein him/herself. In some cases reference is made to other PEST-supported methodologies and software that are used for input dataset construction and for undertaking more advanced analyses. Comprehensive coverage of all of these is provided in PEST documentation. In addition to this, a suite of comprehensive tutorial exercises supplied to PEST course attendees provides many more examples of PEST usage than those discussed herein.

Surface Water Model

PEST Input Dataset

A PEST control file named *precal.pst* is provided. An inspection of this file, and files cited therein, reveals the following.

- The NOPTMAX variable in the “control data” section of the *precal.pst* PEST control file is set to 0. Therefore PEST is asked to run the model only once. (NOPTMAX is the first variable on the ninth line of a PEST control file.)
- Parameters cited in Table 1.1 are listed in the “parameter data” section of *precal.pst*. So are three others. However these extra variables are fixed; meanwhile those cited in Table 1.1 are variable and are declared as log-transformed.
- The *precal.pst* PEST control file features the transformed HSPF parameters AGWRCTRANS and IRCTRANS. Their non-transformed counterparts are calculated by the PEST PAR2PAR utility that is run as part of the model. (PAR2PAR stands for “parameter to parameter transformation”.)
- Suitable bounds are supplied for all parameters; these bounds account for the transformed state of the AGWRCTRANS and IRCTRANS parameters.
- Only one observation appears in the “observation data” section of *precal.pst*. This is, in fact, not an observation at all. It is the value of maximum model-predicted flow over the period 19th August 1986 to 23rd August 1986 (this being outside of the model calibration period that is discussed in the next chapter). It is given a weight of 0.0 and a (completely arbitrary) observed value of 0.0. This is referred to as “the prediction” below, and in subsequent chapters of this document where discussion of this example continues.
- The model run by PEST is actually a batch file named *model_predict.bat*. An inspection of this file reveals that it does the following.
 - First it deletes the PAR2PAR output file *example.sup* which is actually the HSPF supplementary input file. This is done as a precautionary measure to ensure that if PAR2PAR fails to run, then HSPF will not read an input file written during a previous PAR2PAR run.

- Next it runs PAR2PAR.
- It then runs the HSPF simulator.
- Finally it runs a program named TSPROC. TSPROC stands for “time series processor”. This is the flagship of the PEST Surface Water Utility suite. It was built specifically to facilitate PEST usage in conjunction with models, such as surface water models and land use models, that produce lengthy time series and whose calibration requires that these time series be compared with observed counterparts.
- An inspection of the TSPROC input file *tsproc_predict.dat* reveals that TSPROC is asked to obtain observed and modelled time series from the WDM files *observ.wdm* and *output.wdm* respectively. It then evaluates the maximum observed flow for both of these over the period spanning 19th to 23rd August, 1986.
- The instruction file *tsproc_predict.ins* reads the maximum flow over the above time period from the TSPROC output file *tsproc_predict.out*. Note that, upon completion of PEST execution, this maximum flow value is recorded in both the PEST run record file *precal.rec* and the PEST residuals file *precal.res*.

Run PEST using the command:

```
pest precal
```

Then inspect *precal.res*. It should be apparent that the value of the prediction calculated for the set of parameter values provided in the “parameter data” section of the PEST control file *precal.pst* is 196.954 m³/sec.

Generating Random Parameter Values

“Initial values” for parameters provided in the fourth column of the “parameter data” section of file *precal.pst* are considered to be reasonable pre-calibration values for these parameters. They were chosen on the basis of expert knowledge of conditions in the simulated watershed. As such, they are values of minimum error variance, a concept that will be discussed in greater detail in the next chapter. For the moment they can be considered to be those that approach highest probability when assessed in terms of the prior parameter probability distribution, the same distribution that is featured on the right side of Bayes equation. It is not expected that these will lead to correct predictions of system behaviour, for we know that expert knowledge is accompanied by a high degree of expert ignorance. As an outcome of this ignorance we recognise that the “true” watershed parameter values may be significantly different from those which appear in the fourth column of the “parameter data” section of file *precal.pst*. In fact, we recognise that they may each vary between lower and upper bounds supplied in the 5th and 6th columns of this section. Let us assume that expert knowledge informs us that pre-calibration parameter probabilities diminish as these bounds are approached, and that these probabilities peak at the supplied initial parameter values. Thus we decree that the pre-calibration probability distribution is more likely to resemble a multi-normal distribution (centred on initial parameter values), rather than a uniform distribution.

File *param.unc* is a “parameter uncertainty file”. As is explained in PEST documentation, this file can take many forms. In the present case it simply provides the standard deviation of each parameter. Actually, it provides the standard deviation of the log (to base 10) of each parameter, this being necessary because all non-fixed (and hence variable) parameters are declared as log-transformed in the PEST control file *precal.pst*. Standard deviations recorded in *param.unc* were calculated as the difference between the logs of respective parameter

upper and lower bounds divided by 4.0. This rough calculation is based on the premise that if the parameters are log-normally distributed, and if the intervals between respective parameter bounds correspond roughly to 95% pre-calibration parameter confidence intervals, then the differences between these bounds correspond to approximately 4 parameter standard deviations. By providing only standard deviations in file *param.unc* we decree that pre-calibration inter-parameter statistical correlation is zero.

We will now generate 100 sets of random parameter values. Run the RANDPAR utility by typing its name at the screen prompt. Respond to its prompts in the manner shown below.

```
Enter name of existing PEST control file: precal.pst
- 10 parameters read from file precal.pst.
- 7 of these are adjustable.

Use (log)normal or (log)uniform distrib for param generation? [n/u]: n
Compute means as existing param values or range midpoints? [e/m]: e
Respect parameter ranges? [y/n]: y

Enter name of parameter uncertainty file: param.unc
- parameter uncertainty file param.unc read ok.

Enter name of parameter ordering file (<Enter> if none): <Enter>

Enter filename base for parameter value files: random
How many of these files do you wish to generate? 100

Enter integer random number seed (<Enter> if default): <Enter>
```

RANDPAR writes random parameter sets to files *randomN.par* where the integers 1 to 100 are substituted for *N*. These are recorded in “parameter value file” format. Use of this format allows easy manipulation of these parameter sets by PEST utilities. Note that parameters whose values are declared as fixed in the PEST control file are not randomized in these parameter value files; instead their values are maintained at those supplied in the PEST control file.

Examining Predictive Uncertainty

Our next task is to run the model many times using a different one of these random parameter sets on each occasion. This can be accomplished using the batch file *precal_runs.bat*. This file is reproduced below.

```
for /L %i in (1,1,100) do (
del temp.pst
del temp.res
parrep random%i.par precal.pst temp.pst
pest temp
copy temp.res temp.res.%i)
```

Figure 3.1 The batch file *precal_runs.bat* stored in the *sw* folder.

Run this batch file by typing its name at the screen prompt. It will run PEST many times as part of a processing loop in which the same set of commands is issued 100 times. This command sequence is comprised of the following.

- First, to guard against reading old files when expected new ones are not re-written, files produced by programs that are run during the processing loop are deleted.
- The PARREP utility (“PARREP” stands for “parameter replacement”) is used to build a PEST control file named *temp.pst* that is identical to *precal.pst* except for the fact

that initial parameter values in this new file are the same as those recorded in the parameter value file *randomN.par* where *N* is the current loop index.

- PEST is run. As NOPTMAX is set to 0 in *precal.pst* it is also set to 0 in *temp.pst*. Hence PEST runs the model once and then ceases execution.
- The residuals file *temp.res* written by PEST when it was run on the basis of the *temp.pst* PEST control file is copied to a file named *temp.res.N* where *N* is the same index as that assigned to the parameter value file on which basis the predicted maximum flow value contained in the residuals file was calculated.

Once the batch process depicted in Figure 3.1 is complete, we have at our disposal 100 predictions made on the basis of 100 parameter sets. However these predictions are inconveniently stored in 100 separate files. They can be collected into a single file using the RDMULRES utility. (RDMULRES stands for “read multiple results”.) But first an RDMULRES input file must be prepared which tells RDMULRES what files to read, and what numbers to read from those files. Such a file is *rdmulres_precal.in*. It is reproduced in Figure 3.2.

```
* observations
qmax_sim
* instruction file
resfile.ins
* model output file
temp.res.*
* integer list
1 - 100
* rdmulres output file
rdmulres_precal.out
```

Figure 3.2 The RDMULRES input file *rdmulres_precal.in* stored in the *sw* folder.

rdmulres_precal.in instructs RDMULRES to read the sequence of files *temp.res.1* to *temp.res.100*. It tells it to read just one number from each of these files; the instructions to read this number are provided in the instruction file *resfile.ins*. The quantity which is read is assigned the name *qmax_sim*. The 100 instances of *qmax_sim* are recorded in file *rdmulres_precal.out*. To generate *rdmulres_precal.out*, run RDMULRES using the command:

```
rdmulres rdmulres_precal.in
```

As an inspection will readily reveal, 100 values of the *qmax_sim* prediction generated on the basis of our 100 random parameter sets are now listed one after the other in file *rdmulres_precal.out*. If these are imported into a scientific plotting package, a plot such as that depicted in Figure 3.3 can be produced.

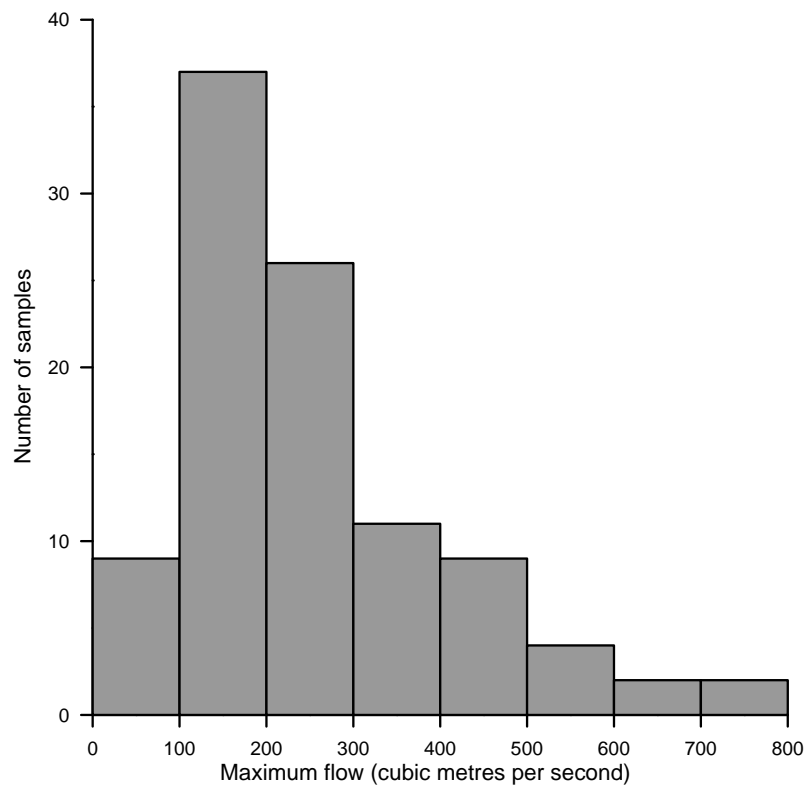


Figure 3.3 Histogram of model-predicted maximum flow calculated using 100 random parameter sets.

Linear Analysis

The PEST control file *precall.pst* is identical to *precal.pst* except for the fact that NOPTMAX is set to -2 rather than to 0 in this file. Hence when PEST is run it will calculate the Jacobian matrix (i.e. the sensitivity matrix) for the current case, and then cease execution. As 7 adjustable parameters are cited in *precall.pst* it will run the model 8 times - once on the basis of initial parameters and once with each parameter incrementally varied. As there is only one model output that PEST actually reads (this being the maximum flow over the period 19th to 23rd August 1986), the Jacobian matrix will have only one row. This row will be comprised of the sensitivities of this single model output to all 7 adjustable parameters. As such, it will comprise the y vector of equation 3.1.

Run PEST using the command:

```
pest precall
```

When PEST has finished execution, a file named *precall.jco* will be present in the model working directory. This is a binary file, and hence is not fit for human consumption. The PEST suite offers a variety of mechanisms for conversion of this file to ASCII format; these include the JACWRIT and JCO2MAT utilities. For the moment however, the JROW2VEC utility will be employed. (JROW2VEC stands for “Jacobian row to vector”.) This utility extracts a single row from the Jacobian matrix and writes it as a vector in PEST matrix file format. Run JROW2VEC using the command:

```
jrow2vec precall.jco qmax_sim qmax_sim_precal.vec
```

to record the extracted vector in file *qmax_sim_precal.vec*. (Note that if you forget the command-line protocol required to run any PEST utility, simply type its name at the screen prompt and the utility will inform you how to run it.)

Now run PREDUNC1 responding to its prompts as follows. (PREDUNC1 stands for “predictive uncertainty #1”.) PREDUNC1 was actually written to assess *post-calibration* predictive uncertainty. However in the process of doing this it also assesses *pre-calibration* predictive uncertainty. Furthermore, when all observation weights are zero in a PEST control file, pre- and post-calibration predictive uncertainty are the same.

PREDUNC1 screen prompts and responses are as follows.

```
Enter name of PEST control file: precall1.pst
Enter observation reference variance: 1

Enter name of parameter uncertainty file: param.unc
Enter name of predictive sensitivity matrix file: qmax_sim_precal.vec

Use which version of linear predictive uncertainty equation:-
    if version optimized for small number of parameters - enter 1
    if version optimized for small number of observations - enter 2
Enter your choice: 1

- reading PEST control file precall1.pst....
- file precall1.pst read ok.

- reading Jacobian matrix file precall1.jco....
- file precall1.jco read ok.

- reading predictive sensitivity matrix file qmax_sim_precal.vec....
- file qmax_sim_precal.vec read ok.

- reading parameter uncertainty file param.unc....
- parameter uncertainty file param.unc read ok.

- computing pre-calibration predictive uncertainty....
- forming XtC-1(e)X matrix....
- inverting C(p) matrix....
- inverting [XtC-1(e)X + C-1(p)] matrix....
- calculating post-calibration predictive uncertainty....

*****
*
*
*   Pre-cal predictive uncertainty =    149.6789
*   Post-cal predictive uncertainty =    149.6789
*
*
*****
```

The “predictive uncertainty” listed by PREDUNC1 is the standard deviation of predictive uncertainty. This actually compares very well with that computed empirically on the basis of the figures provided in the second column of *rdmulres_precal.out*, this being 152.33. Hence, in this case at least, the outcomes of linear and nonlinear analysis are very similar. However linear analysis requires far fewer model runs.

Groundwater Model

Cell-By-Cell Variability

To examine pre-calibration predictive uncertainty for our groundwater model we could follow an identical path to that outlined above for the surface water model. That is, we could

provide a statistical descriptor of pre-calibration uncertainty for the pilot point parameters used by that model, generate many different realisations for those parameters on the basis of this probability distribution, and then examine the variability of results using RDMULRES. However in this case we will introduce cell-by-cell variability to the model domain instead of parameter-by-parameter variability, this providing a better representation of the type of aquifer variability that exists in the real world. This will be done using the FIELDGEN utility. (FIELDGEN stands for “stochastic field generator”.) This utility program is a member of the PEST Groundwater Data Utilities suite; its inner workings are based on software supplied with the GSLIB geostatistical suite (Deutsch and Journel; 1998).

Before FIELDGEN can be run we need a “structure file”. This provides FIELDGEN with a statistical descriptor of heterogeneity within the model domain. This is supplied as a variogram (from which a covariance matrix can be easily derived as is demonstrated later in this document). File *struct.dat* is such a file. The format for this type of file is described in documentation for the PEST Groundwater Data Utility suite. File *struct.dat* informs us that, in the present instance, the entire model domain is characterized by a single geostatistical structure comprised of just one variogram. This is an isotropic exponential variogram having a range of 200m; this variogram describes variability of the *log* of hydraulic conductivity rather than of hydraulic conductivity itself. This same variogram (in conjunction with FIELDGEN) was used to generate the “reality field” of Figure 1.2.

Run FIELDGEN, responding to its prompts as follows.

```
Enter name of grid specification file: rect_model.spc
- grid specifications read from file model.spc
Enter name of conditioning pilot points file (<Enter> if none): <Enter>

Enter name of zonal integer array file: all1.inf
- integer array read from file all1.inf
Enter name of structure file: struct.dat

The following zones have been detected in the integer array:-

For zone characterised by integer value of 1:-
Enter structure name (blank if no field generation for this zone): structure1
Use simple or ordinary kriging [s/o] in field generation: s
Enter maximum number of previously simulated nodes to use: 20

How many realizations do you wish to generate? 100
Enter filename base for real array files: randcond
Write formatted or unformatted files? [f/u]: f

Mean field values within each zone are now requested.

Enter mean field value in zone with integer value 1: 1.0

Enter integer seed for random number generator [324853]: <Enter>
```

The following points will help to explain what you have just done. For further details see documentation for the PEST Groundwater Data Utilities suite.

- A “grid specification file” is a small ASCII file that provides dimensions and geographical specifications for a MODFLOW grid. See documentation for the Groundwater Data Utilities suite for details.
- Model-compatible integer arrays can be used to define zonation within a model domain. FIELDGEN allows different variograms to be employed within different zones. In the present case the model domain contains only one zone, this being associated with an integer index of 1. Thus all elements of the integer array contained in file *all1.inf* are 1.

- The outcomes of FIELDGEN execution are 100 model-compatible real arrays, these being stored in files named *rancondN.ref* where *N* ranges from 1 to 100. They all look a little like the heterogeneous field pictured in Figure 1.2 Any one of these could be imported into a MODFLOW graphical user-interface for user-inspection. Alternatively, with the help of programs such as LOGARRAY, REAL2MIF and REAL2SRF from the Groundwater Data Utilities suite, any one of them could be imported into a GIS or mapping package.

Exploring Predictive Uncertainty

An inspection of the MODFLOW layer property flow package input file *lpf.dat* reveals that MODFLOW reads hydraulic conductivity from a file named *hk.ref*. Hence if we wish to run MODFLOW many different times using all of the different random parameter fields that we have generated, each of them must, in turn, be substituted for *hk.ref*. This is accomplished using the batch file *precal_runs.bat* stored in the *gw* folder. It is reproduced below.

```
for /L %i in (1,1,100) do (
del path.dat
del time.dat
copy randcond%i.ref hk.ref
mf2k < mf2k.in
finaltime < finaltime.in
copy time.dat time%i.dat)
```

Figure 3.4 The batch file *precal_runs.bat* stored in the *gw* folder.

An inspection of file *precal_runs.bat* reveals the following.

- As a precaution against model failure, files written by one model component and read by another model component are deleted at the beginning of each pass of the processing loop provided in this file.
- Keyboard response to the MF2K and FINALTIME programs are supplied through files *mf2k.in* and *finaltime.in*; MF2K and FINALTIME are directed to look to these files for keyboard input (rather than to the keyboard) using the “<” symbol.
- When the entire batch process has reached completion, predicted travel times are stored in files *time1.dat* to *time100.dat*.

Initiate this batch process by typing:

```
precal_runs.bat
```

at the screen prompt. Once the batch process is complete particle travel times computed using the 100 different random hydraulic conductivity fields can be accumulated using the RDMULRES utility. An appropriate RDMULRES input file is depicted in Figure 3.5.

```
* observations
part_time
* instruction file
timefile.ins
* model output file
time*.dat
* integer list
1 - 100
* rdmulres output file
rdmulres_precal.out
```

Figure 3.5 The RDMULRES input file *rdmulres_precal.in* stored in the *gw* folder.

Run RDMULRES using the command:

```
rdmulres rdmulres_precal.in
```

to collect results into a single file named *rdmulres_precal.out*. The frequency distribution of travel times is shown in Figure 3.6.

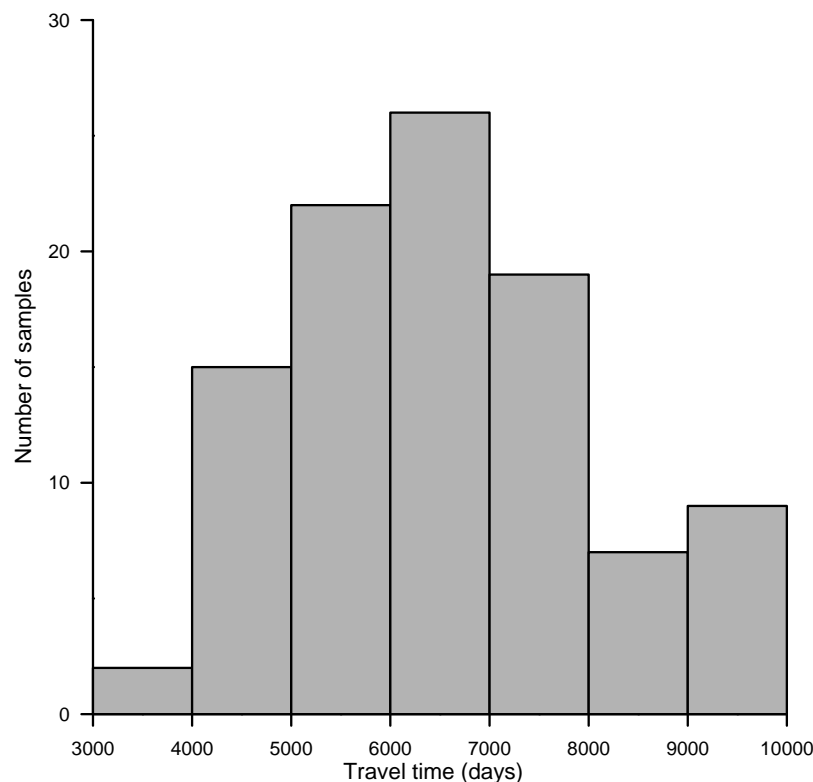


Figure 3.6 Histogram of model-predicted particle travel times based on 100 random parameter fields.

The skewness of the distribution depicted in Figure 3.6 is an outcome of the nonlinearity of the model. This, in turn, is an outcome of the dependence of the prediction of interest on hydraulic property heterogeneity as it exists at a fine scale.

With slightly (but not much) more trouble, stochastic variability of other aquifer properties such as porosity, and of other model characteristics such as the strength of inflow into the northern model boundary, could be included in our analysis in a similar fashion.

4. Getting Information out of Data

History-matching

The process of adjusting model parameters until a good fit between model outputs and field measurements is obtained is often referred to as “calibration”. A model whose parameters have been adjusted in this fashion is often referred to as a “calibrated model”. There is a certain sense of finality in that term, some of this inherited from the fact that the word “calibration” is commonly associated with finely-tuned laboratory instruments. Unfortunately, however, when applied to environmental modelling, the term can be misleading, both in its direct and implied sense.

History-matching is a vital part of preparing a model for use in decision support. However the outcomes of the history-matching process must be viewed in a way that is in harmony with:

- a mathematical description of what history-matching can actually achieve, and
- the practicalities of what history-matching can achieve when applied to a model which is a defective simulator of real-world environmental processes.

History-matching is normally implemented through minimizing a so-called “objective function”. This can be defined in many ways. A common way is as the sum of weighted squared differences between model outputs and field measurements. Ideally greater weights should be given to measurements which are thought to be less contaminated by errors that are incurred in the actual making of those measurements (often referred to as “noise”). However in practice, as will be discussed below, weighting strategies may need to be more flexible than this in order to accommodate the fact that model-to-measurement misfit is usually dominated by so-called structural noise rather than measurement noise.

Bayes Equation

In the Bayesian context the fit between model outputs and historical field observations determines the magnitude of the likelihood term, this being the first term on the right of equation 2.3. Parameters which give rise to a better fit result in a greater likelihood function. In Bayes equation “better” is defined in a statistical sense, for model-to-measurement misfit is assumed to be an outcome of the fact that measurements of system state are accompanied by random error. Betterment of fit, and hence increase in parameter likelihood, is calculated using the probability distribution that is associated with this error.

Two immediate outcomes of this are as follows.

- Bayes equation makes no inference of parameter uniqueness. Where parameters are many and data is scarce, it is not hard to imagine that many different combinations of parameters will provide the same or similar level of fit. The ranking, in terms of posterior probability, of different parameters sets which yield the same likelihood function must then take place on the basis of the prior probability of those parameter sets.
- The statistics of measurement noise are assumed to be known - or “almost known”. In practice, some aspects of the statistical distribution of measurement noise (mainly its overall magnitude) can be estimated through the history-matching process, while others (mainly variables that govern its shape) are assumed to be known because their estimation is difficult or impossible. In doing this it must be realised, however, that

assumptions regarding the statistical properties of measurement noise can have a large influence on what parameter sets are construed to be “better” than others, and hence on the nature of the inferred posterior parameter probability distribution.

From the posterior parameter probability distribution inferred through Bayesian analysis the posterior distribution of any model prediction can be calculated. Parameters of which field measurements are informative may have a significantly narrower posterior distribution than prior distribution. Likewise, predictions that are sensitive to parameters (and parameter combinations) of which the measurement dataset is informative may also have a significantly narrower posterior probability distribution than prior probability distribution. This is illustrated in Figure 4.1.

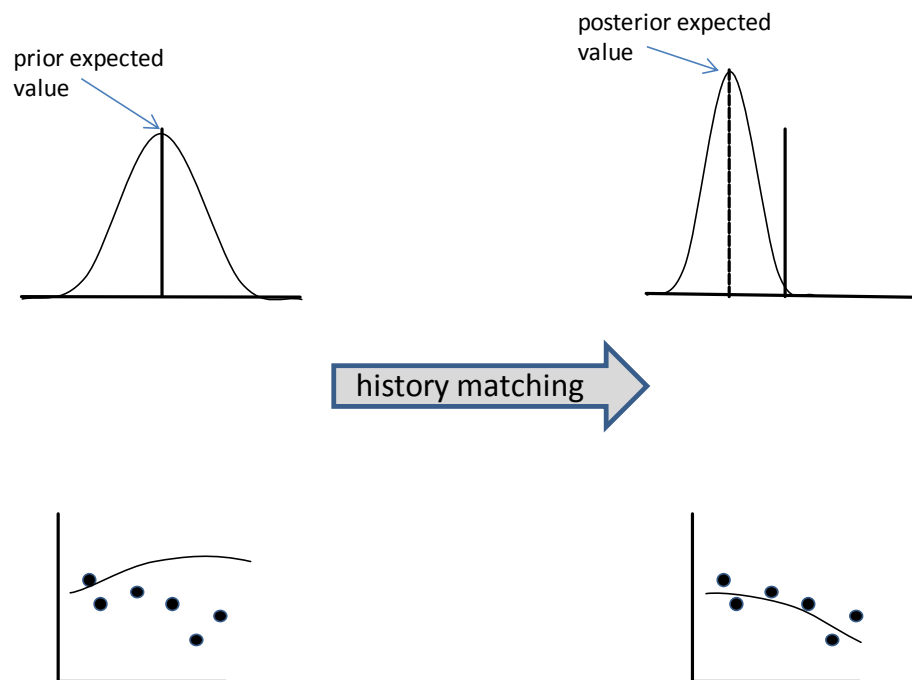


Figure 4.1 Schematic representation of Bayesian analysis.

Figure 4.1 attempts to illustrate a fact that has been emphasized in previous sections of this document. When using an environmental model to inquire into future system behaviour, all that can be expected is a probability distribution at best, or a range of predictive possibilities at worst. The latter is narrower than the range of prior predictive possibilities as all information has been taken into account through supplementing expert knowledge with the information that resides in measurements of system state.

Calibration

Unfortunately, direct manipulation of parameter and predictive probability distributions is a numerically burdensome procedure. To be sure, software is available which can do this. In particular, Markov chain Monte Carlo (MCMC) analysis allows a modeller to define the posterior parameter distribution by directly sampling from it. As such it may be considered to represent the “purest” way to use a model as it constitutes a direct implementation of Bayes equation. A problem with the method, however, is that sampling of the posterior parameter distribution requires many model runs. Furthermore, as the number of parameters used in the analysis increases, the number of model runs required to implement the analysis tends to rise dramatically, especially if the inverse problem that defines the history-matching process is characterised by a high or even moderately-dimensional null space (see below). Furthermore,

attempts to reduce the number of parameters involved in the analysis through devices such as lumping, fixing, tying and averaging often erode the capacity of MCMC analysis to achieve what it sets out to achieve, for a wide posterior predictive uncertainty distribution is often a direct outcome of the fact that many parameters are ill-informed by measurements of system state.

Other approaches to Bayesian-based history-matching include formulation of equations that directly encapsulate prior and posterior probability distributions, and direct solution for the parameters that govern these distributions. This can be a more fruitful approach than MCMC analysis where model run times are high. However it requires considerable model-specific programming and hence specialized software. Furthermore it often requires the making of assumptions pertaining to the nature, type and size of parameter variability that may further reduce the generality of its application.

For these reasons, and for cultural reasons, most history-matching is undertaken as part of the process of “model calibration”. In the environmental modelling context, this is an almost mystical term whose hidden meaning has more strength than its actual meaning. To the non-specialist the term “calibrated model” has overtones of predictive certainty which, in the environmental sphere, few if any models can lay claim to. Given that the term has no place in Bayesian analysis, and given that Bayesian analysis alone provides a complete mathematical characterization of what the history-matching process can and should achieve, it is hardly surprising that the term “calibration” has found meanings which have little or no scientific basis. Unfortunately the term is well suited to advertising campaigns that bestow on existing, or yet-to-be-built, models predictive powers for which justification must be sought in wishful thinking rather than in mathematics.

The word “calibration” infers parameter uniqueness, for the calibration process purports to seek one set of parameters which the model will then employ for the making of predictions of future environmental behaviour. Numerically, it is much easier to find a single set of parameters than a suite of parameters on which basis a posterior probability distribution can be built. Hence the process of model calibration has a physical allure that compliments its metaphysical allure.

If a single set of parameters is sought in lieu of a posterior parameter probability distribution, or a suite of parameters that sample the posterior parameter probability distribution, this raises some serious questions. In particular:

- What properties should the single set of parameters possess?
- How much propensity for error exists when using these parameters to make predictions of future environmental behaviour?

If we are indeed going to select a single set of parameters for predictive model usage (this being the set of parameters which is deemed to “calibrate” the model), it makes sense that these parameters should be “as right as possible”; that is, of all the parameter possibilities that we may employ, the values that we assign to them should be those of minimum error variance. If the posterior parameter probability distribution is symmetric, values assigned to parameters which bestow on them their “calibrated” status should thus be their posterior “expected values” in the statistical sense. If the model provides a correct representation or reality, predictions based on these parameters will then be of minimum error variance.

This then defines the goal of the model calibration process. That is, this process must seek parameter values of minimized error variance so that predictions that depend on these parameters may also be of minimized error variance. It is important to note, however, that

“minimized error variance” does not mean “minimal error variance”. It only means that calibrated parameter values lie somewhere near the centres of their posterior probability distributions, and that a prediction made on the basis of these parameter values lies somewhere near the centre of its posterior probability distribution. Thus the potential for error incurred by making a prediction on the basis of these parameter values is approximately symmetrical with respect to the prediction itself. This is what is responsible for its minimization.

It is important to note that significant reduction of the width of the posterior predictive probability distribution below its prior width may or may not occur through the history-matching process; this depends on the information content of the measurement dataset. The issue of how broad is the posterior predictive probability distribution is separate from that of acquiring, through the history-matching process, an ability to make a prediction which is centrally located with respect to this distribution.

Nevertheless, the potential for predictive error must be quantified if model-based decision-making is to have integrity for reasons that have already been outlined. Ideally, the post-calibration potential for error in a prediction should be the same as the inherent posterior uncertainty of that prediction as calculated from the posterior predictive probability distribution of that prediction achieved through Bayesian analysis applied to a model which is a perfect simulator of environmental behaviour. In practice the potential for predictive error will be somewhat greater than this as an outcome of numerical imperfections that attend both of the processes of model calibration and model simulation. Both of these increase the potential for error in predictions made by a calibrated model; hence both of these must be taken into account when assessing that potential. This, unfortunately, entails an irremovable element of subjectivity.

The Null Space

The process of calculating a single parameter set with a special set of properties from a set of field measurement is often referred to as “inversion”. The problem itself is often referred to as “the inverse problem”. Inverse problems are often difficult to solve, the reason being that they are often characterised as being, in mathematical parlance, “ill-posed”.

Inverse problem ill-posedness arises from the fact that if a model is endowed with parameterization complexity that reflects the complexity and heterogeneity of reality, then rarely, if ever, can these parameters be estimated uniquely on the basis of measurements of system state alone. Linear algebra provides a useful vehicle for analysing this problem.

As in the previous chapter, let model parameters be represented by the vector \mathbf{k} . We will suppose that the elements of \mathbf{k} represent system properties at a level of complexity that is salient to the modelling task at hand. Ultimately “salience” is determined by the fact that critical model predictions may be sensitive to those parameters. Let the matrix \mathbf{Z} represent the action of the model under calibration conditions, and let the elements of the vector \mathbf{h} comprise the set of measurements of system state that comprise the calibration dataset. Then the action of the model when supplied with historical system drivers can be written as:

$$\mathbf{h} = \mathbf{Z}\mathbf{k} + \boldsymbol{\varepsilon} \quad (4.1)$$

where $\boldsymbol{\varepsilon}$ is an (unknown) vector whose elements represent noise associated with the elements of \mathbf{h} . For simplicity, let us assume for the moment that measurement noise is zero. Then:

$$\mathbf{h} = \mathbf{Z}\mathbf{k} \quad (4.2)$$

From the above equations it is apparent that each row of the matrix \mathbf{Z} contains the sensitivities of a given model output (for which there is a corresponding field measurement) to all of the parameters \mathbf{k} . If, because of the level of parameterization complexity represented by \mathbf{k} , there are fewer elements of \mathbf{h} than of \mathbf{k} , then \mathbf{Z} is rectangular with its long direction horizontal. If this is the case, it can be shown that there exist non-zero vectors $\delta\mathbf{k}$ for which:

$$\mathbf{0} = \mathbf{Z}\delta\mathbf{k} \quad (4.3)$$

By adding (4.2) to (4.3) it is easily seen that if \mathbf{k} satisfies (4.2) then so does $(\mathbf{k}+\delta\mathbf{k})$. Hence inference of \mathbf{k} from \mathbf{h} is nonunique.

Nonuniqueness of the inverse problem is the rule rather than the exception. It can readily occur even where there are more observations than parameters, though it is not guaranteed to occur under these conditions as it is when there are less observations than parameters. If equation (4.3) is satisfied by even one non-zero vector $\delta\mathbf{k}$, then the matrix \mathbf{Z} is said to possess a null space. The number of dimensions in parameter space occupied by the null space is at least equal to column-over-row surplus (if one exists). However it is generally much larger than this.

It follows that parameter space can be subdivided into two subspaces - a null space and a so-called "solution space", the latter being the orthogonal complement of the former. A unique \mathbf{k} can be calculated from the \mathbf{h} of equation (4.2) if the search for \mathbf{k} is restricted to the solution space of the matrix \mathbf{Z} . Unfortunately however, it is unlikely that the real parameter set \mathbf{k} falls within the solution space of \mathbf{Z} . In finding \mathbf{k} , all that we in fact find is the projection of \mathbf{k} onto the solution space. Because the orthogonal complement of the solution space, i.e. the null space, contains inestimable system property detail it follows intuitively that the projection of reality onto the solution space constitutes a simplified solution to the inverse problem of model calibration. It can however be shown that, under the right circumstances, it is also the solution to the inverse problem of minimum error variance. This can be verified by intuition, for as soon as we venture from the solution space into the null space we increase our potential for wrongness as we run the risk of venturing into the null space in the wrong direction (for example up rather than down in Figure 4.2).

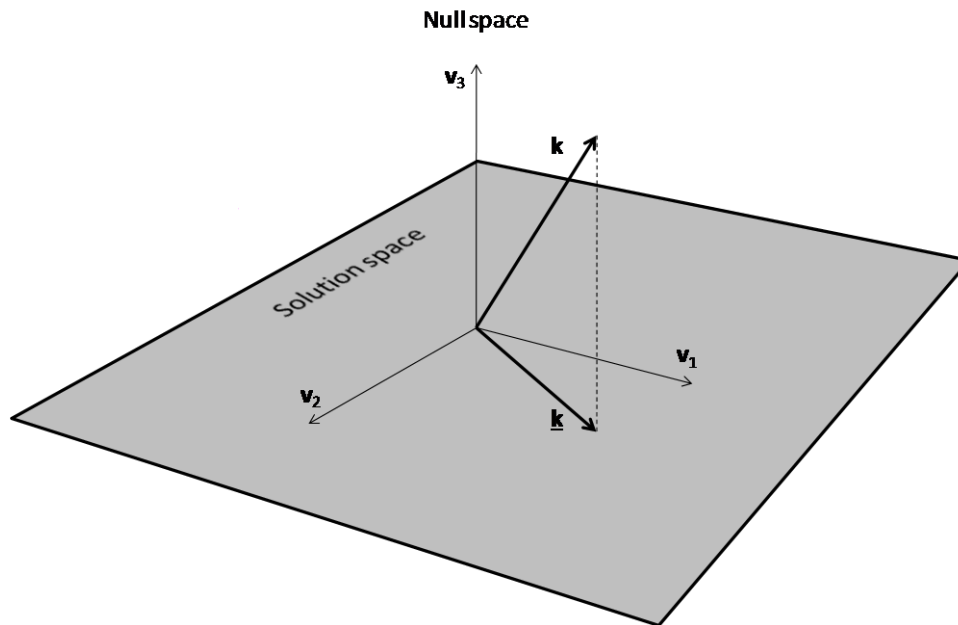


Figure 4.2 Schematic of parameter space showing solution and null spaces. \mathbf{k} represents the true parameter set. All that can be estimated through the calibration process is its projection into the solution space.

In practice, in calibrating a real-world model, the null space needs to have more dimensions than those which strictly satisfy (4.3), for it also needs to include parameter sets $\delta\mathbf{k}$ for which \mathbf{h} is very low, and not just zero. This is because, as Moore and Doherty (2005) show, attempts to estimate parameter sets $\delta\mathbf{k}$ for which $\delta\mathbf{h}$ is nearly zero will lead to a propensity for estimation error that is greater than the pre-calibration uncertainty of these parameter sets.

The partitioning of parameter space into solution and null spaces rarely takes place along neat parameter boundaries. To be sure, some parameters are entirely inestimable and hence lie within the null space. These are parameters to which all model outputs corresponding to field observations are insensitive. In other cases a parameter may lie partly within the solution space and partly within the null space. This indicates that the calibration dataset provides some information pertaining to that parameter - but that this information must be shared with at least one other parameter. The individual parameters amongst which this information is shared therefore show a high or infinite amount of statistical correlation in the posterior parameter probability distribution. This is because the information contained within the calibration dataset is sufficient only for estimation of a *combination* of these parameters, rather than all of them individually.

Figure 4.3 attempts to illustrate this situation. Doherty and Hunt (2009) define the direction cosine between an individual parameter and its projection into the solution space as its “identifiability”. This ranges between 1 for a parameter that lies entirely within the solution space, and 0 for a parameter that lies entirely within the null space.

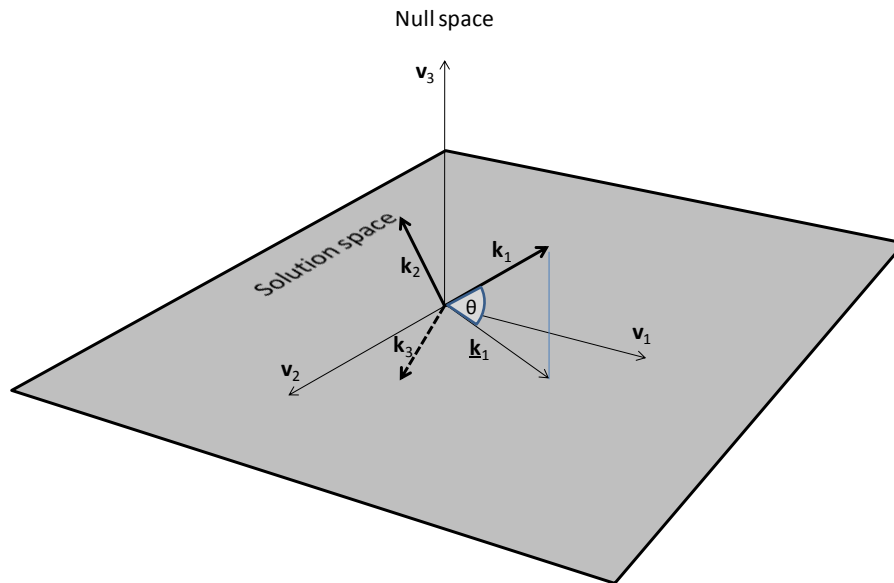


Figure 4.3. Vectors k_1 , k_2 and k_3 point along parameter axes. These are different from the vectors v_1 , v_2 and v_3 which define orthogonal axes through which parameter space can be partitioned into solution and null spaces. The cosine of θ is defined as the identifiability of parameter k_1 . Identifiability is defined in the same way for other parameters.

As stated above, restriction of the search for a solution to the inverse problem of model calibration to parameter combinations that lie within the solution space effectively restricts that search to the simplest set of parameters which allow the model to effectively reproduce historical system behaviour. This accords with the much-repeated precept offered by many calibration sages that the calibration process should pursue the principle of parsimony. It is important to note, however, that parsimony is desirable not as an end in itself, but because it is a means of achieving the only thing that is worth achieving through the calibration process, that is the solution to the inverse problem that is of minimum error variance. As we shall see shortly, too much simplicity, or inappropriately defined simplicity, can act as a barrier to achieving this goal.

Singular value decomposition (SVD) provides a means of subdividing parameter space into solution and null spaces. It can be shown that any matrix \mathbf{Z} can be decomposed according to the formula:

$$\mathbf{Z} = \mathbf{U}\mathbf{S}\mathbf{V}^t \quad (4.4)$$

where the columns of the matrix \mathbf{U} are orthogonal unit vectors which span the range space of \mathbf{Z} , the columns of \mathbf{V} are orthogonal unit vectors which span the domain of \mathbf{Z} (parameter space in our case), and \mathbf{S} is a diagonal matrix composed of positive or zero diagonal elements arranged from highest to lowest. The columns of \mathbf{V} as they pertain to a three-dimensional parameter space are actually depicted in Figures 4.2 and 4.3 as the vectors v_1 , v_2 and v_3 . Columns of \mathbf{V} corresponding to zero-valued singular values span the null space; in practice, columns associated with weakly estimable parameter combinations for which singular values are near zero are also assigned to the null space, this providing a safeguard against amplification of parameter error through over-fitting.

Regularisation

General

“Model calibration” is the search for a unique parameter set. From the above discussion it is apparent that this unique parameter set cannot be the “reality” parameter set, for the latter contains details that are simply not inferable on the basis of the measurement dataset. Nor are they inferable on the basis of expert knowledge which, as we have seen, is characterized by a prior parameter probability distribution (possibly conditioned by point measurements of system properties) rather than by parameter certainty. We have also seen that while the calibrated parameter field will almost certainly be incorrect, it can nevertheless be “optimal” in the sense that its potential for wrongness, though it may be considerable, is minimized. So far in this document we have discussed what this means from both a Bayesian framework (where “optimal” is characterized as leading to predictions of minimized error variance), and from a parameter subspace framework (where “optimal” is characterized as absence of null space components). It can be shown that as measurement noise approaches zero these lead to exactly the same parameter set (Albert, 1972). In practice they can lead to parameter estimates which are slightly different. However these differences are normally small compared with the potential for error which exists in either.

The process of finding a unique solution to an ill-posed inverse problem, and of thereby achieving a parameter set which is deemed to calibrate a model, is called “regularisation”. The ways in which regularisation is most commonly implemented are now briefly discussed.

Tikhonov Regularization

In its simplest form, Tikhonov regularization attempts to guide solution of the inverse problem towards parameter estimates which can be considered to be expected values (in the statistical sense) of the posterior parameter probability distribution (and hence constitute parameter estimates that approach minimum error variance). The modeller must supply a default value for all parameters, and/or default values for relationships between parameters; an example of the latter are default differences of zero between neighbouring spatial parameters - this implying a default condition of parameter field homogeneity. Collectively these parameter values and/or parameter relationships define expectations (in the statistical sense) from the prior parameter probability distribution. Hence they define parameter estimates of minimum error variance based on expert knowledge alone. When using PEST for solution of the inverse problem of model calibration, preferred parameter values and/or conditions can be supplied through the “prior information” mechanism, or as more complex nonlinear “regularisation observations”.

The parameter estimation process is also provided with a suite of measurements of system state. The inverse problem of model calibration is then formulated as a constrained optimisation problem in which a suitable, user-defined value for the target “measurement objective function” (defined through model-to-measurement misfit) is sought subject to the constraint that the “regularisation objective function” (defined as departures of parameters from their default values or preferred condition) is minimized. The target measurement objective function is assigned a value that reflects what is considered to be the ambient level of measurement noise. By seeking (but not exceeding) this level of model-to-measurement fit subject to the constraint that departures from pre-calibration optimality are minimized, the calibration goal of minimized propensity for parameter and predictive error is formally sought.

Tikhonov regularisation has many attractive features, the most obvious of which is that it provides receptacles for information that is forthcoming from both the user and from the calibration dataset. Its chief disadvantage is that it tends to suffer from numerical instability as parameter optimality is approached. Behind the scenes a trade-off is implemented between fitting the data on the one hand and fitting a user's preconceptions as they apply to all parameters involved in the parameter estimation process on the other hand. Numerically, this trade-off is sometimes difficult to apply. It is made no less difficult by the fact that misfit is often dominated by structural noise whose statistical properties are unknown, but whose variance is much higher than that of measurement noise. Avoidance of over-fitting may thus become a trial and error process in which the Tikhonov-based inversion process is repeated with different values assigned to the target measurement objective function. Determination of the relative strengths with which Tikhonov constraints must be applied to different types of parameters can also be problematic. However PEST provides some help with this process; see PEST documentation of the IREGADJ regularization control variable.

Subspace Regularization

As described above, Tikhonov regularization achieves uniqueness by supplementing information within the calibration dataset with information that is born of expert knowledge. Subspace regularisation (of which the flagship is so-called "truncated singular value decomposition") takes the opposite approach. It identifies parameter combinations which are inestimable on the basis of the current calibration dataset, and removes them from the parameter estimation process altogether. These are identified as those that are associated with zero and low singular values, and thus occupy the calibration null space. The solution to the inverse problem is thus comprised entirely of estimable combinations of parameters which, by definition, belong to the calibration solution space. As has been discussed above, these estimable combinations of parameters normally correspond to broad-scale features of a model's parameterization.

When implementing subspace regularisation, a modeller should ensure that initial parameter values which are provided to the parameter estimation process are in fact preferred parameter values from an expert knowledge point of view. Behind the scenes, it is actually departures from these initial values that are estimated. Because departures from these values which occupy the null space are not estimated, the only departures from initial parameter values that are tolerated are those that are supported by the data. For those that are not supported by the data, the user's initial choice prevails. If initial parameter values thus embody user expert knowledge, and therefore constitute pre-calibration minimum error variance parameter estimates, the solution to the inverse problem formulated in this way approaches that of post-calibration minimum parameter error variance status. This is an outcome of the fact that only those parameter combinations whose estimates achieve reduced error variance through the calibration process are adjusted through this process, while those that do not remain unchanged.

Calibration implemented through subspace regularization is unconditionally numerically stable. This is ensured because, by definition, the solution space is comprised only of parameter combinations which are indeed uniquely estimable. The truncated singular value decomposition (SVD) process through which parameter estimation is most easily achieved by this means simply refuses to estimate parameter combinations corresponding to singular values which are below a certain threshold, and hence are not robustly estimable.

Unfortunately, however, truncated SVD as a mechanism for solution of the inverse problem of model calibration suffers from two significant shortcomings. The first is that it is difficult

to link the singular value truncation threshold to the expected level of measurement/structural noise. Hence while unconditional numerical stability is achieved, prevention of over-fitting is less easily achieved as it is not part of the formal definition of the inverse problem as it is for Tikhonov regularisation. Secondly, because expert knowledge is an implicit rather than explicit part of the inversion process (being provided only through parameter initial values rather than in the form of a possibly complex suite of linear and/or nonlinear parameter relationships), the parameter fields that emerge from the truncated SVD process rarely have the same aesthetic appeal as those which emerge from a calibration process that is implemented using Tikhonov regularization.

Hybrid Regularization

Both Tikhonov and subspace regularization have strengths and weaknesses. These tend to complement each other. Hence when used together, the outcome is an inversion scheme that makes best use of the benefits of each of them at the same time as it mitigates the deleterious effects of their weaknesses.

PEST allows a user to estimate parameters using truncated SVD as a solution device to the inverse problem of model calibration while allowing this problem to be formulated as a constrained optimization problem, with constraints supplied as linear or nonlinear parameter relationships in accordance with the Tikhonov approach outlined above. Experience has demonstrated on many occasions that this approach provides pleasing parameter fields at the same time as it maintains numerical stability while simultaneously preventing over-fitting.

Significant gains in computational efficiency can be had through undertaking singular value decomposition of the global sensitivity matrix only intermittently, and on this basis defining a limited number of “super parameters” equal in number to the dimensionality of the calibration solution space. These new super parameters are in fact the projections of the solution to the inverse problem onto the parameter axes which span the calibration solution space. Through estimation of these projections alone, a solution to the inverse problem of model calibration is achieved. Estimation thus takes place on the basis of an often vastly reduced parameter set, this decreasing the numerical burden of the parameter estimation process enormously. At the same time, Tikhonov constraints are exercised on native parameters, thereby ensuring optimal use of expert knowledge in the parameter estimation process. See PEST’s “SVD-Assist” methodology for further details.

Manual Regularization

Despite the ability that PEST provides for implementing highly parameterized inversion using the techniques discussed above, many models are still regularized manually. Thus, prior to initiating the calibration process, the modeller must reduce the number of parameters which he/she estimates by fixing certain parameters at pre-calibration preferred values and amalgamating others into a smaller parameter set. Through examining the nature of model-to-measurement misfit achieved through the parameter estimation process based on this reduced parameter set, and by inspecting statistics that are produced as an outcome of the calibration process, the modeller then assesses whether more or less parameter reduction needs to occur, and/or if parameter reduction needs to take place in different ways.

In some ways, manual regularisation attempts to achieve the same thing as truncated SVD in that the dimensionality of the inversion problem is reduced to that which is estimable. Meanwhile, though it is not referred to as such, parameters that occupy the null space are assigned fixed values, while parameter combinations that are inestimable (these normally expressing fine parameterization detail) are hidden from the purview of the parameter

estimation process through parameter amalgamation. In the spatial parameterization context, parameter amalgamation often takes place through definition of a small number of zones of assumed parameter constancy.

If enough care is taken in implementing manual regularization something approaching the minimum error variance solution to the inverse problem of model calibration can indeed be achieved. However, in the author's opinion, the disadvantages of this approach tend to outweigh its "advantages". In fact, in the author's experience, the main "advantage" to be gained through implementing this kind of regularisation lies in the ability (welcomed by some modellers) to defer the need to learn the details of a superior approach.

Disadvantages of manual regularisation include the following.

- In a spatial setting characterized by system property heterogeneity, it is better to define many rather than few parameters throughout the model domain. This provides the parameter estimation process with a license to introduce heterogeneity to the model domain wherever it needs to, to the extent that it needs to, and in the way that it needs to, rather than having to respect pre-defined (and often inappropriate) mechanisms for expression of heterogeneity that may seriously compromise such expression.
- A considerable amount of trial and error is often required in determining how many parameters can be estimated on the basis of a given calibration dataset. If too few parameters are estimated, less information is obtained from the calibration dataset than it contains. If an attempt is made to estimate too many parameters, over-fitting occurs. In neither case is parameter optimality achieved.
- Proper implementation of either or both of Tikhonov and subspace regularization provides a mathematical guarantee that something approaching parameters of minimized error variance will be achieved through the parameter estimation process. No such guarantee is afforded through manual regularisation.
- Less than optimal definition of spatial parameter variability through use of cumbersome and unrealistic parameterization devices such as zones of piecewise constancy can introduce structural noise to model outputs. This erodes the capacity of the parameter estimation process to obtain information from the calibration dataset, thereby incurring an increased propensity for parameter and predictive error. At the same time, the magnitude of this error is difficult to quantify.
- Where parameterization complexity complements model process complexity at the same time as it represents system property heterogeneity that is salient to predictive variability, it is mandatory that regularization be implemented mathematically rather than manually, for the inverse problem of model calibration is otherwise unsolvable. As a by-product of implementing mathematical regularisation a definition of the null space is achieved (or something approaching the null space in the case of Tikhonov regularization). This is the (unavoidable) source of most parameter and predictive uncertainty on most occasions. Hence use of mathematical regularization based on an underlying parameter field comprised of many parameters provides a far better foundation for post-calibration uncertainty analysis than does manual regularization employing few parameters.

Structural Regularization

Regularization is built into the design of many models by virtue of the lumped nature of their parameters. Implicit in the design of many lumped-parameter models is the fact that they are “made to be calibrated” against measurement datasets comprised of one or many data types gathered at one or many locations over many years. Model design is often specifically aimed at endowing a model with a parsimonious set of parameters that is uniquely estimable on the basis of commonly-available datasets, at the same time as it provides optimal receptacles for the information content of these datasets. As such, model design of this type implicitly attempts to achieve a similar outcome to the use of subspace methods in solving the inverse problem of model calibration (or perhaps an even better outcome as it may provide superior accommodation of the nonlinear relationship between model outputs and model parameters). Calibration of models of this type may therefore indeed achieve parameter values of minimized error variance. A problem arises, however, where post-calibration predictive uncertainty is explored, for while failure to represent parameterization detail comprising the null space may not compromise optimality of the calibration process, it will probably compromise the ability of the uncertainty analysis process to properly define post-calibration predictive variability as it pertains to predictions of system behaviour under extreme conditions. Also, representation of parameters in a lumped manner may detract from a modeller’s ability to explore the effects of certain proposed environmental management strategies on future environmental behaviour. For example it may not be possible to represent land use changes at a farm or sub-regional level in such a parsimonious model as parameterization density is simply too broad to support definition of proposed changes.

To overcome the latter problem, regional lumped-parameter surface water and land use models are often constructed from many submodels that seek to simulate hydrologic processes that are operative at the land management scale. If this is done, some form of (mathematical or manual) regularisation must then be implemented during the calibration process. With the introduction of a higher parameterization density in this manner also comes the ability to conduct post-calibration parameter and predictive uncertainty analysis with greater integrity.

Some Equations

The following equations are presented for completion. They are not derived. Nor is it necessary that they be understood.

We take equation (4.1) as our starting point. When calibrating a model, a vector \mathbf{k} of “calibrated parameter values” is calculated according to:

$$\mathbf{k} = \mathbf{G}\mathbf{h} \quad (4.5)$$

where \mathbf{G} is a matrix that depends on the regularization method employed in the calibration process. If the measurement dataset contains no noise and if parameters are normalized with respect to their innate variability, then ideally \mathbf{G} should be the Moore-Penrose pseudo inverse of \mathbf{Z} , this leading to a parameter set of minimum norm and hence (if norm is defined in an appropriate way) the parameter set of minimum error variance. Where there is noise in the calibration dataset, use of a generalized inverse of \mathbf{Z} (whether this is the Moore-Penrose pseudo inverse or some other generalized inverse) to calculate \mathbf{k} would lead to over-fitting. Hence \mathbf{G} must be derived through other means - any one of these requiring definition of some kind of regularisation. Formulas for \mathbf{G} differ according to the type of regularisation employed. Ideally, however, they should converge to the same formula as the noise associated with \mathbf{h} decreases to zero (as do truncated SVD and properly-designed Tikhonov-based inversion).

For truncated SVD:

$$\mathbf{G} = \mathbf{V}_1 \mathbf{S}_1^{-1} \mathbf{U}_1^t \quad (4.6a)$$

while for Tikhonov regularization:

$$\mathbf{G} = (\mathbf{Z}^t \mathbf{Q} \mathbf{Z} + \beta^2 \mathbf{T}^t \mathbf{W} \mathbf{T})^{-1} \mathbf{Z}^t \mathbf{Q} \quad (4.6b)$$

where:

\mathbf{Q} is a measurement weighting matrix, ideally proportional to $\mathbf{C}^{-1}(\boldsymbol{\varepsilon})$, where $\mathbf{C}(\boldsymbol{\varepsilon})$ is the covariance matrix of measurement noise;

\mathbf{U} , \mathbf{V} and \mathbf{S} are obtained through singular value decomposition of $\mathbf{Q}^{1/2} \mathbf{Z}$, the subscript of “1” on these matrices indicating use of pre-truncation (and hence non-zero) singular values;

\mathbf{T} is a matrix which expresses Tikhonov constraints;

\mathbf{W} is a weighting matrix for Tikhonov constraints; ideally if \mathbf{T} provides preferred pre-calibration values for the parameters \mathbf{k} then \mathbf{W} should be proportional to $\mathbf{C}^{-1}(\mathbf{k})$ where $\mathbf{C}(\mathbf{k})$ is the pre-calibration covariance matrix of innate parameter variability;

β^2 is a factor adjusted during the regularized inversion process; it is equivalent to a Lagrange multiplier employed in the constrained optimisation process described above through which Tikhonov regularization is implemented.

For manual regularization, \mathbf{G} can be expressed as:

$$\mathbf{G} = \mathbf{L}(\mathbf{X}^t \mathbf{Q} \mathbf{X})^{-1} \mathbf{X}^t \mathbf{Q} \quad (4.6c)$$

where:

\mathbf{L} is a matrix through which elements of $\underline{\mathbf{k}}$ are computed from a reduced parameter set $\underline{\mathbf{p}}$; elements of $\underline{\mathbf{p}}$ are few enough for their estimation to formulate a well-posed inverse problem; and

\mathbf{X} expresses the means through which the model-generated counterparts to the measurement dataset \mathbf{h} are calculated from the reduced parameter set $\underline{\mathbf{p}}$; it is related to the model matrix \mathbf{Z} through the equation:

$$\mathbf{X} = \mathbf{Z} \mathbf{L} \quad (4.6d)$$

Substitution of (4.1) into (4.5) leads to the equation:

$$\underline{\mathbf{k}} = \mathbf{G} \mathbf{Z} \mathbf{k} + \mathbf{G} \boldsymbol{\varepsilon} \quad (4.7a)$$

That is:

$$\underline{\mathbf{k}} = \mathbf{R} \mathbf{k} + \mathbf{G} \boldsymbol{\varepsilon} \quad (4.7b)$$

where:

$$\mathbf{R} = \mathbf{G} \mathbf{Z} \quad (4.8)$$

\mathbf{R} is the well-known “resolution matrix”. Where measurement noise is zero it expresses the relationship between estimated parameters $\underline{\mathbf{k}}$ and their real world counterparts \mathbf{k} (which are never known). Where regularization is implemented through truncated SVD, \mathbf{R} is a projection operator as indicated in Figure 4.2. Irrespective of the regularization methodology employed, each row of the matrix \mathbf{R} specifies the averaging relationship through which each element of $\underline{\mathbf{k}}$ is derived from the entirety of elements of \mathbf{k} . Thus it defines the parameter simplification

process that was required for achievement of a unique solution to the inverse problem of model calibration.

The difference between estimated parameters and their real world counterparts is parameter error. From (4.7) this can be formulated as:

$$\underline{\mathbf{k}} - \mathbf{k} = -(\mathbf{I} - \mathbf{R})\mathbf{k} + \mathbf{G}\boldsymbol{\varepsilon} \quad (4.9)$$

The first term on the right side of (4.9) is the “cost of uniqueness” discussed extensively by Moore and Doherty (2005; 2006). It is the contribution made to parameter error arising out of the fact that the calibration process can estimate only a simplified form of reality. The second term of equation (4.9) arises from the fact that the estimated parameter set is calculated from a calibration dataset which is contaminated by noise. Where regularisation is implemented using truncated SVD, the first and second terms of equation (4.9) lie within the null and solution spaces respectively, and are orthogonal to each other. This is illustrated in Figure 4.4.

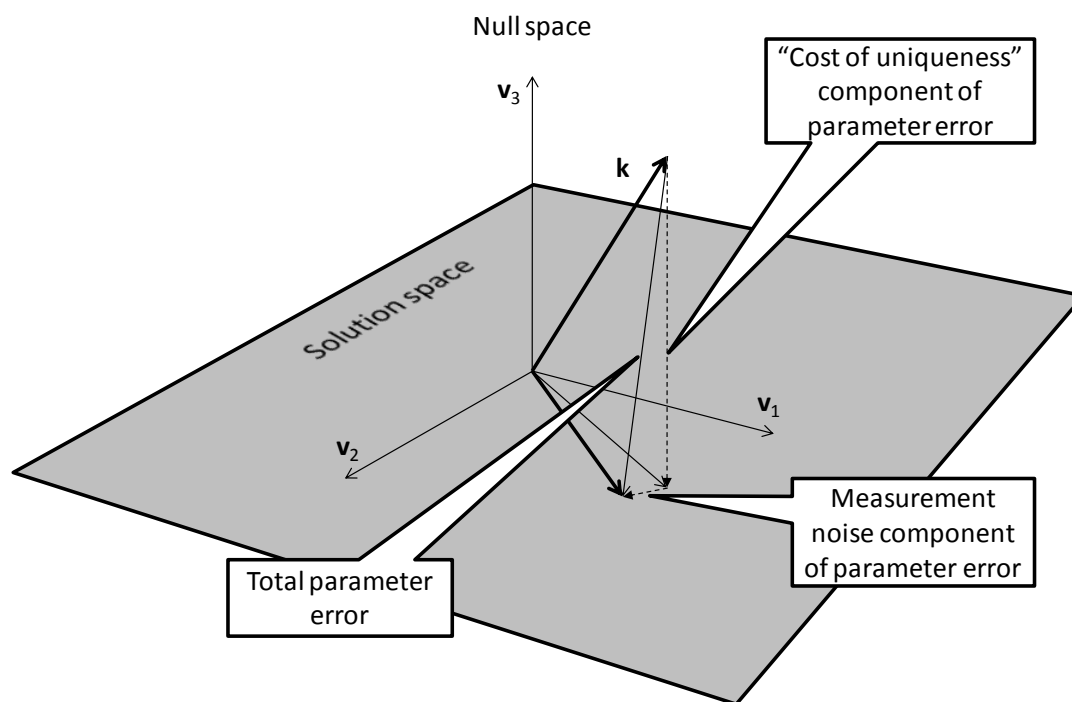


Figure 4.4. The two components of parameter error expressed by the two terms on the right side of equation (4.9).

Unfortunately parameter error cannot be calculated, for neither the real parameters \mathbf{k} nor noise $\boldsymbol{\varepsilon}$ associated with the measurement dataset are known. However, using equation (3.4), the covariance matrix of parameter error can be calculated from the stochastic characterization of pre-calibration parameter variability $C(\mathbf{k})$ and from that of measurement noise $C(\boldsymbol{\varepsilon})$ as:

$$C(\underline{\mathbf{k}} - \mathbf{k}) = (\mathbf{I} - \mathbf{R})C(\mathbf{k})(\mathbf{I} - \mathbf{R})^t + \mathbf{G}C(\boldsymbol{\varepsilon})\mathbf{G}^t \quad (4.10)$$

Equation (4.10) acquires a particularly simple but instructive form when the following conditions are met:

- Regularisation is implemented using truncated SVD;
- $C(\mathbf{k})$ can be expressed as:

$$C(\mathbf{k}) = \sigma_k^2 \mathbf{I} \quad (4.11)$$

where \mathbf{I} is the identity matrix; and

- $C(\boldsymbol{\varepsilon})$ can be expressed as:

$$C(\boldsymbol{\varepsilon}) = \sigma_{\varepsilon}^2 \mathbf{I} \quad (4.12)$$

In this case (4.10) becomes:

$$C(\underline{\mathbf{k}} - \mathbf{k}) = \sigma_k^2 \mathbf{V}_2 \mathbf{V}_2^t + \sigma_{\varepsilon}^2 \mathbf{V}_1 \mathbf{S}^{-2} \mathbf{V}_1^t \quad (4.13)$$

where the columns of \mathbf{V}_2 contain orthogonal unit vectors which span the calibration null space and the columns of \mathbf{V}_1 contain orthogonal unit vectors which span the calibration solution space; both of these are obtained through partitioning of the \mathbf{V} matrix of equation (4.4). As the number of dimensions of the calibration solution space grows, and the null space therefore shrinks, the first contributor to parameter error variance (i.e. the “cost of uniqueness” term) falls, as less and less simplification is being undertaken in an attempt to calibrate the model. However the second term of equation (4.13) rises; furthermore it rises very fast as the magnitudes of singular values fall, until ultimately it becomes infinity as singular values fall to zero. Total parameter error variance therefore falls and then rises as simplification is at first too great, and at last too little (and over-fitting occurs). Minimized total parameter error variance occurs at an intermediate number of singular values; so too does minimized predictive error variance, as will be discussed in the next chapter.

Parameter and predictive error cannot be known. If they could be known, an appropriate correction term could be applied. Thanks to equations (4.10) and (4.13) however, the *propensity* for parameter error (and, as we will see shortly, for predictive error as well) *can* be known. Furthermore, to the extent that $C(\mathbf{k})$ and $C(\boldsymbol{\varepsilon})$ are known or can be surmised, optimality of solution of the inverse problem of model calibration can be achieved through minimizing the propensity for parameter and predictive error.

We finish this subsection with a nice property of singular value decomposition; this property will be demonstrated using one of the exercises provided later in this chapter.

From equations (4.1) and (4.4) with measurement noise ignored:

$$\mathbf{h} = \mathbf{U} \mathbf{S} \mathbf{V}^t \mathbf{k} \quad (4.14a)$$

After removal of zero-valued singular values, this becomes:

$$\mathbf{h} = \mathbf{U} \mathbf{S}_1 \mathbf{V}_1^t \mathbf{k} \quad (4.14b)$$

\mathbf{S}_1 is, by definition, a diagonal matrix with non-zero elements. Hence it has an inverse. From (4.14b) and the fact that, because \mathbf{U} is an orthogonal unit matrix:

$$\mathbf{U}^t \mathbf{U} = \mathbf{I} \quad (4.15)$$

It follows that:

$$\mathbf{S}_1^{-1} \mathbf{U}^t \mathbf{h} = \mathbf{V}_1^t \mathbf{k} \quad (4.16)$$

This formula states that certain linear combinations of parameters are solely and uniquely informed by certain, partnered, linear combination of observations. The number of such partnerships is equal to the number of diagonal elements of \mathbf{S}_1 and hence the number of singular values employed in the inversion process. The sequence of informative linear combinations of observations (which capture the entire information content of the calibration dataset) form an orthogonal set of axes in observation space. The corresponding sequence of informed linear combinations of parameters (which comprise a complete set of receptacles for the information content of the calibration dataset) form an orthogonal set of axes in parameter space; these axes span the calibration solution subspace. The former are given by:

$$\mathbf{u}_1\mathbf{u}_1^t, \mathbf{u}_2\mathbf{u}_2^t, \mathbf{u}_3\mathbf{u}_3^t, \text{ etc} \quad (4.17a)$$

where \mathbf{u}_i is the i 'th column of the \mathbf{U} matrix. The latter are given by:

$$\mathbf{v}_1\mathbf{v}_1^t, \mathbf{v}_2\mathbf{v}_2^t, \mathbf{v}_3\mathbf{v}_3^t \text{ etc} \quad (4.17b)$$

where \mathbf{v}_i is the i 'th column of the \mathbf{V} matrix. The former are referred to as “super observations” in PEST parlance, whereas the latter are referred to as “super parameters”.

Structural Noise

Unfortunately, models are not perfect simulators of system behaviour. This may manifest itself in an inability to achieve a good fit between model outcomes and field measurements during the calibration process. Alternatively, or as well, it may result in parameters being assigned incorrect values as partial compensation for unrepresented processes, this allowing the model to fit the historical data well in spite of the fact that some of the environmental processes that gave rise to that data are not simulated by the model. In addition to this, structural defects may compromise the model's ability to make a desired prediction with integrity, as not all processes on which the prediction depends are represented in the model. If this prediction is of a different type from data comprising the calibration dataset, the model's inadequacies in this regard will not have been detected during the calibration process.

Doherty and Welter (2010) attempt to introduce some rigor to the manner in which structural noise is accommodated in the model calibration and predictive processes. They begin their analysis by stating that equation (4.1) can be used as the basis for analysis of calibration optimality and predictive uncertainty only if the model (represented by the \mathbf{Z} matrix in that equation) is a perfect simulator of environmental reality. In fact, the action of the model is better described by the following equation:

$$\mathbf{h} = \mathbf{Z}_1\mathbf{k}_1 + \mathbf{Z}_2\mathbf{k}_2 + \boldsymbol{\varepsilon} \quad (4.18)$$

where \mathbf{Z}_1 and \mathbf{k}_1 represent the model and parameters used by the model respectively, while \mathbf{k}_2 represents corrections to the model that would allow it to simulate reality perfectly while \mathbf{Z}_2 represents the sensitivity of model outputs to these corrections. Both of \mathbf{Z}_2 and \mathbf{k}_2 are unknown.

Doherty and Welter (2010) draw the following conclusions through mathematical analyses based on this equation.

- Model structural inadequacies can express themselves through model-to-measurement misfit under calibration conditions that can greatly exceed that which arises from measurement noise. However treating that misfit as an additive term to measurement noise (this being commonly referred to as “structural noise”) for the purpose of determining an optimal level of misfit to achieve during the calibration process, and in calculating the penchant for parameter error that arises from that misfit, is fraught with conceptual difficulties. This follows from the fact that the covariance matrix of “structural noise” is unknown, and probably singular.
- Unless the singular nature of the covariance matrix of structural noise is taken into account, post-calibration propensity for parameter error may be seriously underestimated. If historical measurements of system state were contaminated only by measurement noise of non-singular covariance matrix, the potential for error associated with parameter value estimates should decrease as the size of the calibration dataset increases (in accordance with Bayes equation). This is not the case, however, where the model-generated counterparts to field measurements are contaminated with noise of structural origin.

- Calibration strategies can be devised which mitigate the deleterious effects of model structural defects on estimation of at least some parameters. In particular, it is often possible to formulate a multi-component objective function in which different objective function components inform different combinations of parameters. Through ensuring equal visibility of each of these components in the overall objective function, the damage inflicted by the presence of structural noise on at least some parameter estimates can be greatly reduced. Strategies for formulation of a suitable multi-component objective function include the following:
 - use of inter-layer head differences and temporal head differences when calibrating a groundwater model;
 - use of log-transformed flows, together with baseflow-filtered flows, as well as event-based or monthly volumes, when calibrating a surface water model.
- When a defective model is calibrated against a real-world dataset, it is almost certain that some parameters will be assigned values that compensate for model structural defects. This may or may not increase the error variance of model predictions. For predictions that resemble observations used in the calibration process, parameter compensation may reduce the propensity for model predictive error. For predictions that differ in type and location from observations used in the calibration process, parameter compensation may increase the propensity for model predictive error. In either case, the link between parameter optimality and predictive optimality is broken.
- The fact that some estimated parameters can “soak up” the “information that has no other place to go” that is structural noise may, or may not, benefit the calibration process. As stated above, this depends in part on the extent to which predictions required of a model resemble observations used in the calibration process. To the extent that “parameter surrogacy” is judged to be advantageous to the making of some predictions, it may be worthwhile ensuring that observations that most resemble predictions of interest are well fit during the calibration process by endowing them with weights that are great enough to allow this to happen. This raises the spectre of “prediction-specific calibration”.

Doherty and Welter (2010) demonstrate that the presence of model structural defects requires that a modeller make many informed, but necessarily subjective, decisions during both the calibration and predictive phases of model deployment. Their subjective nature arises from the fact that model structural defects, the magnitude and characteristics of structural noise that they incur, and the degree to which parameters may assume advantageous and disadvantageous compensatory roles during their estimation as an outcome of these defects, are all unknown.

It will often be possible to obtain a very good fit between model outputs and field data when calibrating a flawed model. But when is “a good fit” in fact “too good a fit”? “Over-fitting” is often recognized as such when some awkward values are estimated for some parameters. However these values may have been incorrect long before they were recognized as being unrealistic, and hence long before recognition of a model’s “over-fit” status required that the calibration process be repeated with higher levels of parameter simplification introduced in order to prevent such a good fit between model outcomes and field measurements from being obtained again. In many modelling circumstances however (as stated above) the surrogate role that some parameters play may actually benefit the predictive process, this depending on the predictions required of the model. In this case, sacrificing goodness of fit in order to

ensure that no parameter plays any surrogate role whatsoever may leave much important information that is contained in the calibration dataset untapped. Unfortunately, the best path to choose in any given calibration context is often unclear, for there are no universal rules pertaining to this situation (and little expert guidance available). All that a modeller can do is to exercise informed creativity, taking full account of whatever theoretical help is available (including aspects of modelling theory that account for model structural defects). Nevertheless decisions made by a modeller will necessarily be subjective, and will almost certainly vary from modeller to modeller.

Exercises

Surface Water Model

TSPROC

The TSPROC utility was written specifically to expedite the use of PEST in surface water model calibration. TSPROC stands for “time series processor”. Its functionality includes the ability to:

- read model output files written in a number of different formats;
- read observation data files provided in a number of different formats;
- undertake temporal interpolation from model outputs to the times of field measurements so that the two can be directly compared;
- compute statistics pertaining to observed and simulated time series;
- undertake digital filtering of observed and modelled time series;
- accumulated total mass/volume over arbitrary time periods for modelled and observed time series;
- undertake arbitrary mathematical manipulation of modelled and observed time series;
- generate a PEST input dataset in which any or all of the above can feature as components of a multi-component objective function.

As a time series processor, TSPROC is run after the simulator as part of a composite model calibrated by PEST. Commands to run the simulator followed by TSPROC are placed within a batch or script file. As a PEST input dataset constructor TSPROC is run by the user. Full documentation of TSPROC is provided in the manual to the PEST Surface Water Modelling Utility suite.

Calibration Against Flows

A TSPROC input file named *tsproc_calib1.dat* has been provided. The contents of this file can be inspected using any text editor. In accordance with TSPROC protocol, processing instructions within this file are contained within a series of blocks. The processing sequence is as follows.

- Modelled and observed flows are read from files *output.wdm* and *observ.wdm* respectively. Note that flows are read over the one year calibration time span comprising 1985, as well as over the following year.
- Observed flows are converted to units of m^3/sec as they are stored in units of ft^3/sec .

- Maximum observed and simulated flows over the prediction interval (19th to 23rd August 1986) are calculated. These are not used in the calibration process. However they are written to the PEST input dataset with a weight of zero for easy monitoring.
- The time spans of observed and simulated flows stored within TSPROC's memory are reduced to that of the calibration period.
- The logarithms (to base 10) of all simulated and observed flows are computed; a small positive flow of 0.75 m³/sec is added before logarithmic transformation to guard against the presence of any zero flows (of which the log is the negative of infinity).
- The logarithms of simulated flows are written to a file named *model.out*. So too is the prediction (i.e. the maximum flow over 19th to 23rd August 1986).
- A PEST input dataset is generated in which the logs of simulated flows are matched against the logs of observed flows. All weights are 1.0.

As recorded in the `WRITE_PEST_FILES` block, the model batch file is named *model_calib1.bat*; this must be prepared by the user. An inspection of this file reveals that it includes the commands to run HSPF followed by TSPROC. The latter program writes data that PEST reads; the instruction file through which the TSPROC output file is read is generated by TSPROC itself as it implements the directives contained in the `WRITE_PEST_FILES` block. See the manual for the Surface Water Utilities suite for further details.

Run TSPROC, responding to its prompts as follows. (Also respond with “y” if/when asked if it is alright to over-write any existing files.)

```
Enter name of TSPROC input file: tsproc_calib1.dat
Enter name for TSPROC run record file: nul
```

TSPROC echoes to the screen the directives that it reads from its input file. When it has implemented all directives, it records successful completion of execution to the screen. In complying with the directives provided in *tsproc_calib1.dat* it writes a PEST control file named *calib1.pst*, as well as an instruction file named *tsproc_calib1.ins* to read the TSPROC-generated “model output file” *tsproc_calib1.out*. You can check the integrity of the PEST input dataset generated by TSPROC using the following command. (Ignore PESTCHEK's warnings; it is only trying to be helpful.)

```
pestchek calib1.pst
```

Note that TSPROC instructs PEST to use truncated singular value decomposition as a solution mechanism for the inverse problem of model calibration. This is not strictly necessary in the present case as the inverse problem is well posed. However it provides a safeguard against the possible onset of ill-posedness as parameters change their values. (By the way, it is not hard to show that singular value decomposition provides an identical parameter estimation outcome as that provided by the traditional Gauss-Marquardt-Levenberg method when an inverse problem is well posed; the two inversion methodologies are in fact equivalent under these conditions.)

Now run PEST, using the following command.

```
pest calib1
```

From an inspection of the results section of PEST's run record file *calib1.rec* the following is apparent.

- PEST reduces the objective function from an initial value of 14.556 to a final value of 8.042.
- This requires 229 model runs.
- The highest flow over the predictive period, calculated on the basis of optimized parameters, is 412 m³/sec. This compares very poorly with the observed highest flow over this period of 27.24 m³/sec.

The logs of modelled and observed flows are depicted in Figure 4.5a, while native modelled and observed flows are compared in Figure 4.5b. The goodness of fit during 1985 (the calibration period) and the poorness of fit during 1986, are apparent from both of these plots (note that clipping occurs at a flow of 100m³/sec in the latter plot).

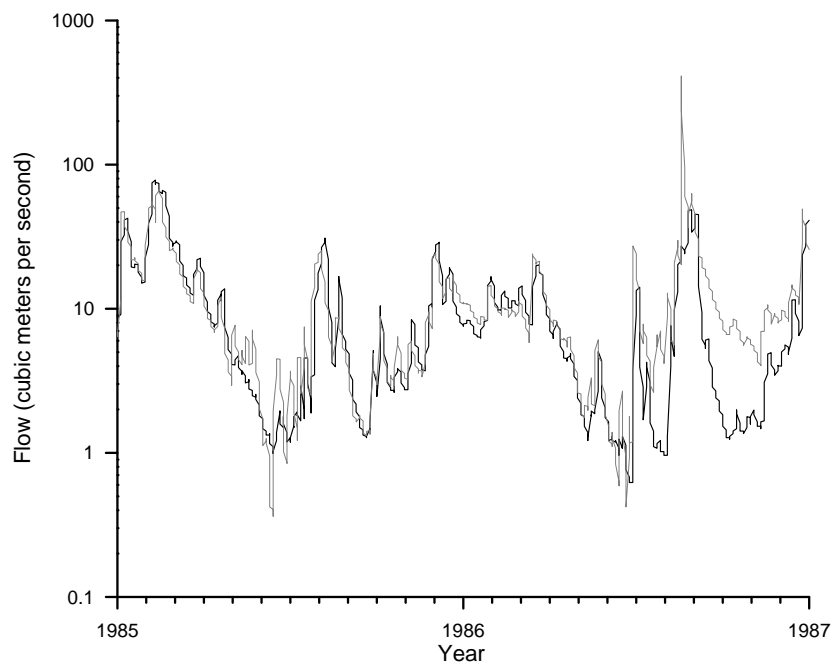


Figure 4.5a. Log of observed (dark line) and modelled (grey line) flows. Modelled flows were calculated using optimised parameters from the *calib1* calibration exercise.

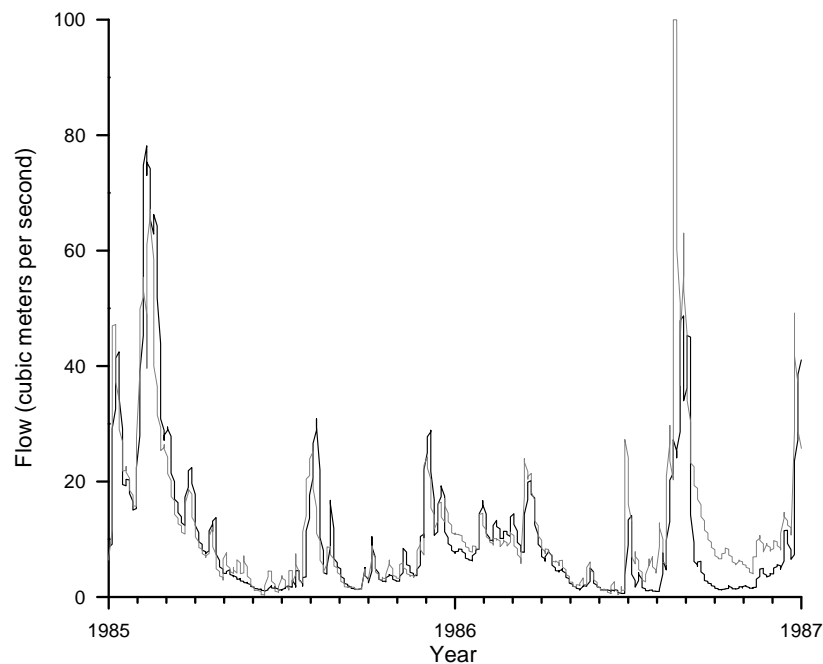


Figure 4.5b. Observed (dark line) and modelled (grey line) flows. Modelled flows were calculated using optimised parameters from the *calib1* calibration exercise. Note that clipping occurs at 100 m³/sec; the simulated peak flow on 20th August 1986 is actually 412 m³/sec.

Optimized parameter values are recorded in the run record file *calib1.rec*. It is apparent that two parameters (namely IRCTRANS and INTFW) are at their bounds. Optimised parameters are also recorded in “parameter value file” format in file *calib1.par*. It is often convenient to build a new PEST control file in which optimized parameters feature as initial parameters. This is easily done using the PARREP utility (which we have already encountered). Run PARREP using the command:

```
parrep calib1.par calib1.pst calib1_soln.pst
```

Now alter the NOPTMAX variable in *calib1_soln.pst* (first variable on the 9th line of this file) to 0. If PEST execution is now initiated using the command:

```
pest calib1_soln
```

PEST will run the model once on the basis of optimized parameters, compute the objective function, and then cease execution.

Multi-Component Objective Function

The TSPROC input file *tsproc_calib2.dat* is a little more complex than *tsproc_calib1.dat*. In addition to directing TSPROC to calculate the logs of observed and simulated flows, it also directs TSPROC to calculate the following quantities over the 1985 calibration period:

- an approximation to baseflow using the digital filtering technique of Nathan and McMahon (1990);
- accumulated monthly volumes;
- exceedence times for various flow thresholds.

In addition to this it directs TSPROC to build a PEST input dataset in which all of these processed flow components are featured. To build this dataset, run TSPROC by typing its

name at the screen prompt. Respond to its prompts as follows (overwriting files as necessary if prompted for permission).

```
Enter name of TSPROC input file: tsproc_calib2.dat
Enter name for TSPROC run record file: nul
```

TSPROC builds a PEST control file named *calib2.pst*. In this file, items that contribute to different objective function components are assigned to different observation groups; see the “observation groups” and “observation data” sections of *calib2.pst*. The “model command line” section of *calib2.pst* directs PEST to run a batch file named *model_calib2.bat* as the model. In this batch file (prepared by the user) the command to run TSPROC using file *tsproc_calib2.dat* as its input file follows the command to run HSPF.

Set NOPTMAX (first variable on line 9 of file *calib2.pst*) to zero. Next check that the PEST input dataset is correct, consistent and complete by running PESTCHEK using the command:

```
pestchek calib2
```

Ignore warnings, and then run PEST using the command:

```
pest calib2
```

PEST runs the model once, calculates the objective function pertaining to each observation group together with the total objective function. Then it ceases execution. Notice that the contribution to the total objective function by the prediction is zero, this following from the fact that this prediction has been given zero weight.

As Doherty and Welter (2010) explain, where a multi-component objective function is formulated in order to promulgate visibility of different aspects of flow data in the overall objective function, it is important to ensure that none of these different flow components either dominate the inverse problem solution process, or are dominated by other flow components. This can be achieved by ensuring that they all contribute roughly the same amount to the initial objective function. An inspection of PEST screen output for case *calib2* reveals that this is not happening at present. Hence inter-group weights adjustment is required. This is most easily achieved using the PWTADJ1 (for “PEST weight addjustment strategy 1”) utility. Run PWTADJ1 as follows.

```
pwtadj1 calib2 calib2a 1000
```

PWTADJ1 writes a new PEST control file in which weighting is such that each observation group contributes exactly 1000 to the overall initial objective function. Now calibrate the model by running PEST using the command:

```
pest calib2a
```

PEST only requires 100 model runs to calibrate the model this time. As before, use the PARREP utility to create a new PEST control file in which initial parameter values are optimized parameter values. Use the command:

```
parrep calib2a.par calib2a.pst calib2_soln.pst
```

to run PARREP. Set NOPTMAX to 0 in the newly-created *calib2_soln.pst* PEST control file and run PEST on the basis of this file to produce a set of model output files based on optimized parameters. Model outputs together with their observed counterparts are shown in Figures 4.6a-d.

Unfortunately, at 405.6 m³/sec the predicted maximum flow over the period 19th to 23rd August 1986 is still very much too high.

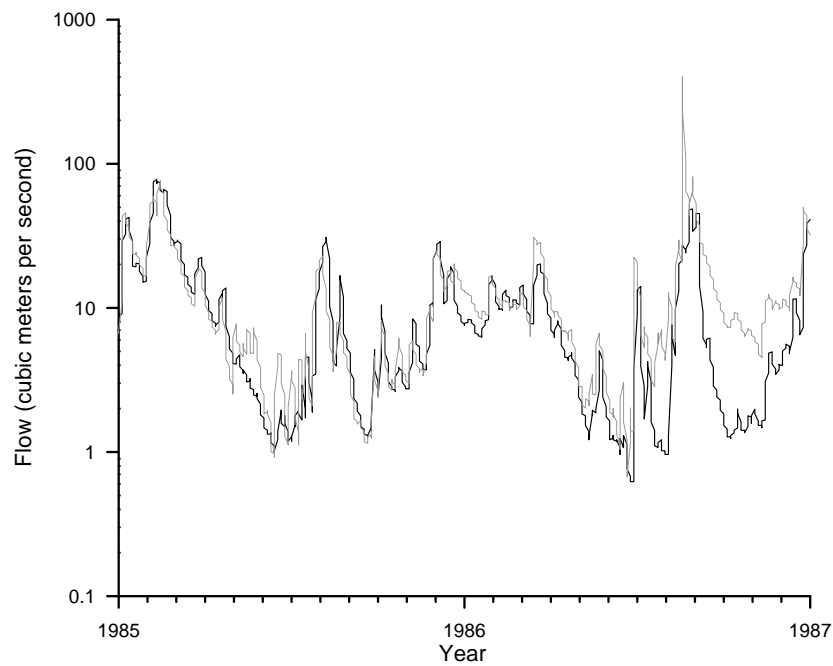


Figure 4.6a. Log of observed (dark line) and modelled (grey line) flows. Modelled flows were calculated using parameters achieved through the *calib2* calibration exercise.

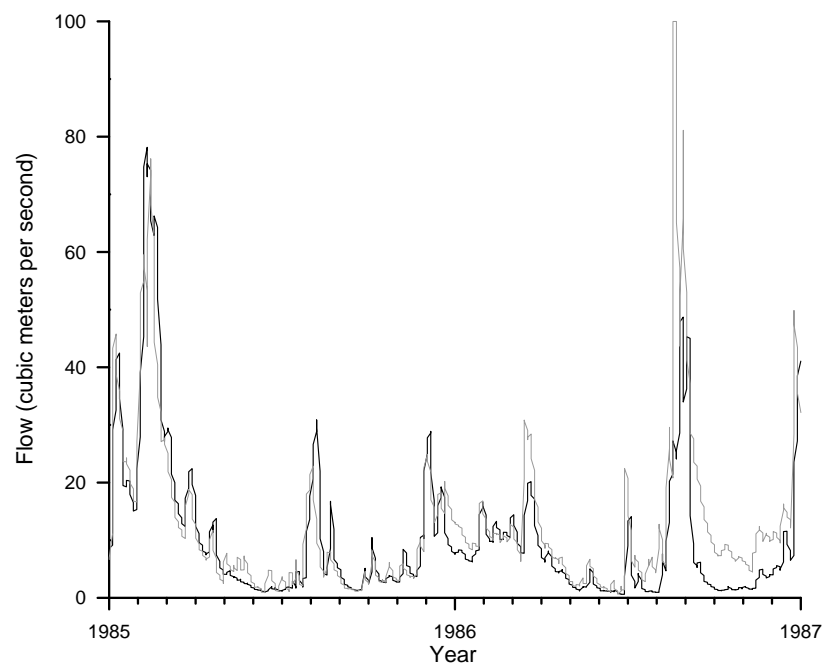


Figure 4.6b. Observed (dark line) and modelled (grey line) flows. Modelled flows were calculated using optimised parameters from the *calib2* calibration exercise. Note that clipping occurs at 100 m³/sec; the simulated peak flow on 20th August 1986 is actually 405.6 m³/sec.

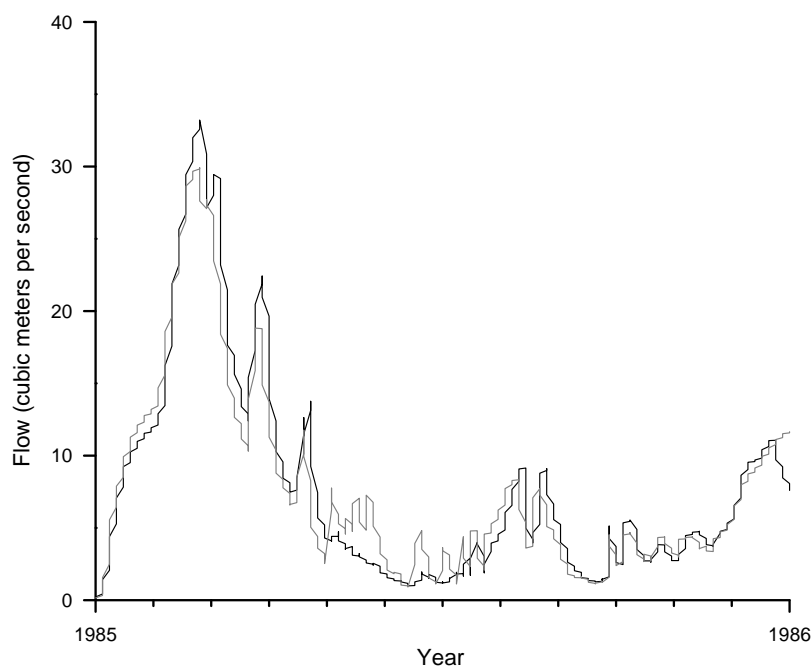


Figure 4.6c Observed (dark) and modelled baseflow-filtered flows over calibration period. Modelled flows were calculated using parameters achieved through the *calib2* calibration exercise.

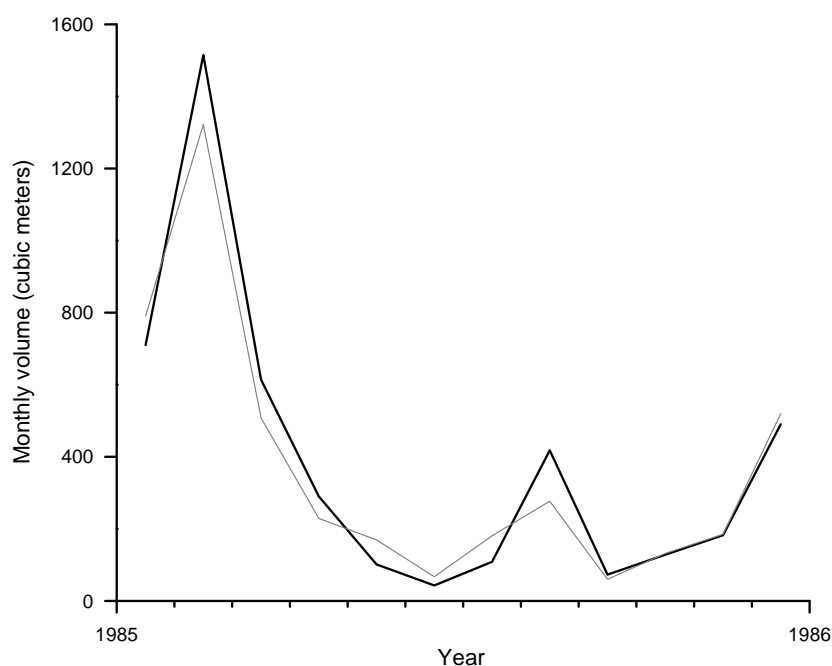


Figure 4.6d Observed (dark) and modelled (grey) monthly volumes over calibration period. Modelled volumes were calculated using parameters achieved through the *calib2* calibration exercise.

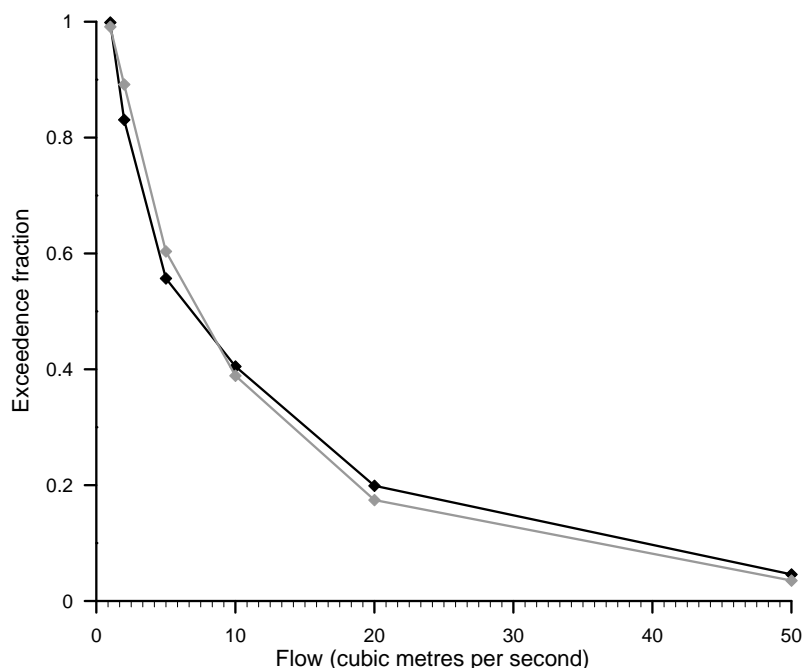


Figure 4.6e Observed (dark) and modelled (grey) exceedence fractions for various flow thresholds over the calibration period. Modelled exceedence fractions were calculated using parameters achieved through the *calib2* calibration exercise.

Super Observations

We will now examine in more detail the flow time series that spans the one year calibration period. We will inquire into where the information content of this time series lies, and to what aspects of model parameterization this information pertains. This will be done through use of the super observation and super parameter concepts referred to in the preceding discussion.

First we must obtain a Jacobian matrix corresponding to the calibrated parameter set. Open file *calib1_soln.pst* and set NOPTMAX (first variable on the ninth line of this file) to -1. With NOPTMAX set to -1 PEST is instructed to fill the Jacobian matrix, and to calculate some statistics based on the Jacobian matrix, and then cease execution.

Run PEST using the command:

```
pest calib1_soln
```

to initiate these calculations.

The SUPOBSPREP utility builds a new PEST control file on the basis of an existing PEST control file. In this new PEST control file super observations replace existing observations. (SUPOBSPREP stands for “super observations input dataset preparation”.) In the present instance we do not actually want a new PEST control file (though we will obtain it anyway). What really interests us are pictures of the super observations and of the super parameters that they inform.

Run SUPOBSPREP, responding to its prompts as follows.

```
Enter name of existing PEST control file: calib1_soln.pst
Enter number of super observations to build from this file: 3

Enter clipping-enforced pre-compression weights range (<Enter> if 1E6): <Enter>
Enable compression/expansion of super observation weights? (y/n): n
Enter minimum super-observation weight: 1

Enter name for new super pest control file: temp.pst
```

```
Enter name for super observation matrix file (<Enter> if none): superobs.mat
```

```
Enter name for super parameter matrix file (<Enter> if none): superpar.mat
```

The two SUPOBSPREP-generated files that interest us the most are *superobs.mat* and *superpar.mat*. These are recorded in PEST matrix file format; hence they are amenable to further processing by other PEST utilities.

As was explained in the theory section of this chapter, super observations and super parameters are actually normalized orthogonal linear combinations of observations on the one hand and of parameters on the other hand. Each such linear combination of observations is entirely and uniquely informative of the corresponding combination of parameters; hence each super observation is uniquely linked to each super parameter. The magnitude of a normalized vector is unity; this must be born in mind when inspecting it. Each component of each super observation pertains to an actual observation. Hence, in the present case, the components of each super observation can be plotted against time so that the composition of each super observation may be more clearly seen. This is done in Figure 4.7a for the first three super observations; the modelled flow over the calibration period is also plotted so that the total flow can be compared with its partitioning into super observations. The composition of the corresponding first 3 super parameters is plotted in Figure 4.7b. Recall that each super observation contains the entirety of the information pertaining to its corresponding super parameter. As for super observations, super parameters are also normalized so that their magnitude is 1.0. The information used to make these plots was obtained from files *superobs.mat* and *superpar.mat*.

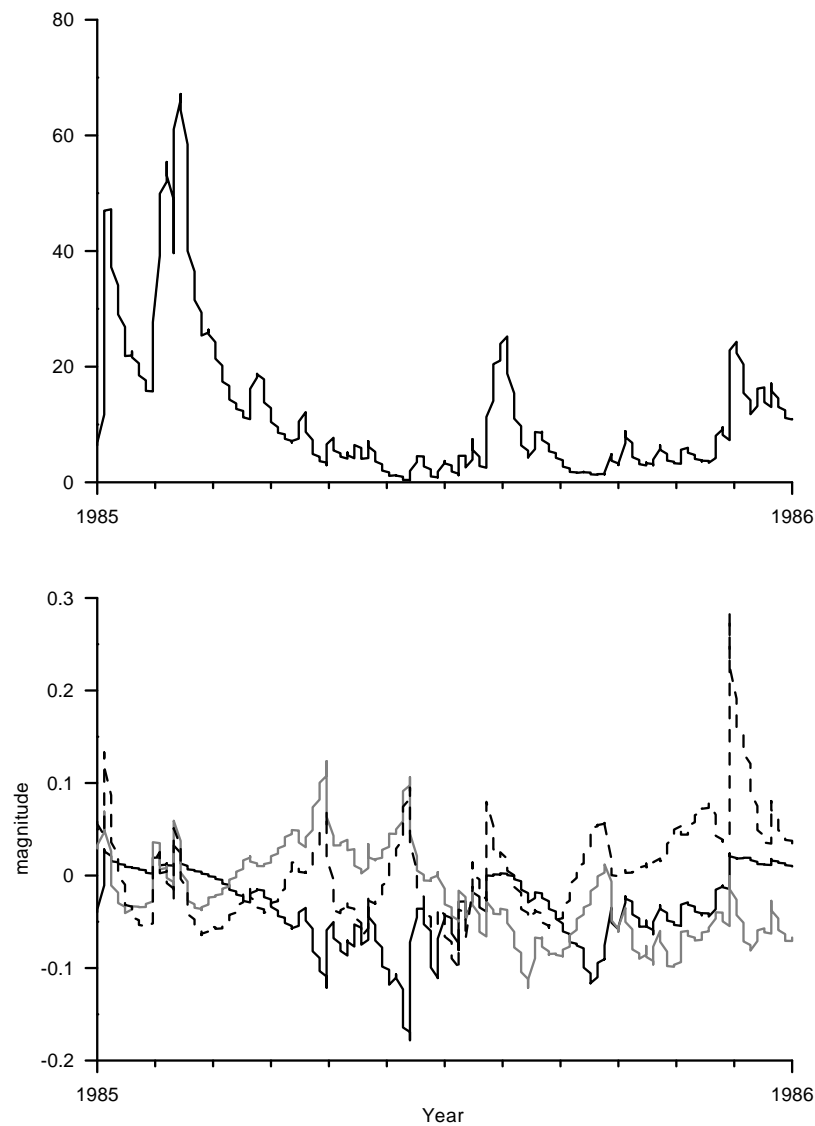


Figure 4.7a. The upper part of the figure shows total modelled flow over the calibration period. The lower part of the figure shows the decomposition of this flow into its first three super observations. The first super observation is plotted as a dark line, the second as a grey line and the third as a dashed line.

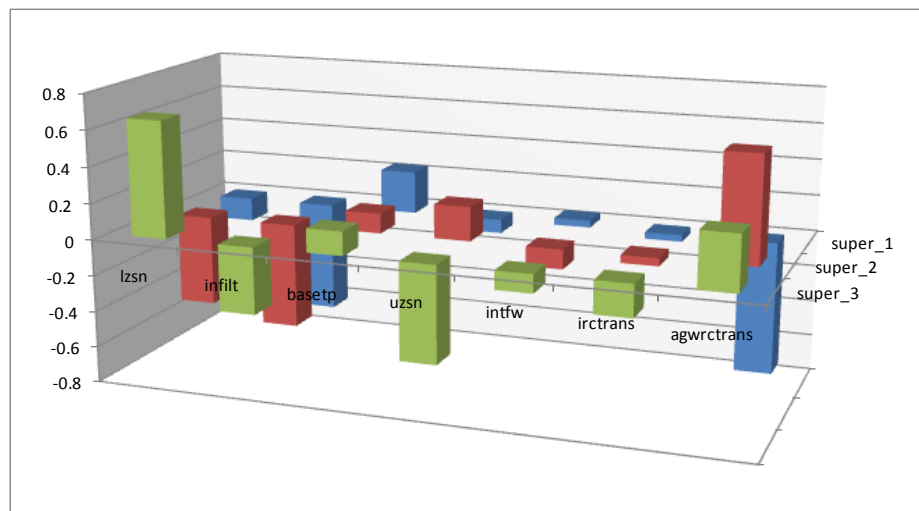


Figure 4.7b. Compositions of the first three super parameters.

Groundwater Model

The Model Batch File

The batch file *model_calib1* is reproduced as Figure 4.8. This is the file that PEST will run as “the model” as it adjusts hydraulic conductivity values assigned to pilot point parameters in order to obtain a good fit between modelled and observed heads at observation wells.

```
@echo off
Rem #####
Rem Some intermediate files are deleted.
Rem #####

del hk.ref > nul
del heads.dat > nul

Rem #####
Rem Now the actual model is run
Rem #####

fac2real < fac2real.in > nul
mf2k < mf2k.in > nul
finaltime < finaltime.in > nul
mod2obs < mod2obs.in > nul
```

Figure 4.8 The model batch file *model_calib1.bat*.

As was discussed in Chapter 1 of this document, hydraulic conductivity parameterization for our synthetic groundwater model takes place on the basis of 104 pilot points. Their locations are provided in Figure 1.4. PEST’s task in calibrating the model is to estimate values for the hydraulic conductivities that are associated with each of these 104 points. These conductivity values are listed in the final column of a file named *hk.pts*. The FAC2REAL utility (a member of the PEST Groundwater Data Utility suite) undertakes spatial interpolation from these points to the MODFLOW finite difference grid, writing the hydraulic conductivity real array file *hk.ref* in the process. (FAC2REAL stands for “interpolation factors to real array”.) It will be recalled that in the previous exercise this same real array file was written by the FIELDGEN random field utility. In contrast to FIELDGEN however, the use of pilot points

as a parameterization device does not allow representation of cell-by-cell heterogeneity, as hydraulic property variability is only allowed on a pilot-point-by-pilot-point basis.

MODFLOW calculates the piezometric head at every active cell of the finite-difference grid. Spatial interpolation from MODFLOW-calculated heads arrays to the sites at which head measurements were actually taken is implemented using the MOD2OBS utility. (MOD2OBS stands for “model to observations”; like FAC2REAL it is also a member of the Groundwater Data Utility suite.) MOD2OBS reads the MODFLOW-generated binary heads file *heads.dat* and interpolates to wells whose coordinates are supplied in file *wells.crd*.

The following points are also noteworthy.

- FAC2REAL implements spatial interpolation using kriging. In the present case kriging factors are housed in a file named *factors.dat* which FAC2REAL reads in order to access these factors. *factors.dat* was written by another Groundwater Data Utility program named PPK2FAC. (PPK2FAC stands for pilot points to kriging factors”.)
- As is described in documentation to the PEST Groundwater Data Utilities suite, MOD2OBS not only undertakes spatial interpolation. Where a model is transient it also undertakes temporal interpolation to the times at which measurements were actually taken. Its task is to read a MODFLOW/MT3D/SEAWAT head or concentration file and to calculate the model-generated counterparts to an historical head or concentration dataset. In the present case of course, our “field data” are synthetic. They are housed in a file named *observ.smp*. In accordance with user instructions provided in the re-directed keyboard input file *mod2obs.in*, MOD2OBS reads *observ.smp* shortly after its execution is initiated.
- Many other programs that facilitate PEST usage with MODFLOW and other models are provided with the Groundwater Data Utility suite. Some of these facilitate automatic PEST input dataset construction. Others assist in formulation of a multicomponent objective function. Different objective function components can include heads, concentrations, budget terms, head and concentration ratios and differences, and other raw or processed data types.

The PEST Control File

A PEST control file named *calib1.pst* has been provided. This file was written by the PESTPREP1 utility from the Groundwater Data Utility suite. (PESTPREP1 stands for PEST input dataset preparation.) PESTPREP1 also wrote the instruction file *model_smp.ins* that is cited in *calib1.pst*. The instructions contained in *model_smp.ins* enable PEST to read the MOD2OBS output file *model.smp*.

Inspection of file *calib1.pst* reveals the following.

- PEST is asked to solve the inverse problem of model calibration using singular value decomposition. Truncation occurs at a singular value that guarantees that the square of the ratio of the lowest to highest singular value of the weighted Jacobian matrix that is retained in the inversion process is greater than 5×10^{-7} . See the EIGHTHRESH variable which is placed at the end of the second line of data within the “singular value decomposition” section of this file.
- The initial value of the Marquardt lambda (i.e. the RLAMBDA1 variable) is set to 10.0. The Marquardt lambda adjustment factor RLAMFAC is set to -3.0. The latter setting allows rapid adjustment of the Marquardt lambda during the parameter

estimation process. RLAMBDA1 and RLAMFAC occupy the first and third positions on the sixth line of the PEST control file.

- Although not part of the calibration process, two predictions of interest, namely the *part_time* and *part_east* predictions (these being the particle travel time to the southern model boundary and the easting of its point of emergence there) are included in the “observation data” section of the PEST control file. Inclusion of these quantities in the PEST input dataset allows us to monitor them easily as parameter values change. As will be seen in the next chapter, inclusion of these predictions as “observations” (whose effect on the calibration process is nullified by endowing them with weights of zero) also allows us to easily obtain the sensitivities of these predictions to parameter values; this will be useful when carrying out linear uncertainty analysis.
- Weights on all head observations are set to 10.0. This is the inverse of 0.1 m, the latter being the standard deviation of noise which is presumed to be associated with each head measurement.

Tikhonov regularization will now be added to the PEST control file. This is most easily accomplished using the ADDREG1 utility. (ADDREG1 stands for “add regularisation”.) ADDREG1 writes a prior information equation for each parameter. This equation is recorded in the “prior information” section of the PEST control file; ADDREG1 creates this section if it is not already present. In these equations the value of each parameter is matched to its initial value. (It is, of course, assumed that user-supplied initial values are pre-calibration preferred parameter values.) A weight of 1.0 is assigned to each new prior information equation.

Run ADDREG1 as follows.

```
addreg1 calib1.pst calib1r.pst
```

An inspection of the new PEST control file *calib1r.pst* reveals that it includes both a “prior information” section and a “regularisation” section. The first line of the “regularisation” section is as follows:

```
1.000000E-10    1.050000E-10    0.1000000
```

These are the regularisation control variables PHIMLIM, PHIMACCEPT and FRACPHIM. The first of these defines the target measurement objective function; the second should be set slightly above that. In the present case the expected value of each of the 12 residuals featured in the calibration process should be about 0.1 as this is the level of noise associated with each head measurement. (A “residual” is the difference between an observation value and its model-generated counterpart.) With a weight of 10 supplied for each of these observations, the expected value of each weighted residual is thus 1.0, so that the expected value of the measurement objective function is about 12. So alter the first line of the “regularisation” section of the PEST control file *calib1r.pst* so that it is as follows.

```
12.0    12.1    0.1000000
```

Parameter Covariance Matrix

As was discussed earlier in this document, the “reality” hydraulic conductivity field is characterized by a variogram whose properties are recorded in the “structure file” *struct.dat*. If we know something about the nature of parameter field variability, the Tikhonov regularization process can be improved through use of this knowledge. Before running PEST to calibrate the model, we will make this improvement. We will do the following.

- First we will build a covariance matrix of innate parameter variability (i.e. a $C(\mathbf{k})$ matrix) using the variogram specifications in file *struct.dat*.
- Then we will inform PEST that it must use this matrix when constraining pilot point parameter values to respect their preferred values. Thus if PEST must introduce heterogeneity to the pilot point parameter field in order to allow the model to replicate the heads measured in observation wells, this heterogeneity should arise in ways that are most geologically reasonable.

The PPCOV Groundwater Data Utility (PPCOV stands for pilot points covariance matrix) can be used to build a covariance matrix for our pilot point parameters. Run this utility by typing its name at the screen prompt. Respond to PPCOV's prompts as follows.

```
Enter name of pilot points file: hk.pts
- data for 104 pilot points read from pilot points file hk.pts

Enter minimum allowable separation for points in same zone: 0

Enter name of structure file: struct.dat

Enter structure to use for pilot point zone 1: structure1

Enter name for output matrix file: hk_cov.mat
Enter pilot point prefix for parameter name (<Enter> if none): k_
```

The covariance matrix recorded in file *hk_cov.mat* respects PEST matrix file format. To instruct PEST to use this matrix in conjunction with the prior information that ADDREG1 provided for pilot point parameters, alter the “observation groups” section of the PEST control file *calib1r.pst* so that it appears as below. The text which you must add to this section is shown bold and highlighted.

```
* observation groups
heads
time
distance
regul_k hk_cov.mat
```

Now check your work with PESTCHEK by typing the command:

```
PESTCHEK calib1r
```

and then run PEST using the command:

```
PEST calib1r
```

PEST ceases execution after completing 322 model runs (don't worry if it is a little different on your machine); an inspection of screen output or the run record file *calib1r.rec* reveals that the desired measurement objective function of 12.0 has been obtained (or a number very close to it). The calibrated parameter field is depicted in Figure 4.9.

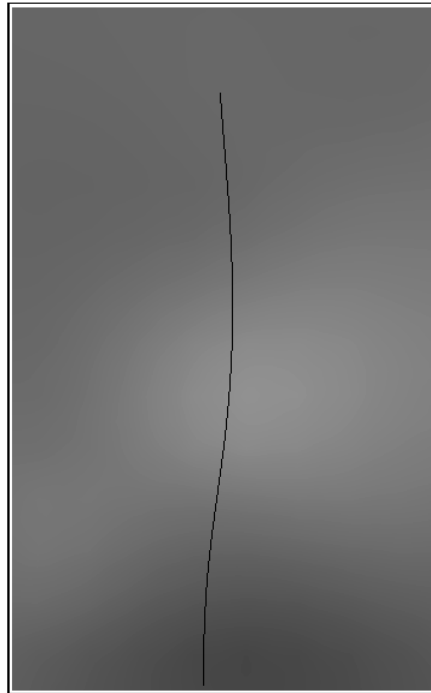


Figure 4.9 Calibrated parameter field. Also shown is the particle path calculated using the calibrated parameter field. The particle travel time is 6864 days.

As usual, a PEST control file which features optimised parameters as initial parameters can be built using the PARREP utility. Run PARREP as follows.

```
parrep calibr.par calibr.pst calibr_soln.pst
```

If you wish, set NOPTMAX to 0 in this file and then run PEST, thereby undertaking one model run on the basis of optimized parameters in order to obtain model outputs calculated on the basis of these optimised parameters. An inspection of file *calibr_soln.rec* (the run record file associated with this PEST control file) or *calibr_soln.res* (the residuals file) will reveal that the calibrated model predicts a particle travel time of about 6864 days. Given that the real travel time is actually 3256 days, this prediction is terribly wrong. However this should not constitute grounds for condemnation of the model. As has already been discussed, and as will be discussed further in the next chapter, there is no mathematical reason why a calibrated model *should* provide the correct answer. This is especially so in the present case for, as will be seen, the information content of head measurements provides very few constraints on parameter combinations which determine particle movement. Recall that a model is only worthy of condemnation if the uncertainty intervals calculated using the model fail to *encompass* reality; an ability to predict reality is a property that should be expected of no model.

Dimensions of Solution Space - the SUPCALC Utility

Model calibration using the highly-efficient SVD-assist methodology will now be demonstrated. But first the PEST SUPCALC utility will be employed to suggest a dimensionality for the solution space. (SUPCALC stands for “number of super parameters calculation”.) When undertaking SVD-assisted parameter estimation, only as many super parameters need be estimated as there are dimensions in the solution space. (In practice more than this should be estimated if resources are available as protection against model

nonlinearity, with estimation of super parameters implemented using truncated SVD to maintain numerical stability.)

SUPCALC uses a similar equation to (4.10) to estimate optimal solution space dimensionality. It obtains a $C(\mathbf{k})$ matrix from the user, or optionally estimates $C(\mathbf{k})$ itself from user-supplied parameter bounds. In the present case we will supply a $C(\mathbf{k})$ matrix ourselves. It computes $C(\epsilon)$ from observation weights, in conjunction with the expected value of the measurement objective function. In doing so it uses the fact that the latter should approximately equal the number of non-zero-weighted observations if the weight associated with each observation is supplied as the inverse of the standard deviation of noise associated with that observation. SUPCALC calculates the pre- and post-calibration error variance of each new super parameter as the dimensionality of the solution space is increased. Pre-calibration error variance is computed on the basis of $C(\mathbf{k})$ alone using a formula similar to (3.5). The calibration solution space is deemed to be spanned only by those columns of the \mathbf{V} matrix of equation (4.4) for which the post-calibration error variance of corresponding super parameters is smaller than their pre-calibration variance. SUPCALC also calculates an upper bound for solution space dimensionality as that which leads to a highest-to-lowest squared singular value ratio of about 5×10^7 .

PEST provides the user with many means of supplying the $C(\mathbf{k})$ matrix. In the present case the covariance matrix of pre-calibration parameter uncertainty has already been calculated using the PPCOV utility; recall that it is housed in file *hk_cov.mat*. In other modelling contexts there may be many types of parameters employed in the calibration process, each with a covariance matrix of its own. These can be collectively furnished to PEST through a “parameter uncertainty file”. File *param.unc* is a simple example of such a file. As there is only one parameter type featured in the present calibration process, it cites only one covariance matrix. In practice it may cite many of these. A parameter uncertainty file also allows the pre-calibration uncertainty of one or many parameters to be defined simply through provision of their individual standard deviations.

Run SUPCALC as follows.

```
Enter name of PEST control file: caliblr.pst
Enter expected value of measurement objective function: 12

To conduct SVD on  $Q^{(1/2)}X$  - enter 1
To conduct SVD on  $XtQX$  - enter 2
Enter your choice: 1

Use uncertainty file or bounds to specify parameter variability? [u/b]: u
Enter name of parameter uncertainty file: param.unc

Enter name for eigenvector gain/loss output file: temp.dat
```

SUPCALC records a recommended usage of 12 super parameters to the screen.

SVD-Assist

The SVDAPREP utility is used to build a PEST input dataset based on super parameters. Run SVDAPREP, responding to its prompts in the manner shown below; most of the responses shown below are in fact acceptances of SVDAPREP defaults.

```
Enter name of existing PEST control file: caliblr.pst
Use pre-defined super-parameter file? [y/n] (<Enter> if "no"): <Enter>
For computation of super parameters:-
  if SVD on  $Q^{(1/2)}X$  - enter 1
  if SVD on  $XtQX$  - enter 2
  if LSQR without orthogonalisation - enter 3
```

```

    if LSQR with orthogonalisation      - enter 4
Enter your choice (<Enter> if 1): 1
Enter number of super parameters to estimate: 12
Enter name for new super pest control file: calib1r_svda.pst
Enter offset for super parameters (<Enter> if 10): <Enter>
Enter value for RELPARMAX (<Enter> if 0.1): <Enter>
Write multiple BPA, JCO, REI, none [b/j/r/n] files (<Enter> if "n")? <Enter>
Parameter scale adjustment [SVDA_SCALADJ] setting (<Enter> if 2): <Enter>
Make new model batch file silent or verbose? [s/v] (<Enter> if "s") : <Enter>
Automatic calc. of 1st itn. super param. derivs? [y/n] (<Enter> if "y") : <Enter>

```

SVDAPREP writes a new PEST control file named *calib1r_svda.pst* based on super parameters. Model calibration can now be undertaken using the command:

```
pest calib1r_svda
```

Monitor PEST's screen output as it runs. It is apparent that the target measurement objective function is achieved in only 35 model runs!

A before, the PARREP utility will be used to store optimized parameter values in a suitable PEST control file for safe-keeping. Run PARREP as follows.

```
parrep calib1r.bpa calib1r.pst calib1r_soln_svda.pst
```

Notice that, even though PEST estimates super-parameters when undertaking SVD-assisted inversion, it still records optimized values of base parameters. These are recorded in parameter value file format in a file with an extension of ".bpa". Now set NOPTMAX to zero in file *calib1r_soln_svda.pst* and run PEST to see the value of the travel time prediction. It is 6889 days. The calibrated parameter field and particle track are very similar to those presented in Figure 4.9.

See PEST documentation and other PEST workshops for a more complete description of PEST calibration techniques.

Parameter Identifiability

We will conclude this chapter by computing the identifiability of each of the 104 pilot point parameters involved in this groundwater model calibration exercise. Because the inverse problem of model calibration is ill-posed the dimensionality of the solution space is less than that of total parameter space. In fact the solution space has 12 dimensions, whereas parameter space has 104 dimensions. It follows that many, if not all, parameters will have an identifiability of less than 1.0; some may have identifiabilities approaching zero.

Before computing identifiability we will compute a new Jacobian matrix for the present problem using optimized parameter values. Recall that these have been recorded as initial parameter values in the PEST control file *calib1r_soln.pst*. (Of course, another similar set has been recorded in file *calib1r_soln_svda.pst*. However we will use the former set; the choice is quite arbitrary really.) Open this file with a text editor, changing NOPTMAX (first variable on the 9th line) from 0 to -2. This NOPTMAX setting instructs PEST to terminate execution as soon as it has completed calculation of the Jacobian matrix based on initial parameter values.

Once NOPTMAX has been changed to -2, run PEST using the command:

```
pest calib1r_soln
```

The new Jacobian matrix is stored in binary format in file *calib1r_soln.jco*. If desired, it can be re-written in ASCII (i.e. text) format using either of the JACWRIT or JCO2MAT utilities. Sensitivities with respect to parameters of any observations (including the two dummy

part_time and *part_east* observations which are actually predictions) can be extracted from *calibr_soln.jco* using the JROW2VEC utility.

To compute parameter identifiability, run the IDENTPAR utility as follows. (IDENTPAR stands for “parameter identifiability”.)

```
identpar calibr_soln 12 null null ident.dat
```

File *ident.dat* lists the direction cosine of the projection of each parameter onto each of the columns of the \mathbf{V} matrix which span the calibration solution space; recall that these columns are unit orthogonal vectors which are the axes of parameter solution space. In its second-to-last column *ident.dat* also lists total parameter identifiability (which is the direction cosine between each parameter and its projection onto the entire solution space).

Identifiability can be displayed in a number of different ways. In Figure 4.10a the identifiabilities of some parameters are displayed as a bar chart.

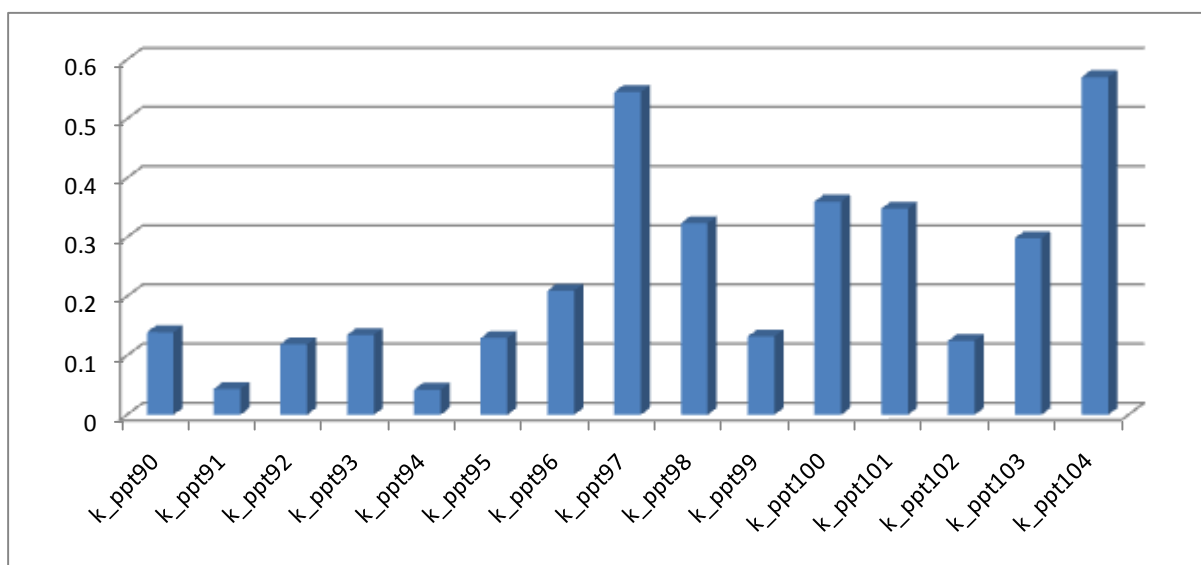


Figure 4.10a. Identifiabilities of some hydraulic conductivity pilot point parameters. Parameter names are derived from pilot point names through use of a “k_” prefix. Pilot point names and locations are provided in file *hk.pts* and plotted in Figure 4.10b.

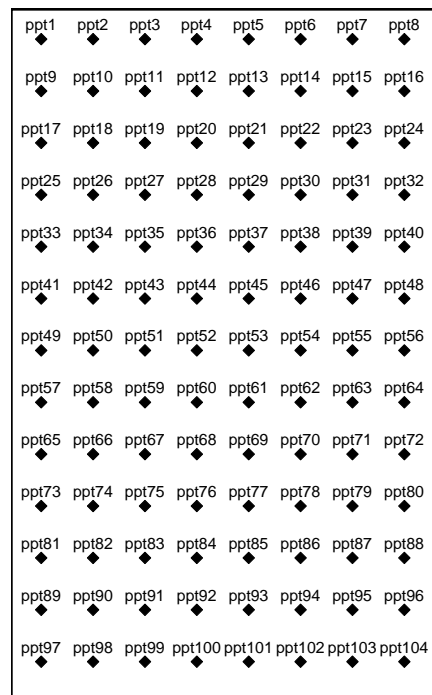


Figure 4.10b. Names and locations of pilot points.

Because pilot point parameters have a spatial setting, it is sometimes instructive to plot their identifiabilities in space. With a little cutting and pasting (pilot point coordinates are available in file *hk.pts*) followed by importation into SURFER for contouring, the following plot is easily produced. Note the enhanced parameter identifiability between measurement wells in the direction of groundwater flow, and between the most southern row of wells and the southern fixed head model boundary.

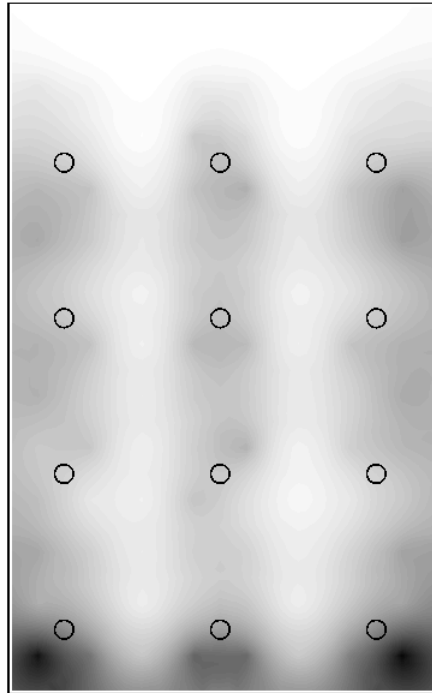


Figure 4.11 Identifiability as a function of position throughout the model domain. Darker coloration indicates greater identifiability, with a maximum of 0.58 occurring at the southern corners of the domain. Identifiability is reduced to almost zero north of the most northern row of measurement wells.

5. How Wrong Can a Prediction Be? Linear Analysis

Error and Uncertainty

Parameter and predictive uncertainty following imposition of constraints on parameter values imposed by the necessity for a model to reproduce historical system behaviour, is described by Bayes equation. It is repeated here (in a slightly different form from that presented in equation 2.3) for completeness.

$$P(\mathbf{k} | \mathbf{h}) = \frac{P(\mathbf{h} | \mathbf{k})P(\mathbf{k})}{\int P(\mathbf{h} | \mathbf{k})P(\mathbf{k})d\mathbf{k}} \quad (5.1)$$

As before, the vector \mathbf{h} represents measurements of system state (referred to herein as the “calibration dataset”) while the vector \mathbf{k} represents parameters. The symbol $P()$ represents probability. $P(\mathbf{k})$ represents the prior probability of parameters \mathbf{k} while $P(\mathbf{k}|\mathbf{h})$ is the posterior probability of parameters \mathbf{k} ; the latter is the probability of parameters \mathbf{k} conditional upon information encapsulated in \mathbf{h} . Meanwhile $P(\mathbf{h}|\mathbf{k})$ represents the likelihood function, this increasing with the extent to which model-to-measurement misfit (defined in a way that accounts for the stochastic properties of measurement noise) decreases. The denominator of the right hand side of equation (5.1) is required for normalization so that the area under the posterior probability distribution is 1.0. In implementation of Bayes equation, the model is employed for calculation of the likelihood term.

In everyday modelling practice direct use of Bayes equation is difficult. This arises from the numerical difficulties involved in handling probability distributions, especially where they do not have analytical formulations. Unfortunately, even if a prior probability distribution can be given an analytical formulation (such as multi-uniform or multi-normal) the nonlinear behaviour of most models will ensure that the posterior parameter probability distribution does not have an analytical formulation.

Another problem that arises in assessing the uncertainty that is associated with the environmental future is that models are imperfect simulators of environmental system behaviour. An immediate repercussion of this is that a model’s calculation of the likelihood function is likely to be in error. This is inescapable in most calibration contexts, so that most model-to-measurement misfit on most occasions arises from model imperfections. To a limited extent the nature of misfit engendered by model imperfections can be assessed through the history-matching process and assimilated into Bayesian characterization of posterior parameter probabilities. However the effects of model imperfections on predictions of interest cannot be assessed in this way, unless those predictions are of identical character, and occur under identical conditions, to those which prevailed when measurements comprising the calibration dataset were made. Unfortunately, this facet of model structural error therefore affects predictive uncertainty in a manner that constitutes an often unknown term that must be added to any model prediction, and thus requires that the probability distribution of that prediction as assessed through Bayesian analysis be modified in an unknown way.

Because of the numerical difficulties involved in working with probability distributions, particularly when these must be assessed using complex models with large run times, everyday model usage normally involves a two step process of calibration followed by parameter and predictive uncertainty analysis. Ideally, as has already been discussed, the calibration process should yield parameters that are the expected values (in the statistical

sense) of real-world parameters. Predictions should be the same. Pursuit of this goal normally requires a highly parameterized approach to inversion accompanied by the use of mathematical regularisation to achieve uniqueness of the calibrated parameter field. However there are numerical and computational limits on the number of parameters which can be included in the calibration process. Hence, in practice, even where calibration is implemented using regularized inversion, many of a model's parameters will be, of necessity, lumped to at least some extent while others will be fixed by the user at reasonable values and remain unadjusted through the calibration process. In proportion to the extent to which reality remains unlumped, and that fixed parameters are set at incorrect values by the user, a model acquires defects. These defects are in addition to those which arise from the imperfect nature of the model as a simulator of real-world environmental processes. Ultimately, model defects induce errors in parameters estimated through the calibration process, and in predictions made by the model. Errors in the latter arise both from their dependencies on error-prone parameters, and from model defects as they directly affect these predictions.

In light of the above, it is probably better to use the term “error” rather than “uncertainty” when discussing environmental modelling. Parameters estimated through the calibration process are in error by virtue of the fact that their estimation necessarily involves regularisation (i.e. simplification). The calibration process must then strive to achieve estimates of parameters which are of minimum error variance. As was discussed in the preceding chapter, where calibration is undertaken through the agency of highly parameterized inversion, the resolution matrix that describes the simplification that underpins the quest for parameter uniqueness is available as a by-product of the regularized inversion process. To some extent then, post calibration parameter error variance can be calculated, and indeed minimized. The same is not true where regularization is undertaken manually; an inability to calculate and minimize the potential for error arising from the necessity to simplify in order to achieve uniqueness is one of the principle drawbacks of its use.

In practice the situation is more difficult than this. The imperfect nature of models engenders structural noise. This can add to the propensity for parameter error by:

- decreasing the dimensionality of the solution space at which parameter error variance is minimized, thereby increasing the first term on the right of equation 4.10;
- increasing the model-to-measurement misfit contribution to the potential for parameter error described by the second term on the right of equation 4.10.

To make matters worse, the increased propensity for parameter error incurred by model structural defects is virtually impossible to quantify, though as Doherty and Welter (2010) point out, steps can be taken to reduce it through intelligent formulation of the calibration objective function. However what is at once disturbing and comforting is that this increased penchant for parameter error may or may not result in an increased propensity for predictive error (in fact the opposite may be the case), this depending entirely on the nature of the prediction and its dependencies on model parameters.

Where predictions are similar in character to observations employed in the calibration process, and where the calibration process allows parameters to adopt values that compensate for model imperfections as they affect its ability to simulate those aspects of environmental behaviour that are recorded in measurements comprising the calibration dataset, the accrual of parameter error through the calibration process can be entirely beneficial to the making of those types of predictions. In other cases, however, particularly those where predictions are of a different character to observations comprising the calibration dataset, the accrual of parameter error in this way should be assiduously avoided in pursuit of the goal of

minimization of predictive error variance. However this variance will probably be inflated beyond its theoretical minimum as the prediction must then be made by a model whose parameters are less constrained by the calibration dataset than they would otherwise be, this being born of the necessity to eschew too low a level of calibration misfit in order to avoid any hint of parameter surrogacy. Total predictive error will, of course, include another term, this being that which describes the model's imperfections in relation to this prediction. Neither this term, nor its stochastic distribution, will in general be known in contexts where a prediction is different in character from members of the calibration dataset, for the model's integrity (or lack thereof) in making predictions of this type will not have been explored through the calibration process.

This rather bleak picture does not provide sufficient grounds for giving up on attempts to evaluate model predictive uncertainty. However, it does suggest the following.

- As stated above, it is probably better to employ the concept of error rather than uncertainty when working in the context of environmental models. Despite the fact that the two terms are often used interchangeably (including in the present document), the distinction should be born in mind.
- The potential for error in a prediction made by an environmental model will always be greater than its inherent uncertainty. The latter is notionally calculable through implementing Bayesian analysis in conjunction with a perfect model. As real-world models are imperfect, their predictions have a potential for error that is greater than the inherent uncertainty of those predictions, this arising out of the necessity to make predictions, and to analyse their potential for wrongness, using an imperfect model.
- Uncertain parameters lead to uncertain predictions. However errors in parameters that compensate for imperfections in a model's simulation capabilities may or may not increase the potential for error in predictions made by that same model. Some models are explicitly designed for parameter compensation to work in a prediction's favour, this normally requiring that predictions of interest resemble observations that comprise the dataset through which that model is calibrated. Other models are not designed in this way, being specified as "physically based". However no "physically based" model is perfect, and hence at least some degree of parameter compensation is unavoidable. This may or may not be a bad thing, as it may be possible to "tune" a model to make good predictions of one type (while unavoidably de-tuning its ability to predictions of other types). This raises the spectre of prediction-specific calibration.
- It will rarely, if ever, be possible to quantify either predictive uncertainty or predictive error variance with a high level of precision. Thus it will rarely, if ever, be possible to make statements such as "these thresholds mark the 95% confidence interval of this prediction" or "there is only a 5% chance that such an event will occur" in the environmental modelling context. Assessment of model predictive uncertainty/error necessarily involves a high degree of subjectivity.

The Predictive Error Term

As even a complex, physically-based model is an imperfect simulator of environmental reality, any prediction made by such a model will contain a component of error that reflects its imperfections. To some extent it may be possible to "calibrate out" at least some model imperfections in some circumstances. As already discussed, with care this may be legitimate where predictions of interest resemble observations used in the calibration process. Alternatively, if model-to-measurement misfit cannot legitimately be reduced in this manner

through allowing some parameters to “soak up” structural noise, the resulting level of model-to-measurement misfit provides quantification of the penchant for error in model predictions that are similar in nature to model outputs used in the calibration process. However in cases where a prediction does not resemble model outputs used in the calibration process it is not possible to quantify the contribution that model imperfections make to predictive error. In this case, recognition of this contribution to model predictive error can only take the form of a judiciously-chosen “engineering safety margin” added to other components of model predictive error potential that can be at least partially quantified.

It is beyond the scope of this document to discuss the means through which a structurally-induced model predictive error correction term should be formulated in different modelling contexts. Indeed, it is the author’s opinion that this is a problem that is yet to be properly addressed and is therefore deserving of research. For the moment however, it is recommended that this predictive structural error term be treated (notionally if not actually) as a parameter. There will be cases where this parameter is directly estimable through the calibration process. In other cases it will be subjectively inferable through studying calibration misfit as it pertains to different observation types at different locations. In still other cases its size will be purely a matter of educated guesswork.

As a “parameter” (and an uncertain one at that) the error term which must be added to a model prediction has an initial and “estimated” value of zero, this being an outcome of the fact that it does not receive information in a formal way through the calibration process, and hence lies entirely in the null space. This parameter can be included in the existing set of parameters comprising the vector \mathbf{k} . Its uncertainty (as inferred through the calibration process or assessed subjectively) is then included in the overall $\mathbf{C}(\mathbf{k})$ matrix of parameter uncertainty along with the other parameters on which a model prediction is dependent.

If the above strategy is followed for accommodation of model predictive structural noise, no further modifications to any of the linear analysis methodologies presented in the present chapter, or to any of the nonlinear analysis methodologies presented in the following chapter are then required. The “model” as it applies to the prediction is simply enhanced with this extra parameter - the value of this parameter being added to the model prediction with which it is associated.

In setting up PEST input files for linear and nonlinear analysis, the predictive structural noise parameter can be included in the PEST control file as a parameter which no component of the model actually reads when the model is run under calibration conditions. In this way its uncertainty is propagated through to predictive uncertainty, unaffected by the calibration process. The parameter is only used under predictive conditions as an additive term to a prediction. Different parameters of this type can be used with different predictions. In certain applications they may be specified as showing statistical correlation with each other, or even with physically-based parameters, this being done through assignment of appropriate values to pertinent elements of the $\mathbf{C}(\mathbf{k})$ matrix of innate parameter variability supplied by the user.

Linear Parameter and Predictive Uncertainty Analysis

Parameter Uncertainty

An uncertain variable is characterized by a probability distribution. The width of a probability distribution can be characterized by its variance, this being the square of its standard deviation. An uncertain vector represents a multiplicity of individual uncertain variables. The uncertainty of a vector is characterized by a multi-component probability distribution. The width of this distribution in any direction of parameter space, and the degree to which random

elements of the vector exhibit statistical inter-relatedness can be characterized by an $m \times m$ covariance matrix, where m is the dimension of the vector (i.e. the number of elements contained in the vector). To date in this document, covariance matrices $C(\mathbf{k})$ and $C(\boldsymbol{\varepsilon})$ have characterized the prior probability distribution of the vector of parameters \mathbf{k} , and the probability distribution of the vector $\boldsymbol{\varepsilon}$ of noise associated with measurements comprising the calibration dataset \mathbf{h} respectively.

Let $C'(\mathbf{k})$ denote the covariance matrix that is associated with the posterior parameter probability distribution. This is the probability distribution that appears on the left side of Bayes equation, sometimes referred to herein as the “post-calibration” parameter probability distribution. Suppose also that the following conditions are met:

- Model outputs are linear with respect to parameters so that the action of the model on its parameters can be represented by a matrix (which we denote as \mathbf{Z}):
- The prior parameter probability distribution is multiGaussian;
- Measurement noise is also characterized by a multiGaussian distribution.

It can be shown that under these circumstances $C'(\mathbf{k})$ can be calculated using either of the following two formulas (which are mathematically equivalent).

$$C'(\mathbf{k}) = [\mathbf{Z}^t C^{-1}(\boldsymbol{\varepsilon}) \mathbf{Z} + C^{-1}(\mathbf{k})]^{-1} \quad (5.2a)$$

$$C'(\mathbf{k}) = C(\mathbf{k}) - C(\mathbf{k}) \mathbf{Z}^t [\mathbf{Z} C(\mathbf{k}) \mathbf{Z}^t + C(\boldsymbol{\varepsilon})]^{-1} \mathbf{Z} C(\mathbf{k}) \quad (5.2b)$$

Use of the first of these two formulas is more computationally efficient when the number of observations exceeds the number of parameters, whereas use of the second is more efficient when the opposite is the case. This follows from the dimensionality of the matrix which must be inverted in each case.

Predictive Uncertainty

Let s be a model prediction of interest. Let the elements of the vector \mathbf{y} denote the sensitivities of this prediction to the elements of the parameter vector \mathbf{k} . As was previously expressed as equation (3.5), the prior variance of predictive uncertainty (i.e. the square of the standard deviation of uncertainty associated with the prediction s) is given by:

$$\sigma_s^2 = \mathbf{y}^t C(\mathbf{k}) \mathbf{y} \quad (5.3a)$$

Obviously the posterior variance of predictive uncertainty σ_s^2 is expressed as:

$$\sigma_s^2 = \mathbf{y}^t C'(\mathbf{k}) \mathbf{y} \quad (5.3b)$$

From (5.2) it then follows that:

$$\sigma_s^2 = \mathbf{y}^t [\mathbf{Z}^t C^{-1}(\boldsymbol{\varepsilon}) \mathbf{Z} + C^{-1}(\mathbf{k})]^{-1} \mathbf{y} \quad (5.4a)$$

$$\sigma_s^2 = \mathbf{y}^t C(\mathbf{k}) \mathbf{y} - \mathbf{y}^t C(\mathbf{k}) \mathbf{Z}^t [\mathbf{Z} C(\mathbf{k}) \mathbf{Z}^t + C(\boldsymbol{\varepsilon})]^{-1} \mathbf{Z} C(\mathbf{k}) \mathbf{y} \quad (5.4b)$$

These two equations are equivalent; however (5.4b) is particularly illustrative. The first term on the right of this equation is the prior uncertainty variance. The second term is the amount by which this term is reduced through the history-matching process. The matrix that appears between \mathbf{y}^t and \mathbf{y} in this second term can be shown to be positive semidefinite. Hence history-matching can never lead to an increase in the uncertainty of a prediction. At worst it can lead

to zero reduction in prior predictive uncertainty; at best it can lead to a considerable reduction in this uncertainty. It all depends on the information content of the calibration dataset with respect to parameters to which the prediction is sensitive. (Note that, as we shall see, the same does not apply to post-calibration predictive *error*; it is indeed possible for predictive error to be higher after calibration than before calibration.)

Equation (5.4) can be used to calculate the post-calibration uncertainty of an individual parameter if desired. In this case the \mathbf{y} vector is comprised of zero-valued elements, except for the element pertaining to the parameter whose uncertainty is desired. This element is assigned a value of 1.0.

Linear Parameter and Predictive Error Analysis

Parameter Error

Equation (4.10) depicts the covariance matrix of post-calibration parameter error. It is repeated below as equation (5.5).

$$\mathbf{C}(\underline{\mathbf{k}} - \mathbf{k}) = (\mathbf{I} - \mathbf{R})\mathbf{C}(\mathbf{k})(\mathbf{I} - \mathbf{R})^t + \mathbf{G}\mathbf{C}(\boldsymbol{\varepsilon})\mathbf{G}^t \quad (5.5)$$

Recall that \mathbf{G} is the matrix which describes the means through which the calibrated parameters set $\underline{\mathbf{k}}$ is computed from the observation dataset \mathbf{h} while \mathbf{R} is the resolution matrix. Both of these are ultimately functions of the model matrix \mathbf{Z} . Hence equations (5.5) and (5.2) involve the same components; they are just manipulated in different ways.

Predictive Error

As before, let a prediction s be calculable from model parameters \mathbf{k} through the linear relationship:

$$s = \mathbf{y}^t \mathbf{k} \quad (5.6a)$$

The prediction calculated using the calibrated model is given by:

$$\underline{s} = \mathbf{y}^t \underline{\mathbf{k}} \quad (5.6b)$$

where $\underline{\mathbf{k}}$ is the calibrated parameter set. Predictive error is given by the difference between these two quantities. That is:

$$\underline{s} - s = \mathbf{y}^t (\underline{\mathbf{k}} - \mathbf{k}) \quad (5.7)$$

Applying equation (3.4) to equation (5.5), predictive error variance is obtained as:

$$\sigma_{\underline{s}-s}^2 = \mathbf{y}^t (\mathbf{I} - \mathbf{R})\mathbf{C}(\mathbf{k})(\mathbf{I} - \mathbf{R})^t \mathbf{y} + \mathbf{y}^t \mathbf{G}\mathbf{C}(\boldsymbol{\varepsilon})\mathbf{G}^t \mathbf{y} \quad (5.8)$$

Pre-calibration predictive error variance can be formulated as a special case of equation (5.8). In this case both \mathbf{G} and \mathbf{R} are zero so that equation (5.8) becomes equation (5.3a). Thus pre-calibration predictive error variance is the same as the pre-calibration variance of predictive uncertainty.

Where a model is a perfect simulator of reality, it can be shown that post-calibration predictive error variance as calculated using equation (5.8) is always greater than post-calibration predictive uncertainty variance, except in the special case where:

- calibration is undertaken using Tikhonov regularization;
- Tikhonov prior information equations specify that the preferred value of each parameter is equal to its pre-calibration value of minimum error variance;

- the weight matrix applied to Tikhonov prior information constraints is equal to $C^{-1}(\mathbf{k})$, where $C(\mathbf{k})$ is the covariance matrix of the prior parameter distribution; and
- the weight matrix applied to the calibration dataset is equal to $C^{-1}(\epsilon)$ where $C(\epsilon)$ is the covariance matrix of measurement noise.

These conditions will rarely be met in real-world modelling practice for a variety of reasons, including the following.

- Neither $C(\mathbf{k})$ nor $C(\epsilon)$ is perfectly known.
- Numerical stability of the inversion process will normally require that singular value decomposition play some role in calculation of the parameter set which is deemed to calibrate a model; use of Tikhonov inversion alone does not provide the same unequivocal guarantee of numerical stability.
- In all modelling contexts, at least some degree of regularisation is encompassed in the simplifications required to build a model and to furnish it with a useable parameterization scheme; no resolution matrix is available for this type of regularisation, though it may make a significant contribution to the potential for parameter and predictive error.

It follows that predictive error variance will always be greater than predictive uncertainty variance. Nevertheless the goal of a well-implemented calibration strategy should be to minimize the difference between these two.

For the special case where calibration is undertaken using truncated singular value decomposition, and where the covariance matrices of prior parameter uncertainty and measurement noise are given by equations (4.11) and (4.12), equation (5.8) becomes:

$$\sigma_{\underline{y}-\underline{s}}^2 = \sigma_{\mathbf{k}}^2 \mathbf{y}^t \mathbf{V}_2 \mathbf{V}_2^t \mathbf{y} + \sigma_{\epsilon}^2 \mathbf{y}^t \mathbf{V}_1 \mathbf{S}^{-2} \mathbf{V}_1^t \mathbf{y} \quad (5.9)$$

The first term of equation (5.9) falls as the number of singular values used in the inversion process increases, whereas the latter term rises. Where truncation occurs at zero singular values (which is equivalent to not calibrating at all) then $\sigma_{\underline{y}-\underline{s}}^2$ is equal to $\sigma_{\mathbf{k}}^2 \mathbf{y}^t \mathbf{y}$; this of course is the pre-calibration uncertainty of the prediction. Where too many singular values are employed in the inversion process the second term of (5.9) becomes very high, thereby showing the deleterious effects of over-fitting. If predictive error variance is plotted against number of singular values used in the calibration process, a graph such as the following should result. (As will be shown later in this chapter, this can be obtained using the PEST PREDVAR1 utility.)

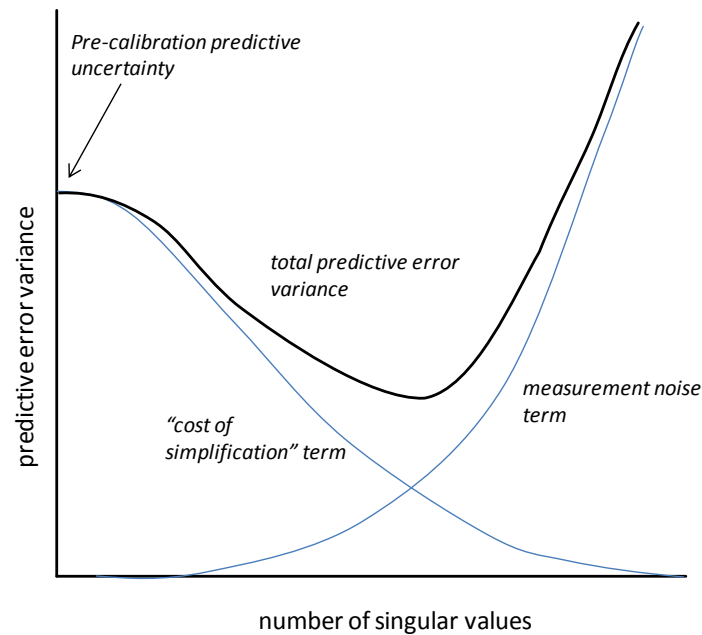


Figure 5.1 Predictive error variance as a function of number of singular values used in the inversion process.

Moore and Doherty (2005) show that graphs such as that depicted in Figure 5.1 can also be drawn to represent the effects on predictive error variance of varying the strength of application of regularization devices other than singular value decomposition. Irrespective of how a model is calibrated, a modeller is always faced with the decision on how strongly to apply his/her chosen regularisation scheme. The horizontal axis in Figure 5.1 can be considered to be the inverse of the strength of application of regularisation. When regularisation is “total”, and hence pre-calibration preferred parameter values are perfectly respected, the potential for predictive error is equal to pre-calibration predictive uncertainty. Where parameters are actually adjusted through the calibration process, but where a modeller is too heavy-handed in application of regularisation constraints, information extracted from the calibration dataset is less than the information that resides in it; the full potential of the calibration process to reduce predictive error variance is therefore not achieved. On the other hand, where too good a fit between model outputs and field measurements is sought, measurement noise is amplified in estimation of too many parameters - some of which should not be estimated at all because the information content of the calibration dataset with respect to these parameters is just too weak. In this case the potential for error in predictions made by the calibrated model may actually exceed the potential for error in predictions that would have been made if the model had not been calibrated at all!

Obviously optimality of regularisation is achieved at the minimum of the error variance curve of Figure 5.1.

Note that equations (5.8) and (5.9) can be used to calculate the error variance of an individual parameter if desired. In this case the \mathbf{y} vector is composed of zero-valued elements, except for the element pertaining to the parameter whose uncertainty is desired; this element should be given a value of 1.

Over-Determined Parameter Estimation

Where the calibration problem is well-posed, parameters can be estimated using the standard equation for Gauss-Marquardt-Levenberg parameter estimation:

$$\underline{\mathbf{k}} = (\mathbf{Z}^t \mathbf{Q} \mathbf{Z})^{-1} \mathbf{Z}^t \mathbf{h} \quad (5.10)$$

It is easily shown that solution of the inverse problem through singular value decomposition leads to the same $\underline{\mathbf{k}}$; however no truncation is necessary under these circumstances. Parameter and predictive error variance are therefore calculable using equations (5.5) and (5.8) respectively with \mathbf{R} equal to \mathbf{I} . For the special case where \mathbf{Q} is chosen to be the inverse of $\mathbf{C}(\epsilon)$ (the covariance matrix of measurement noise) the formula for the post-calibration covariance matrix of parameter error becomes particularly simple:

$$\mathbf{C}(\underline{\mathbf{k}} - \mathbf{k}) = (\mathbf{Z}^t \mathbf{Q} \mathbf{Z})^{-1} \quad (5.11)$$

Where the noise associated with different measurements is statistically independent (as is often assumed to be the case) \mathbf{Q} is diagonal. In this case use of a weighting matrix can be replaced by the use of individual measurement weights. If using PEST, a weighting strategy which assigns to each measurement a weight that is equal to the inverse of the standard deviation of noise associated with that measurement ensures a \mathbf{Q} matrix which is the inverse of $\mathbf{C}(\epsilon)$.

Where prior information on parameter values is weak so that $\mathbf{C}(\mathbf{k})$ specifies large pre-calibration uncertainties, equation (5.2a) becomes:

$$\mathbf{C}'(\mathbf{k}) = (\mathbf{Z}^t \mathbf{Q} \mathbf{Z})^{-1} \quad (5.12)$$

Hence under these circumstances post-calibration parameter and predictive uncertainty variance become equivalent to post-calibration parameter and predictive error variance.

The covariance matrix appearing in equations (5.11) and (5.12) is recorded by PEST at the bottom of its run record file, and in its matrix (*.mtt) file when parameter estimation is implemented using the Gauss-Marquardt-Levenberg method and the inverse problem of model calibration is indeed well-posed.

Derived Quantities

General

As already noted, both of equations (5.4) and (5.8) ultimately rely on the same matrices; the \mathbf{Z} matrix which specifies the action of the model does not appear in (5.8) but is used in calculation of the \mathbf{R} and \mathbf{G} matrices featured in that equation. Where a model is nonlinear its action, of course, cannot be represented by a matrix. In that case the Jacobian matrix is used in place of \mathbf{Z} . The Jacobian matrix mimics \mathbf{Z} ; its elements are comprised of the derivatives of all model outputs used in the calibration process with respect to all parameters adjusted through that process. Ideally calculation of the Jacobian matrix should be based on calibrated parameter values; supposedly these lie somewhere near the centre of the uncertainty/error interval that equations (5.4) and (5.8) seek to explore. The effect of model nonlinearity on estimates of parameter and predictive uncertainty/error is hopefully thereby diminished.

In spite of the approximation that necessarily attends use of a linearity assumption in a nonlinear context, equations (5.4) and (5.8) can provide useful estimates of predictive uncertainty and predictive error variance. As will be now demonstrated, they can also provide estimates of related quantities. While calculation of these related quantities will also be subject to error, it is expected that the effects of the linearity assumption on these calculations will be somewhat diminished as the principle focus of these calculations is to provide a basis for comparing different quantities, rather than in obtaining absolute values. Where compared quantities are affected in similar ways by an erroneous assumption of model linearity, relativity of their values may nevertheless be preserved.

Parameter Contributions to Predictive Uncertainty and Error Variance

The first of these derived quantities is referred to as “the contribution that a particular parameter, or group of parameters, makes to the uncertainty or error variance of a prediction”. This is defined as the fall in predictive uncertainty/error variance that is accrued when perfect knowledge of the parameter, or group of parameters, is gained. This is easily computed using either of equations (5.4) or (5.8) by simply modifying the $C(\mathbf{k})$ matrix to respect acquisition of this perfect knowledge, and calculating the diminution in uncertainty or error variance so accrued. Parameter or parameter group contributions to *pre-calibration* predictive uncertainty can be computed in similar fashion using equation (5.3a).

In general, it is better to use equation (5.4) rather than equation (5.8) in making calculations of this type. It is sometimes found that parameter contributions to predictive error variance (as distinct from predictive uncertainty variance) can be negative. This is an outcome of the non-Bayesian nature of error as opposed to uncertainty. It can also arise from the discrete nature of singular values in contexts where calibration is implemented using truncated singular value decomposition. An outcome of the latter phenomenon is that the minimum of the predictive error variance curve may shift between singular values when the error variance is calculated on the basis of two very different $C(\mathbf{k})$ matrices - one which reflects the true uncertainty of a particular parameter or parameter group, and another which infers that the parameter or parameter group is perfectly known.

“Contribution to predictive uncertainty” analysis need not be restricted to parameters whose values are estimated during the calibration process. It is often fruitful to include in this analysis boundary conditions and system stresses which are fixed at assumed values during the calibration process. When building a model, many details of its construction are only poorly known. Hence a litany of “reasonable assumptions” are made pertaining to various aspects of the model in order to allow model construction to proceed. In normal modelling practice, these “reasonable assumptions” are then retained during both the calibration and predictive phases of model usage, in spite of the fact that they may be erroneous. This can cause errors in model predictions, these arising from:

- the fact that certain parameters may assume erroneous values during the calibration process to compensate for the structural defects that are thus built into the model; and
- the fact that some or many model predictions may be sensitive to features of the model which are subject to error.

Features which are commonly fixed at “reasonable values” during the model construction, calibration and deployment processes include the following:

- inflow into some model boundary segments;
- head, pressure and concentration values assigned to other boundaries;
- historical pumping rates;
- recharge under historical and present land uses;
- elevations of the beds of rivers and streams;
- many other aspects of model design.

When undertaking linear predictive uncertainty/error analysis based on equations (5.4) and (5.8) these quantities can be awarded parameter status and thereby included in the analysis. Sometimes the unknown quantities themselves can be introduced as parameters into these equations. At other times “surrogate parameters” will be required that can simulate the effects

of unknown quantities while not representing them directly. For example a spatial, pilot-point-based multiplier field can be applied to recharge in order to include in the analysis a suite of parameters which represent the fact that the disposition of recharge over a model domain is only imperfectly known. Similarly, seasonal multipliers can be applied to historical rainfall to simulate the effect of limited rain gauge coverage on the integrity of parameters estimated for a rainfall/runoff model and of predictions which depend on these parameters. In all of these cases, the negative impact of having to make possibly erroneous assumptions on overall model predictive performance can be assessed by computing the contribution that parameters pertinent to these assumptions make to the uncertainty/error variance of critical model predictions.

Figure 5.2 shows the outcomes of such an analysis when applied to a regional groundwater model; this is taken from Gallagher and Doherty (2007a). See James et al. (2009) for another example.

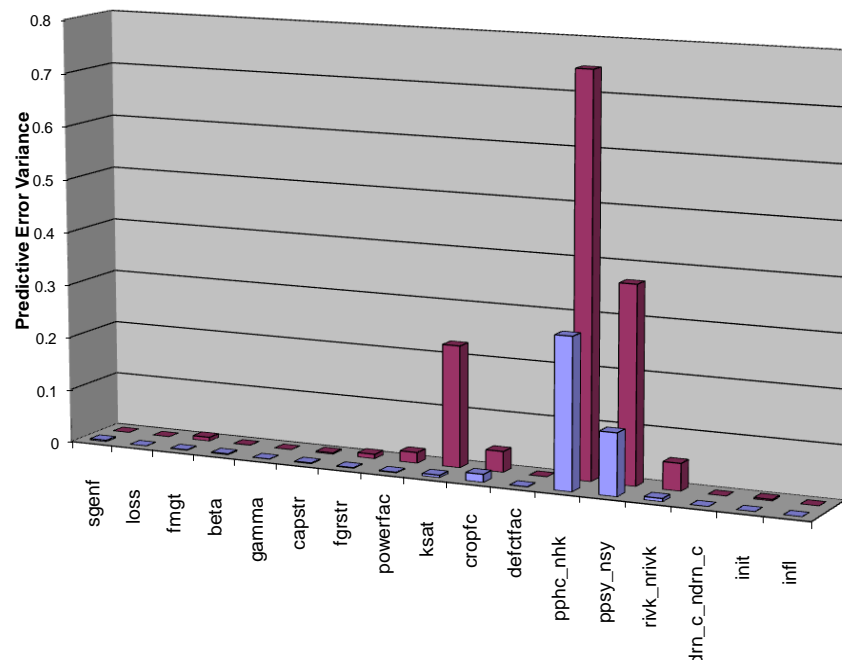


Figure 5.2. Pre-calibration (back row) and post-calibration (front row) contributions to the error variance of a prediction of interest made by different parameter and boundary condition types employed by a regional groundwater management model.

It is sometimes found when computing pre- and post-calibration contributions to predictive uncertainty/error variance that the post-calibration contribution of a particular parameter to the uncertainty/error variance of a particular prediction exceeds its pre-calibration contribution. This seemingly contradictory conclusion arises from the definition of “contribution to predictive uncertainty/error variance” given above. If a prediction is not sensitive to a parameter, then acquisition of perfect knowledge of that parameter does not decrease the uncertainty of that prediction under pre-calibration conditions. However if that parameter can only be estimated in conjunction with another parameter during the model calibration process - one to which the prediction is indeed sensitive - then acquisition of perfect knowledge of the first parameter supplements the information available through the calibration dataset pertaining to the second parameter. Hence acquisition of perfect post-

calibration knowledge of the first parameter reduces the uncertainty/error variance of the prediction.

As will be demonstrated shortly, parameter contributions to predictive uncertainty and error variance can be calculated using the PREDUNC4 and PREDVAR4 utilities supplied with PEST.

Data Worth

Suppose that a particular prediction of future environmental behaviour is important to the management of that environment. The “worth” of a particular item of data in relation to that prediction can be defined as “the reduction in uncertainty of that prediction that is accrued through acquisition of that data”. An immediate outcome of the ability to compute predictive uncertainty is therefore an ability to compute the utility of data in reducing that uncertainty, and hence the worth of that data.

An extremely useful feature of equations (5.4) and (5.8) when used for assessment of data worth in this manner is that neither predictive uncertainty nor predictive error variance depends on the actual value of the measurements that comprise an observation dataset, nor on the actual values of parameters that populate the model. According to these equations, predictive uncertainty and error variance depend only on the stochastic characterization of parameter variability as expressed by $C(\mathbf{k})$, the stochastic characterization of measurement noise as expressed by $C(\epsilon)$, and on the sensitivities of calibration and predictive model outputs to parameters as expressed by \mathbf{Z} and \mathbf{y} . (For a linear model both \mathbf{Z} and \mathbf{y} are independent of parameter values.) It follows that equations (5.4) and (5.8) can be used to calculate the reduction in uncertainty that would be accrued through acquisition of data that has not yet been gathered. All that is needed is the sensitivity of model outputs corresponding to the yet-to-be-acquired measurements to parameters employed by the model. Obviously these can be computed by the model irrespective of whether corresponding field measurements have actually been taken or not.

When assessing data worth, use of equation (5.4) is preferable to that of equation (5.8) for the same reasons as those discussed above for assessment of parameter contributions to predictive uncertainty. That is, use of equation (5.8) can be compromised to some extent by the granular nature of singular values, and by the fact that predictive error is less of an intrinsic quality of a system and its data than is predictive uncertainty.

When simulating the acquisition of data in order to assess its worth, the user must specify elements of the $C(\epsilon)$ matrix pertinent to the tested data. Where measurement noise is independent from one data element to the next, this requires specification only of the uncertainty associated with individual measurements, as the $C(\epsilon)$ matrix is diagonal under these circumstances.

“Data” can include either measurements of system state, or direct measurements of system properties. The sensitivity of data of the later type to model parameters (i.e. the rows of the \mathbf{Z} matrix corresponding to such data) are zeros except for a 1 which corresponds to the parameter whose value is measured. The (normally diagonal) elements of the $C(\epsilon)$ matrix corresponding to these measurements are calculated on the basis of the expected propensity for error in making them.

Data worth computation based on equation (5.4) is undertaken using the PEST PREDUNC5 utility; data worth computation based on equation (5.8) is undertaken using the PEST PREDVAR5 utility. These utilities allow the worth of either individual data elements or of

suites of data to be assessed. In both cases “worth” can be assessed in both of the following ways:

- through the reduction in predictive uncertainty/error variance that is accrued if data were added to an existing dataset (including the null dataset); and
- through the rise in predictive uncertainty that is incurred if data were lost.

It is essential that data worth assessment take place in a highly parameterized context, particularly if assessing the worth of acquiring data at different places within a spatial model domain such as that of a groundwater model. If artificial constructs such as zones of piecewise constancy are employed for regularization purposes, it will be found that optimal locations for acquisition of new data will invariably lie at the boundaries of such zones. If real-world hydraulic properties are not, in fact, piecewise constant, the disinformation involved in the assumption of piecewise uniformity will create an artificial context for assessment of the worth of new data which does little more than reinforce this assumption. See Fienen et al. (2010) for a discussion of this topic.

In most cases, data is of worth to the extent that it reduces the dimensionality of the null space, for the existence of this space is by definition an outcome of data insufficiency. The null space can only be explicitly or implicitly defined where a model is endowed with parameterization complexity that reflects real-world complexity.

Exercises

General

The GENLINPRED Utility

The PEST suite provides a number of tools that analyse predictive uncertainty/error, and that calculate ancillary quantities such as parameter contributions to predictive uncertainty/error variance and data worth. Those that are of most general use belong to the PREDUNC and PREDVAR suite of programs. “PREDUNC” stands for “predictive uncertainty” while “PREDVAR” stands for “predictive error variance”. In general, use of the former suite is to be preferred over use of the latter suite because uncertainty is more of an intrinsic quality of the system and data whereas “error” includes recognition of the imperfect nature in which environmental data is necessarily processed using models. Because the PREDVAR suite bases error variance calculation on the use of truncated singular value decomposition as a calibration device, the potential for predictive error that is additional to that arising purely from uncertainty is reduced to a minimum. However it is not eliminated - especially in cases where pre-calibration uncertainty as expressed by the $C(\mathbf{k})$ matrix of innate parameter variability shows a high degree of correlation between parameters of the same or different types. In some instances this, together with the non-continuous nature of singular values, can lead to spurious processing outcomes, such as negative contributions by certain parameter groups to predictive error variance. These effects are unlikely to be large. However it is best to avoid them altogether, as occurs when basing these analysis on PREDUNC-suite programs.

The PEST suite provides a driver program named GENLINPRED which grants easy user access to members of the PREDUNC and PREDVAR utility suites. (GENLINPRED stands for “general linear predictive analysis”. Part of its ease of use springs from the fact that GENLINPRED provides default inputs for PREDUNC-suite and PREDVAR-suite programs when it runs them. The user should not forget, however, that greater flexibility of

PREDUNC-suite and PREDVAR-suite program usage is gained when they are run without the help of GENLINPRED, and the user can therefore tune them to his/her specific needs.

Predictive Sensitivities

The exercises below are made easy by the fact that one or more predictions of interest are “carried” in PEST control files that were used in calibrating the surface water and groundwater models that constitute our example cases. These predictions appear to PEST as “observations”, but are given weights of zero so that they have no effect on the parameter estimation process. However their presence in the PEST control file forces PEST to calculate sensitivities of these predictions to all parameters. Thus the \mathbf{y} vector that features in equations (5.4) and (5.8) is automatically calculated.

In other contexts a special PEST run will be required for calculation of the elements of the \mathbf{y} vector. The PEST control file on which this run is based will not instruct PEST to calibrate a model, for all of the entities that feature in its “observation data” section will in fact be predictions. However it will feature the same set of parameters as those employed in a previous calibration exercise. NOPTMAX will be set to -1 or -2 in the “control data” section of this PEST control file. Thus PEST will calculate a Jacobian matrix which contains the sensitivities of all predictions to all model parameters, and then cease execution. The \mathbf{y} vector for any particular prediction will comprise a single row of this matrix.

The $C(\mathbf{k})$ and $C(\epsilon)$ Matrices

Both of equations (5.4) and (5.8) require two covariance matrices, namely the $C(\mathbf{k})$ matrix of innate parameter variability (i.e. the covariance matrix of the prior parameter probability distribution), and the $C(\epsilon)$ matrix of measurement noise. The former is normally supplied through a “parameter uncertainty file”. We have already had some experience in using this type of file. However GENLINPRED provides the user with an easy alternative option for obtaining this matrix. Under the assumption that all parameter are statistically independent (and hence that the $C(\mathbf{k})$ matrix has no off-diagonal elements) GENLINPRED can calculate the standard deviation of each parameter as one quarter of the difference between its upper and lower bounds as supplied in a PEST control file. Where parameters are log-transformed this is taken into account in making this calculation. As is explained in PEST documentation, parameter variability as described by the $C(\mathbf{k})$ matrix must pertain to the logs of parameters that are declared as log-transformed in PEST control file(s) on which predictive uncertainty/error analysis is based.

To facilitate ease of input, PREDUNC-suite and PREDVAR-suite programs (and hence GENLINPRED) calculate the $C(\epsilon)$ matrix from measurement weights appearing in the PEST control file with which they are supplied. The user must bear this in mind when preparing this file. Weights associated with measurements are assumed to be proportional to the standard deviation of noise associated with respective measurements; the constant of proportionality (or a variable from which this can be calculated) is supplied to PREDUNC-suite or PREDVAR-suite programs, or to GENLINPRED, when these programs are run. (Note that the user has the option of providing a $C(\epsilon)$ matrix through the PEST control file if he/she desires; see PEST documentation for further details.)

Formulation of a suitable $C(\epsilon)$ matrix for use in predictive uncertainty/error analysis is a problematical matter. As has already been discussed, most model-to-measurement misfit is an outcome of structural noise rather than measurement noise. Ideally, therefore, $C(\epsilon)$ should represent, at least to some degree, the covariance matrix of structural noise. However this is never known. Furthermore it is probably singular. Failing to take its singular nature into

account inevitably results in underestimation of predictive uncertainty. Unfortunately there is no way that it can be properly taken into account. Hence designation of a $C(\epsilon)$ matrix (and hence assignment of weights to observations appearing in the PEST control file) for use in linear uncertainty/error analysis will always be an approximate process.

Fortunately, however, failure to properly characterize $C(\epsilon)$ affects only the solution space component of predictive uncertainty/error. As most uncertainty/error in many modelling contexts is dominated by parameter null space variability rather than solution space variability, an incapacity to characterize $C(\epsilon)$ exactly is often not as big an issue as it first appears.

Predictive Noise

The largest contributor to the uncertainty/error variance of predictions made by many surface water models, particularly those which employ only a few parameters, will ultimately be the predictive structural error term. As has already been discussed, this can be considered to be a random “parameter” that is added to a model output in order to furnish the actual model prediction. (Note that in highly parameterized contexts this term is often of smaller importance than in contexts characterized by only a few parameters. This is because model structural error tends to be reduced by the use of many parameters, and because uncertainty will normally be dominated by null space variability.)

Calibration datasets for surface water models tend to be very large, being comprised of daily or even hourly flow measurements taken over many years. The limited number of parameters that are often estimated in calibrating these types of models tend to look as if they are very well estimated, for their PEST-calculated uncertainty margins are often very small. This is often illusory, for uncertainty analysis cannot take account of the singular nature of the $C(\epsilon)$ covariance matrix, mainly because the exact nature of its singularity is unknown.

It follows that where low uncertainties are attributed to estimated surface water model parameters this rarely means that they lack uncertainty. Even less does it mean that model predictions which depend on them lack uncertainty. It only means that the additive predictive structural noise term may become the dominant contributor to predictive uncertainty. Where a prediction resembles measurements used in the calibration process, this predictive noise term is partially quantified through the calibration process. It can then be formally included in the predictive uncertainty analysis process as a notional parameter as described previously (and as is demonstrated below). Alternatively an appropriate predictive “engineering safety margin” can be employed when basing environmental management on model outcomes. Given the necessarily heuristic assessment of this important contributor to predictive uncertainty, this is often as good an approach to use as any other. The subjective nature of the process is highlighted by the fact that this noise term may be different for different types of prediction, and for predictions pertaining to different aspects of flow. This, in fact, suggests another advantage of the use of a multi-component objective function (with different components based on different aspects of the same flow time series) in calibrating models of this type. The noise associated with these different aspects of flow can be estimated as part of the parameter estimation process, and then included in the predictive uncertainty analysis process.

The JCO2JCO Utility

At the heart of linear uncertainty analysis is a Jacobian matrix. This resides in a binary file with an extension of *.jco after completion of a PEST run. If a model has a large run time, and if many parameters are employed in the uncertainty/error analysis process (as is

recommended so that the null space contribution to these quantities can be properly formulated), acquisition of a Jacobian matrix can be a computationally expensive procedure. Once acquired, a Jacobian matrix is therefore precious.

A user does not want to have to calculate a new Jacobian matrix if he/she alters a PEST control file in a small way. The JCO2JCO utility allows a user to compute a Jacobian matrix for a new PEST control file based on the Jacobian matrix for an existing PEST control file, if the alterations that were required in making the new PEST control file from the existing one are restricted to the following.

- removal, fixing and/or tying of parameters;
- changing the transformation status of parameters;
- changing the SCALE and OFFSET applied to parameters;
- removal of observations;
- alterations to weights assigned to observations;
- addition or removal of prior information.

See PEST documentation for further details.

Surface Water Model

Which Case to Use?

So far we have calibrated our surface water model twice. On one occasion we used a single objective function comprised only of the logarithms of flows. On the other occasion we used a multi-component objective function in which weights were uniform within each observation group, but in which inter-group weighting was adjusted so that the contribution to the initial objective function made by each objective function component was about the same.

Which one of these should be used as a basis for post-calibration uncertainty analysis? Because calibration of this model constitutes a well-posed inverse problem (there is no null space), all parameter and predictive uncertainty is theoretically ultimately attributable to measurement noise (which is mostly structural noise). So the $C(\epsilon)$ matrix is important. However it is also unknown. Implicit in the two definitions of the objective function through which calibration of the model previously took place are two assumptions of the nature of structural noise. Neither assumption is valid; so the choice between the two is necessarily subjective. However, in light of considerations presented by Doherty and Welter (2010), the PEST control file in which calibration was achieved through use of a multi-component objective function is chosen as the basis for the linear uncertainty analysis that follows.

The pertinent PEST control file is *calib2.pst*. However recall that we made a version of this named *calib2_soln.pst* which contains best-fit parameters.

Re-weighting

Inspect file *calib2_soln.pst*. Set NOPTMAX to 0 in this file (if it is not already set to this value). Recall that NOPTMAX is the first variable on the 9th line of a PEST control file. Hence if PEST is now run on the basis of *calib2_soln.pst* it will run the model once, calculate the objective function (and contributions to the objective function made by different observation groups) and then cease execution.

Run PEST using the command:

```
pest calib2_soln
```

PEST's screen output should list the total objective function, and contributions to the total objective function made by different observation groups, as follows.

Sum of squared weighted residuals (ie phi)	=	2019.7
Contribution to phi from observation group "lsim_flow"	=	594.32
Contribution to phi from observation group "bsim_flow"	=	644.62
Contribution to phi from observation group "qmax_sim"	=	0.0000
Contribution to phi from observation group "sim_vol"	=	674.68
Contribution to phi from observation group "sim_ex"	=	106.05

Recall that the contribution from the group *qmax_sim* is zero because this group contains a single “observation” which is actually the prediction which is of interest to us. This “observation” is assigned a weight of zero.

Use of PREDUNC-suite and PREDVAR-suite programs is predicated on the assumption that observation weighting is inversely proportional to the standard deviation of measurement/structural noise as it affects the different observations used in the model calibration process. These standard deviations are normally impossible to know. Nevertheless we could, if we wished, attempt to conform to this precept by adjusting observation weights at this stage if we wished.

Options for inter-observation-group weights adjustment are provided by the PWTADJ1 or PWTADJ2 utilities supplied with PEST. (PWTADJ stands for “PEST weight adjustment”.) The first equalizes contributions to the objective function by different observation groups; we have already used this utility. The second performs inter-group weights adjustment in an effort to have weighting reflect measurement noise as it is suggested by a preceding calibration process; see PEST documentation for further details. However PWTADJ2 does not take account of the fact that a greater number of observations does not necessarily carry a greater amount of information when $C(\epsilon)$ is singular (as it is when structural noise is accounted for). Hence we will leave relative inter-group weighting as it is. Nevertheless, as will be shown shortly, a global weighting factor will nevertheless be applied in order to harmonize overall weighting with model-to-measurement fit achieved through the previous calibration exercise.

Calculating the Jacobian Matrix

Set NOPTMAX to -1 in *calib2_soln.pst* and re-run PEST to calculate the Jacobian matrix. This forms the basis for the linear uncertainty analysis that we will now undertake.

GENLINPRED

Run GENLINPRED, responding to its prompts as follows.

```
Enter name for response file (<Enter> if none): <none>
Use abbreviated or full input? [a/f] (<Enter> if "f"): a

Enter name of PEST control file: calib2_soln.pst

- reading PEST control file calib2_soln.pst....
- file calib2_soln.pst read ok.

Use bounds or uncert file for param uncertainties [b/u] <Enter> if "b": <Enter>

Are weights the inverse of measurement uncertainty? [y/n] <Enter> if "y": n
Enter factor for weights to make this so: 0.609
```

```

Enter name for output file <Enter> if genlinpred.out: <Enter>

Perform comprehensive analysis of a prediction/param? [y/n] <Enter> if "y": y
Provide pertinent information:-

Enter name of prediction/parameter to analyze: qmax_sim_max
Enter file to read its sensitivities ["p" if a parameter]: calib2_soln.jco

Compute predictive uncertainty? [y/n] <Enter> if "y": <Enter>

Compute parameter contributions to uncertainty? [y/n] <Enter> if "y": <Enter>
For indiv parameters or for parameter groups? [i/g] <Enter> if "g": <Enter>

Compute observation worth wrt uncertainty? [y/n] <Enter> if "y": <Enter>
For indiv observations or for observation groups? [i/g] <Enter> if "g": <Enter>

```

As is apparent, default responses are provided to most of GENLINPRED's prompts. The following should also be noted.

- Correct setting of observation weights implies that the objective function pertaining to the calibrated model should be roughly equal to the number of observations featured in the calibration dataset. Under these circumstances squared weighted residuals would have an average value of 1.0 because the weight associated with each observation would be equal to the standard deviation of noise associated with that observation. As stated above, the situation is, of course, terribly compromised by the presence of structural noise. However some effort should be made to adjust weights so that they are at least in the right ball park. The objective function for the calibrated model is 2019.7. The total number of observations is 750 (actually 751 but one of these is a prediction with a weight of zero). If all weights were multiplied by a factor equal to the square root of 750 divided by 2019.7 (i.e. 0.609), the objective function would then be equal to 750. This is the basis for the selection of this factor in response to the pertinent GENLINPRED prompt.
- The prediction of interest is named *qmax_sim_max* in the PEST control file *calib2_soln.pst*; TSPROC gave it this name. Recall that this is the maximum flow over the period spanning 19th to 23rd August 1986. As has already been discussed, sensitivities of this prediction to all parameters are computed by PEST when it computes sensitivities for all calibration-specific model outputs to these parameters as this prediction is "carried" in the PEST control file. Hence the sensitivity of this prediction to all model parameters resides in the Jacobian matrix file which we have just calculated.
- Because each parameter belongs to its own parameter group in the PEST control file *calib2_soln.pst*, calculation of the contribution of each *parameter* to total predictive uncertainty is the same as calculation of the contribution of each *parameter group* to total predictive uncertainty.

Predictive uncertainty

When run as above, GENLINPRED runs programs of the PREDUNC suite behind the scenes. The outcomes of these calculations are recorded in the GENLINPRED output file named, in this case, *genlinpred.out*.

As is apparent from *genlinpred.out*, GENLINPRED calculates pre- and post-calibration uncertainty standard deviations of 432 m³/sec and 56 m³/sec respectively for the predicted maximum flow over the period 19th to 23rd August 1986. The former is much higher than the pre-calibration uncertainty calculated in Chapter 3. This is an outcome of model nonlinearity,

and consequential changes in sensitivities of model outputs to parameter values as the latter vary throughout their range.

Despite the obviously approximate nature of linear uncertainty analysis when applied to a nonlinear model, the assessment of post-calibration predictive uncertainty as being significantly lower than pre-calibration uncertainty is probably robust. GENLINPRED reports that the post-calibration uncertainty standard deviation of the maximum flow prediction is about 56 m³/sec. Actually, in light of the considerable error which is in fact associated with the prediction made by the calibrated model, this may not be a bad assessment. (Recall that for a normal distribution, an interval of 1 standard deviation on either side of the mean corresponds to a confidence interval of about 67%, 2 standard deviations correspond to a confidence interval of about 95%, and 3 standard deviations correspond to a confidence interval of about 99.7%).

Parameter Contributions to Predictive Uncertainty

Pre- and post-calibration contributions to predictive uncertainty variance by the different parameters employed by the model are tabulated by GENLINPRED following its listing of total pre- and post-calibration uncertainty. Note that:

- variance is the square of standard deviation;
- contributions to variance rather than standard deviation are calculated because contributions to variance are additive in the pre-calibration context whereas contributions to standard deviation are not.

These contributions are plotted below. All contributions have been normalized with respect to the pre-calibration uncertainty variance of the maximum flow prediction in this graph.

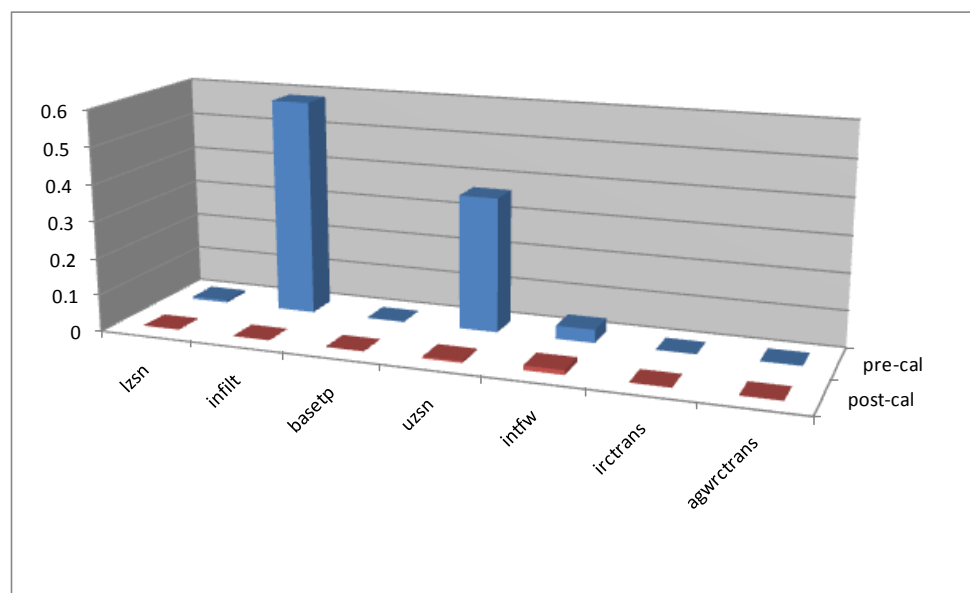


Figure 5.3 Normalized parameter contributions to pre- and post-calibration uncertainty variance of predicted flow over the period 19th to 23rd August 1986.

Data Worth

GENLINPRED was asked to assess the worth of different observation groups to the prediction of interest. In the present instance these groups are comprised of the same data processed in different ways. However ideally such processing should be such as to attempt to “distil” from the calibration dataset information that is salient to different groups of

parameters. If a prediction of interest is sensitive to only one of these parameter groups, then this may result in some of these processing outcomes providing information that is particularly salient to that prediction while others may not.

GENLINPRED provides two different bases for data worth assessment. Firstly it calculates the rise in predictive uncertainty variance that is incurred if each observation group is successively removed from the calibration dataset. This provides a measure of the extent to which information pertaining to the prediction of interest resides exclusively in each group. Secondly it assesses the fall in uncertainty variance from its pre-calibration level that is accrued as each observation group in turn comprises the sole member of the calibration dataset. This indicates the absolute information content of each observation group in relation to the prediction of interest.

GENLINPRED's output tables are graphed in Figure 5.4. As for the previous figure, uncertainty variance is normalized with respect to pre-calibration uncertainty variance.

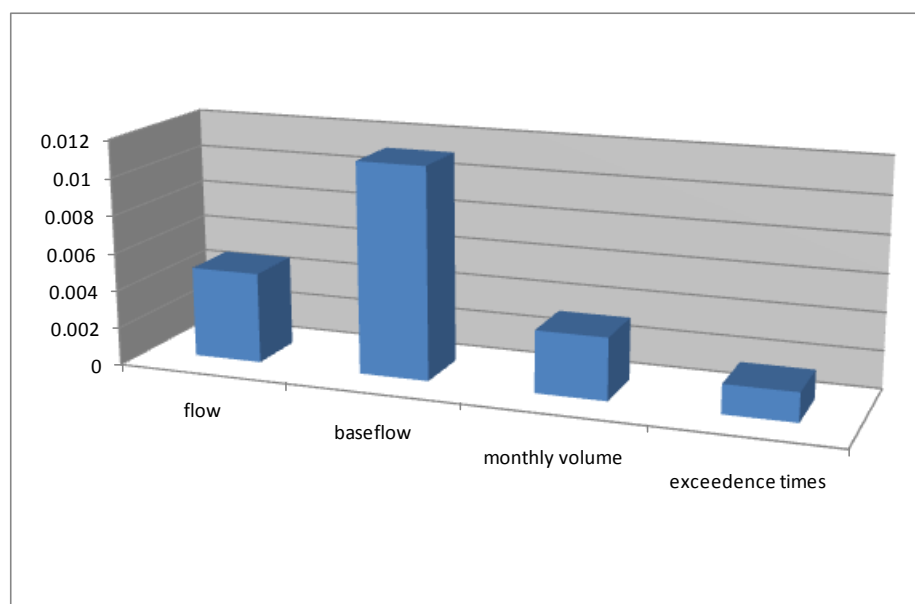


Figure 5.4a Rise in relative (to pre-calibration uncertainty variance) post-calibration predictive uncertainty variance incurred by omission of each observation group from the calibration dataset.

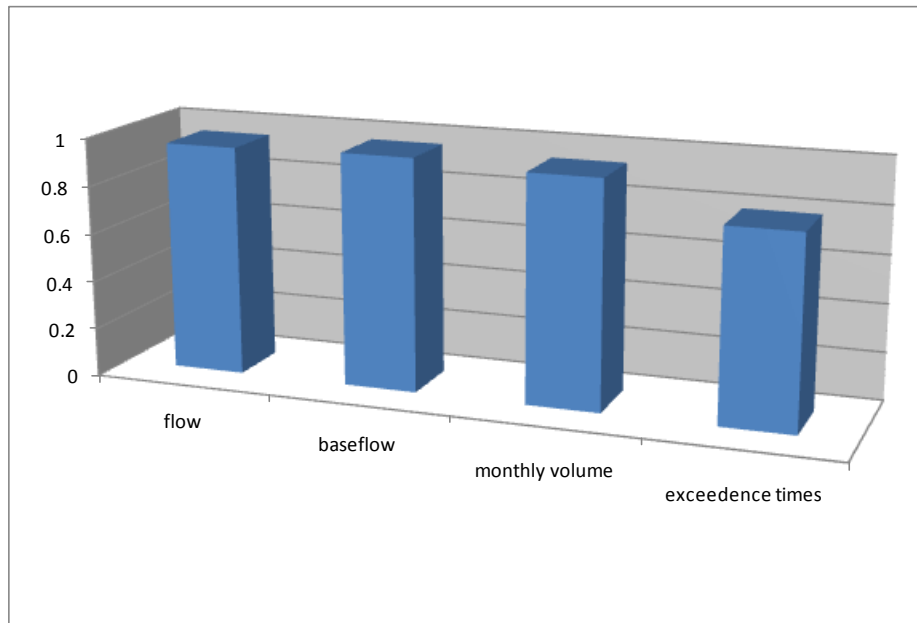


Figure 5.4b. Fall in relative (to pre-calibration uncertainty variance) of uncertainty variance accrued with inclusion of each observation group as the sole member of the calibration dataset.

Figure 5.4 is not very exciting; nevertheless it is informative. It states that information pertaining to the prediction of interest is repeated in the different observation groups, and that use of any one of these groups in the calibration process would achieve about the same reduction in predictive uncertainty from its pre-calibration level as use of any other. Nevertheless the following points are of some interest.

- The information content of exceedence times is very high given the small population of this observation group.
- Baseflow contains a little prediction-pertinent information that is not repeated in other groups.

Groundwater Model

Preparing for GENLINPRED

As an outcome of work we have done previously in calibrating the groundwater model, best-fit parameter values are represented as initial parameter values in the PEST control file *calib1r_soln.pst*. Recall also that we obtained a Jacobian matrix corresponding to this file; this matrix is recorded in binary format in file *calib1r_soln.jco*.

GENLINPRED insists that the PEST control file which it reads does not direct PEST to run in “regularisation” mode. This avoids confusion between expert knowledge supplied through regularization constraints, and that supplied through the pre-calibration covariance matrix $C(\mathbf{k})$; all such information must be supplied through the latter vehicle.

A stripped-down PEST control file named *calib2.pst* can be made from *calib1r_soln.pst* using the SUBREG1 utility; SUBREG1 stands for “subtract regularisation”. To achieve this, run SUBREG1 as follows:

```
subreg1 calib1r_soln calib2
```

Now we need a Jacobian matrix file corresponding to *calib2.pst*. This can be obtained from the Jacobian matrix file that was previously calculated for *calib1r_soln* using the command:

```
jco2jco calib1r_soln calib2
```

Run this command to obtain the new Jacobian matrix file.

Running GENLINPRED

Run GENLINPRED, responding to its prompts as follows. If you make any errors, simply respond to the next prompt with “e” followed by <Enter>; GENLINPRED will then backtrack to its previous prompt.

```
Enter name for response file (<Enter> if none): <Enter>
Use abbreviated or full input? [a/f] (<Enter> if "f"): f

Enter name of PEST control file: calib2.pst

- reading PEST control file calib2.pst....
- file calib2.pst read ok.

Use bounds or uncert file for param uncertainties [b/u] <Enter> if "b": u
Enter name of parameter uncertainty file: param.unc

Are weights the inverse of measurement uncertainty? [y/n] <Enter> if "y": y

Enter name for output file <Enter> if genlinpred.out: <Enter>

Perform global parameter estimability analysis? [y/n] <Enter> if "n": y
Provide pertinent information:-

    Compute parameter identifiabilities? [y/n] <Enter> if "y": y
    Compute relative parameter error reduction? [y/n] <Enter> if "n": y
    Use SUPCALC to estimate soln space dimensions? [y/n] <Enter> if "y": y
    Compute relative parameter uncertainty reduction? [y/n] <Enter> if "n": y

Perform comprehensive analysis of a prediction/param? [y/n] <Enter> if "y": y
Provide pertinent information:-

    Enter name of prediction/parameter to analyze: part_time
    Enter file to read its sensitivities ["p" if a parameter]: calib2.jco

    Compute soln/null space contribs to predict error? [y/n] <Enter> if "n": y
    Compute predictive uncertainty? [y/n] <Enter> if "y": y

    Compute parameter contributions to error? [y/n] <Enter> if "n": y
    Compute parameter contributions to uncertainty? [y/n] <Enter> if "y": y
    For indiv parameters or for parameter groups? [i/g] <Enter> if "g": i

    Compute observation worth wrt error? [y/n] <Enter> if "n": y
    Compute observation worth wrt uncertainty? [y/n] <Enter> if "y": y
    For indiv observations or for observation groups? [i/g] <Enter> if "g": i

- running SCALEPAR to build scaled PEST input dataset...
- program SCALEPAR run ok.

- running SUPCALC to compute dimensions of solution space...
- program SUPCALC run ok.

SUPCALC has recommended the use of 12 solution space dimensions
for computation of parameter identifiability and relative
error reduction.

Do you wish to over-ride this? [y/n] <Enter> if "n": n
```

Because full, rather than abbreviated, GENLINPRED input was requested, GENLINPRED issues many prompts. However in order to ease the burden of running it on subsequent

occasions, GENLINPRED provides the user with the option of recording a response file. This can then be supplied to it on subsequent runs. See GENLINPRED documentation for further details.

The GENLINPRED output file *genlinpred.out* provides a wealth of information. Some of this will now be reviewed.

Parameter Identifiability and Relative Uncertainty/Error Reduction

Parameter identifiability has already been discussed. Relative uncertainty and error variance reduction are similar statistics; these are described in detail by Doherty and Hunt (2010). Like identifiability they can be plotted in the same spatial disposition as the parameters to which they pertain if this proves useful; see file *hk.pts* for pilot point, and hence parameter, coordinates. If this is done plots similar to Figure 4.12 result.

Note the occurrence of some negative values for relative parameter error variance reduction. These are an outcome of the nature of error as discussed above; negative values for relative parameter uncertainty variance reduction are impossible.

Predictive Uncertainty

The outcomes of predictive uncertainty and error variance calculations are provided in the next part of the GENLINPRED output file. The following are noteworthy.

Statistic	Value
Pre-calibration uncertainty	2753
Post-calibration uncertainty standard deviation	1723
Post-calibration error standard deviation	1902

Table 5.1. Pre- and post-calibration uncertainty and error statistics computed by GENLINPRED.

As is expected, post-calibration error standard deviation is higher than post-calibration uncertainty standard deviation, the latter being the aspirational goal of the former. It is apparent that the uncertainty associated with the prediction of particle travel time is still disturbingly high, despite the fact that the model has been “calibrated” against head measurements in the 12 observation wells.

Data forthcoming from GENLINPRED’s running of PREDVAR1 allows us to explore this issue. This data allows us to produce a plot similar in type to that shown in Figure 5.1. Unfortunately the data provided by GENLINPRED in this case does not provide graphs with the same aesthetic appeal as those schematized in Figure 5.1. Nevertheless these data do provide some powerful insights. These include the following.

- The variance of predictive uncertainty is dominated by the null space term. In other words, head measurements simply do not carry enough information to allow accurate predictions of travel time to be made, for the latter is sensitive to hydraulic property details that are not available to us through calibration based on heads.
- The solution space term rises to infinity at the thirteenth singular value. As there are only 12 observations, this marks the point at which the null space *must* begin. In everyday modelling practice the point at which predictive error variance is minimized (and at which the null space is *decreed* to begin for that particular calibration

exercise) often occurs at a much smaller number of singular values than there are observations in the calibration dataset.

Parameter Contributions to Predictive Uncertainty

Parameter contributions to the pre- and post-calibration uncertainty of predicted particle travel time can be visualized through a bar chart as in Figure 5.2. In the present case, however the latter are plotted at respective pilot point locations; see Figure 5.5. These contributions are normalized with respect to pre-calibration uncertainty variance.

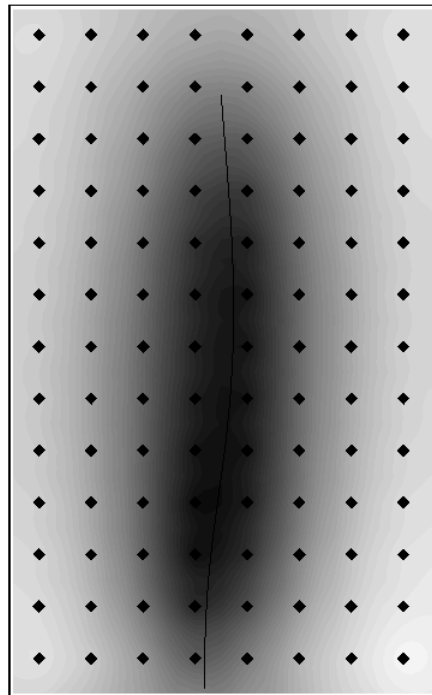


Figure 5.5 Post-calibration contributions by different pilot point parameters to the uncertainty of predicted travel time predictions. Darker shading indicates higher contributions. The maximum value is 0.42 (relative to pre-calibration uncertainty variance) while the minimum value is 0.0. The particle path calculated by the calibrated model is also shown.

Recall the definition of “parameter contribution to predictive uncertainty”. It is the reduction in predictive uncertainty that would be accrued if perfect knowledge were gained of a particular parameter; the same definition can be applied to groups of parameters. It is hardly surprising that acquisition of knowledge of hydraulic conductivity in the centre of the model domain, in the vicinity of the particle trajectory, would achieve greater decreases in particle travel time predictive uncertainty than parameter knowledge acquired elsewhere. Furthermore, the reductions in predictive uncertainty so accrued would be significant.

Data Worth

The head data that presently comprises the calibration dataset does not promulgate a very large reduction in travel time predictive uncertainty. However it is nevertheless worth something, for the above analysis shows that the uncertainty of this prediction is lowered to some extent through the calibration process. If the worth of different head measurements is plotted at respective measurement well locations, the following figures emerge.

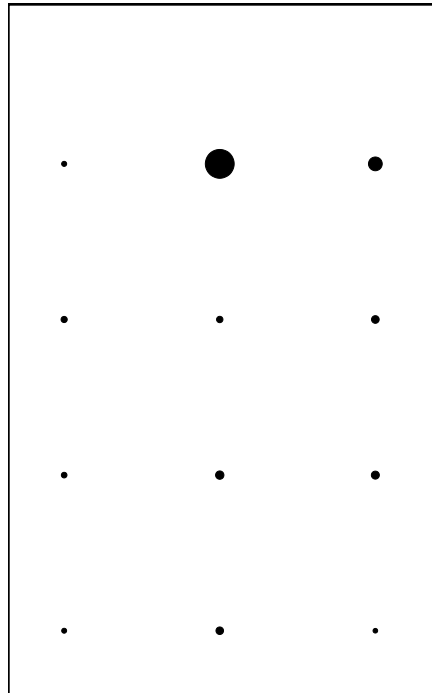


Figure 5.6a. Uncertainty variance gain (relative to pre-calibration uncertainty) of predicted travel time incurred through loss of an individual head measurement from the calibration dataset. Highest value is 0.028 while lowest value is 2×10^{-4} .

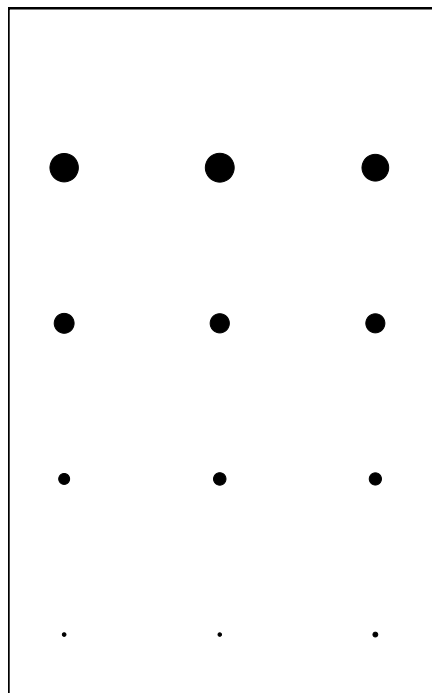


Figure 5.6b. Uncertainty variance loss (relative to pre-calibration uncertainty) of predicted travel time incurred through calibration using a single head measurement at each of the individual well locations. Highest value is 0.55 while lowest value is 0.17.

The back row of wells figures prominently in Figure 5.6b. This is because the head difference between any one of the wells in this row and the lower model boundary says much about the average hydraulic conductivity within a large part of the model domain. However Figure 5.6a shows that the most informative of these is that in the centre of this row.

6. How Wrong can a Prediction Be? Nonlinear Analysis

Error and Uncertainty

Bayes equation makes it clear that a model's parameters retain uncertainty even after they have been subjected to the history-matching process. That is, a model's parameters are still free to "wobble", even though a model has been calibrated. However their post-calibration variability is subject to constraints. Obviously one of these constraints is that the parameters remain realistic, thereby respecting expert knowledge. This constraint is embodied in the prior probability distribution of parameters which is an integral part of Bayes equation. However a second constraint is imposed through the history-matching process. This constraint further restricts "parameter wiggle room"; it maintains that as model parameters wiggle, they must wiggle in such a way that model-to-measurement misfit (as embodied in the likelihood term of Bayes equation) does not rise unduly.

The same notions are expressed in equations (4.10) and (4.13) which focus on error rather than on uncertainty. Post-calibration variability is again subject to two constraints. If regularisation is achieved using singular value decomposition, these constraints on parameters are orthogonal to each other in parameter space. One set of constraints restricts solution space parameter variability to that which allows the model to retain its calibrated state. However parameter combinations that are orthogonal to this, and hence lie within the calibration null space, are given much greater freedom of movement for, by definition, their variation has no effect on model outputs for which there are complimentary field measurements comprising the calibration dataset. Mathematically, for a linear model, their variability has no limits. It is only expert knowledge that imposes limits on their variability.

As has been discussed, while Bayesian analysis provides a complete conceptual framework for assessment of parameter and predictive uncertainty, modelling practicalities normally require a two step process of calibration followed by analysis of potential for parameter and predictive error. Not only is the latter approach generally more computationally tractable than Bayesian analysis. It also acknowledges the fact that sources of error in model predictions include not only information deficits in expert knowledge and hard site data; they also arise from the imperfect nature of a model as a simulator of environmental behaviour. Ideally, as has been stated, the potential for error in model predictions of interest should be reduced to their theoretical lower bounds through the model construction and calibration processes, these lower bounds being the inherent uncertainties of those predictions. Paradoxically, the role played by parameters of a necessarily defective model during the history-matching process is such that the presence of these defects may either help or hinder the attainment of this minimum. However the ability to quantify the potential for predictive error diminishes as the model's relationship with the reality that it purports to simulate becomes less physically-based, even though the potential for predictive error may actually be reduced. This, unfortunately, is the murky world in which we work as modellers.

It is worth repeating here a point that has been made earlier in this document, and which is salient to this chapter and to the next. It is this.

There should be no expectation that a model can provide a correct prediction. However one model, or modelling approach, can be assessed as technically superior to another when, for predictions of interest:

- *it can guarantee that correct predictions are within computed error limits; and*

- *these error limits approach the inherent uncertainty limits of these predictions, and hence approach optimality.*

From this it is apparent that quantification and minimization of predictive uncertainty/error limits should ideally be the focus of a modelling enterprise. Because of the numerical and practical difficulties associated with this endeavour, informal approaches must often be taken. One such approach - that of using a model as a basis for scientific hypothesis-testing – is examined in the following chapter. In this chapter we focus on post-calibration predictive error analysis, keeping in mind the desirability of reducing a model's propensity for predictive error to that of predictive uncertainty as conceptually provided by Bayes equation.

Constraints

The process of post-calibration predictive error analysis involves exploration of parameter variability subject to two constraints. These are that:

- the model “remains calibrated”; and
- that parameters “retain believability”.

Loss of calibration status and loss of parameter credibility will, of course, happen by degrees. A reduction in either of these engenders lower probabilities for predictions that are calculated on the basis of these parameters. The range of predictive values associated with finite probabilities defines a predictive confidence region. Ideally a number should be associated with this region. For example it would be nice to be able to say that “there is a 95% probability that the true value of the prediction lies between an upper limit of x and a lower limit of y ”. Unfortunately, given the imperfect nature of models, it will rarely be possible to say this. Sadly, there is a high degree of uncertainty associated with our assessment of uncertainty.

At the heart of the first term of equation (5.8) is the $C(\mathbf{k})$ matrix of innate parameter variability. At the heart of the second term of equation (5.8) is the $C(\epsilon)$ matrix of measurement noise. Each of these is associated with an explicit or implicit probability distribution (implicit in most modelling studies). These are the “reference distributions” through which diminishing parameter, and hence predictive, probability are assessed as parameters are varied from values that are deemed to “calibrate” the model to those that are required if a specified predictive value is to occur. That is to say, if a number is to be associated with any predictive confidence interval, this number must ultimately arise from these two probability distributions collectively, as these provide the metrics through which parameter credibility (or lack thereof) on the one hand, and acceptability of model-to-measurement misfit (or lack thereof) are assigned numbers from which confidence intervals can be calculated.

Unfortunately the probability distributions which $C(\mathbf{k})$ and $C(\epsilon)$ represent are often complex beyond measure. While simplified geostatistical models are often used to calculate $C(\mathbf{k})$, assumptions such as stationarity make little sense when applied over the disparate and interconnected but non-continuous, fractured and faulted geological materials that comprise the domain of any regional groundwater model, or the patchwork of changing land uses and heterogeneous soil types spread over the uneven topography that comprises the domain of any surface water or land systems model. The situation for $C(\epsilon)$ is no less complicated, for model-to-measurement misfit is an outcome of far more than measurement noise. It owes its origins to model structural defects that are as complex as the natural system of which the model is a necessarily imperfect simulator. The probability distribution of structural noise as

it affects different types of measurements made at different locations within a complex model domain is therefore virtually impossible to quantify, and is probably singular.

Nevertheless, some attempt must be made to limit the range of predictive possibilities to those that are compatible with expert knowledge as it resides in $C(\mathbf{k})$, and with information available to us through historical measurements of system state, the integrity of which is characterized by $C(\epsilon)$.

The situation is depicted in Figure 6.1, which builds on Figure 4.4. We do not know the reality parameter field \mathbf{k} . However we know something of its projection onto the parameter solution space. Our knowledge of this, however, is compromised by the fact that estimation of this projection takes place on the basis of a calibration dataset that is contaminated by measurement and structural noise. Hence there is some “wobble room” in our assessment of this projection, the size of this wobble room being set by the amount and nature of (measurement and structural) noise ϵ associated with this data, that is by $C(\epsilon)$. Any parameter set that projects onto that part of the solution space that is identified in this way as being feasible, and which can be assessed as having non-zero probability in terms $C(\mathbf{k})$, is a contender to be the real parameter field. It moves out of contention when either it is deemed to be unrealistic on the basis of $C(\mathbf{k})$, or is deemed to provide a misfit with the calibration dataset which cannot be explained by measurement/structural noise as characterized by $C(\epsilon)$.

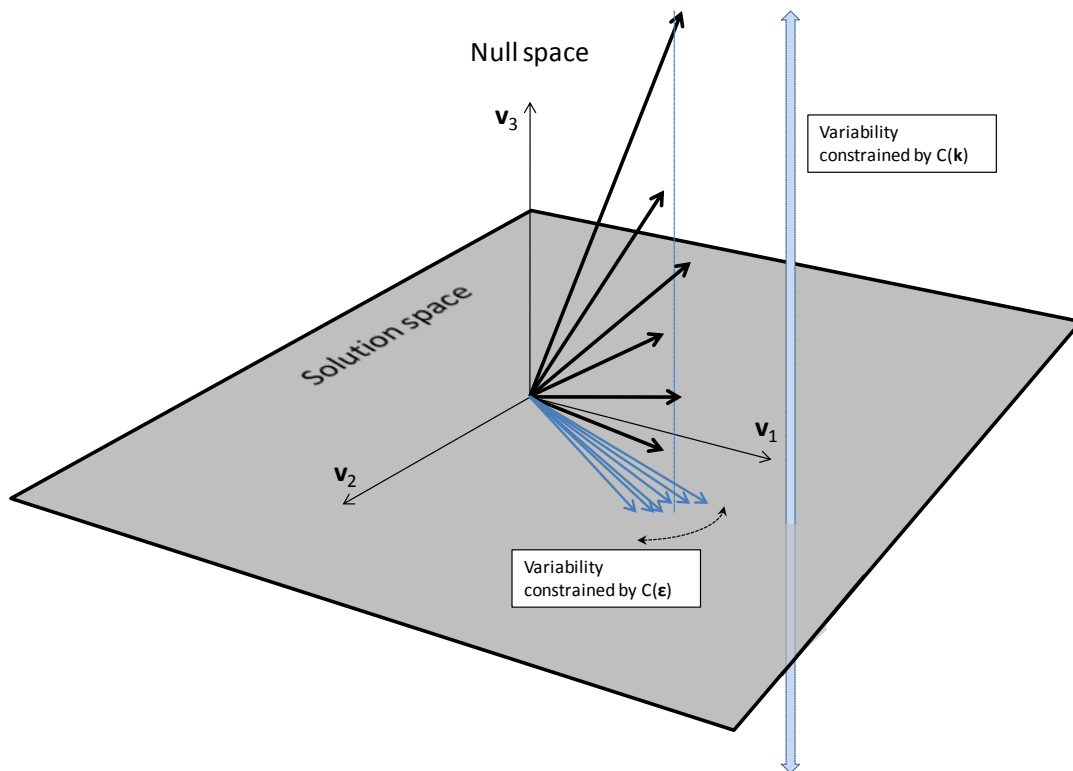


Figure 6.1. Post-calibration parameter variability. All of the parameter sets represented by dark arrows can legitimately be used by the model in exploring post-calibration predictive uncertainty as they are all realistic as assessed in terms of $C(\mathbf{k})$ and they all provide an adequate fit with the calibration dataset as assessed in terms of $C(\epsilon)$.

Well-Posed Inverse Problems

General

The surface water model calibration problem which comprises one of the exercises provided with this document constitutes an over-determined, or well-posed, inverse problem because all parameters are estimable on the basis of the calibration dataset. As has already been discussed, this is an outcome of the fact that regularisation is done:

- manually through estimating only a few of the many parameters offered by the HSPF model; and
- structurally through the fact that complex environmental process are simulated in a lumped and averaged way.

In cases like this only the second term of equation (5.8) is used to calculate post-calibration predictive variability as the first term is zero. Linear analysis based on this term was described in the previous chapter. Two methods that are available through the PEST suite for implementing over-determined nonlinear analysis are now described. A third will be discussed in the following chapter. It should be noted however that if model run times are small, well-posed inverse problems constitute useful candidates for Markov chain Monte Carlo analysis of parameter and predictive uncertainty. Software to implement this analysis is not presently available through the PEST suite.

Constrained Maximization/Minimization

The processes of determining the margin of predictive variability associated with a particular confidence level can be formulated as a constrained maximization/minimization problem. This problem is more easily formulated in the over-determined context than in the under-determined context as only one set of constraints must be applied to parameters in this context, namely those that pertain to model-to-measurement misfit. The methodology is described by Cooley and Vecchia (1987), Vecchia and Cooley (1987), Cooley (2004) and Christensen and Cooley (2006). It is implemented by PEST when run in “predictive analysis” mode.

Suppose that the objective function achieved through the calibration process is Φ_{\min} and that weights used in definition of the objective function are correct in a relative sense, in that they properly reflect the propensity of each measurement to be degraded by measurement error (and structural error to the extent that this is possible). Let s be a prediction of interest. Suppose that we want to determine the 95% confidence interval of this prediction. To do this we must carry out two optimization exercises. First we must maximize the prediction subject to the constraint that the objective function Φ rises no higher than a value which we denote as $\Phi_{95\%}$; then we must minimize the prediction subject to the same constraint.

More generally, suppose that we wish to determine the two-sided $1-\alpha$ confidence interval of a prediction. Then we must maximize and minimize that prediction subject to the constraint that the objective function rises no higher than Φ_0 , where Φ_0 is given by the first of the following two equations if the so-called simultaneous confidence interval of the prediction is explored, and by the second of the following two equations if the so-called “individual” confidence interval of the prediction is explored. The latter provides a narrower, and theoretically more correct statistical bound, though there is a certain level of approximation involved in its usage.

$$\Phi_0 = \Phi_{\min} \left[\frac{m}{n-m} F_{\alpha}(m, n-m) + 1 \right] \quad (6.1)$$

$$\Phi_0 = \Phi_{\min} \left[\frac{t_{\alpha/2}^2(n-m)}{(n-m)} + 1 \right] \quad (6.2)$$

In the first of these equations “ $F(m, n-m)$ ” refers to the F distribution with $(m, n-m)$ degrees of freedom; in the second equation $t(n-m)$ signifies a t distribution with $(n-m)$ degrees of freedom.

The constrained maximization/minimization procedure can include the effects of predictive noise if:

- the noise term is added to the prediction as part of the model;
- the noise term is treated as a parameter;
- the noise term is also treated as an observation whose “observed” value is zero and whose weight is the inverse of its uncertainty.

These are done internally by PEST if the user requests it (see the example at the end of this chapter). Meanwhile the objective function threshold associated with a given confidence interval is still given by (6.2) where the “individual” predictive confidence interval is sought. However equation (6.1) must be replaced by the following equation where the simultaneous predictive confidence interval is sought:

$$\Phi_0 = \Phi_{\min} \left[\frac{m+1}{n-m} F_{\alpha}(m+1, n-m) + 1 \right] \quad (6.3)$$

Conceptually, the constrained maximization/minimization process is an efficient method for determination of the confidence band of a prediction of interest. As Gallagher and Doherty (2006) demonstrate, where the uncertainty band of only a single prediction must be explored its model run requirements are much more parsimonious than those of Markov chain Monte Carlo analysis. However in real-world modelling its use incurs some difficulties. These include the following.

- As has already been discussed, the statistical structure of model-to-measurement misfit induced by model imperfections is unknown; furthermore the greater the extent to which simplifications have been introduced to a model to formulate a well-posed inverse problem, the greater is the magnitude of structural noise likely to be. Unfortunately the veracity of equations (6.1) to (6.3) is dependent on implementation of a weighting scheme which properly complements the statistical structure of measurement/structural noise in ways already described.
- Unless a prediction is very similar to measurements used in the calibration process, the stochastic character of the predictive noise term will not be known.
- Numerical performance of the constrained predictive maximization/minimization process degrades rapidly with the introduction of even a small amount of model numerical malperformance. Though the deleterious effects of model output granularity on calculation of finite-difference derivatives can be mitigated to some extent through use of appropriate PEST derivative control settings, and through implementation of a line search option as part of the constrained predictive maximization/minimization process, these measures all increase the run-time burden of this process considerably.

Notwithstanding these problems, this procedure can provide a means of rapid assessment of “predictive wiggle room” in many modelling contexts.

Calibration-Constrained Monte Carlo

Equation (5.11) provides the covariance matrix of post-calibration parameter error for a linear model. In an over-determined inversion context it also provides the covariance matrix of the posterior parameter probability distribution, as equation (5.12) shows. This matrix is calculated and recorded by PEST whenever it undertakes over-determined parameter estimation. Conceptually this matrix could be used as a basis for random parameter set generation. If predictive model runs were then undertaken using all such random parameter sets, the post-calibration error/uncertainty distribution of that prediction could thereby be explored. In many circumstances this would provide a more efficient means of post-calibration random parameter set generation for a linear model than that provided by Markov chain Monte Carlo as subsequent parameter sets could be very different from each other, and no parameter sets would be rejected (neither of which holds true when sampling the posterior parameter distribution using the Markov chain Monte Carlo methodology). Further model run efficiencies could be gained through using this matrix as a basis for implementation of targeted sampling methodologies such as Latin-hypercube.

Where a model is nonlinear, the covariance matrix of equations (5.11) and (5.12) does not provide a true descriptor of post-calibration parameter variability. Exploration of parameter and predictive uncertainty through random parameter set generation based on this matrix can only therefore be approximate at best. Nevertheless the integrity of such a sampling scheme can be improved if random parameter sets that are generated on the basis of this matrix are subjected to re-calibration. If the model is not too nonlinear, the computational effort required for adjustment of parameters in order to reduce model-to-measurement misfit to a suitable threshold, for example the objective function value described by equations (6.1) and (6.2), will not be large. In most cases this efficiency can be dramatically increased by re-use of a single Jacobian matrix when undertaking parameter adjustment, as is demonstrated in the worked example discussed later in this chapter.

Ill-Posed Inverse Problems

General

Ill-posed inverse problems are those for which parameters cannot be estimated uniquely because of the existence of a null space. While the existence of a null space may make the inverse problem of model calibration a little more difficult to solve than for a well-posed inverse problem, recognition of its presence is often essential to the integrity of model predictive uncertainty analysis.

In the past it has often been recommended that when calibrating an environmental model the principle of parameter parsimony should be respected. It was shown earlier in this document that this precept should not be an end in itself, but may be a logical outcome of the pursuit of a calibrated parameter field of minimum error variance. Parsimonization is an inherent part of the model calibration process. If calibration is based on highly-parameterized inversion, parsimonization is achieved mathematically (and hopefully optimally) as part of the inversion process itself.

Parsimonious parameterization of a model that simulates complex processes in a heterogeneous environment can constitute a profound obstacle to the integrity of predictive uncertainty analysis, however. If parameterization complexity cannot be estimated on the

basis of a given calibration dataset because of a paucity of information within this dataset, this does not make that complexity go away. In fact it makes the need for its inclusion in the uncertainty analysis process even stronger, for to the extent that a prediction is sensitive to parameters, or to parameter combinations, that are inestimable through the calibration process, the uncertainty of that prediction is not diminished through calibration. It is the author's experience that the uncertainty associated with many predictions in many contexts is dominated by the null-space term.

Where the predictions required of a model are similar to measurements used in calibration of that model, a simplified parameter set that emerges from manual and/or structural regularization may provide an adequate basis for post-calibration uncertainty analysis. By definition, in these circumstances the prediction is sensitive mainly to those parameters to which model outputs used in the calibration process are sensitive (provided that conditions that will prevail when a prediction is required are not too different from those which prevailed when calibration was effected). Furthermore, if parameter simplification engenders structural noise, the nature and extent of this noise as it affects model outcomes used for calibration (and hence for predictive) purposes can be determined. Alternatively, if model structural defects require that parameters assume compensatory roles to allow a good fit between model outputs and members of the calibration dataset to be obtained, this same compensatory role is likely to provide a beneficial effect on the ability of the model to make predictions of the same type at the same locations.

However many modelling contexts are very different from this. Models are often built precisely because conditions are about to be changed, or because predictions must be made of a quantity, or at a location, for which little historical hard information is available. It is precisely for this reason that a physically-based model is chosen for environmental simulation, and that a considerable level of numerical complexity may be devoted to the simulation of environmental processes encapsulated in the model. In fact an important design consideration for many models is that a process should not be excluded from the model if a prediction of interest may be sensitive to it. Given that the uncertainty associated with the making of that prediction will probably be high, the same logic must apply to the parameterisation that is associated with prediction-salient processes (and the heterogeneity thereof) if the uncertainty associated with the prediction is to be properly explored. As has been stated, this (and not an illusion of predictive certainty) must then form the basis for the making of decisions to which the prediction pertains.

Highly-parameterized, nonlinear, post-calibration predictive uncertainty analysis is therefore a topic that must be at the heart of modern model usage. Two methods are discussed below. The first is generally impractical, but is an extension of a methodology that was discussed above for use in the over-determined context. The second is more practical. It can (and has) been applied in modelling contexts of considerable parameter and process complexity. A third methodology will be discussed in the next chapter.

Constrained Predictive Maximization/Minimization

In principle, and in practice, PEST can be used in "predictive analysis" mode to undertake constrained predictive maximization/minimization in the highly-parameterized context. In doing this two constraints must be enforced. The first is on model-to-measurement misfit, while the second is on null-space projected parameter departures from their calibrated values. The PEST REGPRED utility (REGPRED stand for "regularized predictive uncertainty analysis) automates construction of a PEST input dataset that can be used to implement this process. Tonkin et al. (2007) discuss this methodology and demonstrate its use.

While of theoretical interest, this process is unlikely to find much application in real-world modelling applications for at least the following reasons.

- In spite of the fact that its efficiency can be increased through the use of “predictive super parameters” as Tonkin et al. (2007) describe, it is a model-run-intensive numerical procedure.
- As for its over-determined counterpart, the integrity and efficiency of the constrained maximization/minimization procedure is easily degraded where model numerical imperfections degrade the integrity of finite-difference derivatives calculations.
- Separate constrained maximization/minimization processes must be undertaken to obtain prediction values corresponding to different confidence levels. Thus the attainment of a relationship between prediction value and confidence level requires that an inordinately large number of model runs be carried out.

Null Space Monte Carlo

The null space Monte Carlo (NSMC) procedure is unique to PEST. It provides a mechanism for rapid generation of diverse parameter fields which satisfy both the model-to-measurement misfit and reality constraints required for exploration of post-calibration parameter uncertainty. By making a model prediction using many such parameter sets, the calibration-constrained variability of that prediction can be explored.

The method is not Bayesian, for it has its roots in equation (5.5) rather than in Bayes equation. However Bayesian analysis is very difficult to implement where models are highly parameterized, nonlinear and possess long run times. Strictly speaking, NSMC provides a methodology for exploration of post-calibration parameter and predictive error rather than of post-calibration parameter and predictive uncertainty. However the outcomes of such an analysis are not expected to be significantly different from the outcomes of Bayesian analysis, and can be acquired with considerably less numerical difficulty.

Another advantage of the NSMC method is that it can be easily adapted to a user’s computing circumstances. If certain compromises are made (these being explained below), the numerical efficiency of the method can be greatly increased. This may be necessary where model run times are high and/or where computing resources are limited. Though the need for such compromises may not always be welcomed, it must be remembered that compromise is a better alternative than doing nothing to explore post-calibration predictive uncertainty for, with the exception of the methodology that is discussed in the following chapter, there is simply no other practical methodology available for use in conjunction with highly parameterized models with long run times.

The NSMC process takes its inspiration from Figure 6.1. It attempts to generate many different parameter fields which have the same solution space projection as that of the parameter field which calibrates the model. (Ideally the latter should have no null space projection at all, for it attempts to provide the most simple means to achieve a desired level of model-to-measurement fit; any attempts to wander off the hyper plane that constitutes the solution space into the null space are likely to increase the potential for predictive error as there is no guarantee that such a journey into the null space is in the right direction.)

The NSMC process is implemented as follows.

- Random parameter fields \mathbf{k} are generated using the prior parameter probability distribution; the covariance matrix of the prior parameter probability distribution is, of course, $C(\mathbf{k})$.

- The calibration parameter field \mathbf{k} is subtracted from each of these random parameter fields to yield a random collection of $\mathbf{k}-\mathbf{k}$ parameter difference fields.
- In each case the difference field is projected onto the calibration null space.
- The projected difference field is then added back to the calibration parameter field. If the model were linear, the process would end here, as the new parameter field would be guaranteed to calibrate the model.
- The model is re-calibrated through adjustment of the new parameter field. However only solution space components of this field are adjusted. Hence re-calibration relies on adjustment of only a limited number of super parameters rather than all parameters used by the model. Further gains in efficiency are achieved through re-use of the same set of super parameter sensitivities for the first iteration of all random field re-calibration exercises. Note that while the need to re-calibrate in this fashion may be seen as an undesirable consequence of model non-linearity, it does provide an opportunity to introduce necessary variability to solution space parameter components.

The outcome of an NSMC exercise is a suite of parameter fields which can be used for the making of any prediction required of the model. The uncertainty associated with that prediction can thereby be assessed through construction of an empirical probability density function.

Means by which the efficiency of this process can be further increased (with some sacrifice to the integrity of that process) include the following.

- The random parameter field generation process based on $C(\mathbf{k})$ can be centred on the calibrated parameter field rather than on the pre-calibration expected parameter field.
- Parameter variability as encapsulated in $C(\mathbf{k})$ can be reduced through use of a surrogate $C(\mathbf{k})$ matrix with narrower probability intervals when generating random parameter fields. This reduces the extent to which different (null space projected) random parameter fields de-calibrate the model, and hence reduces the numerical effort required to achieve model re-calibration.
- The objective function threshold at which a model is deemed to be “recalibrated” can be made higher than that which would be considered to be statistically correct on the basis of the stochastic properties of measurement noise as encapsulated in $C(\epsilon)$. Given the unknown stochastic nature of structural noise which is, in most cases, the dominant contributor to calibration misfit, this is unlikely to introduce a significant loss of integrity to the NSMC process.

As Tonkin and Doherty (2009) explain, the NSMC process can be made even more sophisticated when implemented in conjunction with groundwater and other spatial models by allowing calibration-constrained random heterogeneity to be represented on a cell-by-cell or element-by-element basis. To accomplish this, the NSMC process must be slightly modified as follows.

- A spatial parameter field is generated using a cell-by-cell stochastic field generator such as the FIELDGEN utility supplied with the PEST Groundwater Data Utility suite.
- That field is sampled at a discrete number of points. These samples comprise a random parameter set for a model parameterization scheme based on pilot points.

- A smooth field is interpolated between the pilot points; the difference between the smoothed, interpolated field and the stochastic field is calculated.
- The null space projection operation discussed above is applied to the pilot point parameter field.
- When super-parameter re-calibration is applied to the pilot point parameters the difference field obtained as above is added to the parameter field obtained through spatial interpolation between pilot points.

See Tonkin and Doherty (2009) and the PPSAMP Groundwater Data Utility for further details.

Exercises

Surface Water Model

Objective Function

Our surface water model has been calibrated using two different formulations of the inverse problem, these leading to two different definitions of the objective function that was minimized through the calibration process. In the first case the objective function was composed of flows alone (actually the logarithms of flows). In the second case a multi-component objective function was employed. Design and weighting of this multicomponent objective function was somewhat heuristic. However use of a strategy such as this can often benefit the inverse problem solution process as it makes visible to the inversion process information within the calibration dataset that may otherwise be hidden. If properly done, it can also help mitigate contamination of the inversion process by structural noise. In the previous chapter, linear analysis was employed to obtain some estimate of the uncertainty associated with maximum predicted flow over the period 19th to 23rd August 1986 based on this formulation of the inverse problem.

In the present chapter we will change our tact slightly. In fact we will implement uncertainty analysis “by the book”, partly to demonstrate how this is done, and partly to demonstrate that while there is some comfort to be gained in seeking refuge in a safe theoretical harbour, this may not provide a sound basis for parameter and predictive uncertainty analysis where theory cannot accommodate the structural weaknesses of a model.

We will therefore take as our starting point the PEST control file *calib1_soln.pst* which contains best-fit parameters achieved during a calibration process formulated on the basis of a single objective function comprised of log-transformed flows. As the PEST run record file *calib1_soln.rec* shows, the minimized objective function achieved during this calibration exercise was 8.042.

Constrained Maximization/Minimization

The PEST control file *calib3a.pst* is similar to *calib1_soln.pst* in that it employs the same initial parameter values as *calib1_soln.pst*. However it differs from *calib1_soln.pst* in that PEST is asked to run in “predictive analysis” mode. In doing so it is asked to minimize the value of prediction *qmax_sim_max*, this being the maximum flow over the period 19th August to 23rd August 1986 (which actually occurs on 20th August). This prediction is assigned to the observation group “predict” which identifies it as the quantity to be maximized or minimized during the constrained optimisation process that is undertaken by PEST when run in “predictive analysis” mode.

Further inspection of file *calib3a.pst* reveals that it contains a “predictive analysis” section. The NPREDMAXMIN variable on the first line of this section is set to -1, indicating that minimization, rather than maximization, of the prediction is required. The objective function constraint Φ_0 (PEST variable PD0) is set to 8.19053 in accordance with equation (6.2) calculated for a two-sided confidence interval of 99%. This objective function is obviously only very slightly higher than the minimum obtained during the calibration process. It shows how little wiggle room parameters are granted where the calibration dataset is large and there is no null space. This is almost certain to result in an underestimation of parameter (and hence predictive) uncertainty. This is an outcome of the fact that the highly correlated nature of structural noise is ignored in setting the objective function limit in this manner; hence the assumed information content of the calibration dataset is artificially inflated.

Note also that the “singular value decomposition” section of the PEST control file has been removed, this being a PEST requirement when it is run in “predictive analysis” mode. In the “predictive analysis” section, PEST is asked to enhance the performance of the constrained minimization process by implementing a line search as part of this process; see the third to fifth variables on the third line of this section. This will consume model runs, but will enhance the performance of the optimisation process.

Run PEST using the command:

```
pest calib3a
```

An inspection of screen output and the run record file reveals that PEST is able to lower the predicted flow for 20th August 1986 from an initial value of 412.5 m³/sec (this being calculated using initial parameter values) to a value of 331.9 m³/sec. This is an impressive fall for such a tight objective function constraint. However while a high degree of predictive uncertainty has been exposed through this process, it not enough to encompass the observed flow on that day.

Taking Predictive Noise into Account

The PEST control file *calib3b.pst* is similar to *calib3a.pst*. However it differs from it in the following important respects.

- The PREDNOISE variable in the “predictive analysis” section of the PEST control file is set to 1; this follows NPREDMAXMIN as the second variable on the line immediately following the “predictive analysis” section header. Hence PEST will add predictive noise to the model output that is identified as “the prediction” on each occasion that it runs the model.
- A weight of 1.0 is provided to the *qmax_sim_max* “observation” which is, in fact, the prediction that is subject to constrained minimization. This informs PEST that the predictive noise that is added to this prediction has the same variance as that associated with model outputs used in calibrating the model.
- PEST runs a model named *model_calib1a.bat* instead of *model_calib1.bat*. In this batch file, TSROC is provided with an input file named *tsproc_calib1a.dat*. In this TSPROC input file the *log* of the model output that is identified as “the prediction”, rather than the prediction itself, is calculated. This is because it is to the *log* of the prediction rather than to the native prediction that predictive noise must be added because it was with respect to the *log* of flows rather than native flows that the noise associated with model outputs/field measurements was assessed from model-to-measurement misfit arising from the calibration process. Note that there is no need to

change the “observed value” of this prediction in the “observation data” section of the PEST control file, however, as this “observed value” is ignored by PEST.

Run PEST by typing the command:

```
pest calib3b
```

PEST lowers the prediction to 2.21998. This is equivalent to a flow of 165.95m³/sec. While still not encompassing the correct flow, the substantial broadening of the predictive uncertainty margin is pleasing to see.

Post-Calibration Parameter Covariance Matrix

The PEST control file *calib3c.pst* is identical to *calib1_soln.pst* except for the fact that the “singular value decomposition” section is omitted from this file. Hence upon termination of execution, PEST will print out the post-calibration parameter covariance matrix; this is the matrix featured in both of equations (5.11) and (5.12). NOPTMAX is set to -1 in this PEST control file so that PEST will calculate a Jacobian matrix, calculate statistics on the basis of this matrix (including the post-calibration covariance matrix) and then cease execution.

Actually the Jacobian matrix is already available in file *calib1_soln.jco*; recall that we previously ran PEST specifically to obtain this matrix. So we will save PEST the trouble of having to re-run the model once for each adjustable parameter in order to calculate it by finite differences, by providing this matrix to PEST ourselves. This is accomplished by starting PEST with the “/i” switch - thereby instructing PEST to read this matrix from an existing *.jco file.

Run PEST using the command:

```
pest calib3c /i
```

Respond to the ensuing prompt as follows.

```
Enter name of JCO file for first iteration sensitivities: calib1_soln.jco
```

Upon commencement of execution, PEST runs the model once to calculate the objective function. Then it reads *calib1_soln.jco* to obtain the pre-calculated Jacobian matrix. It then computes parameter statistics before ceasing execution. The post-calibration parameter covariance matrix is available in both the run record file *calib3c.rec* and the “matrix file” *calib3c.mtt*.

The post-calibration parameter covariance matrix can be re-written in PEST matrix file format using the PCOV2MAT utility. (PCOV2MAT stands for “PEST covariance matrix to matrix file format.”) Run PCOV2MAT as follows.

```
pcov2mat calib3c.mtt postcalparcov.mat
```

A file named *postcalparcov.unc* has been supplied. This is a “parameter uncertainty file”; we have seen these before. *postcalparcov.unc* instructs any program which reads it to look to the covariance matrix housed in file *postcalparcovf.mat* that we have just written for a stochastic descriptor of parameter variability. This will now be used as a basis for random number generation.

Obtaining Calibration Constrained Random Parameter Sets

We will now use the RANDPAR utility to generate random parameter sets on the basis of the covariance matrix contained in file *postcalparcov.mat*. These parameter sets will be centred on the calibrated parameter set. Hence, if the model were linear, these random parameter sets should constitute samples of the posterior parameter probability distribution. As such they

could be used as a basis for exploring the uncertainty of any model prediction. This would be accomplished by running the model repeatedly using one of these random parameter sets on each occasion, collecting results, and building an empirical predictive probability distribution on the basis of these model run outcomes.

Run RANDPAR, responding to its prompts as follows:

Enter name of existing PEST control file: **calib3c.pst**

- 10 parameters read from file calib3c.pst.
- 7 of these are adjustable.

Use (log)normal or (log)uniform distrib for param generation? [n/u]: **n**

Compute means as existing param values or range midpoints? [e/m]: **e**

Respect parameter ranges? [y/n]: **y**

Enter name of parameter uncertainty file: **postcalparcov.unc**

- reading covariance matrix file postcalparcov.mat...
- covariance matrix file postcalparcov.mat read ok.
- parameter uncertainty file postcalparcov.unc read ok.

Enter name of parameter ordering file (<Enter> if none): **<Enter>**

Enter filename base for parameter value files: **postcal_random**

How many of these files do you wish to generate? **100**

Enter integer random number seed (<Enter> if default): **<Enter>**

RANDPAR generates 100 sets of parameters, storing these in files named *postcal_randomN.par* where *N* ranges from 1 to 100.

Given that our model is in fact nonlinear, we may ask “how well do these parameter fields respect calibration constraints”? This question can be answered by running PEST 100 times, with NOPTMAX set to zero each time, so that PEST simply calculates the objective function on each occasion that it runs. Ideally, all of these objective functions should be reasonably close to the minimized objective function obtained during the previous calibration process. (Actually they should respect the statistical constraints set by equations (6.1) and 6.2)). A batch file which implements this procedure is *postcal_runs.bat*. It is reproduced below.

```

rem #####
rem Delete an existing record file.
rem #####

del /P record_postcal.dat
echo > record_postcal.dat
pause

rem #####
rem Do all the PEST runs.
rem #####

for /L %%i in (1,1,100) do (
del temp.pst
del temp.rec
parrep postcal_random%%i.par calib3d.pst temp.pst
pest temp
echo ' ' >> record_postcal.dat
echo  PARAMETER SET %%i >> record_postcal.dat
find /I "ie phi" temp.rec >> record_postcal.dat
)

```

Figure 6.2 The batch file *postcal_runs.bat*.

Note the following features of this batch file.

- The PEST control file *calib3d.pst* is identical to *calib3c.pst* except for the fact that NOPTMAX is set to zero in this file.
- On each traversal of the processing loop contained within the batch file *postcal_runs.bat*, the PARREP utility is used to create a reproduction of the PEST control file *calib3d.pst* which contains each of the random parameter sets in turn; this file is named *temp.pst* on each occasion. Thus on each traversal through the loop PEST is asked to run the model only once, calculate the objective function based on pertinent model outputs, and then cease execution.
- After each PEST run, the run record file *temp.rec* is searched for the string “i.e. phi”. The whole line containing this string is then appended to a file named *record_postcal.dat*. This string actually occurs twice within the run record file. On each occasion it is followed by the objective function. Thus when the 100 PEST runs are complete, file *record_postcal.dat* will contain a record of objective functions calculated on the basis of the random parameter sets stored in files *postcal_random*.par*.
- File *record_postcal.dat* is deleted before anything is appended to it (with the user’s permission). This avoids confusion as it prevents the outcomes of the present batch process from being appended to an existing file of the same name.

Run the batch process *postcal_runs.bat* by typing its name at the screen prompt. When the process is complete, inspect file *record_postcal.dat*. It appears that most of the parameter sets respect calibration constraints pretty well, as the objective function is not far above the minimized value of 8.0424 in all cases.

Better Enforcement of Calibration Constraints

The list of objective function values provided in file *record_postcal.dat* indicates a high degree of conformance to calibration constraints by the parameter sets contained in files *postcal_random*.par*. Although the objective functions are not as low as the value of

8.19053 provided by equation (6.2), in a real-world setting we would normally be happy with these objective functions as their extra variability provides room for the effects of structural noise. Nevertheless, we will now demonstrate a process through which these random parameter sets can undergo adjustment in order to enforce greater respect for calibration constraints. The outcome of this process will be another set of parameter value files. We will then make a model prediction on the basis of each member of this set, thereby exploring post-calibration predictive uncertainty.

Inspect file *calib3e.pst*. This is almost identical to *calib3d.pst*. However there are some important differences. These are now outlined.

- NOPTMAX is set to 1 in file *calib3e.pst*. Thus PEST will devote one optimization iteration to improving parameter values.
- A value of 8.3 has been provided for the optional PHISTOPTHRESH variable. This is the seventh variable on the 9th line of file *calib3e.pst*. PEST will not attempt to lower the objective function below this. Thus if the objective function calculated on the basis of initial parameter values is already lower than this, PEST will not undertake even one optimisation iteration. This is still a little higher than the value of 8.19053 provided by equation (6.2). However it is selected in subjective recognition of the contribution that structural noise makes to model-to-measurement misfit under calibration conditions.

The batch file *postcal_recalib.bat* supervises re-adjustment of the random parameter sets recorded in the *postcal_random*.par* parameter value files. It is reproduced below.

```
rem #####
rem Delete an existing record file.
rem #####

del /P record_recalib.dat
echo > record_recalib.dat
pause

rem #####
rem Do all the PEST runs.
rem #####

for /L %%i in (1,1,100) do (
del temp.pst
del temp.rec
parrep postcal_random%%i.par calib3e.pst temp.pst
pest temp /i < response_calib3e.in
echo ' ' >> record_recalib.dat
echo PARAMETER SET %%i >> record_recalib.dat
find /I "ie phi" temp.rec >> record_recalib.dat
find /I "total model calls: " temp.rec >> record_recalib.dat
copy temp.par postcal_random_adjusted%%i.par
)
```

Figure 6.3 The batch file *postcal_recalib.bat*.

The following aspects of file *postcal_recalib.bat* are noteworthy.

- Random parameter sets contained in files *postcal_random*.par* are PARREPEd into the PEST control file *calib3e.pst* to produce a new PEST control file named *temp.pst* on each traversal of the processing loop contained in this batch file. On the basis of

temp.pst, PEST undertakes parameter re-adjustment for a maximum of 1 optimisation iteration as already discussed.

- PEST is run using the “/i” switch. An inspection of the re-directed keyboard input file *response_calib3e.in* supplied to PEST on each occasion that it runs, reveals that PEST is told to read its Jacobian matrix from file *calib1_soln.jco*. Because it is run with the “/i” switch, PEST does NOT undertake a model run for each parameter in calculating a finite-difference Jacobian. It simply reads the pre-recorded Jacobian matrix instead. Hence the computational burden of parameter re-adjustment is very light.
- Adjusted parameters (if indeed adjustment is required) are recorded in parameter value files *postcal_random_adjusted*.par*.
- The number of model runs that PEST undertakes on each occasion of its execution is recorded in file *record_recalib.dat*. So too are the initial and final objective function values (these being outcomes of two *finds* of the “i.e. phi” string).

Run the *postcal_recalib.bat* batch process by typing its name at the screen prompt. When this process is complete, the resulting parameter sets will reside in a suite of parameter value files named *postcal_random_adjusted*.par*. Note that all of these parameter sets could be collected into a single tabular data file using the MULPARTAB utility supplied with PEST if desired.

Exploring Predictive Uncertainty

The batch file *postcal_adjusted_runs.bat* is similar to *precal_runs.bat* which we used in Chapter 3 to explore pre-calibration predictive uncertainty. It runs the model 100 times - once for each of the parameter sets recorded in the parameter value file sequence *postcal_random_adjusted*.par* which we have just generated. Model outcomes corresponding to these random parameter sets are recorded in a series of PEST residuals files named *temp.res.**. Run this batch process by typing:

```
postcal_adjusted_runs.bat
```

at the screen prompt. Finally collect model predictions into a single table using the RDMULRES utility as was done previously when we undertook pre-calibration Monte Carlo analysis. Run RDMULRES using the following command:

```
rdmulres rdmulres_postcal_adjusted.in
```

The RDMULRES input file *rdmulres_postcal_adjusted.in* instructs RDMULRES to accumulate predicted flows in file *rdmulres_postcal_adjusted.out*. A frequency histogram based on these flows is provided in Figure 6.4.

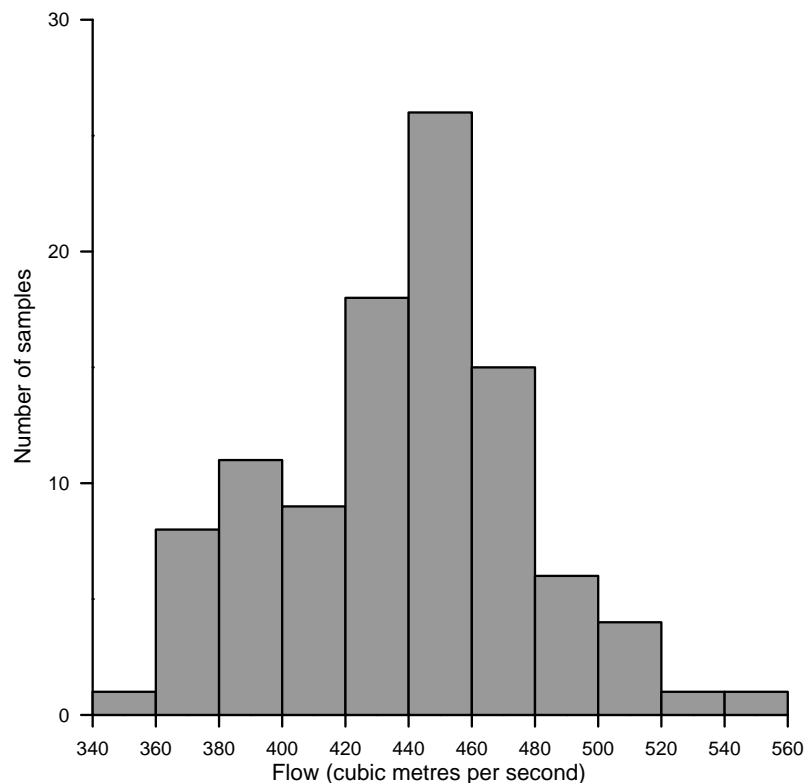


Figure 6.4 Post-calibration predicted maximum flow over the period 19th to 23rd August, 1986. No predictive noise has been added to model-predicted flow.

Not even the lowest flow depicted in Figure 6.4 approaches the observed flow on 20th August 1986. The smaller-than-expected width of the probability distribution depicted in Figure 6.4 is partly an outcome of the fact that a mere 100 samples provides an insufficient basis for identification of predictive extremes. However a bigger problem is that the above analysis did not take account of predictive noise. Hence the predictive variability depicted in Figure 6.4 is a function only of post-calibration parameter variability.

This defect in our analysis could be rectified in a number of ways. Firstly the level of noise that is likely to be associated with the prediction must be evaluated. This could be done (as was done implicitly above when using PEST's predictive analyser) by equating it to the noise encountered during the calibration process. This noise pertains to the log of flows rather than to flows themselves - partly because noise applied to flows log-transformed in this way is more likely to be homoscedastic than noise associated with native flows; hence (theoretically) it can be used in the future as well as in the past to characterise structural noise. This noise could then be formally included in Monte Carlo analysis as another random parameter, this being added to the log of predicted flow on each occasion that the model is run. As it happens we know this noise. In the PEST run record file *calib1_soln.rec* the "standard error of weighted residuals" is recorded as 0.1497. As all logged flows used in the calibration process were provided with a weight of 1.0, this defines the noise associated with logged flows. This equates to a multiplier/divisor of 1.411 (obtained as 10 raised to the power 0.1497). If this is applied to the lowest predicted flow of 358.6 m³/sec obtained in the above Monte Carlo analysis (see file *rdmulres_postcal_adjusted.out*), a flow of about 254 m³/sec is obtained. This is still higher than that obtained using the constrained minimization process to explore the range of predictive uncertainty, this probably being attributable to insufficiency of sampling.

Alternatively, a more heuristic approach could be taken. As Gallagher and Doherty (2007b) show, structural noise is most unlikely to be homoscedastic under any kind of transformation. Instead it is likely to show event-similarity. A modeller may therefore inspect model-to-measurement misfit over the calibration period as it pertains to peaks following dry periods, and thereby obtain an idea of an appropriate range of “correction multipliers” to apply to these types of flows. These could be included either formally (as a random variable in Monte Carlo analysis) or informally (as a worst case correction term as was done above) in calculation of the predictive uncertainty range.

The Groundwater Model

General

As has already been noted, calibration of the groundwater model used in our example constitutes an ill-posed inverse problem. As 104 parameters are estimated on the basis of only 12 observations, the dimensionality of the null space must be at least 92. Recall that the SUPCALC utility estimated the dimensionality of the solution space to be 12 (this being equal to its recommended number of super parameters). It follows that its estimated null space dimensionality is 92. (On most occasions the dimensionality of the null space will be far greater than the observation-to-parameter deficit, this being its theoretical lower limit).

As was stated above, use of constrained maximization/minimization for exploration of post-calibration predictive uncertainty in under-determined calibration contexts is problematical. Hence that methodology will not be used here. Instead the null space Monte Carlo (NSMC) methodology will be employed.

The NSMC methodology is flexible enough to allow hydraulic property heterogeneity right down to the cell-by-cell level to be included in calibration-constrained parameter fields. This will not be demonstrated here; the reader is referred to PEST documentation, and to documentation of the PEST Groundwater Data Utilities (particularly the PPSAMP utility) for further details. See also Tonkin and Doherty (2009) and Herckenrath et al. (2010) for a discussion of this issue. In the example presented below, heterogeneity will be accommodated at the pilot point level, this being the level at which parameters are estimated.

PEST Control and Jacobian Matrix Files

The NSMC process begins with a calibrated model. Our groundwater model has already been calibrated; optimised parameters are recorded as initial parameters in file *calib1r_soln.pst*. A corresponding Jacobian matrix file named *calib1r_soln.jco* was produced on the basis of these calibrated parameters in order to underpin the linear uncertainty analysis that was undertaken in the previous chapter.

Our first task is to build a new PEST control file in which the same initial (and therefore optimized) parameter values are recorded, but from which regularization constraints are removed. Our task in random field generation is, after all, to introduce (reality-constrained) variability rather than to suppress it (as is the role of Tikhonov regularization).

File *calib3.pst.kp1* is a copy of *calib1r_soln.pst* with the appropriate adjustments made. To obtain a Jacobian matrix corresponding to this file we could set NOPTMAX to -2 in its “control data” section and then run PEST. However in order to save model runs, we will use the JCO2JCO utility to build a Jacobian matrix file corresponding to this PEST control file, for sensitivities of the same model outputs with respect to the same parameters have already been calculated and are recorded in file *calib1r_soln.jco*.

First copy *calib3.pst.kp1* to *calib3.pst* using the command:

```
copy calib3.pst.kp1 calib3.pst
```

so that its extension is in accordance with PEST control file protocol. (If your computer tells you that *calib3.pst* already exists, overwrite it.) Then run JCO2JCO as follows:

```
jco2jco calib1r_soln calib3
```

to build the Jacobian matrix file *calib3.jco*.

Random Parameter Set Generation

The RANDPAR utility will now be used to generate random parameter sets centred on the initial values contained in file *calib3.pst*. Run RANDPAR, responding to its prompts as follows:

```
Enter name of existing PEST control file: calib3.pst
- 104 parameters read from file calib3.pst.
- 104 of these are adjustable.

Use (log)normal or (log)uniform distrib for param generation? [n/u]: n
Compute means as existing param values or range midpoints? [e/m]: e
Respect parameter ranges? [y/n]: y

Enter name of parameter uncertainty file: param.unc
- reading covariance matrix file hk_cov.mat...
- covariance matrix file hk_cov.mat read ok.
- parameter uncertainty file param.unc read ok.

Enter name of parameter ordering file (<Enter> if none): <Enter>

Enter filename base for parameter value files: random
How many of these files do you wish to generate? 100

Enter integer random number seed (<Enter> if default): <Enter>
```

Random parameter sets generated by RANDPAR are stored in the parameter value files *randomN.par* where *N* ranges from 1 to 100.

Null Space Projection

The random parameter sets that we have just generated pay no respect to calibration constraints. This will now be rectified by restricting parameter field departures from calibrated values to the null space. This is accomplished using the PNULPAR utility. Run PNULPAR, responding to its prompts as follows.

```
Enter name of PEST control file: calib3.pst
Does PEST control file contain calibrated parameter values? [y/n]: y

Enter number of dimensions of calibration solution space: 12
Would you like to store Q(1/2)X matrix in matrix file format? [y/n]: n

Enter filename base of existing parameter value files: random
Enter filename base for new parameter value files: ns_random
```

The outcomes of PNULPAR's deliberations are 100 parameter sets stored in the parameter value files *ns_randomN.par* where *N* ranges from 1 to 100. If the model were linear, all of these parameter fields would calibrate it; as it is not, they need adjustment. However only solution space components of parameter sets need to be adjusted to achieve this. Hence we

will build a PEST input dataset based on 12 super parameters; then we will automate adjustment of the 100 parameter sets stored in files *ns_random*.par* through a batch process.

Before doing this however, a few moments will be taken to explore the efficacy of null space projection.

Issue the following command.

```
parrep random1.par calib3.pst temp.pst
```

Through this command we build a PEST control file named *temp.pst* whose initial values are the random parameter values contained in the parameter value file *random1.par*. Now edit *temp.pst*, setting NOPTMAX to zero in this file (first number on the ninth line). Then run PEST (with input file *temp.pst*) to calculate the objective function. It is 42120; the model is very “uncalibrated”.

Now let us calculate the objective function using the null-space projected counterpart to the parameter set contained in file *random1.par*. Issue the command:

```
parrep ns_random1.par calib3.pst temp.pst
```

Set NOPTMAX to zero in *temp.pst* and run PEST. The objective function is 56.4. The benefits of null-space projection in achieving an “almost calibrated” random parameter field are readily apparent.

Termination Criteria

In a similar fashion to what was done in Chapter 4, the SVDAPREP utility will now be used to build a PEST input dataset in which only super parameters are adjusted. But before doing this, new termination criteria will be introduced to file *calib3.pst*. These will then be transferred to the new super parameter PEST control file that SVDAPREP writes.

Inspect file *calib3.pst*. It should contain the following termination criteria. If it does not, then edit them to ensure that it does.

- NOPTMAX (the first variable on the 9th line of *calib3.pst*) should be set to 2. Thus PEST is instructed to carry out, at most, two optimisation iterations. The first of these is undertaken at virtually no numerical cost, as PEST automatically calculates super parameter sensitivities from base parameter sensitivities in the first iteration of SVD-assisted parameter estimation. Hence a numerical burden is incurred only if PEST must undertake more than one optimisation iteration in order to effect parameter improvement. However as this is being done using super parameters, only 12 model runs will be required for filling of the Jacobian matrix.
- The PHISTOPTHRESH variable (seventh variable on the 9th line of the PEST control file) is set to 12.1. PEST will consider the model calibrated, and therefore cease execution, if the objective function falls below this value. If the objective function corresponding to initial parameter values is at or below PHISTOPTHRESH, no optimization iterations at all will be carried out.

SVD-Assist

Now run SVDAPREP, respond to its prompts as follows.

```
Enter name of existing PEST control file: calib3.pst
Use pre-defined super-parameter file? [y/n] (<Enter> if "no"): <Enter>
For computation of super parameters:-
  if SVD on Q^(1/2)X          - enter 1
  if SVD on XtQX              - enter 2
```

```

    if LSQR without orthogonalisation - enter 3
    if LSQR with orthogonalisation    - enter 4
Enter your choice (<Enter> if 1): 1
Enter number of super parameters to estimate: 12
Enter name for new super pest control file: calib3_svda.pst
Enter offset for super parameters (<Enter> if 10): <Enter>
Enter value for RELPARMAX (<Enter> if 0.1): <Enter>
Write multiple BPA, JCO, REI, none [b/j/r/n] files (<Enter> if "n")? <Enter>
Parameter scale adjustment [SVDA_SCALADJ] setting (<Enter> if 2): <Enter>
Make new model batch file silent or verbose? [s/v] (<Enter> if "s") : <Enter>
Automatic calc. of 1st itn. super param. derivs? [y/n] (<Enter> if "y") : <Enter>

```

As instructed, SVDAPREP writes a PEST control file named *calib3_svda.pst*.

Now, before doing anything else, make a copy of the base parameter PEST control file *calib3.pst*. Copy this to file *calib3.pst.kp1*. The reason for doing this will become apparent in a moment. (As *calib3.pst.kp1* was provided, you don't really have to do this.)

The Batch Process

Inspect file *adjust_parameters.bat*. It is reproduced below.

```

rem #####
rem Delete an existing record file.
rem #####

del /P record.dat
echo > record.dat
pause

rem #####
rem Do all the PEST runs.
rem #####

for /L %i in (1,1,100) do (
parrep ns_random%i.par calib3.pst.kp1 calib3.pst
pest calib3_svda
find /I "ie phi" calib3_svda.rec >> record.dat
copy calib3.bpa calib3.bpa.%i)

```

Figure 6.5 The batch file *adjust_parameters.bat*.

As in many of our previous batch files, we make provision for keeping a record of initial and final objective function values through use of the “find” command, with screen output from this command being re-directed to file *record.dat*. This file is deleted at the beginning of the batch process to ensure that these outcomes are not appended to an existing file of the same name, thereby creating confusion.

A series of commands contained within a loop is repeated 100 times, once for each null-space-projected random parameter set. First the parameter set is emplaced in a new *calib3.pst* file as initial values in this file. This is accomplished using the PARREP utility which uses file *calib3.pst.kp1* (copied from the original *calib3.pst*) as its base. Next parameter estimation is undertaken on the basis of file *calib3_svda.pst*. When writing this file SVDAPREP instructed it to look to *calib3.pst* for its initial parameter values; hence it inherits null-space-projected random parameter values as initial values in the parameter adjustment process. However, because the *calib3_svda.pst* PEST control file employs super parameters, only solution space parameter components are adjusted through the calibration process (and only 12 of these).

Once calibration is complete (either because 2 optimisation iterations have been completed or because the objective function has fallen below 12.1) the optimized parameter values stored

in file *calib3.bpa* are copied for safekeeping to another parameter value file named *calib3.bpa.N* where *N* is the index of the current parameter set. (Note that when PEST undertakes SVD-assisted parameter estimation the parameter value file that contains optimized base parameter values is named after the base parameter PEST control file, rather than the super parameter PEST control file, and has an extension of “.bpa”.)

To implement the above process, type:

```
adjust_parameters.bat
```

at the command-line. Watch the screen as this process progresses. You will note that rarely does PEST need to undertake a second optimisation iteration; hence the process is very fast. In fact there are many occasions when not even one optimisation iteration is required, as null-space projection has resulted in a random parameter field which already calibrates the model.

Exploring Predictive Uncertainty

An inspection of *record.dat* reveals that all of the parameter fields contained in files *calib3.bpa.** yield an objective function less than 12.1. So there is no need to winnow any of them out because of failure to satisfy calibration constraints. (This could be achieved using the COMFILNME utility supplied with PEST.)

Inspect the *postcal_runs.bat* batch file. This instructs the model to run 100 times, using a different adjusted-null-space-projected random parameter field on each occasion. The model is run through the agency of PEST so that the PARREP command can be used to provide parameter values for each run. At the end of each run, the model prediction is copied to file *timeN.dat* where *N* is the parameter field index. Note that *calib4.pst* featured in this file is identical to *calib3.pst.kp1* except for the fact that NOPTMAX is set to zero, thereby instructing PEST to undertake only one model run.

Initiate the batch process by typing the command:

```
postcal_runs.bat
```

at the command-line prompt.

Now use the RDMULRES utility to collect all the predictions based on different calibration-constrained random parameter sets into a single file, as we did previously when examining pre-calibration uncertainty. An appropriate RDMULRES input file named *rdmulres_postcal.in* has been prepared. Through this file, RDMULRES is asked to read all of the *time*.dat* files produced as an outcome of the previous set of model runs. It is asked to place them into a file named *rdmulres_postcal.out*.

Run RDMULRES using the command:

```
rdmulres rdmulres_postcal.in
```

If the predictions contained in file *rdmulres_postcal.out* are plotted as a histogram, the following graph is obtained.

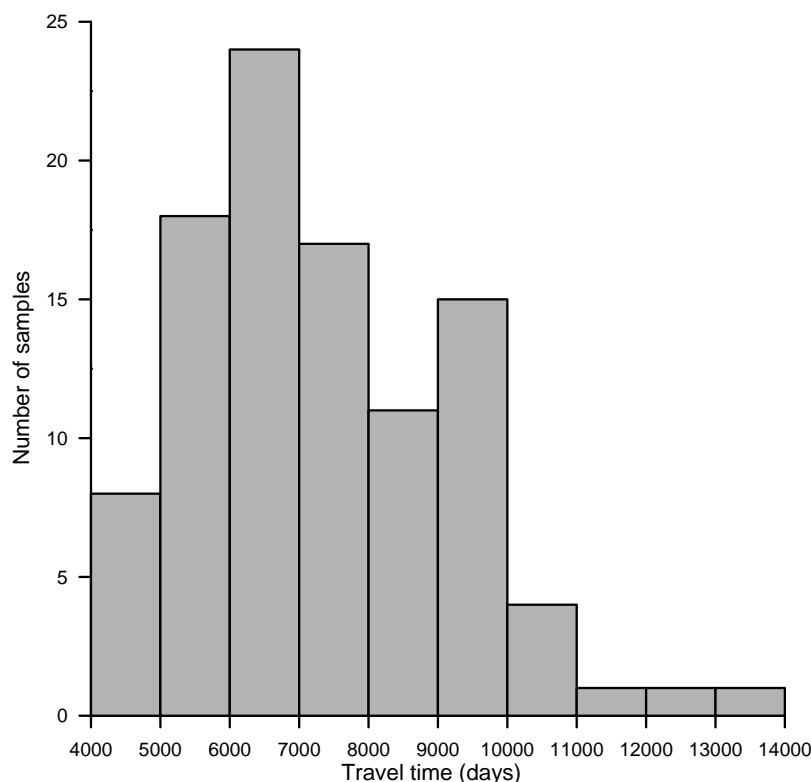


Figure 6.6 Post-calibration uncertainty histogram of predicted travel time.

The minimum prediction listed in file *rdmulres_postcal.out* is 4414 days. This is considerably higher than the true value of the prediction, this being 3256 days. As has already been stated, this example was chosen specifically to be hard; the correct travel time is indeed at the extreme end of the predictive confidence interval. If 2000 parameter sets, rather than a mere 100 parameter sets, were used in the above analysis, this extreme value would have been approached. However another problem with the analysis is that it fails to take account of the effects of cell-by-cell hydraulic conductivity variability on the travel time of the particle. This could be accounted for either through addition of a small, random “predictive noise” term to the prediction, or through using null-space-projection of FIELDGEN-generated parameter fields as a basis for null space Monte Carlo analysis. See documentation of the PPSAMP Groundwater Data Utility for further details.

7. Hypothesis-Testing and Pareto Methods

Where are we at?

Conceptually, Bayes equation provides a mechanism for synthesis of information contained in expert knowledge and in measurements of system state. Unfortunately, problems arise in applying Bayes equation in the environmental modelling context. These include the following.

- Numerically, working directly with probability distributions is difficult, unless models are linear and/or probability distributions are amenable to simple analytical description.
- Encapsulation of expert knowledge in a prior probability distribution is often a difficult matter. In the environmental modelling context expert knowledge of complex geological, environmental, land use and climatic systems is often slight. Furthermore, the variables that describe the nature of spatial and temporal heterogeneity as it applies to influential system properties are categorical rather than continuous, and mathematical simplifications such as stationarity or homoscedasticity are inappropriate.
- Numerical models are defective simulators of environmental behaviour. This has a number of repercussions. It gives rise to significant structural noise under both calibration and predictive conditions. The stochastic character of this noise (which is needed for definition of the likelihood function in Bayes equation) is unknown. Many parameters that are adjusted through the history-matching process must assume, to at least some degree, surrogate roles to compensate for a model's inadequacies in simulating past system behaviour. Depending on the type of prediction required of a model, these roles may either enhance or detract from its ability to make those predictions. In either case, the link between parameter optimality and predictive optimality is broken.

Analysis of predictive uncertainty in the environmental modelling context will therefore be compromised. In fact quantitative exploration of uncertainty may be impossible, as many aspects of the analysis must be heuristic. This does not detract from the importance of assessing model predictive uncertainty. As has already been stated, such an assessment is fundamental to the making of important decisions - decisions that cannot be avoided and that must be made with as high a level of scientific integrity as possible. However it does raise the question of how to approach this matter and whether, given its necessarily subjective, but nevertheless numerically intensive nature, direct application of Bayes equation is the best way to go about it.

A distinction has been made in this document between analysis of the potential for predictive error and analysis of predictive uncertainty. Error is what we, as modellers, carry; our goal is to reduce the potential for error associated with a given prediction to its theoretical lower limit, this being the inherent uncertainty of that prediction given the information that is presently at hand.

Working with error rather than uncertainty allows us many conveniences that can help overcome the problems associated with direct application of Bayes equation. In particular, use of Bayes equation to estimate predictive uncertainty is replaced by a two step process of model calibration followed by post-calibration predictive error analysis. Despite its

convenience, this approach still leaves us with two major problems. As equation (5.8) demonstrates, formal assessment of model predictive error still requires that the user provide assessments of the stochastic character of pre-calibration parameter uncertainty, and of the stochastic character of measurement/structural noise. In equation (5.8) these are represented by the $C(\mathbf{k})$ and $C(\epsilon)$ covariance matrices respectively.

The significance of model structural defects should not be underestimated. Defects arise because no model can provide perfect simulation of all aspects of environmental behaviour that are salient to a particular environmental outcome that we would like to predict. Their existence is not always a cause for concern however, as construction of the “perfect model” may not be a fruitful pursuit anyway. As discussed earlier in this document, while a “perfect model” may, in theory, provide the best mathematical repository for expert knowledge, it may make a very poor tool for extracting vital information from historical measurements of system state. To the extent that a prediction resembles historical measurements of system state, the importance of these measurements to the making of that prediction is increased. While a simplified model may provide receptacles for this kind of information that are only loosely linked to nameable system properties, at least these receptacles are accessible, thanks to manageable model run times and elimination of problematical numerical behaviour. However as model abstraction increases with increased model simplification, and as parameters increasingly assume roles that compensate for model inadequacies, their capacity to act as receptacles for information arising from expert knowledge is diminished. In many circumstances this may be a small price to pay for greatly enhanced predictive ability. In other circumstances, particularly those where predictions are of very different types, or must be made under very different circumstances, from those comprising the calibration dataset, even a small level of abstraction may incur significant and unquantifiable model predictive error.

Where do we go from here?

The above brief analysis suggests that there is no single path forward. Nevertheless the purpose of this chapter is to explain a methodology that, in the author’s opinion, has the potential to provide a useful basis for model usage in many decision-making frameworks. Underpinning its use are a number of assumptions that can be summarized as follows.

- Models that are used as a basis for environmental decision-making are deployed in environments where data have been gathered for many years. History-matching is thus an important part of the model development process.
- The capacity exists to build a model that, though not providing an exact replica of system processes, is nevertheless physically-based to a reasonably high degree.
- Nevertheless, the model that is ultimately used as a basis for environmental management in any particular context will have many numerical imperfections. Some of these imperfections will arise because of the necessity to make assumptions pertaining to elements of the system that are imperfectly known, for example the disposition of geological layering and the nature of historical system stresses. Others will arise from model simplifications that are adopted to forestall numerical instability and/or excessive run times.
- Though a considerable amount of expert knowledge exists over a study area at which an environmental model is employed, this knowledge is frustratingly inadequate - especially as it pertains to the degree and nature of heterogeneity that prevails in

subsurface hydraulic properties, the nature and magnitude of historical system stresses, the spatial and temporal variability of present and historical land uses, etc.

Despite the difficulties that they present, these aspects of real-world environmental modelling do not erode the potential for numerical simulation to provide a sound basis for decision support. However its ability to achieve this potential will depend on the manner in which it is employed, and on the philosophical underpinnings of its use.

The Scientific Method

The basis of the so-called “scientific method” (whose rigorous exposition is credited to the great philosopher of science Karl Popper) is this: an hypothesis is proposed; that hypothesis is then tested using an appropriate experiment. On the basis of outcomes of that experiment it may be possible to invalidate the hypothesis by demonstrating its inconsistency with data gathered and processed through the experiment. If the hypothesis cannot be invalidated, then it remains viable. It can never be validated however. Nevertheless it may achieve something approaching this status as competing hypotheses are successively invalidated through clever, targeted and incisive experiments that are designed to do so.

As discussed earlier in this document, environmental decision-making is often based on avoidance of an unwanted event. Its unwanted status may emerge from the high monetary, social or environmental costs associated with its occurrence. Environmental managers are then charged with preventing its occurrence.

The occurrence of an unwanted event can be considered a scientific hypothesis. Even after management practices are proposed whose intent is to prevent occurrence of the event, the hypothesis that it can nevertheless occur maintains its status until it is invalidated through scientific inquiry. The purpose of a modelling exercise in a management setting that is marked by avoidance of this event must be to provide a basis for rejection of the hypothesis that the event can occur if a certain management option is taken. This is achieved through processing all available information with the model - both expert knowledge and “hard” data arising from measurements of system state. If such processing leads to rejection of the hypothesis that the unwanted event can occur despite the adoption of a proposed management practice, then model-based environmental data analysis has provided support to the decision-making process that this process requires.

In practice, the situation may not be quite as clear-cut as just outlined. For example the hypothesis to be rejected may need to be modified to that of “event occurrence without early enough warning to take preventative action”. Alternatively, while it may not be possible to eliminate the possibility that an untoward event can occur, it may be possible to ascribe such a low probability to its occurrence that society is willing to take the risk in order to receive the benefits of a proposed development that may have created the need to explore the hypothesis that a bad thing may happen in the first place. Notwithstanding these variations, the central premise remains. A modelling exercise should constitute an incisive numerical experiment that attempts to process data optimally in relation to a particular end - this being rejection of the hypothesis that an unwanted event will occur if a certain management practice is adopted.

So on what grounds can a hypothesis be rejected? It can be rejected if its occurrence is incompatible with all available information. That information is composed of expert knowledge, measurements of system properties, and measurements of system state. Expert knowledge and direct measurements of system properties constitute the prior information term of Bayes equation and the $C(\mathbf{k})$ matrix of innate parameter variability. Measurements of system state constitute the calibration dataset. The errors associated with these measurements,

together with model imperfections that impede flow of information from these measurements, form the basis for calculation of the likelihood term of Bayes equation and the $C(\epsilon)$ covariance matrix of measurement/structural noise.

These concepts provide the philosophical basis for deployment of simulation technology as a scientific instrument through which modellers (as scientists) may be able to differentiate between events which can happen and those that cannot. This means of model usage can be summarized as follows.

A model is deployed specifically to test the hypothesis that an unwanted event will occur. This is done by including system states corresponding to that event in the model's calibration dataset, along with historical measurements of system state. The model is then calibrated against this composite dataset. The hypothesis that the proposed state can eventuate can be rejected if:

- *the model cannot replicate the occurrence of the event in the future while simultaneously respecting historical system behaviour;*
- *the model can accommodate the simultaneous occurrence of the proposed and historical system states only through use of parameters that are unrealistic.*

Failure to simulate historical conditions is assessed in terms of $C(\epsilon)$. Unacceptability of parameter fields is assessed in terms of $C(k)$.

The two probability distributions that these covariance matrices represent are (as always) of pivotal importance. However their utility in the decision-making context is enhanced if they are used in a way that is slightly different from that embodied in Bayes equation, or even in the error analysis equations presented in previous sections, as the imperfect nature of numerical models as simulators of environmental behaviour can then be better taken into account.

The Role of Model Calibration

Two calibration contexts are thus proposed. In one of these (which may in fact comprise many calibration exercises) the model is used in an hypothesis-testing capacity in order to assess the likelihood or otherwise of an unwanted future event, and to develop management plans which can forestall the occurrence of that event. However prior to this, the model is calibrated against historical data alone, this constituting model calibration in its traditional sense. The present subsection examines what traditional model calibration should hope to achieve in light of its role as a precursor to model calibration against a dataset which includes an hypothesized event in order to assess the likelihood or otherwise of that event's occurrence.

Calibration against an historical dataset is often a process of compromise, and one that involves a high degree of subjectivity. Where a highly-parameterized approach is taken to model calibration, the process of compromise can be given a more scientific foundation, and can assist in creation of a "sharper instrument" through which hypotheses pertaining to future events can subsequently be tested.

Ideally the degree of parameter complexity ascribed to a model should be commensurate with its process complexity. Presumably the decision to include representation of certain processes within an overall environmental simulation exercise is based on the fact that the model would lose its relevance to decision-making if these processes were omitted. The same logic dictates that if these processes are affected by heterogeneity of system properties that govern them, then this heterogeneity may have an effect on model outcomes on which decisions may rest.

Failure to represent a propensity for system property heterogeneity in a model-based hypothesis-testing procedure may result in the drawing of erroneous conclusions regarding the possibility or otherwise of hypothesised events.

A problem with the inclusion of many parameters however is that their use can result in “over-fitting” of the model to historical system behaviour. In contrast, use of too few parameters can result in failure to sufficiently fit historical data - and therefore failure to extract from it all of the information that resides in it. Ideally, the intelligent use of mathematical regularization should provide the perfect compromise between these two extremes. Thus a model is endowed with parameterization density that is commensurate with the sensitivity of key model outcomes to system property heterogeneity; regularisation ensures that only as much heterogeneity is actually introduced to the model domain as can be supported by the data. Meanwhile, the calibration null space is sufficiently well populated for the range of possibilities associated with a particular model outcome to be assessed, with due recognition paid to the inability of the calibration process to constrain some (or perhaps many) of these outcomes.

In practice, the determination of an appropriate level of regularisation to employ, even when regularisation is mathematically implemented, is often a matter of subjectivity. If regularisation is applied too strongly, model-to-measurement misfit is increased while the emergence of parameter heterogeneity within the model domain is suppressed. On many occasions a modeller will chose to do exactly this if he/she judges that emergent heterogeneity indicates the adoption by model parameters of surrogate roles that provide compensation for model inadequacies. Violation of an explicit or implicit $C(\mathbf{k})$ matrix of acceptable parameter variability provides the modeller with the evidence that he/she needs to identify this occurrence, and hence to take remedial action through imposition of stronger regularisation constraints. Through doing this, he/she denies him/herself as good a fit between model outcomes and historical measurements of system state as he/she would otherwise like. Thus respect for $C(\mathbf{k})$ takes precedence over respect for $C(\epsilon)$, as a greater level of model-to-measurement misfit is tolerated than that which would be expected on the basis of measurement noise alone.

On the other hand, a modeller may have a high degree of confidence that his/her model provides accurate simulation of system behaviour. Furthermore, as is often the case, he/she may have little idea of the innate variability of system properties, and of the propensity for local heterogeneity to prevail within certain parts of the model domain. Model calibration may provide a great deal of information in this regard, especially if it is implemented using a strategy based on highly parameterized inversion that provides the calibration process with the flexibility to introduce complexity to the model domain if and where it is needed. In a case such as this, respect for $C(\epsilon)$ may take precedence over respect for $C(\mathbf{k})$, as a modeller may be loath to reject information pertaining to the existence of system property heterogeneity by ascribing the emergence of such heterogeneity to structural noise.

There is thus a tension between $C(\mathbf{k})$ and $C(\epsilon)$. The highly-parameterized model calibration process plays one against the other as model-to-measurement misfit is traded off against parameter field heterogeneity. This should not be construed as a disadvantage of the highly parameterized approach to model calibration. Rather it is a distinct advantage because it endows the modeller with the ability to apply his/her subjective judgment in a manner that is unimpeded by the necessity to employ parsimonious parameterization schemes in order to create a well-posed inverse problem where none actually exists, this being a requirement of older calibration methodologies.

It is thus apparent that an important outcome of the calibration process (an outcome whose importance has been insufficiently recognized to date) is a subjective reconciliation by the modeller of $C(\mathbf{k})$ with $C(\epsilon)$. However this reconciliation is often implicit rather than explicit because rarely, in real-world modelling practices, is either of these matrices defined (or even needs to be defined). Rather, these matrices are implicit in the heterogeneity that a modeller is prepared to accept within the model domain on the one hand, and in the model-to-measurement misfit that he/she is prepared to tolerate on the other hand. As we have seen, whether implicit or explicit, both of these matrices are crucial to the assessment of model predictive uncertainty, and hence to the hypothesis-testing procedure that constitutes the next stage of model deployment.

It is thus apparent that the task of a properly orchestrated process of model calibration against historical measurements of system state is to provide a platform for the hypothesis-testing that will take place thereafter which, in turn, provides a platform for model-based decision-making. The process of calibrating a model against a historical dataset thus has three important outcomes. These are:

- a parameter field that can be considered to approach that of minimum error variance;
- through this parameter field, an assessment of the degree of system property variability that may prevail within a model domain, this defining an implicit $C(\mathbf{k})$ matrix;
- an assessment of the degree of model-to-measurement misfit that accompanies simulation of environmental processes within a particular study area, this defining an implicit $C(\epsilon)$ matrix.

When hypotheses pertaining to future system behaviour are tested, these will be rejected if simulation of their occurrence requires too great a departure from either the parameter field that emerged during the calibration process, or from the fit with historical data that was achieved during that process, or both. “Too great a departure” will probably be the outcome of subjective assessment. However it will rest heavily on what was learned through the calibration process - this including the estimated parameter field itself, the tolerable level of heterogeneity that may exist in this field, and the tolerable inability of the model to exactly replicate past system behaviour. Introduction of excessive heterogeneity or excessive misfit as a necessary condition for allowing the model to replicate an hypothesised future event provides a basis for deeming that event to be of low likelihood. It is through calibrating the model against historical data alone as a precursor to testing hypotheses of future system behaviour that “excessive” now has a metric, albeit probably a subjective one.

It is important to note that recognition of model structural defects is an implicit part of the calibration and hypothesis-testing processes as thus described. Model-to-measurement misfit will almost certainly be greater than measurement noise. Parameter variability as it is introduced to the calibrated parameter field, may be greater than that which a modeller would ascribe to parameters based on expert knowledge alone. The former recognises the existence of structural noise; the second recognizes the fact that parameters may need to assume unusual values that compensate for a model’s defects and that this may actually enhance a model’s ability to simulate both past and future system behaviour. Both of these phenomena are unavoidable. Both of them operate under both calibration and predictive conditions. Their recognition and accommodation are an essential aspect of model deployment, and for identification of future events as unlikely or otherwise.

Pareto Concepts - Model Calibration

The previous discussion shows that in calibrating a model against an historical system dataset there are too competing objectives. These objectives can be formally encapsulated in two different objective functions, as is done when implementing Tikhonov regularisation. One of these is the so-called “regularisation objective function”. This is zero when parameter values perfectly respect their pre-calibration preferred values or preferred relationships (depending on the way in which the modeller chooses to express his/her preferred parameter condition). The other is the so-called “measurement objective function”. This is zero when a perfect fit is obtained between field measurements and their model-generated counterparts. Between these two extremes lies a subjectively chosen optimal calibration outcome - this defining the parameter field which is accepted as giving rise to predictions of minimized error variance. At the same time the metrics for acceptability of model-to-measurement misfit and for acceptable parameter field variability are defined. Both of these metrics will be applied during future model-based hypothesis-testing.

Whenever two or more objectives compete, a curve such as that shown in Figure 7.1 can be drawn.

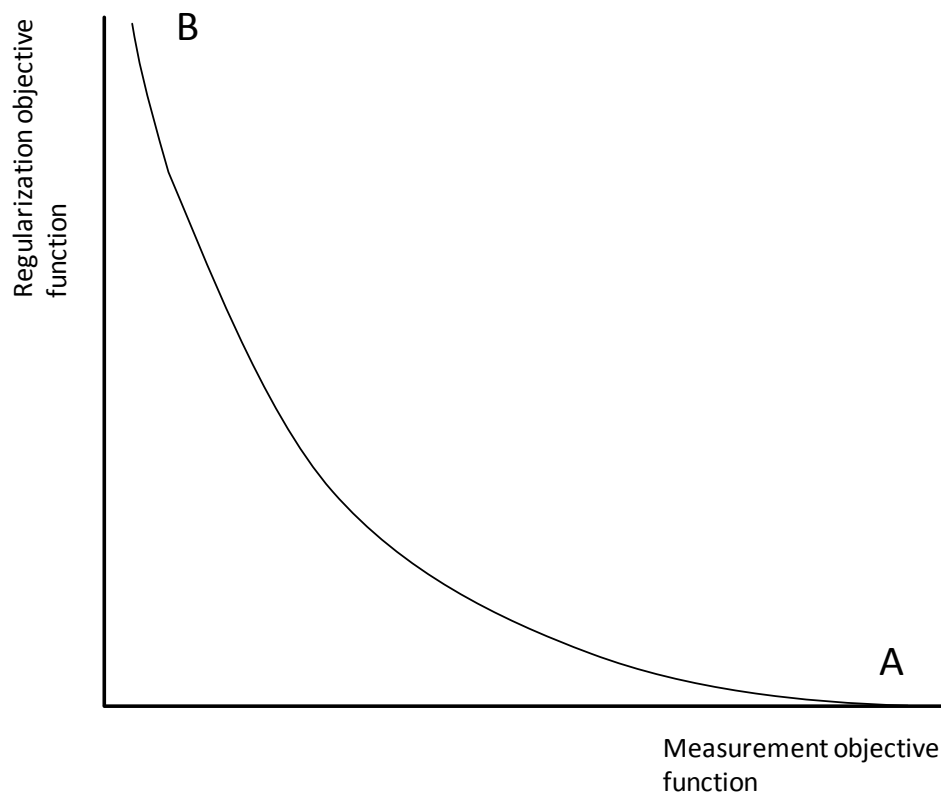


Figure 7.1 The Pareto front as it applies to the model calibration process.

Conceptually, any point to the right of the curve shown in Figure 7.1 is feasible. Parameters can readily be generated that introduce too much system property variability into a model domain at the same time as they provide a poor fit to the calibration dataset. Both the measurement and regularization objective functions associated with such a parameter set will therefore be high.

Ideally the point A is unique. The regularization objective function is zero, implying total respect for regularization constraints. The parameter field that corresponds to point A should therefore be that of minimum pre-calibration error variance. However model-to-measurement

misfit associated with this parameter field may be high, and with it the measurement objective function. At point B the opposite occurs; model-to-measurement fit is as good as can be attained using the current model. It is probable that a multiplicity of parameter sets, all of equal likelihood from a pre-calibration point of view, can provide this level of fit; hence point B is not associated with a unique set of parameter values. But even if it were, we would not be too interested in these values because their departure from preferred pre-calibration values is probably too great, rendering them unrealistic from an expert knowledge point of view.

The curve joining points A and B in Figure 7.1 is referred to as the “Pareto front” if it defines the locus of points in objective function space (and implicitly in parameter space) for which it is not possible to improve both objective function components simultaneously. Hence a better fit with the calibration dataset can only be achieved through lowering parameter pre-calibration likelihood, and vice versa. As a direct consequence of this definition, points cannot exist to the left of the Pareto front. Hence it defines a barrier in objective function space that cannot be crossed.

The Pareto front can also be viewed as implicitly defining the locus of solutions to a set of constrained optimization problems. For any measurement objective function, a unique point on the Pareto front can be selected. The regularization objective function pertaining to that point is the lowest that can be achieved while maintaining the chosen measurement objective function. The parameter set corresponding to that point therefore defines the solution to the same constrained minimization problem as that which is sought through solution of the inverse problem of model calibration through application of Tikhonov regularisation.

When PEST is run in “regularisation” mode, the user must select a target measurement objective function in advance of the Tikhonov solution process. If he/she does not like the outcomes of the regularised inversion process because either the parameter field is too lumpy, or because the fit between model outcomes and historical measurements of system state is not good enough, the process must be repeated using a different target measurement objective function. Eventually a measurement objective function is selected that represents the best compromise between excessive parameter field lumpiness and excessive model-to-measurement misfit.

Conceptually, the process of choosing an optimum point of compromise between goodness of fit and parameter field lumpiness is the process of travelling along the Pareto front. It follows that if software can be designed to traverse this front, recording parameter sets as it does so, this should provide the modeller with all of the information that he/she needs to choose the optimal point of compromise that constitutes the outcomes of a properly-conducted calibration process. Furthermore, if passage along the front can be continuous rather than discrete, the modeller is provided with maximum flexibility in choosing this point. When run in “Pareto” mode, PEST attempts to provide this outcome.

When using PEST’s Pareto capabilities to calibrate a model, measurement and regularisation objective functions are defined in the usual way. PEST starts with a parameter set for which the regularisation objective function is zero. Weights applied to observations comprising the calibration dataset are then slowly increased. During the ensuing sequence of optimisation iterations PEST crawls along the Pareto front, with the measurement objective function slowly decreasing and the regularisation objective function slowly increasing as it does so. Meanwhile the user inspects the changing nature of model-to-measurement misfit on the one hand, and of parameter field variability on the other hand. Eventually he/she selects a point along the front that he/she deems to express the best compromise between the two. In doing so, as stated above, he/she explicitly chooses the parameter field of minimum error variance.

At the same time he/she implicitly selects metrics through which departures from this field will be judged when subsequently testing an hypothesised prediction, both in terms of what extra model-to-measurement misfit he/she will tolerate, and what extra level of heterogeneity he/she is willing to endure in attaining this prediction.

Pareto Concepts - Model Prediction

Not only the process of model calibration, but the process of model-based hypothesis-testing, can be formulated as that of seeking solutions to a series of constrained minimization problems. Formulation of the hypothesis-testing process in this way can take place either formally or informally. In either case, through traversal of a Pareto front, an optimal outcome of the hypothesis-testing process can be obtained, possibly aided by a high degree of subjective judgement based on information that becomes available to a modeller through traversal of the Pareto front.

Suppose that a model is being used to explore the possibility that an untoward event will occur. Let that event be associated with a value of χ_0 for a certain model outcome. This outcome is then “observed” to occur; hence it can be included in the objective function along with other observations that comprise the calibration dataset and still other observations and/or prior information equations that encapsulate the preferred system condition as applied through regularisation constraints. Let the hypothesised prediction be assigned to its own observation group, with its own objective function component. This is calculated as $w^2(\chi_m - \chi_0)^2$ where w is the weight associated with the prediction and χ_m is the model-calculated prediction. This objective function component will diminish as the prediction is approached. Meanwhile the other component of the objective function (comprising measurement and regularization constraints) will increase as calibration model-to-measurement misfit and/or departures of parameters from their calibrated values increase. At some point these departures may be considered to be too unlikely for the corresponding value of the prediction to have reasonable likelihood. Let the value of the model prediction corresponding to this point be denoted as χ'_m . Hypotheses that predictive outcomes are closer to χ_0 than χ'_m can then be rejected. As stated earlier, this point may be formally selected on the basis of known or assumed $C(\mathbf{k})$ and $C(\epsilon)$ matrices. In most cases however, it will be informally selected. In all cases, the previous calibration process, in which model outcomes are matched against historical measurements of system state alone, will have played a large part in determining the explicit or implicit $C(\mathbf{k})$ and $C(\epsilon)$ matrices through which predictive credibility (or lack thereof) is assessed.

Figure 7.2 depicts the Pareto front that is applicable to this aspect of model usage.

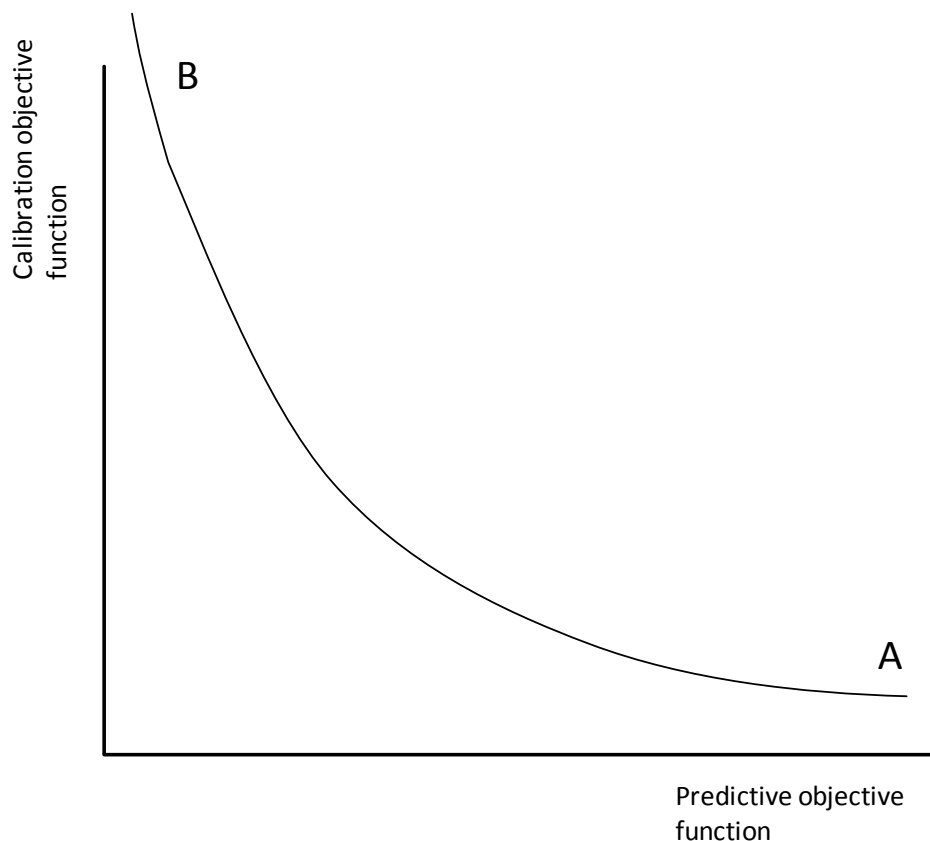


Figure 7.2 The Pareto front as it applies to model-based hypothesis-testing.

Once again, traversal of the Pareto front commences at point A. At this point the “calibration objective function” is that achieved through calibration. Meanwhile the calibration dataset has a new member - this being the prediction whose value is hypothesized. At point A this has a weight of zero. Slowly it is given a greater weight so that traversal of the Pareto front can occur. For a given calibration objective function, each point along the Pareto front represents the minimized prediction objective function for which that calibration objective function is capable of being attained. As such, it represents the maximum or minimum value of the prediction subject to the constraint that the calibration objective function is no higher than that which corresponds to that position on the Pareto front. Hence traversal of the Pareto front constitutes solution of a series of constrained maximization/minimization problems that are exactly equivalent to those solved by PEST when it runs in “predictive analysis” mode. However because a series of problems is solved rather than an individual one, the modeller is able to associate likelihood or otherwise with a series of predictive outcomes instead of just one. Furthermore he/she is able to make subjective decisions pertaining to likelihood if he/she judges (as he/she mostly will) that exact mathematical characterization of pre-calibration parameter variability through an explicit $C(\mathbf{k})$, and measurement/structural noise through an explicit $C(\epsilon)$ is impossible.

Variations of the above theme are possible. For example the calibration objective function could be formulated in terms of *differences* between parameter fields and model outcomes necessary to achieve a certain prediction and those which were achieved at calibration. The initial calibration objective function would therefore be zero. If desired, parameter departures from their calibrated status could be decomposed into solution space and null space components, with likelihood of the latter formally assessed using a null-space projected $C(\mathbf{k})$ matrix. Meanwhile, solution space parameter components would be limited by constraints on

model-to-measurement misfit that are incurred as a predictive model outcome approaches its hypothesised value. Such an exploration of predictive variability would be more in keeping with an analysis of potential for predictive *error* rather than with exploration of predictive *uncertainty*, and has its roots in equation (5.8). PEST setup for this exercise could be aided through use of the REGPRED and OBSREP utilities; see PEST documentation for further details.

Pareto Methods - Some Final Words

To date, practical experience in using PEST's "pareto" mode in real-world settings has proved very rewarding. The following points are noteworthy.

- When using PEST's "pareto" mode to apply Tikhonov constraints, it has been found that it is generally possible to attain a lower regularisation objective function corresponding to a given measurement objective function than that which is attainable when PEST is run in "regularisation" mode. Normally the outcome is a smoother parameter field. It seems that the process of moving slowly along the Pareto front provides a stronger defence against the appearance of unnecessary heterogeneity than that provided by direct solution of a constrained optimization problem pertaining to a given target measurement objective function.
- The ability of Pareto constraintment of parameters to subjugate introduction of spurious heterogeneity is diminished somewhat when SVD-assisted parameter estimation takes place, this arising from the fact that the combinations of parameters that emerge from definition of super parameters may not necessarily be those required to preserve maximum parameter field smoothness.
- When using the Pareto method to explore predictive possibilities that are compatible with an historical calibration dataset, it is sometimes found that the Pareto curve does not resemble that depicted in Figure 7.2. Instead, the curve may rise for a while from point A and then settle into a new minimum to the left of point A in that figure. Not only does this indicate significant nonlinearity of model predictive behaviour; it also appears to indicate the existence of at least two separate system "states" whose likelihood is difficult to separate on the basis of currently available information.
- Ideally, model parameters should change in a continuous fashion as the Pareto front is traversed. However it is often found that discontinuous parameter changes are encountered as traversal of the Pareto front causes parameters to cease congregating about one local objective function minimum and to start congregating about another. See Moore et al. (2010) and the following exercise for further details.

Exercises

Surface Water Model

The Hypothesised Prediction

We will continue to examine as our prediction of interest the maximum flow over the period 19th August to 23rd August 1986 (which actually occurs on 20th August). We will "hypothesize" that this maximum flow is low; in doing so, we will "observe" its value to be 20 m³/sec. The calibration dataset is thus supplemented with this "observation". Note that in all PEST runs undertaken in previous exercises this prediction has been included in the calibration dataset. However up until now it has been given a weight of 0.0; furthermore its "observed" value has been ignored.

The PEST Control File

To maintain compatibility with the analysis of predictive uncertainty that was undertaken in the previous chapter, and to maintain compatibility with the example presented in Moore et al. (2010), the calibration objective function will be computed on the basis of log-transformed flows alone. Thus we will not be employing other objective function components such as baseflow, monthly volumes and exceedence times that were used in a previous calibration exercise.

File *calib4.pst* forms the basis of PEST usage in the current case. The following aspects of this file are salient.

- The PESTMODE variable (the second item on line 3 of file *calib4.pst*) is set to “pareto”.
- The measured value of the *qmax_sim_max* “observation” is set to 20 m³/sec. This observation is assigned a weight of 1.0.
- A “pareto” section appears at the end of the PEST control file.

The “pareto” section of file *calib4.pst* specifies that variable weighting be applied to the *predict* observation group (the sole member of which is the *qmax_sim_max* “observation”). The initial weight applied to this group is zero. However this weight will be increased in 30 increments to a maximum value of 0.8; see descriptions of the PARETO_WTFAC_START PARETO_WTFAC_FIN and NUM_WTFAC_INC variables in PEST Pareto documentation. In practice, some trial and error may be required in selecting these settings. Exploration of the Pareto front is more detailed than necessary in this exercise; as will be seen however, we will be rewarded for our strong numerical commitment to this problem through having some of its difficulties exposed.

It is also specified in the “pareto” section of the PEST control file that the *qmax_sim_max* observation be monitored. Hence values computed for this particular model output will be recorded on the *.*pod* and *.*ppd* files written by PEST when it runs in “pareto” mode.

Running PEST

Run PEST using the command:

```
pest calib4
```

While PEST is running you may wish to monitor its traversal of the Pareto front by opening file *calib4.pod*. The objective function corresponding to each observation group is recorded in this file. In this case there are only two observation groups. As the *predict* observation group is given greater and greater weight (this happening behind the scenes) to promulgate movement along the Pareto front, the value of the prediction diminishes as time goes on. This value is recorded in the third column of file *calib4.pod*.

The Pareto Front

Data recorded in file *calib4.pod* is plotted in Figure 7.3. In this plot the value of the prediction (rather than the predictive objective function) is plotted against the value of the calibration objective function. The former is described by the vertical axis whereas the latter is depicted on the horizontal axis. The disposition of axes in Figure 7.3 is such that traversal of the Pareto front starts at the top left of the graph and continues down and to the right.

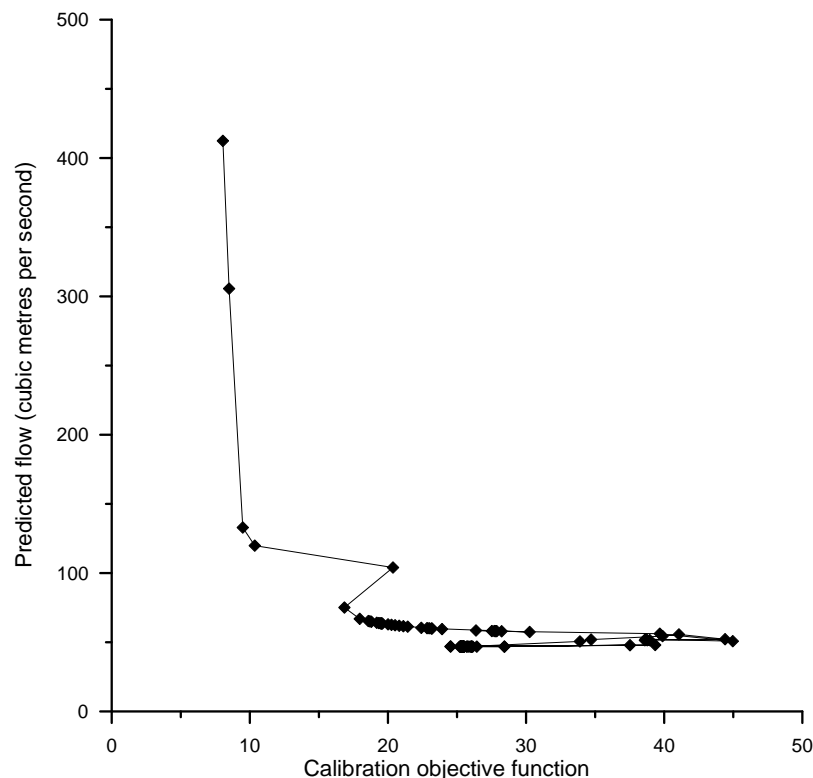


Figure 7.3 The Pareto front: predicted flow vs. calibration objective function.

A number of interesting points emerge from an inspection of Figure 7.3 and of data contained in file *calib4.pod*. These include the following.

- For very little change to the calibration objective function, the predicted maximum flow changes by over 250 m³/sec. This makes the prediction very uncertain indeed. It appears to have greater uncertainty than has been exposed in explorations of predictive uncertainty undertaken in previous exercises.
- In its traversal of what appears to be a discontinuous Pareto front, PEST actually takes two detours, one at a calibration objective function of about 10 and the other at a calibration objective function of about 25. These are an outcome of the presence of local optima in the objective function surface. PEST initially encountered some difficulties in jumping between these optima (as would any method of predictive uncertainty analysis), but eventually made the jump when given sufficient incentive to do so.
- The value of the prediction cannot be lowered beneath about 46 m³/sec (recall that the true value of the prediction is 27.24 m³/sec). This indicates either model structural defects, or the fact that insufficient parameters have been allowed to vary during the predictive analysis process (including perhaps a parameter that may need to have been employed to correct for unrepresentative estimation of catchment rainfall on the basis of a single rain gauge). The fact that so many adjustable parameters hit their bounds (as is evident from the run record file and from PEST's screen output) evinces compensatory behaviour on the part of these parameters. This is a strong indication that while the precepts of parameter parsimony may sometimes be useful in model calibration; they are more likely to be a hindrance than a help in model predictive uncertainty analysis. Reasons for this have already been discussed.

The last point is particularly salient. The more that parameters need to assume roles that compensate for model structural inadequacies, or for lack of adjustability of parameters that should in fact be declared as adjustable, the more likely are they to encounter their bounds. This often requires that other parameters then move large distances in order to play the compensatory roles that are thrust upon them if a good fit between model outputs and field measurements is to be attained. These parameters may then encounter their bounds as well. This can create numerical difficulties, at the same time as it creates the need to explore a parameter space that is gradually losing its relationship with hydraulic property reality.

Parameter Trajectories

When PEST is run in “pareto” mode it records the parameters pertaining to every point that it encounters along the Pareto front in a file whose filename base is the same as that of the PEST control file, but whose extension is “.ppd”. Data is stored in binary format as it can be voluminous in highly-parameterized contexts. The *.ppd file can be converted to ASCII format using the PPD2ASC utility. (PPD2ASC stands for “Pareto parameter data to ASCII format.) Run PPD2ASC using the following command.

```
ppd2asc calib4.ppd calib4.dat
```

Data contained in file *calib4.dat* can be used to inspect changes in parameter values with position along the Pareto front. See Figures 7.4a and 7.4b.

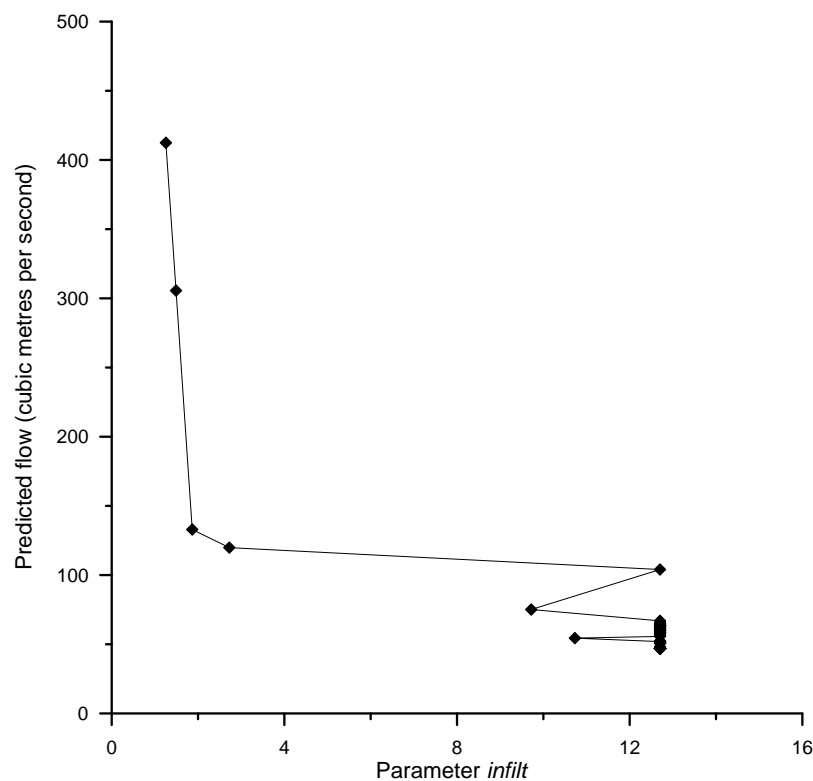


Figure 7.4a Movement of the *infiltr* parameter as the Pareto front is traversed. Notice its repeated encounters with its upper bound.

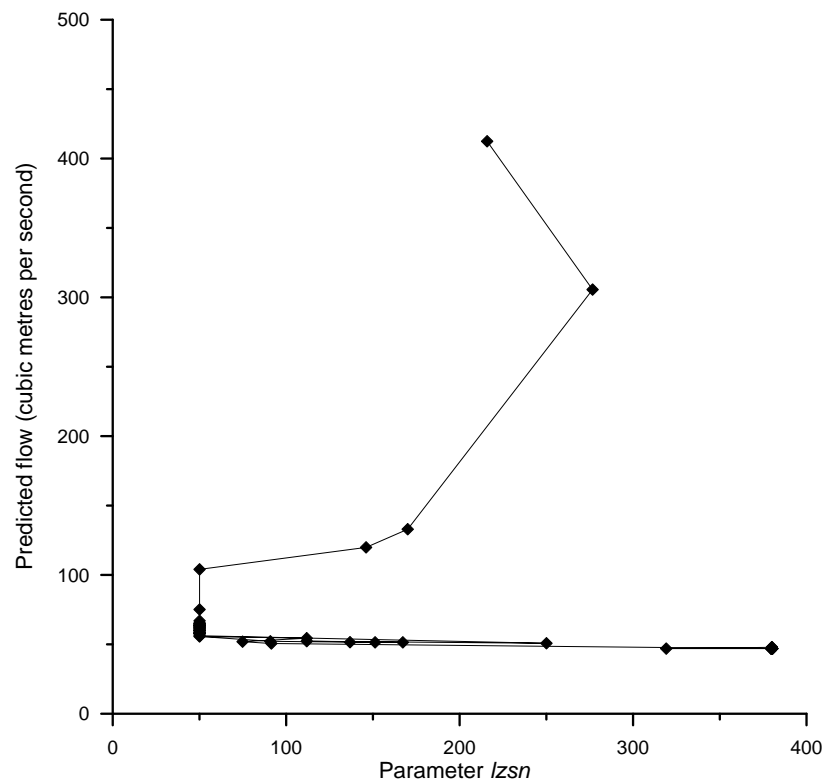


Figure 7.4b Movement of the l_{zsn} parameter as the Pareto front is traversed. Notice its encounters with both its lower and upper bounds.

Selection of Parameters

We will finish our analysis of the surface water model by extracting parameters pertaining to the 4th point on the Pareto curve of Figure 7.3. This point is encountered just before a jump is made to another objective function minimum. At this point, according to *calib4.pod*, the objective function is still at a reasonably low value of 10.35. Meanwhile the predicted maximum flow is 119.9 m³/sec. Following from the logic presented earlier in this chapter, we cannot reject the hypothesis that peak flow forthcoming from rain which fell on and just before 20th August 1986 will be as low as 119.9 m³/sec if the fit between model outputs and observed flows over the calibration period appears to be visibly good when calculated on the basis of the corresponding parameter set.

The PPD2PAR utility can be used to extract the pertinent parameter set from the binary *calib4.ppd* file. (PPD2PAR stands for “Pareto parameter data to parameter value file”.) We will extract the parameter set with index 3. (Indices correspond to iteration numbers; they begin at 0, this corresponding to the initial parameter set.) Run PPD2PAR using the following command:

```
ppd2par calib4.ppd temp.par 3
```

Now build a new PEST control file in which this parameter set is featured as the initial parameter set using the command:

```
parrep temp.par calib1.pst calib4_soln.pst
```

Set NOPTMAX to 0 in *calib4_soln.pst* and then run PEST using the command:

```
pest calib4_soln
```

Verify that the objective function is indeed 10.351. Figure 7.5 depicts modelled and measured flows over the 1985 calibration period, and then over the ensuing year. The fit over the

calibration period isn't too bad at all. A predicted maximum flow that is as low as 119.9 m³/sec over the period 19th to 23rd August 1986 is indeed compatible with the data.

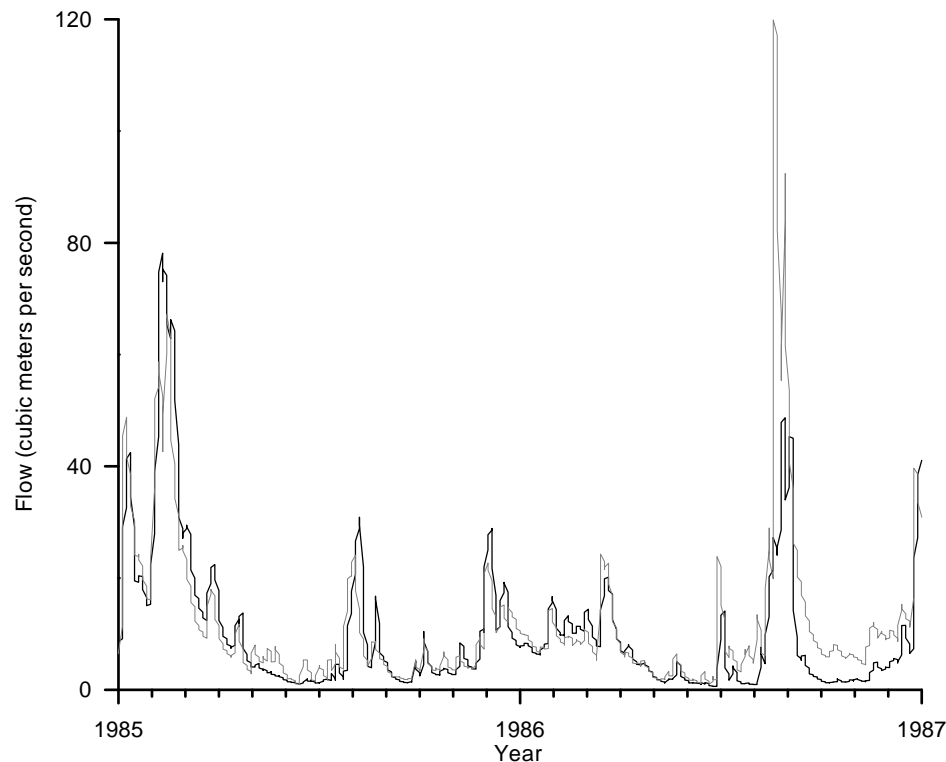


Figure 7.5 Measured and modelled flows over the calibration and prediction periods. Parameters used for model simulation were chosen from the Pareto front.

Groundwater Model

The Hypothesized Prediction

As for the surface water model discussed in the previous subsection, we have the luxury of knowing the value of the model prediction whose compatibility with existing soft and hard knowledge we now wish to explore. However in real life, not knowing the true value of the prediction does not present too much of a problem. In practice we would choose an extreme predictive value, or a value that has some regulatory significance. By traversing the Pareto front we have the ability to assess the likelihood of all predictive values between that calculated on the basis of the calibrated parameter set and that selected as a basis for hypothesis-testing.

In the present case we will select a value of 3000 days for particle travel time to the bottom model boundary as our test value.

The PEST Control File

File *calib5.pst* is the PEST control file that will form the basis for our traversal of the predictive Pareto front. This is slightly modified from the PEST control file *calib1r_soln.pst* that contains optimized values obtained from Tikhonov-based inversion. The following features of this file are salient.

- PEST is asked to run in “pareto” mode, this being the setting of the PESTMODE variable on line 3 of this file.

- The *part_time* observation now has an “observed value” of 3000. A weight of 0.004 is assigned to this “observation”. (It may require a little experimentation in order to find a suitable value for this weight, as well as for some of the variables appearing in the “pareto” section of the PEST control file.)
- In the “pareto” section of the PEST control file, PEST is told to adjust the weight factor applied to members of the observation group *time*. The *part_time* particle travel time prediction is the sole member of this group.
- In the “pareto” section of the PEST control file, PEST is asked to provide an initial weight factor of 0.0 to the *time* observation group, and to then increase this weight factor up to a maximum of 1.0 in 20 increments; see the PARETO_WTFAC_START, PARETO_WTFAC_FIN and NUM_WTFAC_INC control variables on the second line of this section. Two optimisation iterations are to be performed for each one of these increments (see the NUM_ITER_GEN control variable on the third line of the “pareto” section). Meanwhile the value of the *part_time* prediction will be made available for monitoring purposes while PEST is running. See PEST documentation for further Pareto setting details.

Some explanation for other PEST settings is warranted. All observations in file *calib5.pst* are assigned a weight of 10.0 as this is the inverse of 0.1, the standard deviation of noise associated with head measurements. Parameter credibility is enshrined in prior information. Each prior information equation contained in file *calib5.pst* ascribes to one parameter a preferred value equal to its initial value. All prior information is collectively assigned a covariance matrix based on the known stochastic description of geological heterogeneity prevailing in the synthetic study area. (See Chapter 4.) Overall, this strategy ensures that observations and prior information equations are weighted correctly with respect to each other. Collectively these comprise the calibration objective function. These are played against the predictive objective function, this being calculated from the discrepancy between the model prediction and an “observed” particle travel time of 3000 days, in order to create the conditions for existence of a Pareto front which is then traversed.

In everyday modelling practice we do not know correct relative weighting between field data and regularisation constraints encapsulated in prior information equations. Theoretically, this can emerge from a regularized inversion process; PEST prints out its optimized regularisation weight factor at the end of this process. A new PEST control file could then be created in which all regularisation observations and regularisation prior information equations were provided with new weights obtained as old weights multiplied by this weight factor. Meanwhile all elements in covariance matrices assigned to prior information equations would be divided by the square of this weight factor. In practice this can be a troublesome procedure as PEST adjusts inter-regularisation group weighting during the Tikhonov inversion process if the IREGADJ regularisation control value is set to a positive number. This is recommended practice as it renders the inversion process much more numerically stable than it would otherwise be. In this case, final regularisation weights can be extracted from the “residuals file” written by PEST upon termination of execution. These can simply be cut and pasted into the new PEST control file used for the running of PEST in “pareto” mode.

This, too, is a little cumbersome and may not be necessary. It is the author’s experience that regularisation constraints may not be required at all when using PEST in “pareto” mode to test the credibility of hypothesized predictions. This is an outcome of the fact that regularisation is already implied to at least some extent in most of the parameterization devices that are used in groundwater models (including pilot point parameters) by virtue of the fact that numerical considerations limit the number of these parameters that can be used,

and by virtue of the fact that concepts such as stationarity on which a covariance matrix will probably be based are generally inapplicable.

Another alternative is to use the ADDREG1 utility to provide new regularisation constraints in which preferred parameter values are re-assigned as calibrated values. Thus the new regularisation objective function is zero at the start of the Pareto process. Weights should then be assigned to these preferred values that are in harmony with what has been learned from the calibration process about spatial parameter variability. Alternatively, build a covariance matrix from what you think may be a suitable variogram using the PPCOV utility. For proper relativity, adjust weights applied to field measurements so that the total measurement objective function is roughly the same the total number of field measurements.

In the end it matters little. Your task in using the Pareto method is to find out what it takes for a certain prediction to arise. The real world is seriously non-stationary. If an hypothesised prediction can be made to happen easily, then not much variability will need to be introduced to the calibrated parameter field to make it happen. In contrast, if the hypothesised prediction is improbable, then a high degree of local parameter variability will be required to make it happen. A modeller's assessment of whether this is too much variability or not will almost certainly be based on visual inspection of the parameter field that is required for occurrence of the prediction, and not on a statistical test applied to the prior information component of the objective function. Perfection in defining prior probability constraints is therefore of secondary importance. Meanwhile devices such as anisotropic interpolation from pilot points to the model grid can help to guarantee that heterogeneity is introduced in ways that conform as much as possible to what is known of the prevailing geology when exploring the likelihood or otherwise of the prediction of interest.

Running PEST

Run PEST by typing the command:

```
pest calib5
```

at the command-line prompt. While PEST is running you can monitor some of the details of its Pareto journey by inspecting file *calib5.pod* as these details are continually updated. This records the contribution to the objective function made by every observation group, as well as the value of the *part_time* prediction (as was requested in the “pareto” section of the PEST control file). The total calibration objective function is obtained by adding together the contributions made by the *heads* and *regul_k* observation groups. (Recall that the *distance* observation group contains the particle exit time prediction that is not used in any of the present exercises but is nevertheless “carried” through the inversion processes described herein and provided with a weight of 0.0).

The Pareto Front

Figure 7.6 records PEST's Pareto front journey as a plot of particle travel time against calibration objective function.

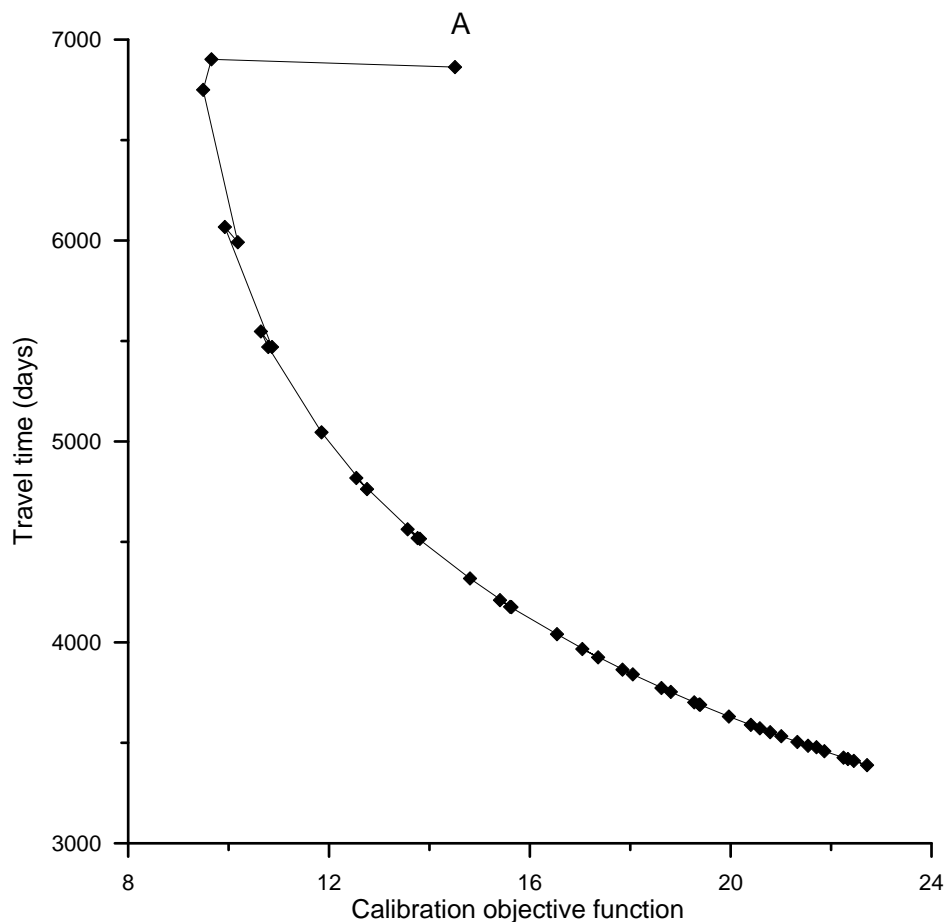


Figure 7.6 The Pareto front - predicted particle travel time vs. calibration objective function.

It is immediately apparent from an inspection of Figure 7.6 that PEST's journey along the Pareto front did not actually begin on the Pareto front. This should not be construed as indicating failure of the Tikhonov regularisation scheme that defined this starting point. Rather it is an outcome of the fact that we chose a target measurement objective function of 12.0 when calibrating the model using Tikhonov regularisation and instructed PEST to stop the calibration process as soon as this measurement objective function target was achieved. This saved model runs. It is apparent from Figure 7.6 and from an inspection of file *calib5.pod* that a lower measurement objective function value could have been sought without over-fitting. Alternatively, the REGCONTINUE variable in the "regularisation" section of the PEST control file could have been set to "continue" to force PEST to make all attempts possible to minimize the regularisation objective function subject to maintaining the selected measurement objective function constraint before ceasing execution. In real-world modelling practice these measures are not usually warranted as numerical convenience and subjectivity tend to play a larger role than rigorous application of geostatistical principles. Furthermore, in the present case they would not have lead to a better model prediction of total particle travel time, as the error in this prediction has its source in the null space. Fine-tuning of the calibrated parameter field would not have made this prediction any better.

Once situated on the Pareto front, it appears that PEST's journey along this front was smooth. Traversal of this front exposed no other objective function minima as occurred for the surface water modelling example. Nor did any parameters hit their bounds (as an inspection of parameters within file *calib5.ppd* would readily reveal if this were converted to ASCII format using the PPD2ASC utility).

So where does particle travel time predictive credibility run out? The fastest travel time prediction encountered on that part of the Pareto front that is pictured in Figure 7.6 is 3389 days, this having been calculated on the second-to-last optimisation iteration. For this travel time the measurement objective function is 4.30 - which is not unreasonable for an ambient measurement noise of 0.1 m and a measurement weight of 10 applied to each of 12 observations. The prior information objective function is 18.4 - also not unreasonable given that prior information equations are weighted using the inverse of the true covariance matrix and that there are 104 of these.

Pilot point parameter values corresponding to this prediction can be extracted from the binary *calib5.ppd* file using the command:

```
ppd2par calib5.ppd temp.par 39
```

These can be inserted into a new PEST control file as initial values in that file using the command:

```
parrep temp.par calib1r.pst calib5_soln.pst
```

Set NOPTMAX to 0 in *calib5_soln.pst* and then run PEST using the command:

```
pest calib5_soln
```

PEST's screen output will verify that the above objective function components are correct and that we have indeed extracted from *calib5.ppd* the parameter set that we desire. The hydraulic conductivity field corresponding to these pilot point values is pictured in Figure 7.7; so too is the particle path.

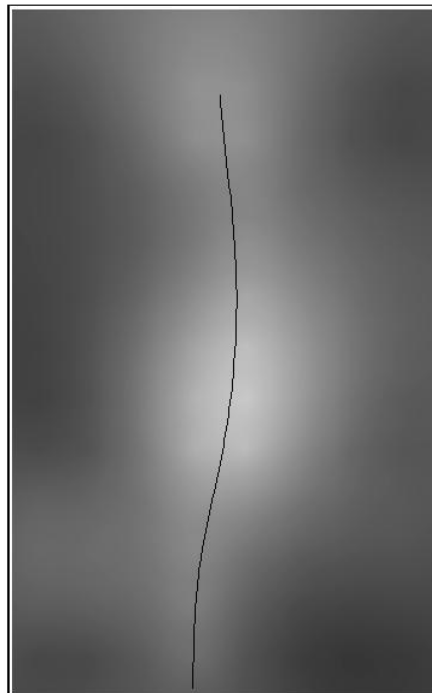


Figure 7.7 Hydraulic conductivity field giving rise to a prediction of 3389 days. The particle path is also shown.

A more quantitative analysis of this same problem is presented by Moore et al. (2010). As has been extensively discussed herein however, rarely in real-world modelling practice can assessment of predictive likelihood be more than subjective. This does not diminish the need

to make this assessment, nor the powerful means that modelling provides to make such assessment possible. In fact, scientifically-based environmental management depends on it.

8. Conclusions

This document has attempted to describe the range of possibilities that are offered by PEST and its associated utility support software for exploration of the uncertainty associated with predictions of future environmental behaviour, and for reducing that uncertainty to its theoretical lower limit. In doing this, it has attempted to make this exploration as salient as possible to the way in which numerical simulation should be used to underpin real-world environmental management.

Modelling cannot provide certainty where none exists. However, if used properly, it can minimize our potential for error when making predictions of future environmental behaviour by providing proper receptacles for all available information. This information includes expert knowledge, point measurements of system properties, and historical measurements of system state. Modelling can then be used to quantify the potential for error that remains once all of this information has been assimilated. This quantification is essential to risk assessment which, in turn, is essential to good decision-making.

From these considerations it is apparent that if environmental management is to benefit from numerical modelling, two types of software are required. Obviously, simulation capabilities must be provided by numerical models. Indeed the art of modelling is a mature art, and while models will continue to improve, a solid foundation has been built over the last 30 years of model usage for their continued development and improvement. However for models to achieve their full potential in environmental management, they must be partnered with software which can use them to extract information from all available sources, and to quantify the nature and ramifications of gaps in this information as it pertains to the assessment of future environmental behaviour under different possible management regimes. Unfortunately, the development of this kind of software has not yet reached maturity. However there is growing recognition within the industry that there is an urgent need for this to happen. Perhaps, as software developers respond to this need, the next 10 years will see modelling “come of age” as a consequence.

Meanwhile, another obstacle to the general acceptance of methodologies and tools such as that provided by the PEST suite is fast disappearing. The numerical burden of having to undertake hundreds, thousands, or even tens of thousands of model runs will not be too great of a numerical burden for much longer. Parallelisation of model runs is essential if runs are to be done in these numbers. At the time of writing, the development of computing technology is exactly in this direction. Within a few years massive parallelisation of model runs within a single machine, or across multiple real or virtual machines in unknown places across the world will be a trivial undertaking, available to all modellers in all places.

It is the author’s hope that modellers find the methodologies and exercises provided herein useful, and that they convey the ideas which underpin them to their colleagues, and to those who must make decisions on the basis of models. It is hoped that these exercises and ideas provide at least a small contribution to the continued evolution of a modelling culture that abandons the magical aura that is sometimes implicitly associated with models, replacing it instead with a scientific understanding of the modelling process that guarantees it an indispensable role in hard-nosed, risk-based decision-making.

9. References

- Albert, A., 1972. Regression and the Moore-Penrose pseudo-inverse. *Academic Press*, New York.
- Anderman, E.R. and Hill, M.C., 2001. MODFLOW-2000, The U.S. Geological Survey modular ground-water model—Documentation of the advective-transport observation (ADV2) package, version 2. *U.S. Geological Survey Open-File Report*, 01-54, 69p., U.S. Geological Survey, Reston, Va.
- Bicknell, B.R., Imhoff, J.C., Kittle, J.L., Jobes, T.H. and Donigan, A.S., 2001. HSPF User's Manual. *Aqua Terra Consultants*, Mountain View, California.
- Cooley, R.L. and Vecchia, A.V., 1987. Calculation of nonlinear confidence and prediction intervals for ground-water flow models. *Water Resour. Bull.* 23 (4), 581-599.
- Cooley, R.L., 2004. A theory for modeling ground-water flow in heterogeneous media. *US Geological Survey Professional Paper* 1679, 220p.
- Cooley, R.L. and Christensen, S., 2006. Bias and uncertainty in regression-calibrated models of groundwater flow in heterogeneous media. *Adv. Water Resour.*, 29, 639-656.
- Deutsch, C.V. and Journel, A.G., 1998. GSLIB: Geostatistical software library. *Oxford University Press*.
- Doherty, J., and Hunt, R.J., 2010. Response to comment on “Two statistics for evaluating parameter identifiability and error reduction”. *J Hydrol.* 380, 489-496.
- Doherty, J. and Johnston, J.M., 2003. Methodologies for calibration and predictive analysis of a watershed model. *J. Am. Water Resour. Assoc.*, 39(2), 251-265.
- Doherty, J. and Welter, D., 2010, A short exploration of structural noise. *Water Resour. Res.*, 46, W05525, doi:10.1029/2009WR008377.
- Fienen, M.N., Doherty, J.E., Hunt, R.J. and Reeves, H.W., 2010. Using prediction uncertainty analysis to design hydrologic monitoring networks: example applications from the Great Lakes Water Authority availability pilot project. *USGS Scientific Investigations Report* 2010-5159.
- Freeze, R.A., Massmann, J., Smith, L., Sperling, T. and James, B., 1990. Hydrogeological decision analysis: 1 A Framework. *Ground Water* 28 (5), 738 - 766.
- Freeze, A., James, B., Massmann, J., Sperling, T. and Smith, L., 1992. Hydrogeological decision analysis: 4. The concept of data worth and its use in the development of site investigation strategies. *Ground Water*, 30 (4), 574 - 588.
- Gallagher, M.R. and Doherty, J. 2006. Parameter estimation and uncertainty analysis for a watershed model. *Environ Modell. Softw.*, 22, 1000-1020.
- Gallagher, M. and Doherty, J., 2007a. Predictive error analysis for a water resource management model. *J Hydrol*, 34(3-4), 513-533.
- Gallagher, M. R., and Doherty, J., 2007b. Parameter interdependence and uncertainty induced by lumping in a hydrologic model. *Water Resour. Res.*, 43, W05421, doi:10.1029/2006WR005347.
- Harbaugh, A.W., Banta, E.R., Hill, M.C. and McDonald, M.G., 2000. The U.S. Geological Survey Modular Ground-Water Model – User Guide to Modularization Concepts and

- the Ground-Water Flow process. *U.S. Geological Survey Open File Report* 00-92. Reston, Virginia.
- Herckenrath, D., Langevin, C.D., and Doherty, J., 2010. Predictive uncertainty analysis of a salt water intrusion model using null space Monte Carlo. *Submitted to Water Resources Research*.
- James, S.C., Doherty, J. and Eddebarh, A.-A., 2009. Post-calibration uncertainty analysis: Yucca Mountain, Nevada, USA. *Ground Water*. 47 (6), 851-869.
- Koch, K-R., 1999. Parameter estimation and hypothesis testing in linear models. *Springer-Verlag*, Berlin, Heidelberg. Third Edition.
- Massmann, J., Freeze, R.A., Smith, L., Sperling, T. and James, B., 1991. Hydrogeological decision analysis 2. Applications to Ground-Water Contamination. *Ground Water* 29 (4), 536-548.
- Moore, C. and Doherty, J., 2005. The role of the calibration process in reducing model predictive error. *Water Resour. Res.*, 41 (5), W05050.
- Moore, C. and Doherty, J., 2006. The cost of uniqueness in groundwater model calibration. *Adv. Water Resour.*, 29 (4), 605–623.
- Moore, C., Wöhling, T., and Doherty, J., 2010. Efficient regularization and uncertainty analysis using a global optimization methodology. *Water Resour. Res.*, In press.
- Nathan, R.J. and McMahon, T.A., 1990. Evaluation of automated techniques for base flow and recession analysis. *Water Resour. Res.*, 26 (7), 1465-1473.
- Orrell, D., 2007. *Apollo's Arrow. The Science of Prediction and the Future of Everything*. Harper Collins Publishers, Toronto, Canada.
- Sperling, T., Freeze, R.A., Massmann, J., Smith, L. and James, B., 1992. Hydrogeological decision analysis: 3. Application to design of a ground-water control system at an open pit mine. *Ground Water*, 30 (3), 376-389.
- Tonkin, M., Doherty, J. and Moore, C., 2007. Efficient nonlinear predictive error variance evaluation for highly parameterized models. *Water Resour. Res.*, 43, W07429, doi:10.1029/2006WR005348.
- Tonkin M., J. and Doherty, J., 2009. Calibration-constrained Monte Carlo analysis of highly parameterized models using subspace techniques. *Water Resour. Res.*, 45, W00B10, doi:10.1029/2007WR006678.
- USEPA, 2000. BASINS Technical Note 6: Estimating Hydrology and Hydraulic Parameters for HSPF. *EPA-823-R-00-012*.
- Vecchia, A.V. and Cooley, R.L., 1987. Simultaneous confidence and prediction intervals for nonlinear regression models with application to a ground water flow model. *Water Resour. Res.*, 23 (7), 1237-1250.

Appendix 1. PEST Utilities

This appendix presents a series of tables listing utility software provided with PEST, together with the function that each program serves. Programs are grouped into different tables according to similarity of function. Complete descriptions of the following utilities can be found at <http://www.pesthomepage.org/>.

Checking Utilities

Program	Purpose
TEMPCHEK	Checks the integrity of a PEST template file.
INSCHEK	Checks the integrity of a PEST instruction file.
PESTCHEK	Checks an entire PEST input dataset for correctness and consistency.

Classical parameter estimation preprocessing and postprocessing

(Note: Some of these can also be employed for regularized inversion preprocessing and postprocessing.)

Program	Purpose
PARREP	Builds a new PEST control file whose initial values are optimized values from a previous PEST run.
PARAMFIX	Alters prior information pertaining to one or a number of parameters as these parameters are tied or fixed.
EIGPROC	Collects uncertainty, sensitivity, and eigencomponent information pertinent to a nominated parameter from PEST output files.
PCOV2MAT	Extracts a parameter covariance matrix from a PEST control file, rewriting it in matrix file format.
INFSTAT	Computes a suite of observation influence statistics, including DFBETAS and Cook's D.
PESTGEN	Builds a basic PEST control file based on a parameter value file and an INSCHEK output file.

Regularized inversion preprocessing and postprocessing

Program	Purpose
ADDREG1	Adds preferred-value regularization to a PEST control file based on initial parameter values.
SUBREG1	Subtracts regularization from a PEST control file.
SUPCALC	Estimates number of superparameters to employ in SVD-assisted parameter estimation.
SVDAPREP	Writes a PEST input dataset for SVD-assisted parameter estimation.
PARCALC	Run as part of a model employed for SVD-assisted parameter estimation; computes base parameter values from superparameter values.
PICALC	Run as part of a model employed for SVD-assisted parameter estimation; computes prior information expressed in terms of base parameter values.
IDENTPAR	Computes parameter identifiability.
PCLC2MAT	Computes base parameter composition of SVD-assist superparameters.
GENLINPRED	Automates running of PREDUNC* and PREDVAR* utilities. Undertakes linear predictive uncertainty/error analysis; also computes parameter identifiability and relative uncertainty/error variance reduction.
RESPROC	Processes information written in binary form to *.rsd file during any PEST run in which regularization of any kind is employed.
RESWRIT	Processes information written by RESPROC; stores resolution and "G" matrices in PEST matrix file format.
REGERR	Computes the covariance matrix of regularization-induced structural noise.

Weights and covariance matrix manipulation

Program	Purpose
COV2COR	Calculates a correlation coefficient matrix from a covariance matrix.
COVCOND	Calculates a conditioned covariance matrix from an unconditioned covariance matrix.
PWTADJ1	Alters weights in a PEST control file so that the contribution to the initial objective function by all observation groups is equal based on residuals calculated at initial values.
PWTADJ2	Attempts to create observation-group-specific weights, which are the inverse of measurement-error standard deviations.
WTFACOR	Multiplies the weights pertaining to all observations belonging to a selected observation group by a specified factor.

Linear uncertainty analysis

Program	Purpose
PREDUNC1	Computes the uncertainty of a user-specified prediction.
PREDUNC4	Computes contributions to predictive uncertainty by different parameters or parameter groups.
PREDUNC5	Computes observation worth through its effect in lowering predictive uncertainty.

Linear error analysis

Program	Purpose
PARAMERR	Computes the covariance matrix of parameter error after a calibration exercise involving any form of regularization.
PREDERR	Computes the error variance of a prediction whose sensitivities are available after a calibration exercise involving any form of regularization.
PREDERR1	Similar to PREDERR, but slightly different in its input-file requirements.
PREDERR2	Similar to PREDERR, but slightly different in its input-file requirements.
PREDERR3	Similar to PREDERR, but slightly different in its input-file requirements.
PREDVAR1	Computes the error variance of a model prediction based on a notional calibration exercise implemented using truncated SVD; also finds the minimum of the predictive error variance curve.
PREDVAR1A	As for PREDVAR1, but undertakes SVD on $\mathbf{Q}^{1/2}\mathbf{X}$ rather than $\mathbf{X}^t\mathbf{Q}\mathbf{X}$.
PREDVAR4	Computes contribution made to the error variance of a prediction by different parameters and/or groups of parameters.
PREDVAR5	Computes observation worth through its efficacy in lowering predictive error variance.

Nonlinear error analysis

Program	Purpose
VECLOG	Computes the log of all elements of a vector (normally used as part of nonlinear highly parameterized predictive maximization/minimization).
PEST2VEC	Facilitates preparation for nonlinear highly parameterized predictive uncertainty analysis done by way of constrained maximization /minimization.
VEC2PEST	Facilitates preparation for nonlinear highly parameterized predictive uncertainty analysis done by way of constrained maximization /minimization.
OBSREP	Replaces observations in a PEST control file with best-fit model-generated equivalents. (This is normally run just prior to REGPRED.)
REGPRED	Builds a PEST control file in which postcalibration nonlinear predictive uncertainty analysis is effected by constrained prediction maximization/minimization.
RANDPAR	Computes random parameter values, placing these values into a series of parameter-value files.
PNULPAR	Undertakes null-space projection of random parameter fields to remove solution-space component; replaces it with solution space component from calibrated model.
RDMULRES	Reads multiple output files produced as an outcome of Monte Carlo analysis and collates results.

MULPARTAB	Builds a table of multiple sets of parameter values produced through null-space Monte Carlo analysis.
COMFILNME	Facilitates post-null-space Monte Carlo file management.

Pareto analysis

Program	Purpose
PPD2ASC	Stores all data recorded in a Pareto parameter data file in ASCII format.
PPD2PAR	Extracts single parameter sets from the Pareto parameter data file.

Sensitivity data manipulation

Program	Purpose
JACTEST	Undertakes serial or parallel model runs to test the integrity of finite-difference-calculated derivatives.
POSTJACTEST	JACTEST postprocessor; provides index of derivatives corruptness for different model outputs.
JACWRIT	Rewrites the contents of a *.jco file in ASCII format.
JCO2JCO	Writes a Jacobian matrix corresponding to a new PEST control file on the basis of information contained in an existing *.jco/*.pst file pair.
JCO2MAT	Rewrites the contents of a *.jco file in PEST matrix file format.
JCOADDZ	Adds sensitivities to an existing *.jco file.
JCOCOMB	Builds a new *.jco file from an existing one, in which observations from the first are combined in user-supplied ratios in the second.
JCODIFF	Subtracts the contents of one *.jco file from that of another.
JCOORDER	Reorders rows and/or columns in a *.jco file.
JCOPCAT	Concatenates two *.jco files; thus sensitivities with respect to some parameters can be computed on one machine and those with respect to other parameters can be computed on another.
JCOTRANS	Translates from old *.jco storage format to new (compressed) storage format employed by PEST.
JROW2MAT	Extracts a row of a Jacobian matrix file and writes it in PEST matrix file format.
JROW2VEC	Extracts a row of a Jacobian matrix file, transposes it, and writes it in PEST matrix file format.
DERCOMB1	Combines two external derivatives files (supplied by models that can calculate their own derivatives) into one, before being read by PEST.
MULJCOSIN	Reads multiple *.jco files as written on a single PEST run (if PEST is instructed to write such multiple files); calculates composite sensitivity of nominated parameter or observation from iteration to iteration.
WTSENOUT	Computes a weighted Jacobian matrix and a weighted observation vector.

Matrix manipulation

Program	Purpose
MAT2SRF	Writes a matrix in SURFER grid format.
MATADD	Performs matrix addition.
MATCOLEX	Extracts a column of a matrix.
MATDIAG	Extracts the diagonal of a matrix.
MATDIFF	Performs matrix differencing.
MATINVP	Computes the inverse of a positive definite matrix.
MATJOINC	Joins matrices which possess the same number of columns.
MATJOIND	Joins two matrices in a diagonal sense (useful in forming a composite covariance matrix).
MATJOINR	Joins matrices which possess the same number of rows.
MATORDER	Reorders the rows or columns of a matrix.
MATPROD	Performs matrix multiplication.
MATQUAD	Evaluates the quadratic form $\mathbf{y}^t \mathbf{M} \mathbf{y}$.
MATROW	Extracts a single row of a matrix.
MATSMUL	Multiplies a matrix by a scalar.
MATSPEC	Lists matrix specifications.
MATSVD	Undertakes singular value decomposition of an arbitrary matrix.

MATSYM	Forms a symmetric matrix as $(\mathbf{M} + \mathbf{M}^t)/2$.
MATTRANS	Computes the transpose of a matrix.
MATXTXI	Computes $(\mathbf{X}^t\mathbf{X})^{-1}$ where \mathbf{X} has more rows than columns.
MATXTXIX	Computes $(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}$ where \mathbf{X} has more rows than columns.

Global optimization

Program	Purpose
SCEUA_P	Global optimization by use of the SCEUA algorithm.
CMAES_P	Global optimization by use of the CMAES algorithm.

General

Program	Purpose
PAR2PAR	Undertakes arbitrary mathematical manipulation of model parameters; normally run as part of a model calibrated by PEST.
SCALEPAR	Builds a PEST input dataset based on parameters scaled by their innate variability.
GENLIN	Generalized linear model.
PESTLIN	Reads a general PEST input dataset and accompanying *.jco file; creates a GENLIN model and accompanying PEST input dataset for calibration of that model.
SENSAN	Undertakes basic sensitivity analysis through repeated model runs.
SENSCHEK	Checks the integrity of a SENSAN input dataset.
PAUSE	Pauses PEST execution.
PUNPAUSE	Unpauses PEST execution.
PSTOP	Stops PEST execution.
PSTOPST	Instructs PEST to cease execution with a full statistical printout.
PSLAVE	PEST slave program.

Appendix 2. PEST Groundwater Data Utilities

This appendix presents a series of tables listing utility software provided with the PEST Groundwater Data Utilities suite. Programs are grouped into different tables according to similarity of function. Complete descriptions of these utilities can be found at <http://www.pesthomepage.org/>.

Implementation of pilot-point parameterization

Program	Purpose
FAC2FEFL	Uses PPKFAC_FEFL-generated kriging factors to modify a FEFLOW input data file on the basis of spatial interpolation from a set of pilot points.
FAC2FEM	Uses PPK2FAC-generated kriging factors to produce a MicroFEM input file on the basis of spatial interpolation from a set of pilot points.
FAC2MF2K	Modifies an existing set of MODFLOW-2000 input files, replacing parameters cited in that file with pilot-point-based parameters (often a first step in pilot-point-based model calibration).
FAC2REAL	Uses PPKFAC-generated kriging factors to produce a MODFLOW-compatible real array on the basis of spatial interpolation from a set of pilot points.
FAC2RSM	Uses PPKFACR-generated kriging factors to produce an RSM model input data file on the basis of spatial interpolation from a set of pilot points.
PPK2FAC	Calculates kriging factors for use in spatial interpolation from a set of pilot points to model grid cell centers.
PPK2FACF	Calculates kriging factors for use in spatial interpolation from a set of pilot points to the nodes of a MicroFEM finite element mesh.
PPK2FAC1	Identical to PPK2FAC except the regularization data file it writes is suitable for the use of PPKREG1.
PPK2FACR	Calculates kriging factors for use in spatial interpolation from a set of pilot points to the nodes of an RSM mesh. Regularization data file protocol is identical to that of PPK2FAC1.
PPK2FAC_FEFL	Calculates kriging factors for use in spatial interpolation from a set of pilot points to the elements of a FEFLOW mesh. Regularization data file protocol is identical to that of PPK2FAC1.
PARM3D	Assists in pilot-point parameterization of a 3-D model domain where hydrogeological units intersect grid layers.

MODFLOW/MT3D array manipulation

Program	Purpose
ARR2BORE	Undertakes spatial interpolation from a single array to a set of points.
INT2MIF	Generates MAPINFO MIF and MID files based on a MODFLOW/MT3D-compatible integer array.
INT2REAL	Builds a MODFLOW/MT3D-compatible real array based on the contents of a MODFLOW/MT3D-compatible integer array.
LOGARRAY	Evaluates the log (to base 10) of all elements of a real array.
PT2ARRAY	Builds a MODFLOW-compatible real array; the value assigned to each array element is calculated from information pertaining to points lying within the respective element.
REAL2INT	Builds a MODFLOW/MT3D-compatible integer array based on the contents of a MODFLOW/MT3D-compatible real array.
REAL2MIF	Generates MAPINFO MIF and MID files based on a MODFLOW/MT3D-compatible real array.
REAL2SRF	Translates a MODFLOW/MT3D-compatible real array into a SURFER grid file.
REAL2TAB	Translates a MODFLOW/MT3D-compatible real array into three-column real array table format.
SRF2REAL	Re-writes a SURFER grid file as a MODFLOW-compatible real array.

TAB2INT	Generates a MODFLOW/MT3D-compatible integer array from an integer array stored within a GIS.
TAB2REAL	Generates a MODFLOW/MT3D-compatible real array from a real array stored within a GIS.
TABCONV	Translates between integer or real array table files using row/column identifier format and those using cell number identifier format.
TWOARRAY	Combines two real arrays by addition, subtraction, multiplication, division and partial replacement.

MODFLOW/MT3D/SEAWAT preprocessing

Program	Purpose
MOD2ARRAY	Reads a MODFLOW or MT3D input file, extracting real or integer arrays from that file and storing them in separate files.
ELEV2CONC	Computes the elevation of the freshwater-saltwater interface on the basis of a sequence of concentration arrays.
ELEV2CONC1	Similar to ELEV2CONC, but computes “zero flow head” arrays as well.
REPARRAY	“Pastes” a MODFLOW- or MT3D-compatible real array into an existing MODFLOW or MT3D input file.

MODFLOW/MT3D/SEAWAT/FEFLOW postprocessing

Program	Purpose
ARRDET	Lists the contents of a MODFLOW or MT3D binary head/drawdown/concentration output file.
BUD2HYD	Extracts flow data from a MODFLOW binary cell-by-cell flow term file. Rewrites this data in a form suitable for plotting against time.
BUD2SMP	Extracts flow data from a MODFLOW binary cell-by-cell flow term file. Rewrites this data in bore sample file format.
CONC2ELEV	Computes the elevation of the freshwater/saltwater interface on the basis of a sequence of concentration arrays.
DAR2SMP	Translates system states computed by a FEFLOW model to bore sample file format.
GETMULARR	Extracts arrays from MODFLOW/MT3D binary output files at user-nominated simulation times and stores these arrays in separate binary files.
GETMULARR1	Extracts all arrays for a nominated simulation time from a MODFLOW/MT3D binary output file and writes these to another binary MODFLOW/MT3D output file.
MANY2ONE	Splits MODFLOW/MT3D-generated binary files comprised of multiple two-dimensional results arrays into individual ASCII/binary files.
MOD2OBS	Interpolates model-generated data to the same times and locations as those cited in a user-supplied bore sample file; writes another bore sample file.
LAYDIFF	Evaluates head value differences in different layers based on contents of a bore sample file, bore coordinates file and bore listing file.
MOD2SMP	Interpolates the information contained in a binary MODFLOW/MT3D output file to a set of user-specified bores, rewriting the bore-specific data as a bore sample file.
MOD2SMPDIFF	Interpolates the information contained in a binary MODFLOW/MT3D output file to user-specified bores, calculating the difference or ratio between heads/concentrations at user-nominated pairs of bores.
SECTION	Interpolates the data contained in multiple MODFLOW-compatible real arrays to an arbitrary transect line through all or part of the finite-difference grid.

Processing and manipulation of field and model time series

Program	Purpose
PMP2INFO	Builds a bore information file from a bore pumping file, the former containing cumulative pumped volumes between two user-specified dates for a user-supplied list of bores.
PMPCHK	Checks the integrity of the data contained in a bore pumping file.

SMP2HYD	Rewrites the contents of a bore sample file for a user-specified list of bores in a form suitable for plotting borehole data against time.
SMP2INFO	Time-interpolates the information contained in a bore sample file to a user-specified date for a list of user-specified bores, thus writing a bore information file ready for access by commercial contouring software.
SMP2SMP	Interpolates data contained within one bore sample file to the dates and times represented in another bore sample file.
SMPCAL	Calibrates one time-series dataset on the basis of another.
SMPCHEK	Checks the integrity of a bore sample file.
SMPDIFF	Writes a new bore sample file in which differences are taken between successive values in an existing bore sample file, or between values in an existing file and a reference value.
SMPTREND	Writes a new bore sample file in which differences are taken between samples within an existing bore sample file and either the first sample for each bore in that file or a reference sample. However, sampling is restricted to a yearly sample window.

Construction of a PEST input dataset

Program	Function
ADJOBS	Adjusts observation weights for different observation groups in a PEST control file according to user-defined formulas.
ARRAYOBS	Facilitates the introduction of model outputs consisting of MODFLOW/MT3D-compatible real arrays into a PEST parameter-estimation process.
PESTPREP	Automates construction of a PEST control file and PEST instruction file for a model comprised of MODFLOW and/or MT3D followed by MOD2OBS, or MODFLOW followed by BUD2SMP followed by SMP2SMP.
PESTPREP1	Similar to PESTPREP but provides extra flexibility in observation naming.
PESTPREP2	Similar to PESTPREP1 but allows extra observation data to be added to an existing PEST input dataset.

Adding regularization to a PEST input dataset

Program	Purpose
GENREG	Inserts prior information pertaining to many different types of regularization into an existing PEST control file.
PPCOV	Builds a covariance matrix pertaining to pilot point parameters based on one or a number of geostatistical structures.
PPKREG	Adds a “prior information” and “regularization” section to a PEST control file where parameterization is based on pilot points.
PPKREG1	Similar to PPKREG but more powerful in that it facilitates the use of both “difference regularization” (same as PPKREG) and “preferred-value regularization.”
ZONE2VAR1	Computes a parameter variogram where parameterization is based on a large number of zones of piecewise constancy, and is defined through a ZONMDEF output file. Assists in undertaking “variogram regularization” as described by Johnson and others (2007).
ZONE2VAR2	Computes a parameter variogram much more quickly than ZONE2VAR1 because it employs the results of the parameter search process done by the latter program as read from a binary file written by it.
VERTREG	Adds “vertical regularization” prior-information equations to a PEST control file where parameterization is based on pilot points.

Working with the MODFLOW adjoint process

Program	Function
ASENPROC	Reads a “distributed parameter sensitivity file” written by the adjoint state version of MODFLOW; formulates sensitivities for PEST parameters and writes them to a PEST “external derivatives file.”

MKMHOB	Reads a bore sample file. Writes a MODFLOW 2005 heads observation file, as well as an instruction file to read a MODFLOW heads output data file and a “PEST building block file” containing pertinent fragments of a PEST control file.
PPMDEF	Builds a parameter definition file for the use of ASENPROC, linking distributed parameters as employed by the adjoint process of MODFLOW to pilot-point parameters.
ZONMDEF	Assists in the preparation of input files for the use of PEST in conjunction with the MODFLOW-2005 adjoint process where parameters are based on a large number of zones of piecewise constancy.

Uncertainty Analysis

Program	Function
FIELDGEN	Generates a stochastic field in each zone of a model domain using the sequential Gaussian simulation method.
PPSAMP	Used in calibration-constrained Monte Carlo analysis. Samples stochastic fields at pilot point locations, interpolates between the pilot points, and generates difference fields.

Geographical data manipulation

Program	Function
GRID2ARC	Writes ARCINFO generated files of the active part of the finite-difference grid as defined by a user-supplied integer array.
GRID2BLN	Writes a SURFER blanking file of the active part of the finite-difference grid as defined by a user-supplied integer array.
GRID2DXF	Writes a DXF file of the active part of the finite-difference grid as defined by a user-supplied integer array.
GRID2PT	Tabulates the coordinates of the cell centers of the finite-difference grid within an active window defined by a user-supplied integer array.
INT2MIF	Generates MAPINFO MIF and MID files based on a MODFLOW/MT3D-compatible integer array.
PTINGRID	Locates the finite-difference cells in which arbitrary, user-supplied points lie; optionally provides the value of an integer or real array element pertaining to the cell containing each such point.
QDIG2DXF	Translates the output of the shareware digitizing program, QDIGIT, into DXF format.
QDIG2XYZ	Translates the “contours” output of QDIGIT to an “xyz” data file.
RDAT2TAB	Reads an RSM element data file or index file. Adds mesh centroid coordinates to respective data elements and rewrites data in tabular format.
ROTBLN	Rotates a SURFER blanking file about the top left corner of a finite-difference grid so that the component elements of the file can be overlain over the grid when the latter has been rotated such that its row direction is oriented directly east.
ROTDAT	Rotates a data file about the top left corner of a finite-difference grid so that the component elements of the file can be overlain over the grid when the latter has been rotated such that its row direction is oriented directly east.
ROTDXF	Rotates a DXF file about the top left corner of a finite-difference grid so that the component elements of the file can be overlain over the grid when the latter has been rotated such that its row direction is oriented directly east.
RSM2SRF	Reads an RSM (also GMS) 2D mesh file. Writes files through which SURFER can plot mesh design, outer mesh boundary, as well as nodes and element centroids.
ZONE2BLN	Writes a SURFER “blanking” file of finite-difference grid zonation as defined by a user-supplied, MODFLOW-compatible integer array.
ZONE2DXF	Writes a DXF file of finite-difference grid zonation as defined by a user-supplied, MODFLOW-compatible integer array.

Reference Cited

Johnson, T.C., Routh, P.S., Clemo, T., Barrash, W., and Clement, W.P., 2007, Incorporating geostatistical constraints in nonlinear inversion problems: *Water Resources Research*, v. 43, no. 10, W10422, doi:10.1029/2006WR005185.