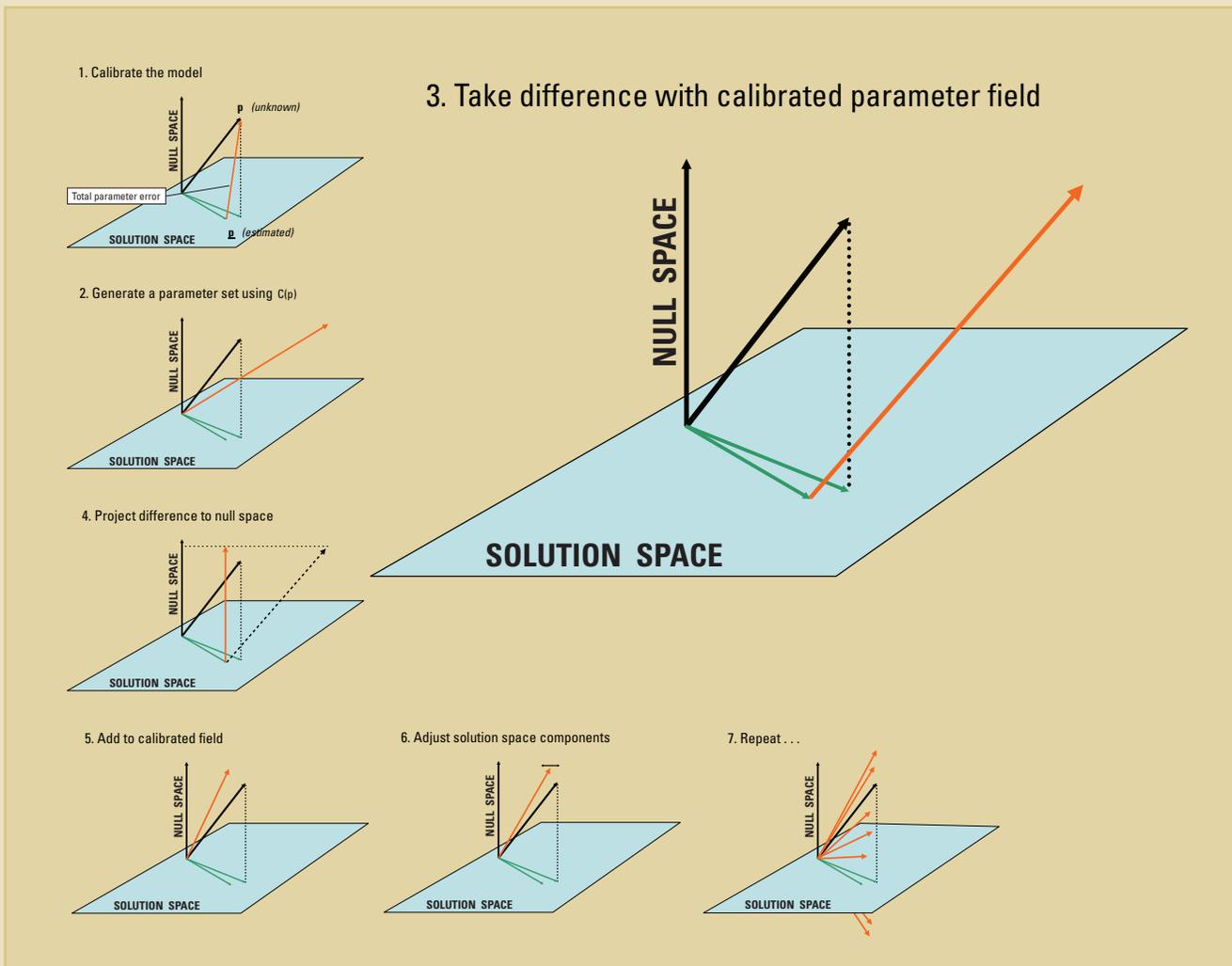


**Groundwater Resources Program
Global Change Research & Development**

Approaches to Highly Parameterized Inversion: A Guide to Using PEST for Model-Parameter and Predictive-Uncertainty Analysis



Scientific Investigations Report 2010–5211

Cover figure Processing steps required for generation of a sequence of calibration-constrained parameter fields by use of the null-space Monte Carlo methodology

Approaches to Highly Parameterized Inversion: A Guide to Using PEST for Model-Parameter and Predictive-Uncertainty Analysis

By John E. Doherty, Randall J. Hunt, and Matthew J. Tonkin

Groundwater Resources Program
Global Change Research & Development

Scientific Investigations Report 2010–5211

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
KEN SALAZAR, Secretary

U.S. Geological Survey
Marcia K. McNutt, Director

U.S. Geological Survey, Reston, Virginia: 2011

For more information on the USGS—the Federal source for science about the Earth, its natural and living resources, natural hazards, and the environment, visit <http://www.usgs.gov> or call 1-888-ASK-USGS.

For an overview of USGS information products, including maps, imagery, and publications, visit <http://www.usgs.gov/pubprod>

To order this and other USGS information products, visit <http://store.usgs.gov>

Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Although this report is in the public domain, permission must be secured from the individual copyright owners to reproduce any copyrighted materials contained within this report.

Suggested citation:

Doherty, J.E., Hunt, R.J., and Tonkin, M.J., 2010, Approaches to highly parameterized inversion: A guide to using PEST for model-parameter and predictive-uncertainty analysis: U.S. Geological Survey Scientific Investigations Report 2010-5211, 71 p.

Contents

Abstract.....	1
Introduction.....	1
Box 1: The Importance of Avoiding Oversimplification in Prediction	
Uncertainty Analysis.....	2
Purpose and Scope	5
Summary and Background of Underlying Theory	6
General Background	6
Parameter and Predictive Uncertainty	6
The Resolution Matrix	7
Parameter Error.....	9
Linear Analysis.....	11
Parameter Contributions to Predictive Uncertainty/Error Variance	11
Observation Worth.....	11
Optimization of Data Acquisition.....	12
Nonlinear Analysis of Overdetermined Systems.....	12
Structural Noise Incurred Through Parameter Simplification	13
Highly Parameterized Nonlinear Analysis	14
Constrained Maximization/Minimization	14
Null-Space Monte Carlo	14
Hypothesis Testing.....	16
Uncertainty Analysis for Overdetermined Systems.....	16
Linear Analysis Reported by PEST.....	17
Linear Analysis Using PEST Postprocessors.....	18
Other Useful Linear Postprocessors	18
Nonlinear Analysis—Constrained Optimization	19
Nonlinear Analysis—Random Parameter Generation for Monte Carlo Analyses	19
Linear Analysis for Underdetermined Systems	20
Postcalibration Analysis	20
Regularized Inversion and Uncertainty.....	21
Regularized Inversion Postprocessing	21
Predictive Error Analysis.....	21
Calibration-Independent Analysis	22
The PREDVAR Suite.....	23
PREDVAR1	23
PREDVAR4.....	23
PREDVAR5.....	24
The PREDUNC Suite	24
Identifiability and Relative Uncertainty/Error Variance Reduction	24
The GENLINPRED Utility	26
Box 2: GENLINPRED—Access to the Most Common Methods for	
Linear Predictive Analysis within PEST.....	26
Non-linear Analysis for Underdetermined Systems	27

Constrained Maximization/Minimization	27
Problem Reformulation	28
Practical Considerations	28
Null-Space Monte Carlo	28
Box 3: Null-Space Monte Carlo—A Flexible and Efficient Technique for Nonlinear Predictive Uncertainty Analysis	30
Method 1—Using the Existing Parameterization Scheme	32
RANDPAR	32
PNULPAR.....	32
SVD-Assisted Parameter Adjustment	32
The Processing Loop.....	32
Postprocessing	33
Making Predictions	34
Method 2—Using Stochastic Parameter Fields	34
General	34
Generation of Stochastic Parameter Fields	34
Pilot-Point Sampling	34
Null-Space Projection.....	34
Model Recalibration	35
Some Practical Considerations	35
Pareto-Based Hypothesis Testing.....	35
Hypothesized Predictive Value.....	35
Pareto Control Variables.....	36
Mapping the Pareto Front	36
Multiple Recalibration.....	36
Uses and Limitations of Model Uncertainty Estimates.....	37
References.....	38
Appendix 1. Basic PEST Input	42
Structure of the PEST Control File	42
Files used by PEST.....	51
Appendix 2. PEST Utilities.....	53
Appendix 3. Groundwater Data Utilities.....	57
Reference Cited.....	60
Appendix 4. Background and Theory of PEST Uncertainty Analyses.....	61
General	61
Parameter and Predictive Uncertainty	61
Parameter and Predictive Error	63
Calibration	63
The Resolution Matrix	64
Parameter Error.....	64
Predictive Error	66
Regularization-Induced Structural Noise	66
Predictive Uncertainty Analysis—Underdetermined Systems.....	67
Parameter and Predictive Error	67
Nonlinear Analysis.....	67
Predictive Uncertainty Analysis—Highly Parameterized Systems.....	69

Constrained Maximization/Minimization	69
Null-Space Monte Carlo	69
References Cited.....	70

Figures

B1–1. Local model domain and the locations of the pumping well, head prediction (H115_259), and streamgage.....	3
B1–2. Parameter discretization, hydraulic conductivity field seen by model, and results of data-worth analysis.....	4
1. Relations between real-world and estimated parameters where model calibration is achieved through truncated singular value decomposition.	8
2. Schematic depiction of parameter identifiability, as defined by Doherty and Hunt (2009)	8
3. Components of postcalibration parameter error.....	8
4. Contributions to total predictive error variance calculated by use of the PEST PREDVAR1 and PREDVAR1A utilities from Hunt and Doherty (2006).....	10
5. Precalibration and postcalibration contribution to uncertainty associated with the drought lake-stage prediction shown in figure 4.....	12
6. Schematic description of calibration-constrained predictive maximization/minimization.	13
7. Processing steps required for generation of a sequence of calibration-constrained parameter fields by use of the null-space Monte Carlo methodology.....	15
8. A Pareto-front plot of the tradeoff between best fit between simulated and observed targets (objective function, x-axis) and the prediction of a particle travel time.....	17
9. Utilities available for postcalibration parameter and predictive error variance analysis.....	20
10. PEST utilities available for calibration-independent error and uncertainty analysis	22
11A. Identifiability of parameters used by a composite groundwater/surface-water model	25
11B. Relative error reduction of parameters used by a composite groundwater/surface-water model	25
12. Processing steps required for exploration of predictive error variance through constrained maximization/minimization.....	27
13. Schematic of null-space Monte-Carlo methodologies.	29
B3–1. Distribution of objective functions computed with 100 stochastic parameter realizations (a) before null-space projection and recalibration; (b) after null-space projection.....	31
14. A batch process that implements successive recalibration of 100 null-space-projected parameter fields	33
A1.1. Structure of the PEST Control File	43
A4.1. Marginal and conditional probability distributions of a random variable x^1 that is correlated with another random variable x^2	63
A4.2. Relation between real-world and estimated parameters where model calibration is achieved through truncated SVD	65

A4.3.	Components of postcalibration parameter error.....	65
A4.4.	Schematic description of calibration-constrained predictive maximization/minimization	68

Approaches to Highly Parameterized Inversion: A Guide to Using PEST for Model-Parameter and Predictive-Uncertainty Analysis

By John E. Doherty^{1,2}, Randall J. Hunt³, and Matthew J. Tonkin⁴

Abstract

Analysis of the uncertainty associated with parameters used by a numerical model, and with predictions that depend on those parameters, is fundamental to the use of modeling in support of decisionmaking. Unfortunately, predictive uncertainty analysis with regard to models can be very computationally demanding, due in part to complex constraints on parameters that arise from expert knowledge of system properties on the one hand (*knowledge constraints*) and from the necessity for the model parameters to assume values that allow the model to reproduce historical system behavior on the other hand (*calibration constraints*).

Enforcement of knowledge and calibration constraints on parameters used by a model does not eliminate the uncertainty in those parameters. In fact, in many cases, enforcement of calibration constraints simply reduces the uncertainties associated with a number of broad-scale combinations of model parameters that collectively describe spatially averaged system properties. The uncertainties associated with other combinations of parameters, especially those that pertain to small-scale parameter heterogeneity, may not be reduced through the calibration process. To the extent that a prediction depends on system-property detail, its postcalibration variability may be reduced very little, if at all, by applying calibration constraints; knowledge constraints remain the only limits on the variability of predictions that depend on such detail. Regrettably, in many common modeling applications, these constraints are weak.

Though the PEST software suite was initially developed as a tool for model calibration, recent developments have focused on the evaluation of model-parameter and predictive uncertainty. As a complement to functionality that it provides for highly parameterized inversion (calibration) by means of formal mathematical regularization techniques, the PEST suite provides utilities for linear and nonlinear error-variance and

uncertainty analysis in these highly parameterized modeling contexts. Availability of these utilities is particularly important because, in many cases, a significant proportion of the uncertainty associated with model parameters—and the predictions that depend on them—arises from differences between the complex properties of the real world and the simplified representation of those properties that is expressed by the calibrated model.

This report is intended to guide intermediate to advanced modelers in the use of capabilities available with the PEST suite of programs for evaluating model predictive error and uncertainty. A brief theoretical background is presented on sources of parameter and predictive uncertainty and on the means for evaluating this uncertainty. Applications of PEST tools are then discussed for overdetermined and underdetermined problems, both linear and nonlinear. PEST tools for calculating contributions to model predictive uncertainty, as well as optimization of data acquisition for reducing parameter and predictive uncertainty, are presented. The appendixes list the relevant PEST variables, files, and utilities required for the analyses described in the document.

Introduction

Suppose that the algorithmic basis of a numerical model is such that the model's ability to simulate environmental processes at a site is perfect. Such a model would, of necessity, be complex. Furthermore, it would need to account for the spatial variability of hydraulic and other properties of the system that it is to simulate. If these properties were all known and the model was parameterized accordingly, the model would predict with perfect accuracy the response of the system under study to a set of user-supplied inputs.

In this document, the word "parameter" is used to describe a number specified in a model that represents a property of the system that the model purports to represent. For spatially distributed models such as those used to describe movement of groundwater and surface waters and/or contaminants contained therein, many hundreds, or even hundreds of thousands, of such numbers may be required by a model.

¹Watermark Numerical Computing, Brisbane, Australia

²National Centre for Groundwater Research and Training, Flinders University, Adelaide SA, Australia.

³U.S. Geological Survey.

⁴S.S. Papadopoulos & Associates, Bethesda, MD

Furthermore, in many models, parameters show time as well as spatial dependence, this adding further to the number of parameters that models may require. To the extent that any one of these numbers is wrong, so too may be any model outcome that depends on it.

Inevitably, the model is not a perfect simulator as the parameter field used by a model is a simplified representation of real-world system property variability. This parameter-field simplification is partly an outcome of simplifications required for model algorithmic development and/or for numerical implementation of the model algorithm. For example, there is a computational limit to the number of active nodes that a two- or three-dimensional discretized numerical model can employ, system property averaging is implicit in the algorithmic design of lumped-parameter hydrologic models, and time-stepping schema used by transient models require temporal averaging of model inputs and the time-varying parameters through which those inputs are processed. To the extent that a model's predictions depend on finer spatial or temporal parameter detail than is represented in a model, those predictions have a potential for error. As used here, "error" refers to the deviation of the best estimate possible of the quantity compared to the true value; recognition of this potential for error constitutes acceptance of the fact that model predictions are uncertain.

Rarely, however, is the potential for parameterization-based model predictive error limited by the inability of a model to represent spatial and temporal heterogeneity of its parameters—especially in modern computing environments where computational limits on cell and element numbers are rapidly shrinking. Instead, in most cases, the potential for model predictive error is set by an inability on the part of the modeler to supply accurate parameterization detail at anything like the fine spatial and temporal scale that most models are capable of accommodating. Expert site knowledge, supported by point measurements of system properties, simply cannot furnish knowledge of these properties at the level of detail that a model can represent. Hence, the assignment of parameters to a complex, distributed parameter model is not an obvious process. Moreover, meeting some minimum level of parameter complexity is critical because model oversimplification can confound uncertainty analyses, and the appropriate level of simplification can change as a model objective changes (Box 1).

Box 1: The Importance of Avoiding Oversimplification in Prediction Uncertainty Analysis

Simplified models can be appropriate for making environmental predictions that do not depend on system detail and for exploration of the uncertainty associated with those predictions. However, to the extent that the model's simplification misrepresents or omits salient details of the system simulated, the prediction is not only susceptible to error: in fact, the extent of this possible error cannot be quantified (Moore and Doherty, 2005). Given the direct relation to the model objective, there is a concern that a model might be simplified appropriately for one modeling objective but then misused in subsequent analysis that depends on parameterization or process detail omitted.

For example, one robust approach for extracting the greatest value from limited monitoring resources is linear analysis of the difference in prediction uncertainty with or without specified observation data. Because of its linear basis, this evaluation does not require that the actual observation *values* are known at proposed monitoring locations. Rather, it requires only that the *sensitivities* of proposed observations to model parameter perturbation be known. This sensitivity can be calculated at any stage of the calibration process (even before this process commences). Such an analysis can thus be done during either an early or late phase of an environmental investigation. Misapplication of the simple model, however, can lead to error when assessing the worth of data collection, because confounding artifacts in the calculated sensitivities that result from oversimplification can cloud insight resulting from inclusion of data sensitive to unrepresented detail. For example, how can the subtle information contained in a series of closely spaced proposed observation well locations "be heard above the clamor of misinformation encapsulated in the requirement that large parts of a model domain possess spatially invariant properties, that the boundaries between these parts are at exactly known locations, and that these boundaries are marked by abrupt hydraulic property changes" (Doherty and Hunt, 2010)? The concern centers on the possibility that outcomes of data-worth analysis in such oversimplified models are more reflective of parameter-simplification devices than of the true information content of hypothetical data collected.

To illustrate this issue, Fienen and others (2010) used a model developed by Hoard (2010) to optimize future data-collection locations in order to maximize reduction in the uncertainty of a prediction on the amount of groundwater level change in area of interest (containing, for example, an endangered species) in response to regional pumping. For the case discussed by Fienen and others, the goal of the monitoring is to assess the effect of a new high-capacity pumping well (500 gal/min) situated near a headwater stream in an ecologically sensitive area (figure B1–1). The value of future data is estimated by quantifying the reduction in prediction uncertainty achieved by adding potential observation wells to the existing model-calibration dataset. The reduction in uncertainty is calculated for multiple potential locations of observations, which then can be ranked for their effectiveness for reducing uncertainty associated with the specified prediction of interest. In Fienen and others (2010), a Bayesian approach was implemented by use of the PEST PREDUNC utility (Doherty, 2010).

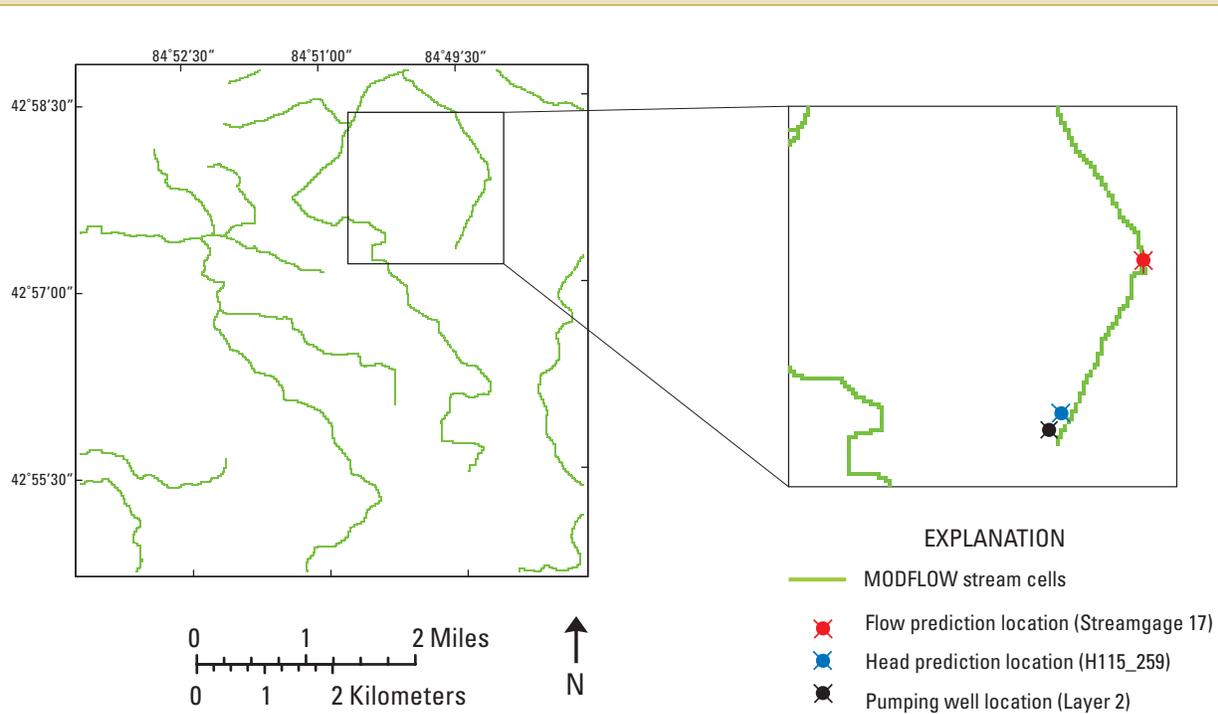


Figure B1-1. Local model domain and the locations of the pumping well, head prediction (H115_259), and streamgage. (Figure modified from Hoard, 2010).

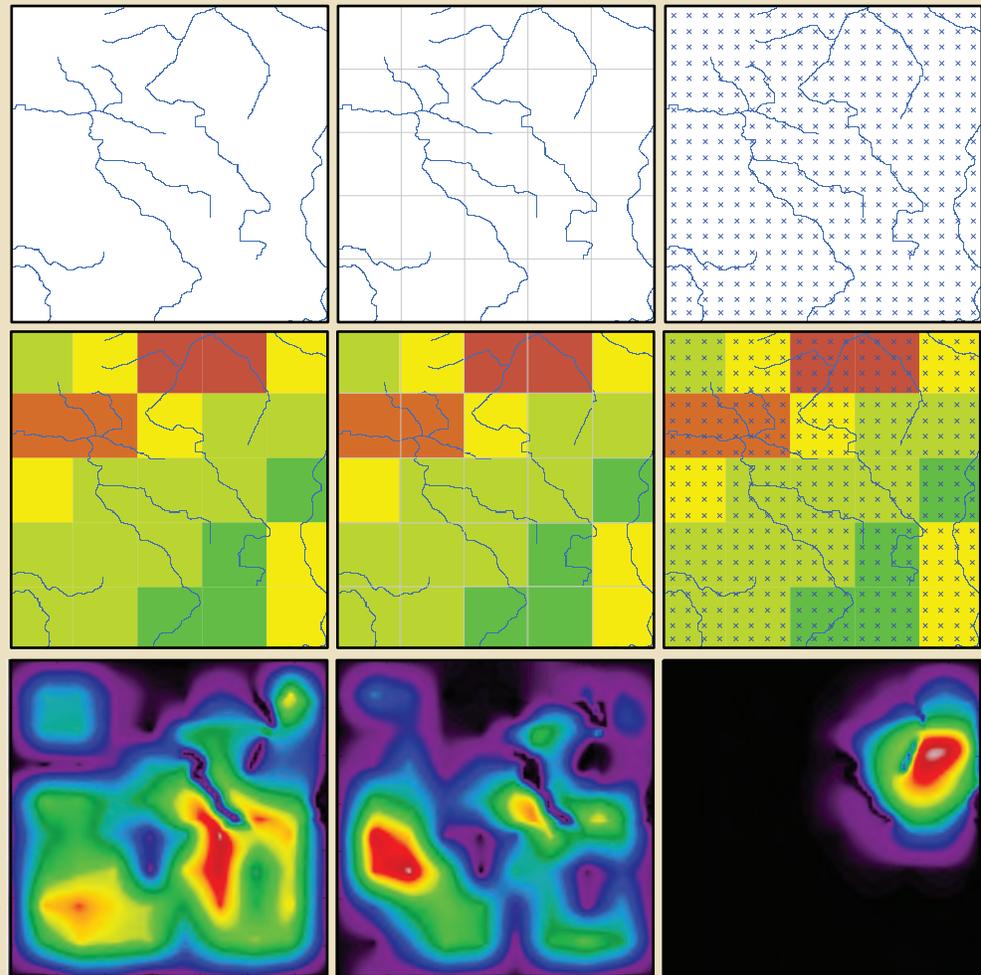
To demonstrate the effect of parameterization flexibility on data-worth analyses, three parameterization resolutions were investigated (fig. B1-2): (1) a hydraulic conductivity (K) layer-multiplier ("1-parameter") approach in which a single multiplier is applied to all horizontal and vertical hydraulic-conductivity values in each layer inherited from the regional model, (2) a 25-parameter version of the K field ("25-parameter") in which the 5×5 zone parameterization inherited from the regional model was used to directly define 25 K zones, and (3) a 400-parameter approach using a 20×20 grid of pilot points to represent hydraulic-conductivity parameterization. It is important to note that only the parameter flexibility specified for the data-worth analysis was being varied in the three cases described above; the actual hydraulic conductivity values input into the hydrologic model were exactly the same in all three cases, and equal to those inherited from the calibrated regional model.

The 1-parameter case represents an end extreme of oversimplification, as might happen when the model used for data-worth analysis adheres closely to the parameterization scheme obtained through regional model calibration; that is, the inset model allows the surface-water features to be more refined, but the local aquifer properties are not. The 25-parameter case was chosen as the more typical case; that is, the parameter flexibility appropriate for the regional model is assumed to be appropriate for the data-worth calculation, when used in conjunction with the additional surface-water-feature refinement of the higher grid resolution of the inset model. This level of parameterization can be thought of as typifying the number of zones that might be used in a traditional calibration approach. The 400-parameter case represents a highly parameterized example typical of a regularized inversion approach that aims to interject sufficient parameter flexibility such that the confounding artifacts associated with oversimplification of a complex world are minimized.

The results of data-worth calculations pertaining to the addition of new head observations for the head prediction by the model are contoured and displayed on a map in figure B1-2. The extent of the map is the same as the model domain and panels in figure B1-1 depict results for the first layer for all three parameterization strategies. The differences in the displayed values from left panel to right reflect the progressively more flexible parameterization of hydraulic conductivity, from a single layer multiplier at left (1 parameter) through a 5×5 grid of homogeneous zones (25 parameters) to a 20×20 grid of pilot points (400 parameters) at the right. Two major trends are evident when comparing the parameterization scenarios: first, counterintuitive artifacts are encountered in the low level (1-parameter) and intermediate levels (25-parameter) of parameter flexibility. These artifacts are

Box 1: The Importance of Avoiding Oversimplification in Prediction Uncertainty Analysis (continued)

Figure B1–2. Parameter discretization (top row), hydraulic conductivity field seen by model (middle row), and results of data-worth analysis (bottom row; warm colors = higher reduction in prediction uncertainty). Figure modified from Fienen and others (2010).



counterintuitive results because the areas reported as most important for reducing the prediction uncertainty of groundwater levels between the well and the headwater stream are distant from both the stress and the related prediction. Inspection of the locations of greatest data worth suggests that high reported data worth is associated with zone boundaries and intersections—a factor introduced by the modeler when the parameter flexibility was specified. When same data-worth analysis is performed by using the highly parameterized 400-parameter case, locations of higher values of data worth are in places where intuition suggests—the area near both the stress and the prediction. In other words, the parameter flexibility afforded by the 400-parameter case reduces structural noise sufficiently so that one can discern the difference in importance of a potential head location.

This work demonstrates that the resolution of the parameter flexibility required for a model is a direct result of the resolution of the question being asked of the model. When the model objective changed to the worth of future data collection and became smaller scale (ranking the data-worth of one observation well location over a nearby location), a parameter flexibility level was needed that was commensurate with the spacing of the proposed observation wells, not the regional model calibration targets. Note that it is the parameterization *flexibility* that is required, not different parameter *values* specified in the model input (because the actual parameter values were identical in all three cases). ■

The uncertainty associated with model parameterization can often be reduced by constraining parameters such that model outputs under historical system stresses reproduce historical measurements of system state. However, as is widely documented, these constraints can be enforced only on the broad-scale spatial or temporal variability of a limited number of parameter types; meanwhile, the typical calibration process exercises few, if any, constraints on fine-scale spatial or temporal parameter variability. To the extent that a model prediction is sensitive to parameter detail, its uncertainty may therefore be reduced very little by the need for model outputs to replicate historical system behavior as observed at a few, or even many, locations and times.

Because parameter and predictive uncertainty is unavoidable, justification for the use of a model in environmental management must not rest on an assumption that the model's predictions will be correct. Rather, justification for its use must rest on the premises that its use (a) enables predictive error and/or uncertainty to be quantified and (b) provides a computational framework for reducing this predictive error and/or uncertainty to an acceptable level, given the information that is available. As such, by quantifying the uncertainty associated with predictions of future environmental behavior, associated risk can be brought to bear on the decisionmaking process.

Purpose and Scope

The intent of this document is to provide its reader with an overview of methods for model-parameter and predictive-uncertainty analysis that are available through PEST and its ancillary utility software. PEST is public domain and open source. Together with comprehensive documentation given by Doherty (2010a,b), it can be downloaded from the following site:

<http://www.pesthomepage.org>

As is described in Doherty and Hunt (2010), PEST is model-independent in the sense that it communicates with a model through the model's input and output files. As a result, no programming is required to use a model in conjunction with PEST; furthermore, the "model" that is used in conjunction with PEST can be a batch or script file of arbitrary complexity, encompassing one or a number of discrete executable programs. Other software suites implement model-independent parameter estimation, and to some extent, model predictive-uncertainty analysis; see for example OSTRICH (Matott, 2005) and UCODE-2005 (Poeter and others, 2005). However, PEST is unique in that it implements model calibration and uncertainty analysis in conjunction with highly parameterized models. A unique solution to the inverse problem of model calibration is achieved through the use of mathematical regularization devices that can be implemented individually or in concert (see, for example, Hunt and others, 2007). Some benefits of a highly parameterized approach to model calibration

and uncertainty analysis versus more traditional, overdetermined approaches include the following:

1. In calibrating a model, maximum information can be extracted from the calibration dataset, leading to parameters and predictions of minimized error variance.
2. The uncertainty associated with model parameters and predictions is not *underestimated* by eliminating parameter complexity from a model to achieve a well-posed inverse problem.
3. The uncertainty associated with model parameters and predictions is not *overestimated* through the need to employ statistical correction factors to accommodate the use of oversimplified parameter fields.

As a result, the tendency for predictive uncertainty to rise in proportion to its dependence on system detail is accommodated by the explicit representation of parameter detail in highly parameterized models, notwithstanding the fact that unique estimation of this detail is impossible.

The topic of model-parameter and predictive uncertainty is a vast one. Model-parameter and predictive-uncertainty analyses encompass a range of important factors such as errors introduced by model-design and process-simulation imperfections, spatial- and/or temporal-discretization artifacts on model outputs and, perhaps most unknowable, contributions to predictive uncertainty arising from incomplete knowledge of future system stresses (Hunt and Welter, 2010). Therefore, comprehensive coverage of this topic is outside of our scope. Rather, we present tools and approaches for characterizing model-parameter and predictive uncertainty and model-parameter and predictive error that are available through the PEST suite. In general terms, *predictive-error* analyses evaluate the potential for error in predictions made by a calibrated model using methods based upon the propagation of variance, whereas *predictive-uncertainty* analysis is used herein as a more encompassing and intrinsic concept, which acknowledges that many realistic parameter sets enable the model to reproduce historic observations.

Strictly speaking, many of the methods described in this document constitute error analysis rather than uncertainty analysis, because the theoretical underpinnings of the methods are based upon error-propagation techniques; however, the application of some of these methods blurs the distinction between error and uncertainty analysis. In particular, the null-space Monte-Carlo technique described later incorporates several developments that render it more akin to uncertainty analysis than error analysis. Throughout this document, the term "predictive uncertainty" is used as a catchall term; however, we have tried to use the terms "predictive error" and "predictive uncertainty" appropriately when discussing specific methods of analysis.

The PEST software suite is extensively documented by Doherty (2010a,b); as such, lengthy explanation of all PEST functions and variables is beyond the scope of this report. Rather, the focus is on guidelines for applying PEST tools to

groundwater-model calibration. The presentation is intended to have utility on two levels. Advanced PEST users can go directly to specific sections and obtain guidelines for specific parameter-estimation and uncertainty-analysis operations; intermediate users can read through a logical progression of typical issues faced during calibration and uncertainty analysis of highly parameterized groundwater models. Appendixes document the relation of PEST variables and concepts used in the report body to the broader PEST framework, terminology, and definitions given by Doherty (2010a,b). Thus, this document is intended to be an application-focused companion to the full scope of PEST capabilities described in the detailed explanations of Doherty (2010a,b) and theory cited by references included therein. Given the similar presentation style and approach, it can also be considered a companion to the regularized inversion guidelines for calibration of groundwater models given by Doherty and Hunt (2010).

Summary and Background of Underlying Theory

Descriptions given herein are necessarily brief, and mathematical foundations are referenced rather than derived, so that we may focus on appropriate application rather than already published theoretical underpinnings of regularized inversion. Detailed description of the theoretical basis of the approach described herein can be found in Moore and Doherty (2005), Christensen and Doherty (2008), Tonkin and others (2007), Tonkin and Doherty (2009), and Doherty and Welter (2010). For convenience, a summary mathematical description of the material discussed below is presented in appendix 4.

In general, errors associated with important predictions made by the model derive from two components:

1. *Effects of measurement noise.*—Exact estimation of appropriate parameter values is not possible because of noise inherent in measurements used for calibration. Thus, uncertainty in predictions that depend on these parameter combinations can never be eliminated—it can only be reduced.
2. *Failure to capture complexity of the natural world salient to a prediction.*—This component represents the contribution to error that results from the conceptual, spatial, and temporal simplifications made during modeling and model calibration. Predictive uncertainty from uncaptured complexity reflects heterogeneity that is beyond the ability of the calibration process to discern.

This second term is often the dominant contributor to errors in those predictions that are sensitive to system detail (Moore and Doherty, 2005).

In order to develop representative estimates of parameter and predictive uncertainty, both of the above components must be considered. In the brief overview presented here, the focus is on estimates in which a linear relation between model

parameters and model outputs is assumed. Linear approaches are more computationally efficient than nonlinear approaches; however, linear approaches have the disadvantages that they (a) rely on differentiability of model outputs with respect to adjustable parameters and (b) can introduce errors into the uncertainty-analysis process of nonlinear systems.

General Background

The foundation for most methods of linear uncertainty analysis is the Jacobian matrix, a matrix that relates the sensitivity of changes to model parameters to changes in model outputs. Model outputs are those for which field measurements are available for use in the calibration process, or those that constitute predictions of interest. The model is parameterized to a level of appropriate complexity, defined here as a level of parameter density that is sufficient to ensure that minimal errors to model outputs of interest under calibration and predictive conditions are incurred through parameter simplification. Thus, *all* parameter detail that is salient to model predictions of interest has been incorporated into the model's parameterization scheme. In practice, this condition is often not met, of course. Because a high level of parameterization is needed to reach this appropriate complexity thus defined, it is unlikely that unique estimates for all parameters can be obtained on the basis of the calibration dataset. As a result, the inverse problem of model calibration is underdetermined, or ill posed.

Before calibration begins, a modeler can estimate the precalibration uncertainty associated with parameters, often by using a geostatistical framework such as a variogram. More often than not, however, estimated precalibration uncertainty will be the outcome of professional judgment made by those with knowledge of the site modeled. This information can be encapsulated in a covariance matrix of uncertainty associated with model parameters. This matrix referred to herein as the “ $C(\mathbf{p})$ covariance matrix of innate parameter variability.” This estimate of uncertainty should reflect the fact that exact parameter values are unknown but that some knowledge of the range of reasonable values of these properties does exist. Precalibration predictive uncertainty can then be calculated from precalibration parameter uncertainty through linear propagation of covariance (if the model is linear) or through Monte Carlo analysis based on many different parameter sets generated on the basis of the $C(\mathbf{p})$ matrix of innate parameter variability.

Parameter and Predictive Uncertainty

Calculation of predictive uncertainty in this way does not account for the fact that parameter sets that do not allow the model to replicate historical measurements of system state should have their probabilities reduced in comparison with those that do. The idea of calibration-constrained parameter variability is formally expressed by Bayes' equation

(Eq. A4.4 - see appendix 4 for further details). It is interesting to note that Bayes' equation makes no reference to the term "calibration," notwithstanding the ubiquitous use of "calibrated model" in environmental management. In fact, Bayes' equation suggests that use of a single parameter set to make an important model prediction should be avoided because this practice does not reflect the degree of parameter and predictive uncertainty inherent in most modeling contexts. It is more conceptually consistent with Bayes' equation to make a prediction from many different parameter sets, all of which are plausible on the basis of the user-specified $C(\mathbf{p})$ matrix and all of which provide an acceptable replication of historical system behavior by providing an adequate fit to historical observation data (where "adequate" is judged on the basis of errors associated with these observations). Nevertheless, justification for use of a single parameter set in the making of model predictions of interest may be based on the premise that this set is of minimum error variance. However, minimum error variance does not necessarily mean small error variance. As Moore and Doherty (2005; 2006) point out, predictions made by means of a calibrated model can be considerably in error, even though they are of minimum error variance and hence constitute "best estimates" of future system behavior.

The goal of the calibration process is then to find a unique set of parameters that can be considered to be of minimum error variance and that can be used to make predictions of minimum error variance. Formal solution of the inverse problem of model calibration in a way that attains parameter fields that approach this condition can be achieved by using mathematical regularization techniques. These techniques often incorporate soft knowledge of parameters, thereby making reference to the prior-probability term of Bayes' equation. However, because the parameter field so attained is unlikely to be correct, even though it has been tuned to reduce its wrongness to the level possible, predictions made on the basis of this parameter field will probably be in error. Quantification of this error requires that parameter values be explicitly or implicitly varied over a range that is dictated by their $C(\mathbf{p})$ matrix of innate parameter variability while maintaining acceptable replication of historical system behavior. The means by which this can be achieved is the subject matter of this document. Furthermore, as has been stated above, it will be assumed that in quantifying the strong potential for error associated with predictions made on the basis of a complex parameter field, the uncertainty of those predictions as described by Bayes' equation will be approximately quantified. Though computationally intensive, this result will be achieved at a far smaller computational cost than through direct use of Bayes' equation.

The Resolution Matrix

The exact values of parameters attained through regularized inversion depend on the means through which mathematical regularization is implemented. Some regularization methodologies are better used in some modeling contexts than

in others; the "correct" method in any particular context is often a compromise between attainment of strictly minimum error variance solution to the inverse problem on one hand and maintenance of numerical stability on the other. Regardless of the regularization methodology used, the postcalibration relations between estimated parameters and their real-world counterparts is given by the so-called resolution matrix, which is available as a byproduct of the regularized inversion process; see appendix 4, as well as texts such as Aster and others (2005). For a well-posed inverse problem, the resolution matrix is in fact the identity matrix. Where the inverse problem is ill-posed and parameters cannot be estimated uniquely, the resolution matrix is rank-deficient (that is, there is not perfect knowledge of all estimated parameters and their real-world equivalents). In most cases of practical interest, the resolution matrix will have diagonal elements that are less than unity and will possess many off-diagonal elements. In such cases, each row of the resolution matrix is composed of factors by which real-world parameters are multiplied and then summed in order to achieve the estimated parameter corresponding to that row. As such, the resolution matrix depicts the manner in which the complex parameter distribution within the real world is simplified or "smudged" in order to attain the unique set of parameters that is associated with the calibrated model.

Where regularization is achieved through singular value decomposition, the resolution matrix becomes a projection operator onto a subset of parameter space comprising combinations of parameters that can be estimated uniquely on the basis of the current calibration dataset (Moore and Doherty, 2005); this subset is referred to as the "calibration solution space" herein. Orthogonal to the solution space is the "calibration null space," which can be thought of as combinations of parameters that cannot be estimated on the basis of the calibration dataset. Their inestimability is founded in the fact that any parameter set that lies entirely within the null space can be added to any solution of the inverse problem with no (or minimal) effects on model outputs that correspond to field measurements; hence, the addition of these parameter combinations to a calibrated parameter maintains the model in a calibrated state. Figure 1 depicts this situation graphically.

So-called "singular values" are associated with unit vectors defined by the singular-value-decomposition process that collectively span parameter space. Those associated with high singular value magnitudes span the calibration solution space, whereas those associated with low or zero singular value magnitudes span the calibration null space. "Truncation" of low singular values provides a demarcation or a threshold between solution and null spaces. Including parameters associated with low singular values in the solution space would lead to an increase, rather than a decrease, in the error variance of parameter or prediction of interest due to the amplification of measurement noise in estimating parameter projections into unit vectors associated with small singular value (Moore and Doherty, 2005; Aster and others, 2005).

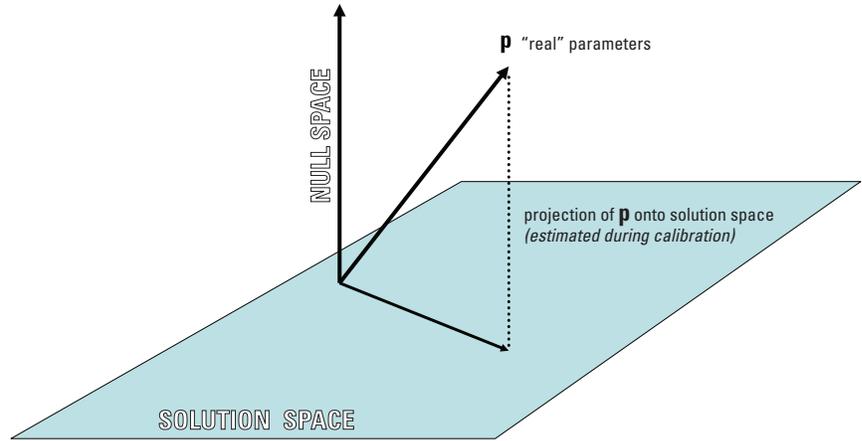


Figure 1. Relations between real-world and estimated parameters where model calibration is achieved through truncated singular value decomposition.

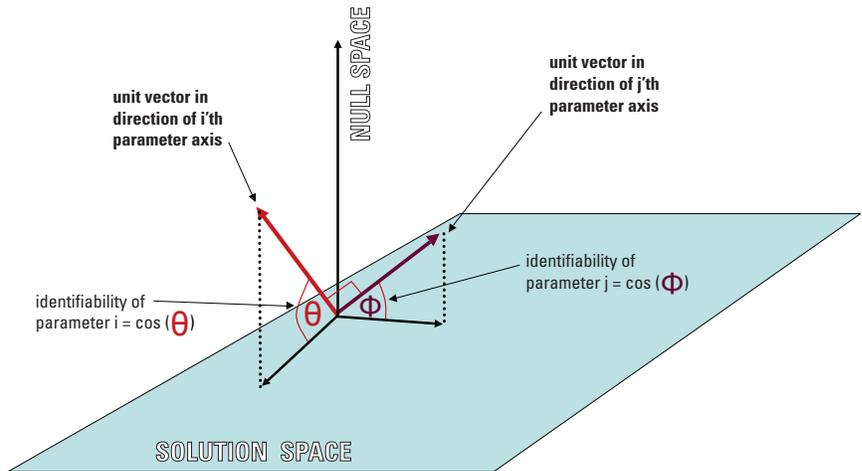


Figure 2. Schematic depiction of parameter identifiability, as defined by Doherty and Hunt (2009).

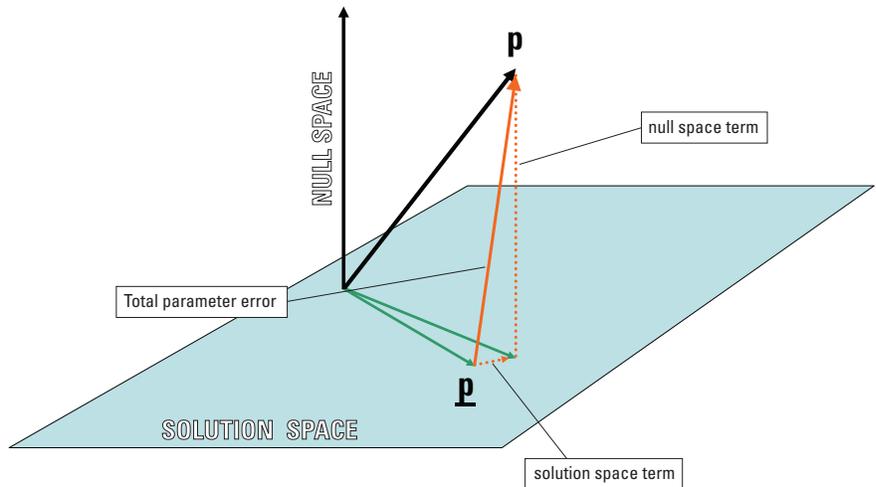


Figure 3. Components of postcalibration parameter error.

As stated above, each row of the resolution matrix defines the averaging or integration process through which a single estimated parameter is related to real-world parameters. In most real-world calibration contexts, the resolution matrix is not strongly diagonally dominant, and it may not be diagonally dominant at all. As a result, each estimated parameter may, in fact, reflect real-world properties over a broad area, or even properties of entirely different type. This “bleed” between parameters is an unavoidable consequence of the need for uniqueness that is implicit in the notion of model calibration (Moore and Doherty, 2006; Gallager and Doherty, 2007a).

The threshold between the null and solution spaces is a general mathematical construct that allows a formal investigation of model error and uncertainty. This general concept can be used to explore the tension between the need for model uniqueness and a world that often cannot be constrained uniquely. A useful insight gained from singular value decomposition of a resolution matrix is its implicit definition of the “identifiability” of each model parameter. As Doherty and Hunt (2009a) discuss, the identifiability of a parameter can be defined as the direction cosine between each parameter and its projection into the calibration solution space; this is calculable as the diagonal element of the resolution matrix pertaining to each parameter. Where this value is zero for a particular parameter, the calibration dataset possesses no information with respect to that parameter. Where it is 1, the parameter is completely identifiable on the basis of the current calibration dataset (though cannot be estimated without error because its estimation takes place on a dataset that contains measurement noise). These relations are diagrammed in figure 2.

Moore and Doherty (2005) demonstrate that even where regularization is implemented manually through precalibration parameter lumping, a resolution matrix must inevitably accompany the requirement for uniqueness of model calibration. In most cases, however, the resolution matrix achieved through this traditional means of obtaining a tractable parameter-estimation process will be suboptimal, in that its diagonal elements will possess lower values, and its off-diagonal elements higher values, than those achieved through use of mathematical regularization. See Moore and Doherty (2005) for a more complete discussion of this issue.

Parameter Error

Where calibration is viewed as a projection operation onto an estimable parameter subspace, it becomes readily apparent that postcalibration parameter error is composed of two terms, irrespective of how regularization is achieved. The concept is most easily pictured when implemented by means of truncated singular value decomposition, where these two terms are orthogonal. Using this construct, the two contributors to postcalibration parameter error are the “null-space term” and the “solution-space term” (see fig. 3). As detailed in Moore and Doherty (2005), the “null-space term” arises from the necessity to simplify when seeking a unique parameter field to calibrate a model. The “solution-space term” expresses

the contribution of measurement noise to parameter error. It is noteworthy that uncertainty analysis performed as an adjunct to traditional parameter estimation based on the solution of a well-posed inverse problem cannot encompass the full extent of the null-space term (Doherty and Hunt, 2010), notwithstanding that regularization is just as salient to attainment of parameter uniqueness, because it is implemented manually rather than mathematically.

Because the parameter set that characterizes reality cannot be discerned, parameter error (that is, the difference between the estimated and true parameter set) cannot be calculated. However, the *potential* for parameter error can be expressed in probabilistic terms, as can the *potential* for error in predictions calculated on the basis of these parameters. As stated previously, these are loosely equated to parameter and predictive *uncertainty* herein. It is thus apparent that model predictive uncertainty (like parameter uncertainty) also has two sources: that which arises from the need to simplify (because of an inability to estimate the values of parameter combinations composing the calibration null space) and that which arises from measurement noise within the calibration dataset.

The need to select a suitable singular value truncation point where regularization is implemented via truncated singular value decomposition has already been mentioned. Moore and Doherty (2005), Hunt and Doherty (2006) and Gallager and Doherty (2007b) discuss this matter in detail. Conceptually, selection of a suitable truncation point represents a tradeoff between reduction of structural error incurred through oversimplification on the one hand and contamination of parameter estimates through overfitting on the other hand. The damaging effect of overfitting arises from the fact that the contribution of measurement noise to potential predictive error rises as the truncation point shifts to smaller and smaller singular values; the ratio of measurement noise to potential predictive error eventually becomes infinity where singular values become zero. However, the temptation to overfit results from the fact that more parameter combinations can be included in the solution space as the truncation point moves to higher and higher singular values, with the result that the contribution of the null-space term to overall predictive error decreases as the truncation point shifts to smaller singular values. The sum of these two terms (that is, the total error variance associated with the prediction) typically falls and then rises again (fig. 4). Ideally, truncation should occur where the total error variance associated with predictions of interest is minimized.

The predictive error that is associated with a singular value of zero is equal to the precalibration uncertainty of that prediction. The difference in error variance between this error variance and that associated with the minimum of the predictive-error-variance curve is a measure of the benefits gained through calibrating the model. In most circumstances the total predictive-error-variance curve will show a monotonic fall to its minimum value, then a monotonic rise with increasing singular value. In some circumstances, however, the total predictive-error-variance curve can rise before it falls.

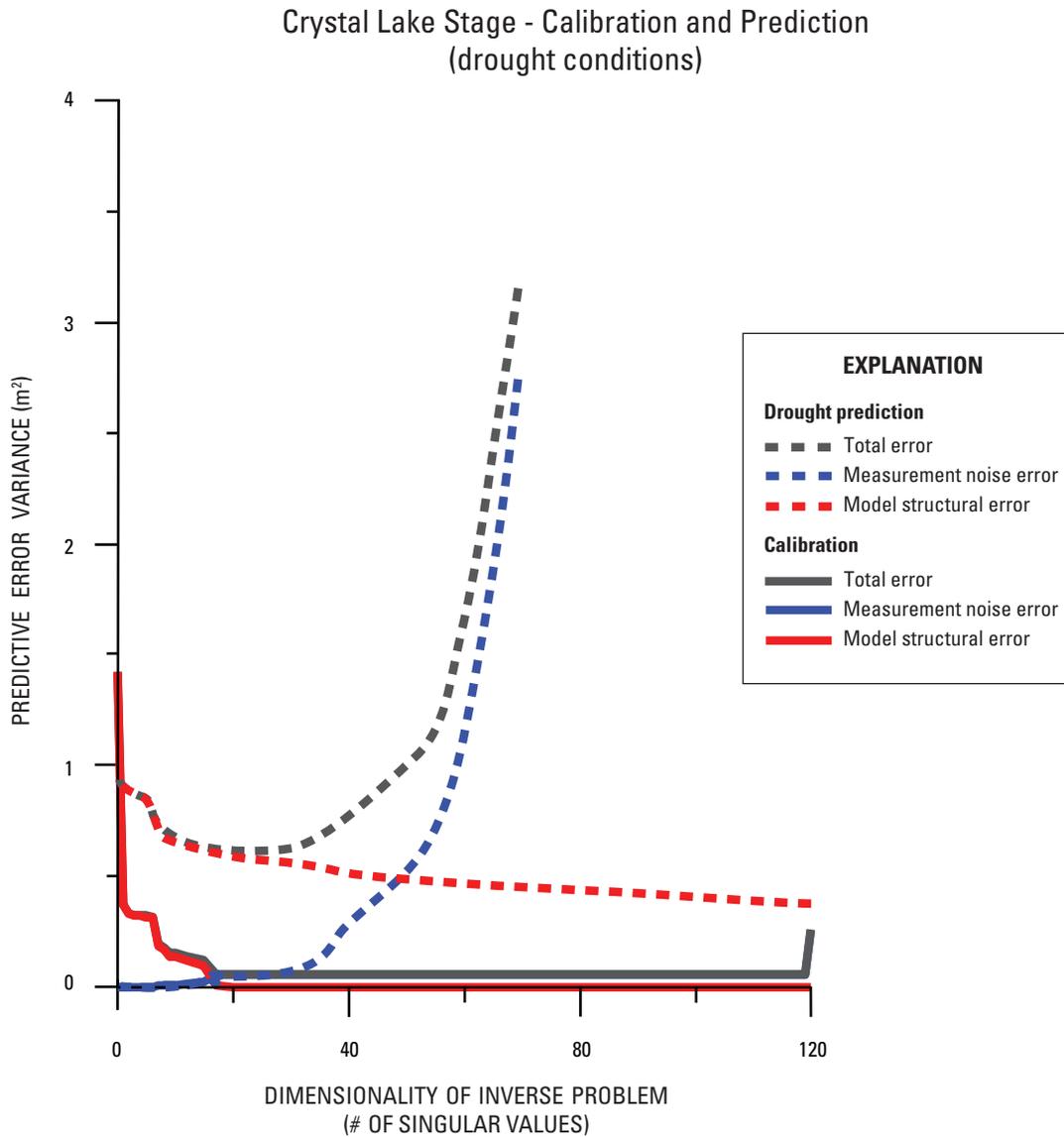


Figure 4. Contributions to total predictive error variance calculated by use of the PEST PREDVAR1 and PREDVAR1A utilities from Hunt and Doherty (2006). Total predictive error variance is the sum of simplification and measurement error terms; both of these are a function of the singular value truncation point.

This behavior can occur where parameters have appreciably different innate variabilities. Under these circumstances, the calibration process may actually transfer a potential for error from parameters whose precalibration uncertainty is high to those for which it is not, thereby endowing the latter with a potential for error after calibration that they did not possess before calibration.

Though not common, this phenomenon can be prevented if the calibration is formulated in such a way that it estimates scaled rather than native parameter values, with scaling being such as to normalize parameters by their innate variabilities

(Moore and Doherty, 2005). Alternatively, if regularization is achieved manually through precalibration fixing of certain parameters, those whose pre-calibration uncertainty is smallest should be fixed, leaving those with greater innate variability to be estimated through the calibration process. If parameters with greater innate variability are fixed, those with less innate variability may inherit error from those with more, and the calibration process may fail to achieve a minimum-error-variance parameter field, which is needed for making minimum-error-variance predictions.

Linear Analysis

Many of the concepts discussed above have their roots in linear analysis, where a model is conceived of as a matrix acting on a set of parameters to produce outputs. Some of these outputs are matched to measurements through the process of model calibration. Others correspond to forecasts required of the model. Equations relevant to this type of analysis are presented in the appendix 4.

Reference has already been made to several outcomes of linear analysis; for example, the resolution matrix, parameter identifiability, and—indeed—the concept of solution and null spaces. The concepts that underpin these linear-analysis outcomes are just as pertinent to nonlinear analysis as they are to linear analysis. However, the actual values associated with the outcomes of linear analysis will not be as exact as those forthcoming from nonlinear analysis. Nevertheless, linear analysis provides the following advantages:

1. It is, in general, computationally far easier than nonlinear analysis.
2. The outcomes of this analysis provide significant insights into the sources of parameter and predictive error, as they do into other facets of uncertainty analysis (as will be demonstrated shortly).
3. The outcomes of the analysis are independent of the value of model parameters and hence of model outcomes. As will be discussed shortly, this makes outcomes of the analysis particularly useful in assessing such quantities as the worth of observation data, for the data whose worth is assessed do not need to have actually been gathered.

Much of the discussion so far has focused on parameter and predictive error and their use as surrogates for parameter and predictive uncertainty. By focusing in this way, insights are gained into the sources of parameter and predictive uncertainty. Linear formulations presented in appendix 4 show how parameter and predictive error variance can be calculated. Other formulations presented in the appendix demonstrate how the variance of parameter and predictive uncertainty also can be calculated. As is described there, as well as requiring a linearity assumption for their calculation, they require an assumption that prior parameter probabilities and measurement noise are described by Gaussian distributions. In spite of these assumptions, these uncertainty variance formulas (like their error variance counterparts) can provide useful approximations to true parameter and predictive uncertainty, at the same time as they provide useful insights. Of particular importance is that parameter and predictive uncertainties calculated with these formulas are also independent of the values of parameters and model outcomes. Hence, they too can be used for examination of useful quantities such as the observation worth, and they hence form the basis for defining optimal strategies for future data acquisition.

Parameter Contributions to Predictive Uncertainty/Error Variance

Each parameter is not expected to contribute equally to error variance and prediction uncertainty. By use of formulas derived from linear analysis, the uncertainty or error variance associated with a prediction can be calculated both with and without assumed perfect knowledge of one or a number of parameters used by the model. The reduction in predictive uncertainty/error variance accrued through notionally obtaining perfect knowledge of one or more parameters in this manner can be designated as the “contribution” that the parameter or parameters make to the uncertainty or error variance of the prediction in question. Such perfect knowledge can be ascribed to an individual parameter (such as the hydraulic conductivity associated with a single pilot point), or to a suite of parameters (such as all hydraulic conductivities associated with a particular model layer). It can also be ascribed to quantities that would not normally be considered as model parameters (for example, imperfectly known values associated with particular model boundary conditions).

Figure 5 illustrates the outcome of a study of this type. The back row of this graph shows precalibration contributions to predictive uncertainty variance made by different types of parameters, inputs, and boundary conditions used in the water resources model prediction shown in figure 4. The front row shows postcalibration contributions to the uncertainty variance of this same prediction.

Observation Worth

Beven (1993), among others, has suggested that an ancillary benefit accrued through the ability to compute model parameter and predictive uncertainty/error variance is an ability to assess the “worth” of individual observations, or of groups of observations, relative to that of other observations. This process becomes particularly easy if uncertainty/error variance is calculated by using formulas derived from linear analysis.

One means by which the worth of an observation or observations can be assessed is through computing the uncertainty/error variance of a prediction of interest with the calibration dataset notionally composed of only the observation(s) in question. The reduction in uncertainty/error variance below its precalibration level then becomes a measure of the information content, or worth, of the observation(s) with respect to the prediction. A second means by which worth of an observation or observations can be assessed is to compute the uncertainty/error variance of a prediction of interest twice—once with the calibration dataset complete, and once with the pertinent observation(s) omitted from this dataset. The increase in predictive uncertainty/error variance incurred through omission of the observation(s) provides the requisite measure of its (their) worth.

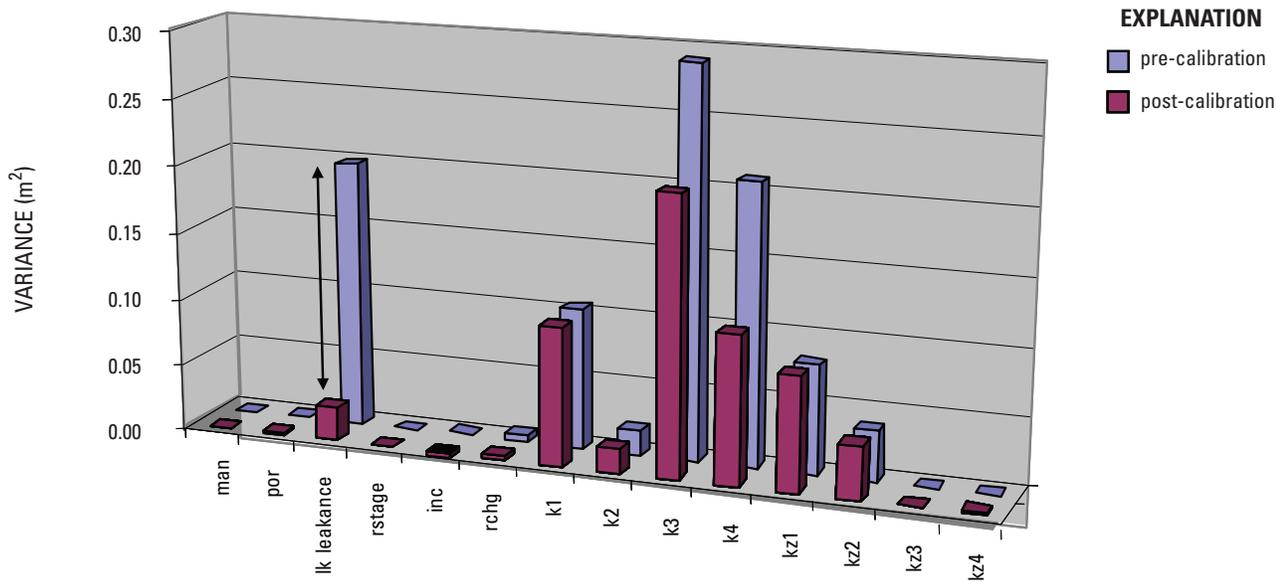


Figure 5. Precalibration and postcalibration contribution to uncertainty associated with the drought lake-stage prediction shown in figure 4. Parameter types used in the model are the following: man=Manning’s n, por=porosity, lk leakage=lakebed leakage, rstage=far-field river stage boundary condition, inc=stream elevation increment boundary condition, rchg=recharge, k1 through k4=Kh of layers 1 through 4, kz1 through kz4=Kz of layers 1 through 4. Note that reduction in the prediction uncertainty accrued through calibration was due primarily to reduction in uncertainty in the lakebed leakage parameter. Thus, less gain is expected from future data-collection activities targeting only this parameter (modified from Hunt and Doherty, 2006).

Optimization of Data Acquisition

As has already been stated, linear analysis is particularly salient to the evaluation of observation worth because the evaluation of the worth of data is independent of the actual values associated with these data. Hence, worth can be assigned to data that are yet to be gathered. This evaluation is easily implemented by computing the reduction in uncertainty/error variance associated with the current calibration dataset accrued through acquisition of further data (for example, Box 1 fig. B1–2). Different data types (including direct measurements of hydraulic properties) can be thus compared in terms of their efficacy of reducing the uncertainty of key model predictions.

Dausman and others (2010) calculated the worth of making both downhole temperature and concentration measurements as an aid to predicting the future position of the saltwater interface under the Florida Peninsula. They demonstrated that in spite of the linear basis of the analysis, the outcomes of data-worth analysis did not vary with the use of different parameter fields, thus making the analysis outcomes relatively robust when applied to this nonlinear model.

Nonlinear Analysis of Overdetermined Systems

Vecchia and Cooley (1987) and Christensen and Cooley (1999) show how postcalibration predictive uncertainty analysis can be posed as a constrained maximization/minimization problem in which a prediction is maximized or minimized subject to the constraint that the objective function rises no higher than a user-specified value. This value is normally specified to be slightly higher than the minimum value of the objective function achieved during a previous overdetermined model calibration exercise.

The principle that underlies this methodology is illustrated in figure 6 for a two-parameter system. In this figure, the shaded contour depicts a region of optimized parameters that correspond to the minimum of the objective function. The solid lines depict objective function contours; the value of each contour defines the objective function for which parameters become unlikely at a certain confidence level. Each contour thus defines the constraint to which parameters are subject as a prediction of interest is maximized or minimized in order to define its postcalibration variability at the same level of confidence. The dashed contour lines depict the dependence of

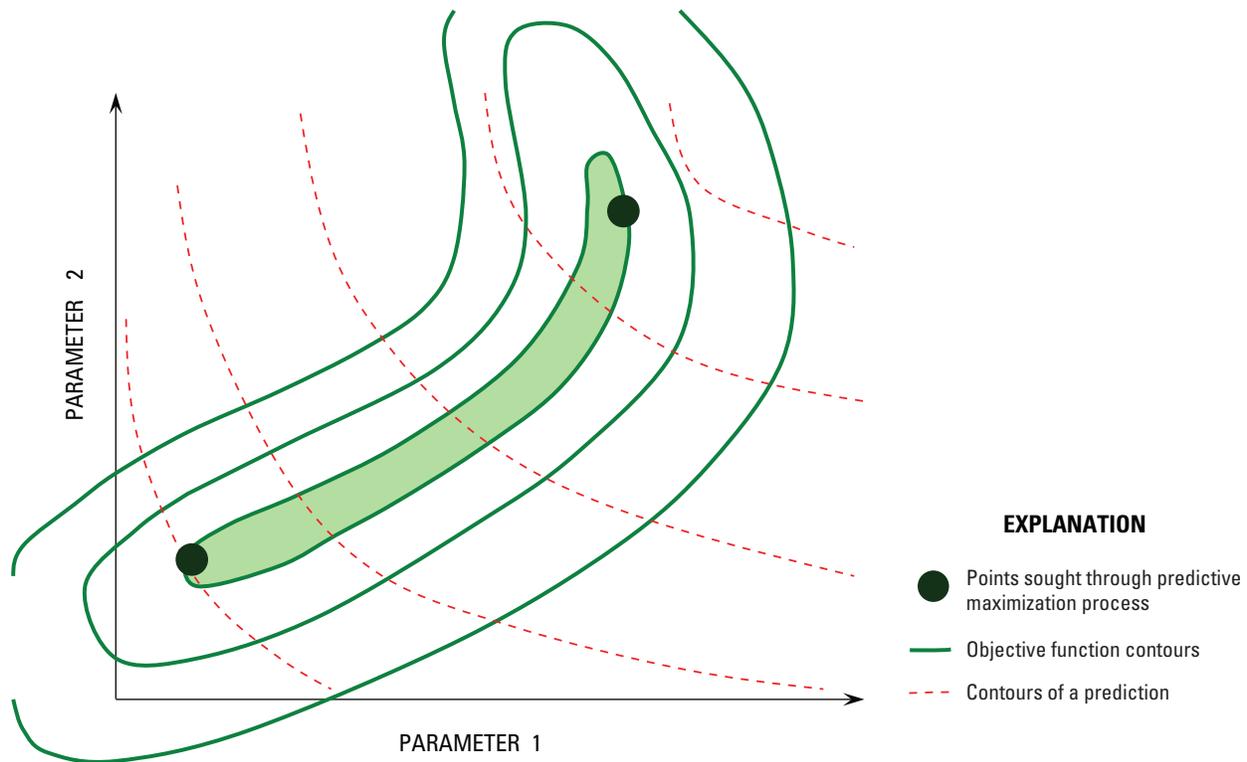


Figure 6. Schematic description of calibration-constrained predictive maximization/minimization.

a prediction on the two parameters. The constrained maximization/minimization process through which the postcalibration uncertainty of this prediction is explored attempts to find the two points marked by circles on the constraining objective-function contour. These points define parameter sets for which the prediction of interest is as high or as low as it can be, while maintaining respect for the constraints imposed by the calibration process. See appendix 4 for further details.

Structural Noise Incurred Through Parameter Simplification

In environmental modeling, calibration problems are overdetermined only because a modeler has done the parameter simplification necessary to achieve this status prior to embarking on the calibration process (Hunt and others, 2007). However, there is a cost associated with this mechanism of obtaining a well-posed inverse problem: a rigorous assessment of postcalibration parameter or predictive error variance must include an assessment of the cost of precalibration regularization that is embodied in this simplification.

Cooley (2004) and Cooley and Christensen (2006) demonstrate a methodology through which one can calculate the potential for nonlinear model predictive error incurred through manual precalibration simplification of complex parameter fields. This methodology constitutes an expansion of the constrained maximization/minimization process described above. They take as their starting point a model domain in which hydraulic properties are assumed to be characterized by a known covariance matrix of innate parameter variability. This matrix will generally include off-diagonal terms to describe continuity of hydraulic properties over appropriate distances. Prior to embarking on model calibration, they first complete a large number of paired model runs. In each case, one of these pairs employs a stochastic realization of the model parameter field generated by use of the assumed covariance matrix of innate parameter variability. The other parameter field is the lumped or averaged counterpart to the complex field. By undertaking many such runs on the basis of many such pairs of parameter fields, they empirically determine the covariance matrix, that forms the “structural noise” induced by parameter simplification under both calibration and predictive conditions. The former is added to measurement noise and used in overdetermined uncertainty analyses formulation thereafter;

hence the constraining objective function of figure 6 includes the contribution of structural noise to calibration model-to-measurement misfit. The structural noise affecting the prediction is also taken into account when its potential for error is computed through the constrained maximization/minimization process.

Highly Parameterized Nonlinear Analysis

The PEST suite provides two methodologies through which nonlinear uncertainty analysis can be done in the highly parameterized context; as Doherty and Hunt (2010) point out, this is the most appropriate context for environmental modeling. Strictly speaking, these methods constitute error analysis rather than uncertainty analysis, because the methodologies discussed above (and elaborated upon in appendix 4) for computation of linear parameter and predictive error provide a basis for their formulation. However, the distinction between error and uncertainty analysis becomes somewhat blurred when applying these methodologies in the nonlinear context. Implementation of both of these methodologies commences with a calibrated model. In both cases, the uncertainty/error analysis process is therefore best served if care is taken in implementing the calibration process to ensure that the calibrated parameter field approaches that of minimum error variance.

Constrained Maximization/Minimization

Tonkin and others (2007) present a methodology for model predictive error analysis through constrained maximization/minimization in the highly parameterized context. Once a model has been calibrated, predictive maximization/minimization takes place in a similar manner to that described previously for the overdetermined context. However, it is different from the overdetermined context because:

1. the maximization/minimization problem is formulated in terms of parameter *differences* from their calibrated values rather than in terms of actual parameter values, and
2. the effect that these parameter differences have on model outputs is also expressed in terms of *differences* between those outputs and those that correspond to the calibrated parameter set.

As predictive maximization/minimization takes place, differences between current parameter values and their calibrated counterparts are projected onto the calibration null space. These differences constitute a “null-space objective function” where projected parameter departures from calibrated values are informed by the null-space formulation of Moore and Doherty (2005). Projected differences are weighted by use of a null-space-projected parameter covariance matrix of innate variability $C(\mathbf{p})$ supplied by the modeler. At the same time, a “solution-space objective function” is calculated such that when the model employs the calibrated parameter field,

the total objective function is zero. As in the overdetermined case, a prediction of interest is then maximized or minimized subject to the constraint that the total objective function rises no higher than a predefined value. See Tonkin and others (2007) for additional discussion.

Null-Space Monte Carlo

Null-space Monte Carlo analysis (Tonkin and Doherty, 2009) is a more robust explanation of the constrained maximization/minimization process described above. Prior to implementation of null-space Monte Carlo analysis, it is assumed that a model has been calibrated, this resulting in computation of an optimal parameter field (that is, a parameter field that approaches that of minimum error variance). Similar to standard Monte Carlo approaches, stochastic parameter fields are then generated by using an appropriate $C(\mathbf{p})$ matrix of innate parameter variability supplied by the modeler. In each case, the calibrated parameter field is then subtracted from the stochastically generated parameter field. The difference between the two is then projected onto the calibration null space. Next, the solution-space component of the stochastically generated parameter field is replaced by the parameter field arising from the previous regularized-inversion calibration. Ideally, because (by definition) null-space parameter components do not appreciably affect model outputs that correspond to elements of the calibration dataset, the null-space processing of the optimal parameter set in this manner should result in a similar model calibration. However, in practice, the null-space-processed parameters commonly result in a slightly decalibrated model, this being an outcome of model nonlinearity and the fact that the cutoff between calibration solution and null spaces does not equate with the location of zero-valued singular values. Recalibration of the model is then effected by adjusting only solution-space parameter eigencomponents until the objective function falls below a user-specified level.

The choice of the target or threshold objective function level at which the model is deemed to be “calibrated” is often subjective. Generally, choice of a higher objective-function threshold can reduce the computational effort required to achieve it. Another factor that may influence the choice of a high objective-function threshold over a low one is that in most modeling contexts, “measurement noise” is dominated by structural noise of unknown stochastic character. Because of this, the “theoretically correct” objective function threshold is difficult, if not impossible, to determine. Moreover, in most modeling circumstances, parameter error is dominated by the null-space term; hence, errors incurred by selection of an inappropriately high objective function threshold in computation of the solution-space contribution to overall parameter and predictive error degrade the assessment of overall model predictive uncertainty to only a small degree.

Figure 7 illustrates the sequence of steps required to compute calibration-constrained stochastic fields by use of the null-space Monte Carlo methodology.

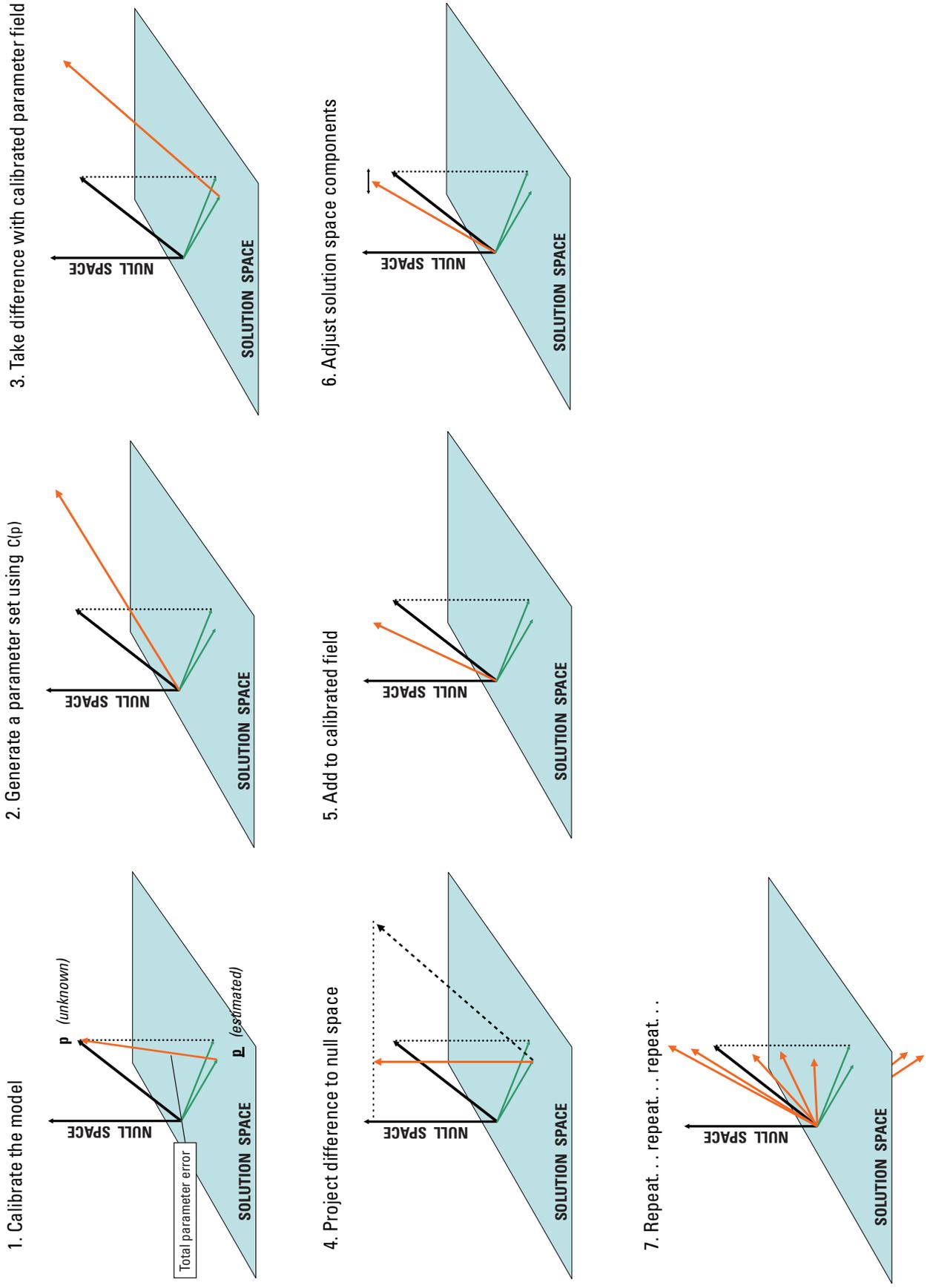


Figure 7. Processing steps required for generation of a sequence of calibration-constrained parameter fields by use of the null-space Monte Carlo methodology.

Recalibration of null-space-projected stochastic fields by solution-space adjustment ensures that variability is introduced to solution-space components. This adjustment is necessary because a potential for error in solution-space parameter components is inherited from noise associated with the calibration dataset. Although performed on highly parameterized models, the numerical burden of stochastic field recalibration is normally negligible for the following reasons:

1. Null-space projection, followed by replacement of solution-space components of stochastically generated parameter fields by that obtained through the antecedent regularized-inversion process, guarantees a near-calibrated status of all parameter fields produced in this manner.
2. As the dimensionality of the solution space is normally considerably smaller than that of parameter space, recalibration normally requires only a fraction of the number of model runs per iteration as there are adjustable parameters; recalibration can be efficiently attained by use of PEST's SVD-Assist functionality (Tonkin and Doherty, 2005).
3. The first iteration of each SVD-assisted recalibration process does not require that any model runs be done for computation of superparameter derivatives, because the derivatives calculated on the basis of the calibrated parameter field are reused for all stochastic fields.

Hypothesis Testing

One useful means through which a model can be used to test the likelihood or otherwise of a future (unwanted) environmental occurrence is to incorporate that occurrence as a member of an expanded calibration dataset. If a prediction-specific recalibration exercise can then find a parameter field that:

1. is defined by the modeler to be reasonable through terms of an explicit or implicit $C(\mathbf{p})$ covariance matrix of innate parameter variability and
2. allows the model to respect the tested predictive occurrence without incurring excessive misfit with the calibration dataset as assessed by using an explicit or implicit $C(\varepsilon)$ covariance matrix of measurement noise,

then the hypothesis that the undesirable occurrence will be realized cannot be rejected on the basis of currently available knowledge and data pertaining to the study site.

In practice, many hypotheses cannot be unequivocally rejected; however, they can be rejected at a certain level of confidence. The confidence level may be judged on the basis of a composite objective function that includes both the $C(\mathbf{p})$ covariance matrix of innate parameter variability and the $C(\varepsilon)$ covariance matrix of measurement noise. Furthermore, in many instances the concept of what constitutes an unwanted future occurrence is not a binary condition but is rather a matter of degree. Hence, a modeler will be interested in relating, for example, the increasing magnitude of the value of a prediction to the decreasing likelihood of its occurrence.

Analyses of predictive value versus likelihood are most easily made where PEST's Pareto functionality is invoked (Figure 8). The Pareto concept is borrowed from optimization theory. The so-called "Pareto front" defines an optimal tradeoff between two competing objectives whereby neither objective can be better met without simultaneously degrading the success of meeting the other. Moore and others (2010) show that it also represents a locus of solutions to a succession of constrained optimization problems whereby a prediction is maximized/minimized subject to the constraints of respecting a specified objective function. As such, the Pareto concept can be used to explore predictive uncertainty. However, its use in this context can be more rewarding than methodologies presented above that are based on similar principles for the following reasons:

1. Because Pareto analysis solves a succession of constrained optimization problems, a modeler is able to relate a continuous series of predictive outcomes to a continuous series of confidence limits.
2. Where the covariance matrix of innate parameter variability is unknown, and where model-to-measurement misfit is dominated by structural noise of unknown covariance matrix, the continuous nature of PEST's outputs when run in Pareto mode provide a sound basis for qualitative assessment of predictive confidence intervals.

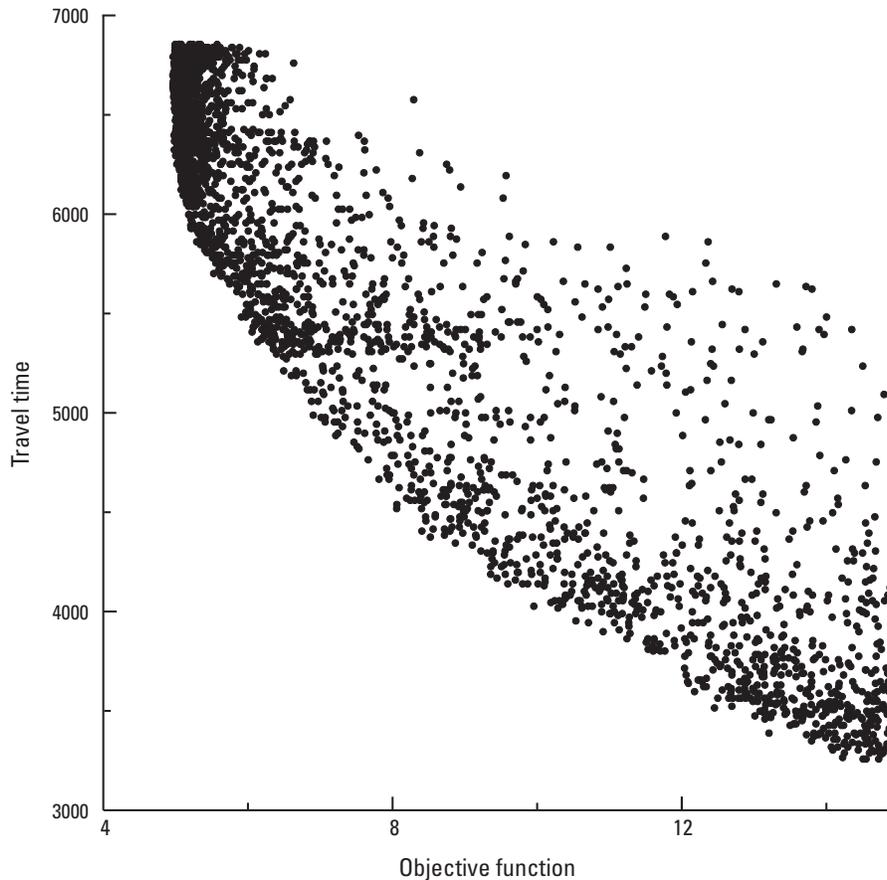


Figure 8. A Pareto-front plot of the tradeoff between best fit between simulated and observed targets (objective function, x-axis) and the prediction of a particle travel time. The correct prediction is 3,256 days, which requires an appreciable degradation of the calibrated model in order to be simulated (modified from Moore and others, 2010).

Uncertainty Analysis: Overdetermined Systems

Given the background presented in previous sections of the report, specific PEST software tools can now be explained. The following presentation is intended to be useful on two levels: advanced PEST users can obtain guidelines for specific uncertainty-analysis operations, whereas intermediate users can become acquainted with programs in the PEST suite that are intended for use in calibration-constrained uncertainty analysis. The appendices relate PEST variables and concepts mentioned below to the broader PEST framework, terminology, and definitions of Doherty (2010a,b).

Linear Analysis Reported by PEST

Upon completion of a parameter-estimation process PEST automatically records the following information to its run record (*.rec) file:

1. the parameter covariance matrix,
2. the parameter correlation coefficient matrix,
3. eigenvectors and eigenvalues of the parameter covariance matrix,
4. the standard deviations of estimated parameters (the square roots of respective diagonal elements of the parameter covariance matrix), and
5. the 95-percent individual linear confidence interval for each estimated parameter, calculated under the assumption that measurement noise (and hence estimated parameters for a linear model) is normally distributed.

If requested, this information is also recorded in a matrix (*.mtt) file at the end of each iteration of the parameter-estimation process. When viewing this information, it is important that the following points be kept in mind.

First, the validity of these results is predicated on the assumption that the user has supplied observation weights that represent the covariance matrix of measurement noise. However, as Cooley (2004) points out, the process of precalibration parameter lumping necessary to achieve well-posedness of the

inverse problem of model calibration induces structural noise with a complex spatial (and temporal) covariance structure. Rarely will this structure be known; thus, rarely can a weight matrix be selected that fulfills these requirements.

Secondly, even if an appropriate weight matrix is determined by using, for example, the precalibration Monte Carlo procedure described by Cooley (2004), parameter uncertainties calculated as an adjunct to overdetermined parameter estimation pertain to properties that are necessarily averaged over large areas and/or large times. This coarseness incurs unquantified errors in quantities that are sensitive to smaller scale parameter detail (Doherty and Hunt, 2010; Fienen and others, 2010).

Thirdly, for the sake of attaining well-posedness of the inverse problem of model calibration, insensitive parameters, and/or those parameters that are highly correlated with other parameters, are generally fixed by the user at values that are deemed to be reasonable so that they do not take part in the calibration process. Because these particular parameters are fixed, the associated uncertainties cannot be recorded by PEST. Furthermore, the fact that values estimated for parameters that are retained in the calibration process may depend on the values at which some of these parameters are fixed is not accommodated in computation of parameter or predictive uncertainty.

Linear Analysis Using PEST Postprocessors

The PCOV2MAT utility reads a PEST run record (*.rec) file, or a matrix (*.mtt) file that contains a parameter covariance matrix. It rewrites this matrix in PEST matrix file format. Once in this format, the covariance matrix can be subjected to matrix manipulation of any type supported by PEST matrix manipulation utilities. (See appendix 2 for a list of processing options offered by these utilities.)

Linear computation of predictive uncertainty/error variance requires that the sensitivity of a prediction of interest to all parameters be available. This can be computed by including the prediction as a dummy observation (an arbitrary value specified) in a PEST input dataset and by then obtaining the Jacobian matrix by running PEST with NOPTMAX set to -1 or -2. Sensitivities for any number of predictions can be computed in this manner. The command cited in the “model command line” section of the PEST control file must be such that

the model reads input files that contain stresses on which basis the prediction of interest is calculated. If these are the same as calibration stresses, then one or a number of predictions can be “carried” throughout a calibration exercise by including them in the calibration dataset with weights of zero. PEST records sensitivities of model outputs to adjustable parameters in its Jacobian matrix (*.jco) file. An individual row of this matrix can be extracted and written in PEST matrix file format using the JROW2VEC or JROW2MAT utilities supplied with PEST. Predictive error variance can then be computed by using the MATQUAD matrix manipulation utility. (See appendix 4 for computational details.)

Other Useful Linear Postprocessors

The INFSTAT (INFLuence STATistic) utility computes a number of observation influence statistics including observation leverage, Cook’s D (see Cook and Weisberg; 1982), an influence statistic proposed by Hadi (1992), and “difference in Betas” or DFBETAS. Belsley and others (1980) and Hunt and others (2006) demonstrate the use of these statistics in the groundwater-modeling context; Yager (1998) and Hadi (1992) provide calculation details for these statistics.

Statistics computed by INFSTAT are designed for use in the overdetermined parameter-estimation context where well-posedness is achieved through precalibration parameter lumping. Thus, in practice insights given by these statistics should often be considered more qualitative than quantitative. Moreover, the integrity of information given by them depends on use of a $C(\epsilon)$ matrix that, in practice, includes the substantial contribution made to measurement noise by precalibration simplification.

Like INFSTAT, the EIGPROC (EIGen PROCessor) utility is designed for use following an overdetermined PEST-based parameter-estimation exercise. This utility reads files written by PEST during the estimation process, accumulating information recorded in these files that is pertinent to one or more parameters whose estimation is most prone to uncertainty. This information is then recorded in a single output file. Thus, all information calculated during the parameter-estimation process that pertains to all parameters whose estimation has been identified as problematic is immediately available for inspection by the modeler.

Nonlinear Analysis—Constrained Optimization

When PEST is run in “predictive analysis” mode (by setting the PESTMODE variable in the “control data” section of the PEST control file to “prediction”; see appendix 1), it implements constrained predictive maximization/minimization as described above and in appendix 4. The following features of PEST’s operation in this mode are notable:

1. If PEST is run in “predictive analysis” mode, the PEST control file must possess a “predictive analysis” section (see appendix 1).
2. The model output that PEST must maximize or minimize must be the sole member of an observation group whose name is “predict.”
3. Maximization or minimization as an optimization alternative is selected by using the NPREDMAXMIN variable in the “predictive analysis” section of the PEST control file.
4. The objective function constraint is supplied as the PDO variable in the “predictive analysis” section of the PEST control file.
5. Predictive noise is optionally taken into account in the constrained predictive maximization/minimization process through use of the PREDNOISE variable.
6. Termination criteria, specific to this mode of PEST’s operation, must also be supplied in the “predictive analysis” section of the PEST control file.

Experience with the Gauss-Marquardt-Levenberg (GML) method of objective function minimization during calibration has shown that it can work reasonably well in contexts where model outputs are contaminated by numerical noise (provided that the level of this contamination is not unduly high). However, the same cannot be said for the constrained predictive maximization/minimization process implemented by PEST when run in “predictive analysis” mode because this is a much more numerically demanding procedure. Problematic model derivatives lead to “serrated” objective-function contours. The geometry of these contours can become even more irregular where they are long and narrow as a result of a high degree of parameter correlation or insensitivity. Contours of model prediction values can also become serrated. The locations where these sets of contours intersect (especially where intersection takes place at a shallow angle) may thus be difficult to compute without error.

The following advice may prove useful when working in numerically troublesome environments:

1. Like parameter estimation, the constrained maximization/minimization process implemented by PEST when it is run in “predictive analysis” mode is an iterative process that requires successive linearizations of the constrained optimization problem. Each iteration requires that at least as many model runs be done as there are adjustable parameters for the purpose of finite-difference derivatives calculation; hence, the process can be numerically expensive. In general, it is best to commence the process by using parameter values that coincide with the objective-function minimum. Following a parameter estimation run in which this minimum is determined, the PARREP (PARAmeter REPlacement) utility can be used to build a PEST control file in which optimized parameter values become initial parameter values for an ensuing predictive maximization/minimization run.
2. Where it is suspected that the constrained maximization/minimization process may not have raised/lowered the prediction as much as it is possible to do, alternative starting points (some of which may lead to objective functions that are considerably higher than the constraining objective function) should be tested.
3. Implementation of the optional predictive maximization/minimization line-search option may allow progress to be made in difficult numerical circumstances where progress would not otherwise be possible. However the line search is numerically expensive and is not parallelizable. If implemented, useful settings for the INITSCHFAC, MULSCHFAC and NSEARCH variables that control the line-search process have been found to be 0.3, 1.3 and 15 respectively.

Nonlinear Analysis—Random Parameter Generation for Monte Carlo Analyses

A fast but approximate method of generating a suite of null-space-processed parameter sets that approximately calibrate a model is to generate these fields by using the postcalibration parameter error covariance matrix. This matrix can be extracted from a PEST run record (*.rec) file by means of the PCOV2MAT utility; it can then be cited in a parameter uncertainty file (see Doherty 2010a) for access by the RANDPAR (RANDOM PARAmeter) utility supplied with PEST. When deployed in this way, RANDPAR generates a sequence of parameter-value files, each containing a parameter set that approximately calibrates the model; the approximation deteriorates with increasing model nonlinearity and with the degree of ill-posedness of the inverse problem.

Linear Analysis for Underdetermined Systems

This section provides a brief description of tasks performed by PEST utilities that implement linear uncertainty/error analysis, and processing related thereto, in the highly parameterized context. Some of these utilities are designed for use in a calibration postprocessing capacity. Others can be used independently of the calibration process.

Postcalibration Analysis

Figure 9 is a schematic description of utilities available in the PEST suite for postcalibration analysis of parameter and predictive error variance. A summary of their use is given in the text that follows.

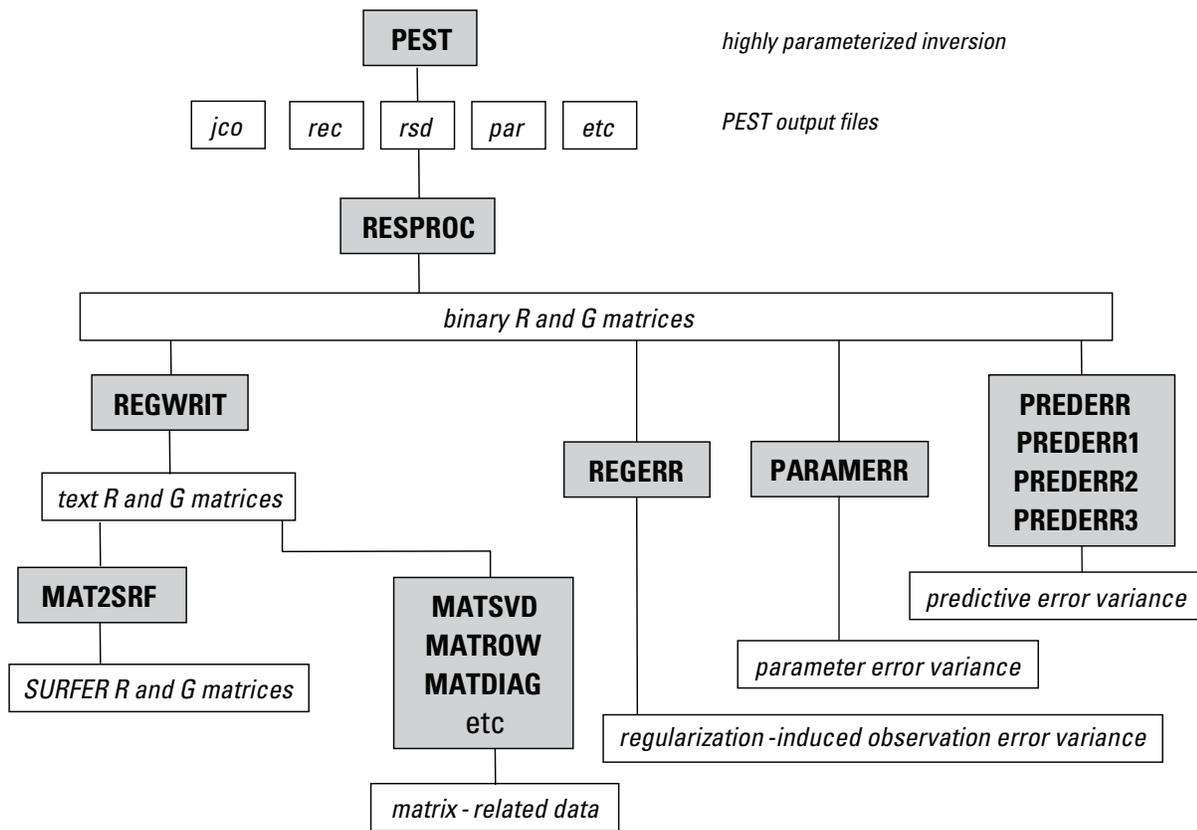


Figure 9. Utilities available for postcalibration parameter and predictive error variance analysis. (See appendix 2 for additional description of these utilities.)

Regularized Inversion and Uncertainty

As is discussed in Doherty and Hunt (2010), PEST offers various means through which mathematical regularization can be implemented in highly parameterized model calibration. These include the following:

- Tikhonov regularization,
- truncated singular value decomposition (SVD),
- PEST’s “SVD-Assist” methodology, and
- the LSQR methodology of Paige and Saunders (1982a, 1982b).

These can often be used in combination. For example, Tikhonov regularization can be used in an inverse problem in which parameters are estimated through truncated SVD. Superparameters used in SVD-assisted inversion can be estimated by using the Gauss Marquardt Levenberg solution method, truncated SVD, or LSQR.

Whenever PEST runs any kind of regularized inversion, it records a resolution data (*.rsd) file. This is a binary file that contains information required to build the resolution matrix; the construction details of this matrix are a function of type of regularization implemented in solving the inverse problem of model calibration.

Regularized Inversion Postprocessing

Regularized inversion postprocessing is a multistep procedure. Where many parameters are featured in the inversion process, calibration postprocessing can be numerically burdensome, and memory requirements can be high. By subdividing postprocessing activities into discrete tasks, and by storing the outcomes of these discrete tasks as individual files, these outcomes can be used in different ways by different application-specific postprocessors without the need to repeat common steps.

The first step in regularized-inversion postprocessing is the running of the RESPROC (RESolution matrix PROCessor) utility. This utility reads a PEST control file, as well as the resolution data (*.rsd) file and the corresponding Jacobian matrix (*.jco) file produced during a previous inversion exercise. RESPROC writes its own binary output file, which contains the resolution matrix that results from solution of the highly parameterized problem. This file contains intermediate data that are then read by other utility programs that compute quantities of direct interest to the modeler.

The RESWRIT (RESolution matrix WRITing) utility reads the resolution-data matrices stored in a RESPROC binary output file, rewriting these matrices in PEST’s ASCII matrix file format. These matrices (and the files that hold them) can be very large where a parameter-estimation process employs many parameters and/or observations. The MAT2SRF matrix manipulation utility can be used to rewrite these matrices in SURFER (<http://www.goldensoftware.com/products/surfer/surfer.shtml>) grid format; SURFER can then be used to inspect the resolution matrix for presence or absence of diagonal dominance.

Once stored in matrix file format, other options for processing the resolution matrix become available. These include the following:

1. The MATROW (MATrix ROW) utility can be used to extract a row of the resolution matrix. As stated in section 2 of this document, each row of the resolution matrix depicts the averaging weights through which each estimated parameter is related to real-world hydraulic properties.
2. Diagonal elements of the resolution matrix can be extracted by using the MATDIAG (MATrix DIAGonal) utility. This allows an inspection of the degree to which these do, or do not, approach unity for different parameters and/or for different parameter types.

Like RESWRIT, the PARAMERR (PARAMeter ERROR) utility reads a RESPROC output file. It also requires a “parameter uncertainty file” that reflects the modeler’s estimate of the innate parameter variability. This estimate can consist of user-supplied standard deviations for the parameterization scheme used (that is, the more lumped the parameters, the larger the likely standard deviation of the properties within that entity) or a formally calculated covariance matrix constructed by means of a variogram or other geostatistical information. This estimate of innate parameter uncertainty is then handled with mathematical rigor via the $C(\mathbf{p})$ matrix (see appendix 4); this matrix, along with an observation uncertainty file required to obtain the $C(\epsilon)$ matrix describing measurement noise, provide the fundamental information for uncertainty analysis. Doherty (2010a) describes the format of these two files. PARAMERR then computes the null-space and solution-space components of the parameter error covariance matrix $C(\mathbf{p}-\mathbf{p})$. The modeler can add these matrices together to obtain the overall parameter error covariance matrix using the MATADD (MATrix ADD) utility. Further manipulation of any of these matrices can be run by use of matrix utilities supplied with the PEST suite (appendix 2).

The covariance matrix of regularization-induced structural noise (see equation A4.33, appendix 4) can be computed by using the REGERR (REGularization ERRor) utility. Like other utilities discussed above, REGERR reads a RESPROC output file. It obtains a $C(\mathbf{p})$ matrix of innate parameter variability from a user-prepared parameter uncertainty file; it also reads the Jacobian matrix (*.jco) file recorded by PEST during the previous parameter-estimation process.

Predictive Error Analysis

PREDiction ERRor Utilities (PREDERR, PREDERR1, PREDERR2, and PREDERR3) are also available to compute the error variance of a prediction (by means of equation A4.27 in appendix 4). In order to accomplish this, they must employ

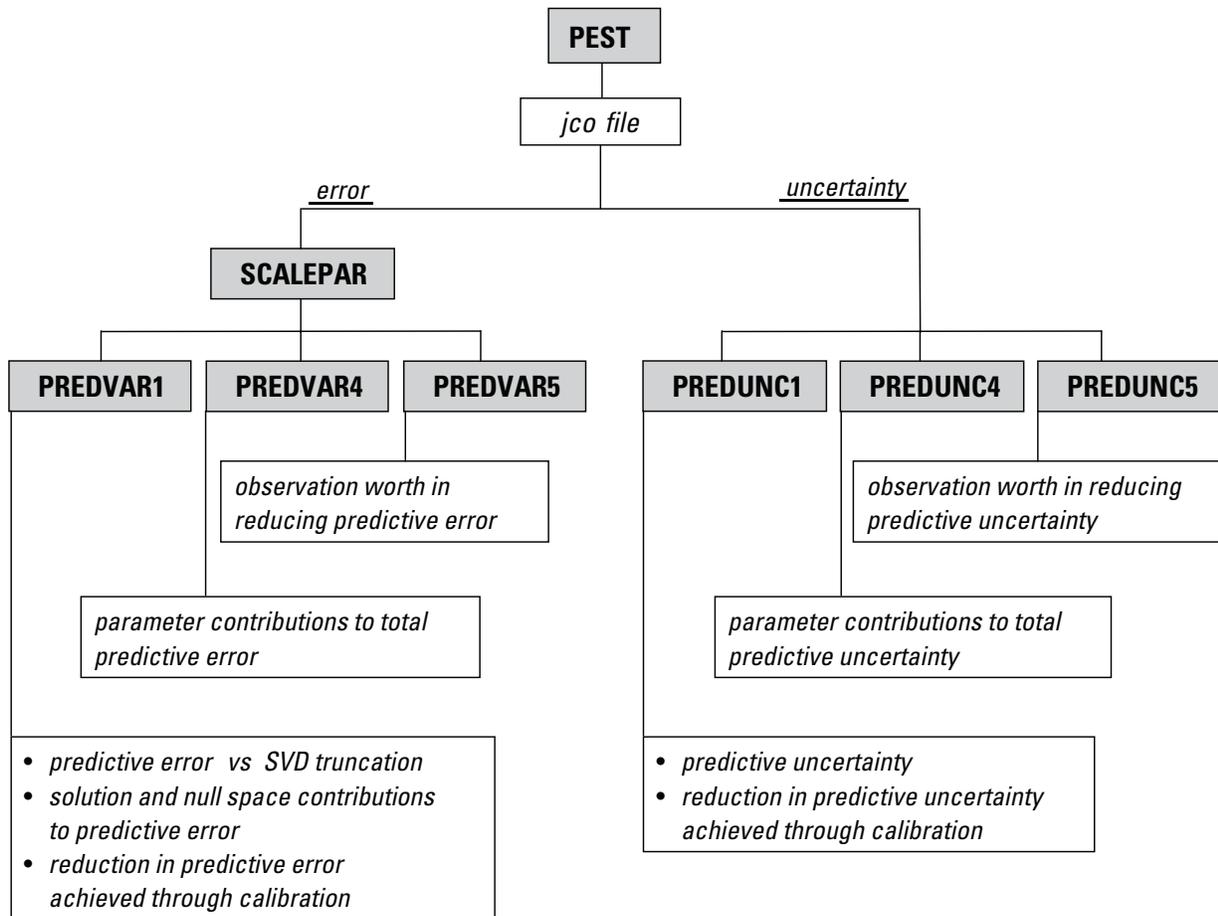


Figure 10. PEST utilities available for calibration-independent error and uncertainty analysis.

the predictive-sensitivity matrix read from a Jacobian matrix (*.jco) file. This file may have been written during a previous calibration exercise, in which case the prediction will probably have appeared as an observation of zero weight in the corresponding PEST control file. Alternatively, the Jacobian matrix (*.jco) file may have been written by PEST on a special run in which no calibration was actually done but in which the model was run under predictive conditions m times, where m is the number of adjustable parameters featured in the previous parameter-estimation process; optimized parameter values should have been used in this PEST run.

Each member of the PREDERR suite of programs differs slightly from other members of this suite in how it obtains input and in the nature of the calculations that it performs. PREDERR and PREDERR3 read the resolution and \mathbf{G} matrices of appendix 4 equation A4.27 from a binary RESPROC output file, whereas PREDERR1 and PREDERR2 read these matrices from ASCII matrix files (that were presumably written by RESWRIT). PREDERR2 and PREDERR3 can compute error variances for multiple predictions, whereas PREDERR and PREDERR1 compute the error variance associated with just a single model prediction.

Calibration-Independent Analysis

The utility programs discussed in the preceding section are designed for use as an adjunct to model calibration. In contrast, utilities discussed in the present section are designed for use independently of the calibration process. Their use depends only on the existence of a PEST control file, an associated Jacobian matrix (*.jco) file, and a parameter uncertainty file that contains the user-specified matrix of innate parameter variability $\mathbf{C}(\mathbf{p})$. There is no requirement for a PEST calibration operation, because these programs do not need the resolution data (*.rsd) file that results from calibration. If desired, parameter values cited in the PEST control file can be those estimated through a previous calibration exercise; alternatively, the PEST control file can list parameter values that are considered by the modeler to be reasonable.

A major distinction between approaches is shown in figure 10. In practice, the uncertainty utilities (PREUNC) require less user input and are easier to apply. The error utilities (PREVAR) are able to handle larger numbers of parameters using singular value decomposition but require the user to provide additional information. Figure 10 shows the relations between and within these utilities schematically. A summary of their capabilities is given in the subsequent text.

The PREDVAR Suite

Each member of the PREDVAR (PREdictive VARiance) suite of programs differs slightly from other members of this suite in the nature of the calculations that it performs.

PREDVAR1

The PREDVAR1 utility reads a PEST control file, its corresponding Jacobian matrix (**jco*) file, a user-supplied parameter uncertainty file, and a PEST matrix file containing sensitivities of a *single* prediction to all model parameters. PREDVAR1 computes the error variance of that prediction under the assumption that calibration is effected by use of truncated singular value decomposition. The user provides a suite of integers denoting singular value numbers at which truncation may occur; the lowest of these numbers should be no lower than zero, whereas the highest should be no higher than the maximum number of adjustable parameters featured in the PEST control file. PREDVAR1 computes the error variance of a user-specified prediction at all such truncation levels. On the basis of these calculations, graphs such as that depicted in figure 4 can be produced. As was discussed previously, from graphs such as this the following information can be gleaned:

1. the minimum attainable error variance expected for the prediction,
2. the singular value number at which predictive error variance minimization occurs,
3. the contributions made by null- and solution-space terms to predictive error variance,
4. precalibration predictive error variance (equivalent to precalibration predictive uncertainty), and
5. the efficacy of the calibration process in reducing predictive error variance below its precalibration level.

As discussed previously, where the $C(\mathbf{p})$ covariance matrix of innate parameter variability contains very different precalibration uncertainties, graphs such as that shown in figure 4 can sometimes rise before they fall as the singular-value-truncation number is increased. Though not common, this can be avoided by reformulating the inverse problem in terms of normalized parameters, rather than native parameters, with normalization taking place with respect to the innate variability of each parameter. Reformulation of the inverse problem in terms of normalized parameters can be accomplished by using the SCALEPAR (SCALE PARAMeter) utility. SCALEPAR reads an existing PEST control file and corresponding Jacobian matrix (**jco*) file and writes a new version of each of these based on an inverse problem in which parameters are normalized by their innate variability. Utility programs of the PREDVAR suite should then be directed to read these files instead of those pertaining to native parameters. If the inverse problem is posed in terms of normalized parameters, predictive sensitivities should also pertain to normalized parameters.

Hence, SCALEPAR should be used to do a transformation of predictive sensitivities, as well as calibration sensitivities. Moreover, once parameters have been normalized, the $C(\mathbf{p})$ matrix supplied to PREDVAR1 must be revised so that it describes the innate variability of normalized parameters rather than of native parameters.

PREDVAR4

PREDVAR4 computes the contribution made to the error variance of a particular prediction by individual parameters or by groups of parameters. For this purpose, “contribution” is defined as the fall in predictive error variance accrued through notionally acquiring perfect knowledge of the value(s) of the parameter(s) in question. Using a “parameter list file” (Doherty, 2010b), the user defines one or multiple sets of parameters for which such contributions to predictive error variance are sought. In each case, PREDVAR4 minimizes the error variance of the nominated prediction with respect to singular value number when calculating predictive error variance with and without inclusion of the nominated parameters in the predictive error variance calculation process.

Graphs such as that shown in figure 5 are readily created on the basis of PREDVAR4 analysis. The back row of figure 5 depicts precalibration contributions by different parameter types to the error variance of a particular prediction; these are computed for truncation at zero singular values. The front row depicts postcalibration contributions to the error variance of the same prediction made by the same parameters; these are computed for minimized (with respect to singular value) error variance. Graphs such as this can provide a modeler with valuable insights into the role of the model-calibration process in enhancing (or not) the reliability of predictions of interest. In addition, they may suggest strategies for future data acquisition aimed at reducing the error with which predictions are made. Alternatively, if postcalibration contributions to the error variance of predictions of a certain type are dominated by parameters whose contributions are unlikely to be reduced either by direct measurement of system properties or by measurements of system state, the recalcitrant nature of the error variance associated with predictions of that type will have been identified.

As discussed previously, parameters included in an analysis of this type do not need to be restricted to those that are adjusted through the calibration process. Any facet of the modeled system that is imperfectly known can be included in the analysis, including boundary conditions for which pertinent values would otherwise be assumed (for example, *rstage* in fig. 5). If an analysis shows that contributions to the error variance of certain predictions accruing from uncertain knowledge of such boundary conditions is minimal or low compared to contributions made by other facets of the system under study, this indicates that reasonable values can be assigned to such boundary conditions without adversely affecting the use of the model in making those predictions.

In rare circumstances, use of PREDVAR4 can yield results that suggest that the postcalibration contribution to predictive error variance made by a particular parameter is greater than its precalibration contribution. This seemingly illogical result arises from the definition of “contribution to predictive error variance” as the loss in predictive error variance accrued through acquiring perfect knowledge of a parameter’s value through means outside of the calibration process. If the value of a certain parameter can only be estimated through the calibration process in conjunction with that of another parameter (that is, if the parameters have a high degree of correlation based on currently available data), then acquisition of perfect knowledge of the first parameter can yield better estimates for the second parameter. Hence, acquisition of perfect knowledge of the first parameter will yield better estimates of any prediction that is sensitive to that second parameter, even if the prediction is not sensitive to the first parameter.

As was described above for PREDVAR1, parameters should be scaled with respect to their innate variabilities before using PREDVAR4. This is most easily achieved by using the SCALEPAR utility.

PREDVAR5

PREDVAR5 is used to assess the worth of individual or grouped observations. PREDVAR5 assesses “worth” in two ways. First, the worth of an observation, or group of observations, can be assessed as the increase in predictive error variance incurred through omitting the observation(s) from the current calibration dataset. Second, the worth of an observation, or group of observations, can be assessed as the decrease in predictive error variance from its precalibration level accrued through including the observation(s) as the sole member(s) of a calibration dataset. PREDVAR5 can also compute the reduction in predictive error variance accrued through adding to the current observation dataset one or more new observations. Because linearity is assumed, it is not necessary for these observations to have actually been made, for only the sensitivities of model outputs corresponding to these observations, together with an assessment of the noise with which they are likely to be associated, are required. PREDVAR5 can thus form a basis for optimizing the acquisition of future data. As for PREDVAR1 and PREDVAR4, parameters should be scaled with respect to their innate variabilities before use of PREDVAR5. This is most easily achieved by using the SCALEPAR utility.

The PREDUNC Suite

The PREDiction UNCertainty utilities PREDUNC1, PREDUNC4, and PREDUNC5 perform similar roles as the PREDVAR1, PREDVAR4, and PREDVAR5 utilities, respectively. However, their calculations are based on Bayes’ equation (appendix 4, equation A4.10) for predictive uncertainty rather than for predictive error variance (appendix 4, equation A4.27). Their use requires identical input files to those of their

respective PREDVAR counterparts. However, there are some important differences between these two suites of utilities:

1. Uncertainty is an intrinsic quality of a prediction and the data used to support that prediction, whereas error variance is not (and is always higher than uncertainty variance). PREDUNC4 should be used in preference to PREDVAR4 for computation of parameter contributions to predictive variability. Similarly, PREDUNC5 should be used in preference to PREDVAR5 for assessment of observation worth.
2. There is no need to scale parameters with respect to their innate variability before using members of the PREDUNC suite; hence, use of SCALEPAR is unnecessary.
3. Unlike members of the PREDVAR suite, PREDUNC utilities do not need to minimize predictive uncertainty with respect to number of singular values, because singular values do not feature in calculation of predictive uncertainty.

Identifiability and Relative Uncertainty/Error Variance Reduction

Through use of the IDENTPAR (IDENTifiability PARAmeter) and GENLINPRED (GENeral LINear PREDiction) utilities supplied with PEST, parameter identifiability and relative parameter error reduction are readily computed. Doherty and Hunt (2010) provide the derivation of these statistics. Plots based on outputs from these utilities are given in figures 11A and 11B. A map-view of identifiability is given in James and others (2009).

In figure 11A, identifiability bars are color coded according to the magnitude of their projection onto different solution-space eigenvectors (columns of the V1 matrix of appendix 4 equation A4.14). Warmer colors indicate projection onto eigenvectors associated with higher singular values. Cooler colors indicate projections onto eigenvectors associated with lower singular values. (Values for the former are estimated with a smaller potential error than are those for the latter.)

Relative parameter error variance reduction is computed with the prediction being the value of a parameter normalized by its standard deviation. Relative parameter uncertainty variance reduction can be computed in a similar fashion (using appendix 4, equation A4.10). Graphs of relative parameter-uncertainty-variance reduction are similar to those of relative parameter-error-variance reduction. In general, use of relative parameter-uncertainty-variance reduction is preferable to that of relative parameter-error-variance reduction because

1. parameter uncertainty does not depend on the number of pre-truncation singular values,
2. minimization of parameter error with respect to number of pre-truncation singular values is not required, and
3. unlike relative parameter-error-variance reduction, the relative uncertainty-variance reduction of a parameter can never become negative; the latter can occur if the number of pre-truncation singular values is set too high.

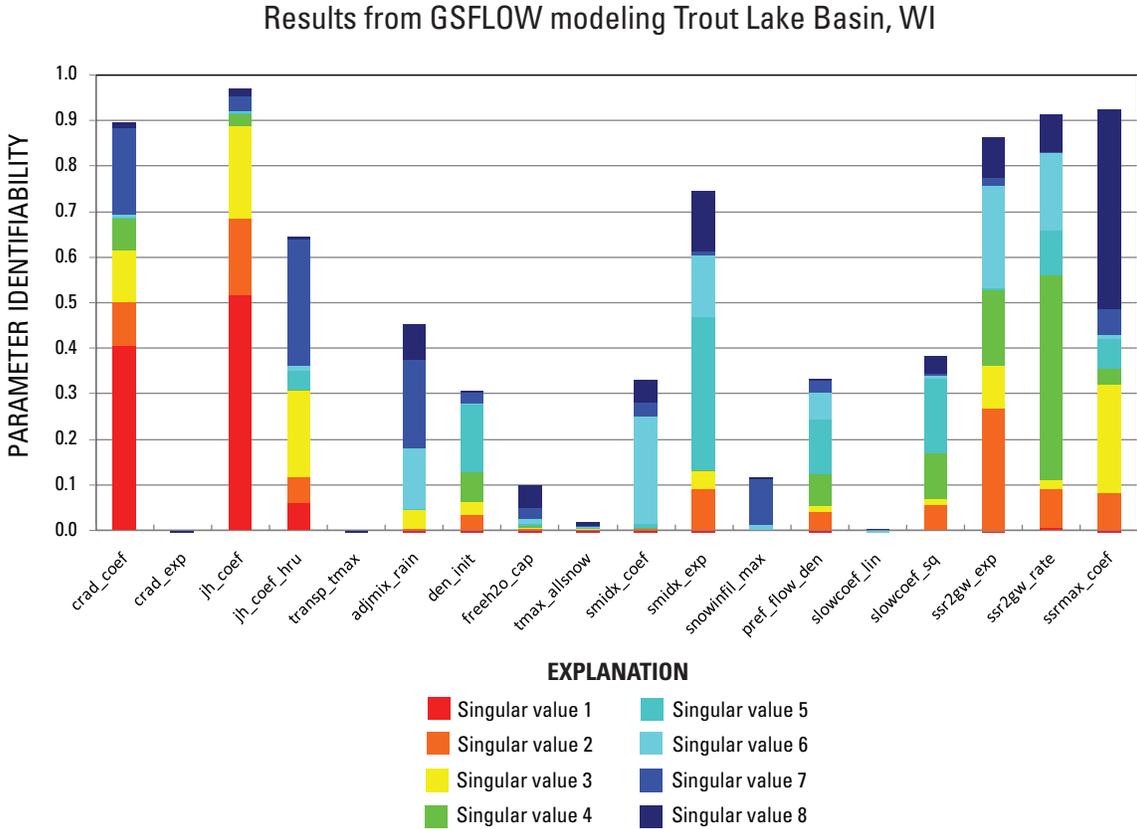


Figure 11A. Identifiability of parameters used by a composite groundwater/surface-water model (modified from Doherty and Hunt, 2009a).

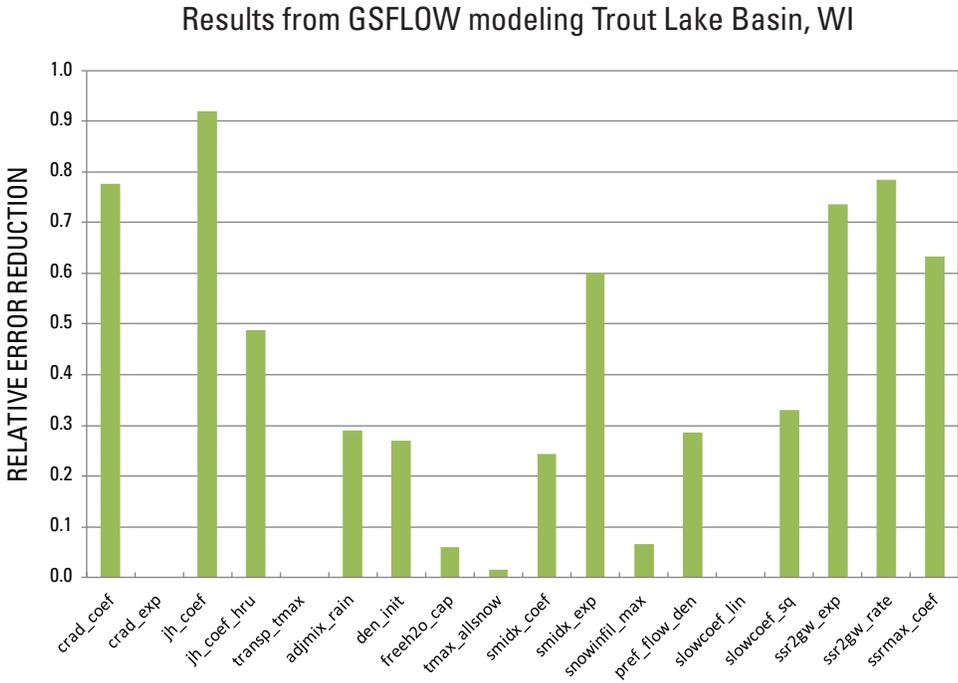


Figure 11B. Relative error reduction of parameters used by a composite groundwater/surface-water model (modified from Doherty and Hunt, 2009a).

The GENLINPRED Utility

The GENLINPRED utility can be used for computation of parameter identifiability, relative parameter- and prediction-error variance reduction, and relative parameter- and prediction-uncertainty-variance reduction. It calls the IDENTPAR, SCALEPAR, PREDVAR1A, and PREDUNC1 utilities.

If specified by the user, GENLINPRED can also run the PREDUNC4 and PREDVAR4 utilities to compute parameter contributions to predictive uncertainty/error variance, as well as the PREDUNC5 and PREDVAR5 utilities to compute observation worth. GENLINPRED outputs are recorded in a series of tables written to its single text output file. These tables can be pasted into a spreadsheet for immediate plotting.

Box 2: GENLINPRED—Access to the Most Common Methods for Linear Predictive Analysis within PEST

GENLINPRED—the *GENeral LINear PREDictive uncertainty/error analyzer*—is a master utility program that enables a modeler to undertake many of the linear parameter and predictive error (uncertainty) analyses described in this chapter. GENLINPRED accomplishes this by automatically executing other PEST utilities—including SCALEPAR, JROW2VEC, SUPCALC, IDENTPAR, PREDUNC1/4/5 and/or PREDVAR1/4/5. GENLINPRED saves the modeler the task of running each of these utilities individually by automatically piping intermediate results from one utility to the next utility in sequence and, at the end, gathering information produced by these utilities into a single output file. The contents of this output file are readily pasted into Microsoft Excel and other programs for numerical and graphical postprocessing.

From a computational standpoint, use of GENLINPRED can sometimes be inefficient because of some redundancy in the actions performed by the various utilities that GENLINPRED executes. From the modeler's perspective, however, GENLINPRED performs a multitude of tasks necessary to error/uncertainty analysis with minimal user input and/or input file preparation. GENLINPRED accomplishes all of these operations on the basis of a conventional PEST input dataset supplied by the user.

Default operations carried out by GENLINPRED reflect those that are most likely to be requested (IDENTPAR to calculate parameter identifiability and PREDUNC to calculate relative parameter uncertainty reduction), but these defaults can be overridden by the user to select more comprehensive processing options. Depending on the user's requests, GENLINPRED will calculate some or all of the following quantities:

- the optimum dimensionality of the calibration solution and null spaces (using SUPCALC);
- parameter identifiability (IDENTPAR), relative error variance reduction (PREDVAR1), and relative uncertainty variance reduction (PREDUNC1);
- the solution-space and null-space components of the total error variance of a prediction (or parameter) at different singular value truncation levels (PREDVAR1);
- precalibration and postcalibration uncertainties of a nominated prediction (PREDUNC1);
- contributions to precalibration and postcalibration error and/or uncertainty variance of a prediction (or parameter) made by different parameter groups, or by different individual parameters (using PREDVAR4 and/or PREDUNC4, respectively); and/or
- the worth of individual, or groups of, observations in lowering the postcalibration error or uncertainty variance of a prediction (or parameter) (using PREDVAR5 and/or PREDUNC5). ■

Nonlinear Analysis for Underdetermined Systems

Constrained Maximization/Minimization

As is discussed by Doherty and Hunt (2010), when calibrating a model by means of mathematical regularization, an “appropriate” fit, rather than the best possible fit, is sought between model outputs and field measurements. Where Tikhonov regularization is used, an appropriate target measurement objective function is set by the user before starting the regularized inversion process (using the PEST control file variable PHIMLIM; see appendix 1, Doherty, 2003, and Fienen and others, 2009, for more information). If regularization is implemented by use of truncated SVD, or SVD-Assist without Tikhonov regularization, the singular value truncation level and/or the number of estimated superparameters is set below that which would achieve too good a fit, thereby achieving a level of fit that is commensurate with measurement noise. On many occasions, assessment of the level of measurement noise is made through inspection of parameter fields that are realized through the calibration process; if too much heterogeneity is introduced to a calibrated parameter field, the level of measurement/structural noise is judged to be higher than originally thought, and a higher level of model-to-measurement misfit is sought in an ensuing calibration exercise.

In contrast to this, where parameter estimation is performed in the overdetermined context, the objective function is minimized. Where the level of postcalibration uncertainty associated with a particular model prediction is then explored, a threshold objective function (Φ_0 of equations A4.43 and A4.46) is selected according to a desired level of predictive confidence; the prediction of interest is then maximized or minimized subject to the constraint that the previously minimized objective function rises no higher than this threshold.

Where constrained predictive maximization/minimization follows highly parameterized inversion (where the objective function is not minimized but reduced to an appropriate level), the objective function constraint that is applied in the maximization/minimization process must be expressed not in terms of misfit between model outputs and corresponding field measurements but in terms of changes to model outputs, as well as changes to model parameters, that are incurred through this process. As when undertaking a traditional (overdetermined) minimization/maximization, the value of this objective function threshold will depend on the confidence limit that is to be associated with the maximized/minimized prediction.

The sequence of processing steps required for exploration of predictive error variance through constrained maximization/minimization is depicted schematically in figure 12 and described briefly in the following text.

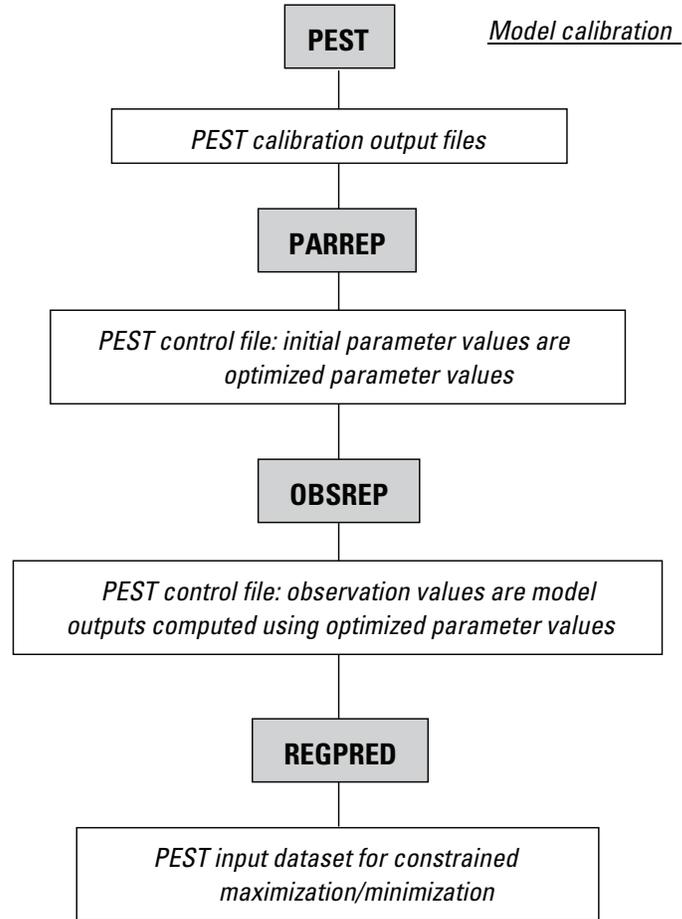


Figure 12. Processing steps required for exploration of predictive error variance through constrained maximization/minimization.

Problem Reformulation

As for overdetermined predictive maximization/minimization, the prediction to be maximized or minimized must be included in the PEST input dataset as the sole member of an observation group named “predict.” Furthermore, PEST must be run in “predictive analysis” mode in order to run constrained predictive maximization/minimization.

Upon completion of the preceding highly parameterized calibration process, the PARREP utility should be used to build a new PEST control file in which initial parameter values are actually optimized parameter values estimated through the previous calibration exercise. The OBSREP utility must then be used to build yet another PEST control file in which observation values are replaced by model outputs corresponding to these estimated parameter values. No Tikhonov regularization must appear in this file. Thus, if NOPTMAX is set to zero and PEST is invoked in order to run the model once and simply compute an objective function, that objective function will be zero (or very close to it).

The REGPRED utility must then be run. It performs the following tasks.

1. It writes a PEST control file in which PEST is instructed to run in “predictive analysis” mode. A target objective function (PD0) of the user’s choice is included in this file; a value of 9.0 (REGPRED’s default) is suggested.
2. It adds a new template file to those already listed in the “model input/output” section of the existing PEST control file. This new file contains a list of parameter names; the model input file that is written on the basis of this template file thus contains a list of parameter values.
3. It modifies the model batch or script file by adding commands to compute the projection of parameter-value differences (with respect to calibrated parameter values) onto the calibration null space.
4. It adds a new instruction file to the modified PEST control file through which these null-space-projected parameter differences are read as “observations.” These are assigned to their own observation group to which a covariance matrix $C(\mathbf{p}-\mathbf{p})$ computed through equation A4.48 is assigned (in the “observation groups” section of the modified PEST control file).
5. Measurement weighting is assigned as $C^{-1}(\epsilon)$, where $C(\epsilon)$ is the covariance matrix of measurement noise, as supplied by the user.
6. If desired, the PEST control file is configured to allow noise associated with the prediction that is to be maximized/minimized to feature as an adjustable element within this process.

The REGPRED-generated PEST input dataset is ready for use. With PD0 set to 9.0, the two-sided confidence interval associated with a maximized and then minimized predic-

tion is 99.7 percent. Null-space parameter components are constrained by the $C(\mathbf{p}-\mathbf{p})$ covariance matrix of projected parameter difference variability (calculated from $C(\mathbf{p})$ by GENPRED), while solution-space components are constrained by the necessity for model-to-measurement misfit to remain “statistically similar” (as assessed using $C(\epsilon)$) to that attained through the previous calibration. See Tonkin and others (2007) and Doherty (2010b) for a further discussion of this method.

Practical Considerations

Drawbacks of this approach to constrained prediction maximization/minimization in the highly parameterized context include the following;

1. The constrained maximization/minimization technique enables evaluation of the uncertainty of a single prediction only: the analysis must be repeated in order to evaluate the uncertainties associated with multiple predictions.
2. Constrained maximization/minimization based on equations A4.41, A4.42, and A4.44 requires a high level of numerical integrity on the part of the model. Any numerical granularity of model outputs will degrade the accuracy of finite-difference derivatives calculation and, with it, PEST’s ability to maximize/minimize a specified prediction.
3. Where many parameters are used in the parameter-estimation process, the number of model runs required for predictive maximization/minimization can be very large. Tonkin and others (2007) discuss how this burden can be reduced (a) by employing superparameters, in much the same way that the burden of calibration can be reduced through the use of SVD-assisted parameter estimation and/or (b) by employing predictive superparameters computed on the basis of “surrogate predictions.” Although use of these schemes can speed the constrained maximization/minimization process, the prediction maximum/minimum computed by PEST may, unfortunately, be greater/smaller than that which would have been computed in the absence of these numerical conveniences.

Null-Space Monte Carlo

The null-space Monte Carlo methodology presents an efficient means of generating many different parameter fields, all of which (a) respect parameter soft-knowledge “reality constraints” as expressed by the $C(\mathbf{p})$ matrix of innate parameter variability and (b) respect the fact that the model must replicate historical measurements of system state to within a misfit margin set by the $C(\epsilon)$ matrix of measurement noise.

Presently, null-space Monte Carlo analysis can be implemented at two different levels of parameterization. Implementation at the first level allows generation of a suite of calibration-constrained parameter sets where parameterization

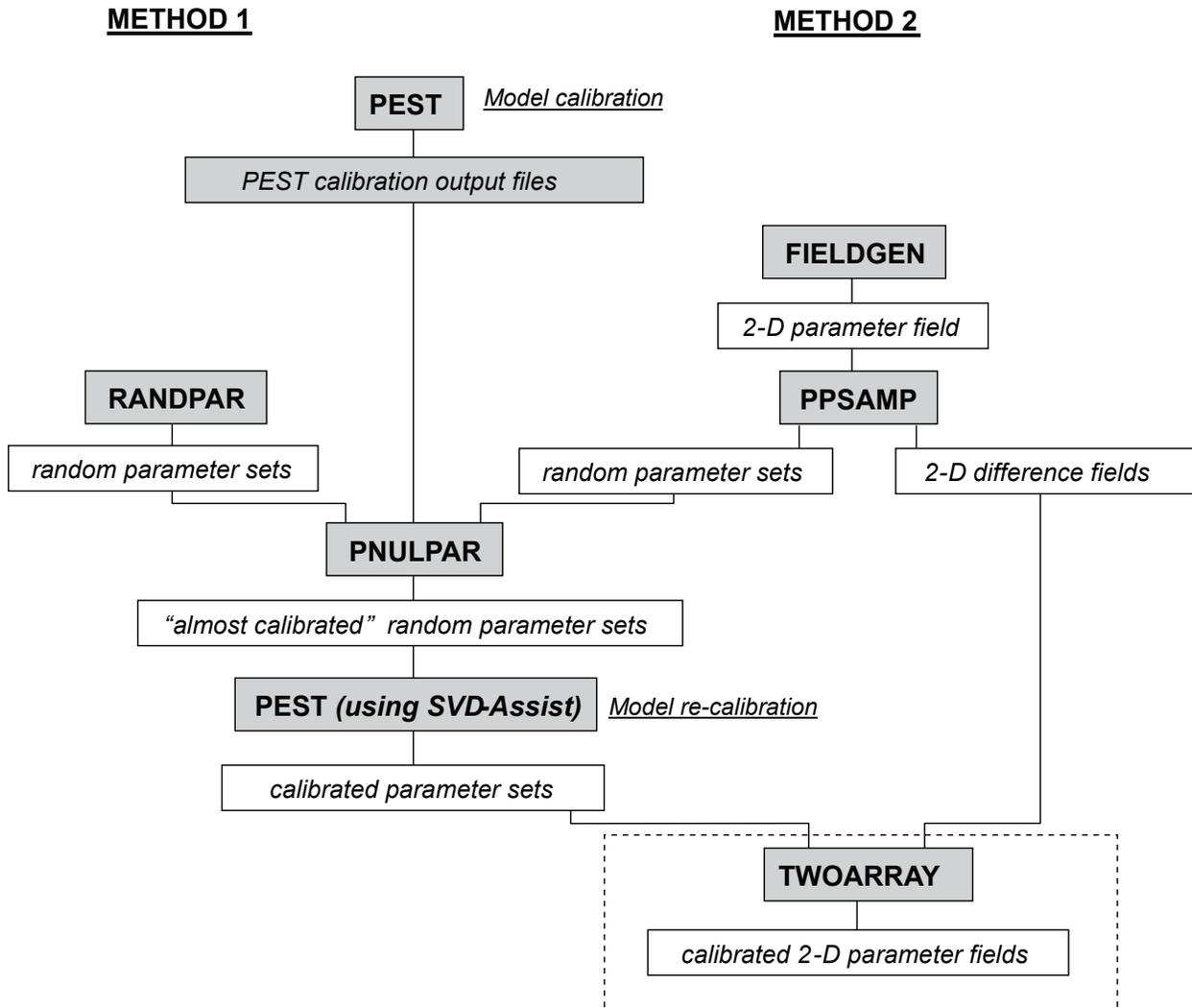


Figure 13. Schematic of null-space Monte-Carlo methodologies. Note that in method 2, maximum effectiveness is achieved if the boxed segment through which parameters and two-dimensional difference fields are recombined is included in the model because it is subject to recalibration.

density is the same as that represented in the calibrated model. Implementation at the second level is applicable to spatial models with continuous or semicontinuous parameter fields; for example, finite-difference groundwater models such as MODFLOW and MT3DMS, where a different value of any property can be assigned to every cell or element within the model domain. Even though parameters are not represented to this level of detail in a governing PEST input dataset (where devices such as pilot points may be used), the null-space Monte Carlo process can explore the effect of cell-by-cell stochastic fields at little additional computational cost.

Practical implementation of both of these methodologies will now be discussed; see figure 13 for a schematic representation. In both cases, it is assumed that calibration has already taken place through highly parameterized, regularized inversion using any (or all) of the regularization techniques available through PEST. It is also assumed that

1. a new PEST control file has been created in which initial parameter values are previously optimized parameter values (the PARREP utility can be used to achieve this);
2. no Tikhonov regularization is used in the new PEST control file;
3. a corresponding Jacobian matrix (*.jco) file has been produced (either through running PEST with NOPTMAX set to -1 or -2, or through use of the JCO2JCO utility to build it from an existing Jacobian matrix file), and
4. the user is able to make a reasonable estimate of the dimensionality of the calibration solution and null spaces; this estimate does not have to be exact, and help in this regard can be obtained by using the SUPCALC utility.

Box 3: Null-Space Monte Carlo—A Flexible and Efficient Technique for Nonlinear Predictive Uncertainty Analysis

Monte Carlo methods have a wide appeal for analysis of predictive uncertainty. Unfortunately, the computational burden of calibration-constrained Monte Carlo usually restricts Monte Carlo analysis to problems involving a small number of parameters: as such, the parameter ensembles cannot reproduce or interrogate fine-scale detail. Null-Space Monte Carlo (NSMC) was developed to overcome this shortcoming. As a result, it can be employed when model run times are long and/or where large numbers of parameters are included.

NSMC analysis takes advantage of the calibration efficiency of the “SVD-Assist” technique (Tonkin and Doherty, 2005; Doherty and Hunt, 2010). It uses the null space defined by singular value decomposition to define stochastic parameter sets that maintain calibration. It honors calibration constraints by projecting differences between stochastic and calibrated parameter values onto the calibration null space before adding each of them to the calibrated parameter field. Thus, rather than postconditioning Monte Carlo results by removing runs that do not meet a calibration criteria, null-space projection of the stochastic parameter set acts to precondition them such that they have the same projection onto the calibration solution space as does the parameter set obtained through the previous calibration process (Figure B3-1).

Unfortunately, because of the nonlinear nature of underlying models, null-space projection of stochastically generated parameter fields does not ensure that such fields provide as good a fit to the calibration dataset as does the field obtained through the preceding calibration process. This problem, however, can be rectified through polishing the original calibration with recalibration. Adjustment of these fields during recalibration is highly efficient because only solution-space components of these fields need to be adjusted by using the SVD-Assist methodology. Therefore, nonlinearity in the underlying model can be accounted for with a minimal computational burden. Reuse of the existing superparameter sensitivities (derived during the original calibration process) allows for further enhancements to computational efficiency.

The NSMC technique is well suited to include fine-scale (even node-scale) parameter detail in the predictive uncertainty analysis process, with reduced computational overhead associated with inclusion of this detail. Including such high levels of parameter flexibility is important if calibration-constrained Monte Carlo analysis is to serve its purpose, because it is this very detail that is generally unconstrained by the calibration process. To the extent that a prediction of interest may be sensitive to this detail, its full range of variability must therefore be represented in the uncertainty analysis process.

The ability of the NSMC to accommodate high-dimensional null spaces is the source of its strength, for in the environmental modeling context, this is often the primary source of predictive uncertainty. Alternative methods such as Markov Chain Monte Carlo often face difficulties in exposing the full extent of the null-space contribution to parameter and predictive uncertainty because of the reliance they place on random-number generation to explore tightly constrained, though nevertheless complex and pervasive, null spaces. The problem is severely compounded where a model takes more than a few seconds to run; hence, making use of the NSMC methodology becomes especially attractive, because other methodologies cannot currently accommodate the numerical burden required in varying a prediction, and the parameters on which it depends, while ensuring a good fit with field data.

Despite NSMC’s computational advantage, the modeler must still balance the number of runs performed with the need to more fully explore the extent of nonuniqueness-generated parameter and predictive uncertainty. The following guidelines can help the modeler decide on the sufficient number of calibration-constrained parameter sets for robust exploration of parameter and predictive uncertainty with a manageable computational burden:

- Take advantage of NSMC’s ability to accommodate different parameterization scales: use enough parameters to fit the field data well during the calibration process. Introduce extra parameters only when needed to explore postcalibration (yet calibration-constrained) predictive uncertainty.
- Employ the Broyden Jacobian update algorithm implement in PEST (see Doherty, 2010b; and Doherty and Hunt, 2010): doing so incurs limited additional computational cost but has demonstrable benefits in effecting reconditioning of null-space-projected parameter fields, particularly during the first reconditioning iteration.
- Increase the number of superparameters employed in this reconditioning process: if NSMC reconditioning results in unsatisfactory recalibration of the new parameter sets, or fails to attain a satisfactory fit to the observed data (that is, a sufficiently low objective function), specification of more superparameters interjects more parameter flexibility into the problem.

In spite of these measures, very long runtimes may sometimes preclude NSMC analysis from providing a full characterization of calibration-constrained predictive uncertainty. If this is the case, the following measures may be considered.

- If the interest is the central tendency, recondition the best-fit parameter sets obtained through null-space projection.
- If the interest is the upper and lower predictive limits (that is, the tails of the distribution), recondition only those parameter sets that lead to predictions that are far from the central tendency after null-space projection. ■

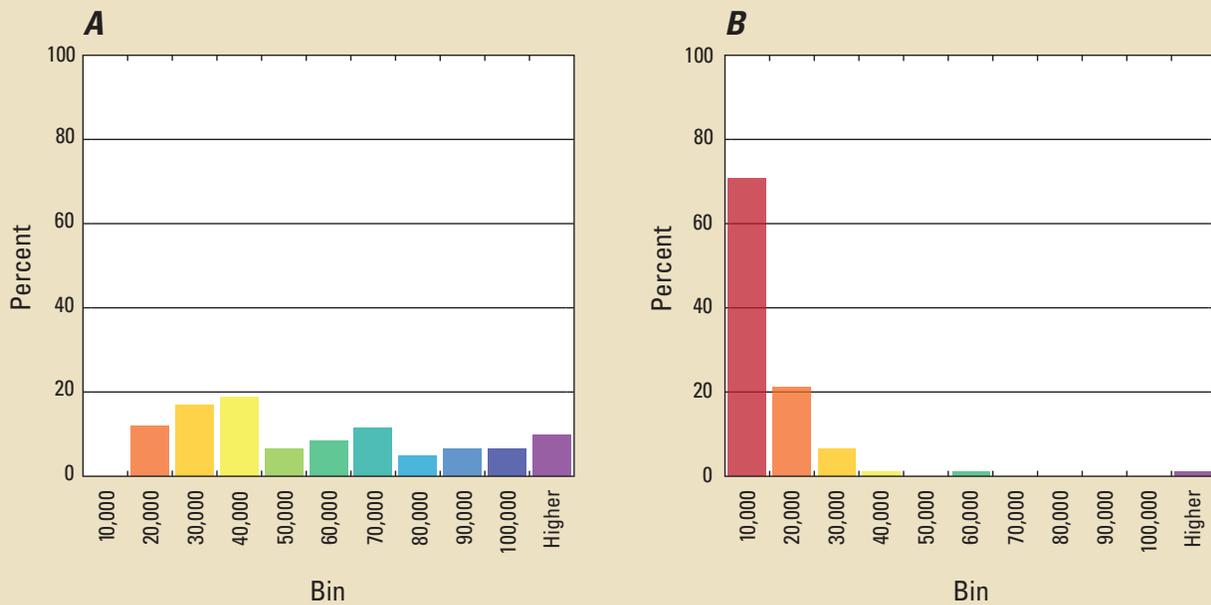


Figure B3-1. Distribution of objective functions computed with 100 stochastic parameter realizations (a) before null-space projection and recalibration; (b) after null-space projection. Generally, the post-null-space projected parameter sets can be reconditioned with only a small computational burden.

Method 1—Using the Existing Parameterization Scheme

As stated above, it is supposed that a model has been calibrated and that the set of parameters estimated by PEST has been placed into a new PEST control file as the initial parameter values featured in this file. It is also supposed that a corresponding Jacobian matrix file exists.

RANDPAR

The RANDom PARameter (RANDPAR) utility can be used to generate random parameter sets based on a covariance matrix of innate parameter variability (that is, $C(\mathbf{p})$) specified in a user-supplied parameter uncertainty file; see PEST documentation for options available for design of this file. RANDPAR places these random parameter sets into a series of PEST parameter value (*.par) files. This is a convenient storage mechanism for these parameter sets, because any one of them can then be deposited into a new PEST control file as an initial parameter set by using the PARREP utility. If NOPTMAX is set to zero in that PEST control file and PEST is initiated, PEST will run the model once using that parameter set, compute the objective function, and then cease execution. If this is done, the objective function will normally be found to be far higher than that computed by using the optimized parameter values attained through the previous calibration exercise.

PNULPAR

The Prepare NULl-space PARameter (PNULPAR) utility should then be used to produce a set of parameter value (*.par) files that complement those produced by RANDPAR. In production of these new files, parameter values stored in the RANDPAR-generated parameter value files undergo null-space projection, excision of solution-space parameter components, and replacement of these components by those pertaining to the calibrated parameter field. The outcomes of PNULPAR execution are multiple parameters sets stored in parameter value files that “almost calibrate” the model: this result can be verified through use of the PARREP utility to repeat the procedure described in the previous paragraph. Each PNULPAR-modified parameter-set counterpart should give rise to a much lower objective function than its RANDPAR-produced counterpart.

SVD-Assisted Parameter Adjustment

SVDAPREP should then be run to produce an input dataset for SVD-assisted parameter estimation based on the PEST control file whose initial parameter values are calibrated parameter values. A copy of this base PEST control file should be kept. Let it be assumed for explanatory purposes that *case_base.pst* is the name of the base parameter PEST control file containing calibrated parameter values, and that *case_base.pst.copy* is the name given to the copy of this file. Let it also be assumed that the name of the superparameter

PEST control file produced by SVDAPREP is *case_svda.pst*. When SVDAPREP is asked whether superparameter sensitivities for the first iteration of the SVD-assisted parameter estimation process should be calculated from existing base parameter sensitivities, the response should be affirmative (as is the default). Although the second iteration will require one model run for each superparameter, the computational cost of the second and subsequent iterations will be still comparatively light because there are fewer superparameters than base parameters.

The modeler must now decide how much effort PEST should devote to recalibrating each null-space-projected random parameter field and at what objective function the model can be deemed to be recalibrated. The NOPTMAX and PHISTOPTHRESH termination variables in the “control data” section of the super parameter PEST control file (*case_svda.pst* in this example) should be set accordingly. Often it is useful to set NOPTMAX to 2. This setting allows the first iteration to be completed at minimal computational cost and a second to be completed at somewhat greater cost. If the objective function has not yet fallen below the designated threshold, recalibration of that parameter field is then abandoned, and recalibration of the next parameter field is commenced. More complex “abandonment schedules” can be programmed by using the PHIABANDON control variable and/or an “abandonment schedule file.” See PEST documentation for further details.

The following control settings are recommended for the superparameter PEST control file (*case_svda.pst* in this above example).

1. JACUPDATE should be set to 999 so that the Broyden update will be used to improve the Jacobian matrix on each parameter upgrade attempt.
2. A “singular value decomposition” section should be given in the superparameter PEST control file, and the EIGTHRESH variable set to between 10^{-7} and 10^{-6} in that section. This setup prevents numerical instability if the number of superparameters is set too high.
3. The initial Marquardt lambda (RLAMBDA1) should be set to 0.01 and the lambda multiplier (RLAMFAC) set to -3 or -4 . These settings enable broad changes to be made to the Marquardt lambda during each optimization iteration, if necessary, thus increasing the opportunities for objective-function reduction.

The Processing Loop

Sequential recalibration of null-space-projected parameter fields is implemented by undertaking repeated PEST runs by use of a batch file such as that illustrated in figure 14; a similar procedure can be embodied in a script file in the UNIX environment. The result is a sequence of parameter-value files, many of which result in an objective function that is low enough for the model to be considered calibrated. However, each of these parameter-value files will contain a appreciably different set of parameters.

```

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 nsp_random%%i.par case_base.pst.copy case_base.pst
pest case_svda
find /I "Sum of squared weighted residuals" case_svda.rec >> record.dat
copy case_svda.rec case_svda.rec.%%i
copy case_base.bpa case_base.bpa.%%i)

```

Figure 14. A batch process that implements successive recalibration of 100 null-space-projected parameter fields

The batch process illustrated in figure 14 commences with deletion of a file named *record.dat*. This file is reinitiated, and then regularly updated, as the sequential recalibration process continues. Regular inspection of this file allows the modeler to monitor the progress of the sequential recalibration processes implemented by the batch process. (Initial deletion of *record.dat* prevents old records from being mistaken for new ones because information is appended to this file at the completion of each recalibration exercise.) The processing loop then encompasses the following steps:

1. Null-space-projected parameter values contained in a PNULPAR-produced parameter value file (named *nsp_random1.par*, *nsp_random2.par*, etc.) are used to make a new base PEST control file, the name of which is *case_base.pst*. In each case, the newly created *case_base.pst* cites initial parameter values that are in null-space-projected parameter values that should “almost calibrate” the model.
2. SVD-assisted parameter estimation is initiated on the basis of the SVDAPREP-produced superparameter PEST control file *case_svda.pst*. The superparameter adjustment process alters the solution-space component of the initial parameter field to achieve recalibration.
3. Once each calibration process is terminated according to user-defined criteria, the *find* command is used to append the initial and final objective functions pertaining to each completed recalibration exercise to file *record.dat*. As stated above, this file is always available for user inspection.
4. The SVDA run record file is copied to a file whose name includes the index of the null-space-projected parameter set subjected to recalibration.
5. Updated base parameter values are copied to a similarly indexed parameter-value file for safekeeping.

Each recalibration process can employ serial or parallel PEST. In the latter case, execution of PEST’s slaves should be initiated before beginning the batch-file processing. Furthermore, slaves should be run by using the “/i” command-line switch so that they do not shut down at the completion of each parallel PEST run.

Postprocessing

Parameters recorded in the parameter value files produced through the process depicted in figure 14 can be collected into a single table by using the MULtiple PARAmeter TABLE (MULPARTAB) utility. Options available through this utility can exclude parameter sets that do not achieve an objective function that is low enough for the model to be considered calibrated. The resulting table can then be imported into a spreadsheet or plotting program where parameter frequency histograms can be produced.

The COMBine FILENaME (COMFILNME) utility can be used to resequence indexed parameter-value filenames to omit rejected parameter sets if desired.

Making Predictions

The outcome of a null-space Monte Carlo exercise is a sequence of parameter sets that (a) calibrate the model and (b) can be considered to be “reasonable.” When the model is used to predict system behavior, predictions should be made from all of these parameter sets. Simulated predictions can then be collected into tables for plotting as histograms or correlograms or for generating plots of other types that allow easy visualization of their statistical properties.

Multiple model runs based on different parameter sets can be made by using a batch processing loop that is similar to that depicted in figure 14. In that loop, the PARREP utility can be used to sequentially place parameter sets into a PEST control file that is set up for a dummy calibration exercise but that actually runs the model just once under prediction conditions. Alternatively, through a series of TEMPlATE CHECk (TEMPCHEK) commands, model input files can be written on the basis of parameter values stored in a sequence of parameter-value files and the model then run on the basis of each parameter set. After each PEST or model run, model output files containing predictions of interest should be copied to run-indexed files for later processing.

The ReaD MULtiple RESults (RDMULRES) utility can be used to read a sequence of model output files produced in this manner, obtaining from those files model outputs of interest. These outputs are tabulated in a single file that can then be imported into a spreadsheet or plotting program for histogram generation or for undertaking other statistical analyses.

Method 2—Using Stochastic Parameter Fields

General

Where spatial models such as groundwater models are subject to calibration, parameterization devices such as pilot points are often used to represent spatially varying property fields. As is described in Doherty, Fienen, and Hunt (2010), values are assigned to these pilot points through the calibration process. Property values are then assigned to cells or elements composing a model grid or mesh through spatial interpolation from these points. This interpolation process results in smoother parameter fields than the property fields that exist in reality. However, because simplification-induced smoothness is a natural outcome of the quest for uniqueness that is integral to the model calibration process, nothing is lost to the calibration process through the use of pilot points that is not already lost through the need to regularize in order to attain parameter uniqueness of the calibrated parameter field. See Moore and Doherty (2006) and Doherty, Fienen, and Hunt (2010) for further details.

When using the null-space Monte Carlo methodology to produce a suite of parameter fields which, on the one hand, calibrate a model while, on the other hand, are hydrogeologically realistic, one can legitimately expect that such parameter

fields include not only components of the null space that are lost through the calibration process (and that can be represented by use of pilot points or a similar parameterization scheme) but also the components of the null space that are lost through the use of the specific parameterization scheme. Fortunately, these components can be included by using a slight modification of the procedure discussed in the previous subsection.

Generation of Stochastic Parameter Fields

The FIELD GENeration (FIELDGEN) utility supplied with the PEST Groundwater Data Utilities suite (Doherty, 2008) is a MODFLOW/MT3D-compatible stochastic field generator. It is based on the *sgsim* algorithm supplied with the GSLIB geostatistical software library (Deutsch and Journel, 1998). By use of FIELDGEN, stochastic fields can be generated for model grids of arbitrary complexity. If desired, different stochastic fields can be generated within different user-defined zones within a model domain on the basis of different variograms.

Using FIELDGEN, a modeler can generate any number of stochastic hydraulic-property field realizations. These fields are stored in MODFLOW-compatible array files whose names include the field index.

Pilot-Point Sampling (PPSAMP)

The PPSAMP utility (another Groundwater Data Utility) assigns values to pilot-point parameters through sampling of FIELDGEN-generated stochastic fields. Two methods of pilot-point value assignment are given: the first, through direct sampling; the second, through assignment of values such that the grid-based field that is interpolated from the pilot points shows minimum differences with the sampled stochastic field. The latter is the better alternative to use in the present context.

For each sampled stochastic field, PPSAMP writes a parameter value file in which sampled pilot-point values are recorded. It also records a MODFLOW-compatible array file that houses a “difference field”; this difference field is formed by computing differences on a cell-by-cell basis between the sampled stochastic field and the hydraulic-property field that is interpolated between pilot-point values sampled from this field. If pilot-point spatial density is sufficiently high, this difference field lies within the calibration null space.

Null-Space Projection

Once a sequence of pilot-point parameter sets have been obtained and stored in parameter-value files, the PNULPAR utility can be used for null-space projection of these parameter sets and replacement of their solution-space components with those obtained through the calibration process in the manner discussed above. As before, the outcome of PNULPAR processing is a revised set of parameter-value files.

Model Recalibration

For each sampled parameter set, recalibration is implemented in a similar manner to that described above using SVD-assist in conjunction with base PEST control files containing null-space-projected pilot-point parameter values as initial parameter values. However, the batch model that is undergoing calibration is altered slightly such that after interpolation from pilot points to the model grid is complete, the difference field pertinent to the current sampled pilot-point parameter set is added to the pilot-point-interpolated field to produce a new property array that is then used by the model. Thus, model recalibration is effected with this difference field “riding on the back” of the sampled (and adjusted through recalibration) pilot-point parameter set. At the end of the recalibration process, the resulting parameter field thus calibrates the model while retaining cell-by-cell detail present in the original FIELDGEN-generated stochastic field.

This sampling and recalibration process is repeated for each FIELDGEN stochastic field.

Some Practical Considerations

In many modeling contexts, generation of a sequence of calibration-constrained parameter fields by use of either of the methods outlined above can be implemented with moderate to high levels of model-run efficiency. Aspects of the above procedures that contribute to this efficiency include the following:

1. Null-space projection of a random parameter field to remove solution-space components that differ from those of the calibrated parameter field results in a parameter field that may be very different from that achieved through the calibration process but that decalibrates the model to only a small extent.
2. Parameter-field recalibration is achieved by adjusting a limited number of superparameters that collectively span the calibration solution space.
3. Reuse of existing parameter sensitivities for the first iteration of all parameter-field recalibration processes reduces the numerical burden of this first iteration to that of undertaking only one or two model runs.

In some modeling contexts, however, particularly those where nonlinearity is high, the numerical burden of recalibrating many null-space-projected random parameter fields may present obstacles to the use of the null-space Monte Carlo methodology. This will occur in modeling contexts that require more than one optimization iteration for recalibration. In such cases, compromise is warranted. The following strategies can reduce this burden; albeit, the cost of reducing this burden may be underestimation or overestimation of postcalibration parameter-field variability, depending on the strategy chosen:

1. The threshold objective function at which a model is deemed to be calibrated could be raised.

2. When using RANDPAR or FIELDGEN to generate random parameter fields, a relatively small variance can be assigned to each parameter in the $C(\mathbf{p})$ matrix on which field generation is based.
3. When using PNULPAR to perform null-space projection of parameter sets, a user can inform this program that the dimensionality of the calibration solution space is larger than it really is. With parameter sets projected onto a smaller dimensional null space, the likelihood of violating calibration constraints can be reduced, as will the burden of recalibrating these parameter sets.

Choice of the value of the threshold objective function at which model recalibration can be considered to have occurred is normally a matter of some subjectivity. In theory, an objective-function threshold should be applied to model output *differences* rather than to model-to-measurement residuals, as was discussed for the constrained predictive maximization/minimization methodology; alternatively, equation A4.43 can be used as a guide. However, in almost all calibration contexts, the dominance of structural noise over measurement noise makes the quest for mathematical rigor somewhat irrelevant. Added to this is the fact that choice of an appropriate $C(\mathbf{p})$ matrix of innate variability as a basis for stochastic field generation will always be a matter of some subjectivity.

Pareto-Based Hypothesis Testing

Predictive hypothesis testing by means of the Pareto methodology is implemented by running PEST in Pareto mode. This is done by setting the PESTMODE variable in the control data section of the PEST control file to “pareto.” A “pareto” section must also be added at the end of the PEST control file, this providing PEST with the values of control variables that are required for implementation of this kind of analysis. As is discussed in Moore and others (2010) and Doherty (2010b) PEST’s Pareto functionality can be deployed as an aid to evaluate regularized inversion and as an aid to predictive uncertainty analysis; only its role in the latter context is considered here.

Hypothesized Predictive Value

As was documented previously, use of PEST in a hypothesis-testing capacity requires that a prediction of interest be “observed.” The calibrated parameter field, as well as the capacity of the model to realize adequate replication of historical system behavior, is then tested for its compatibility with this prediction. In many cases, the prediction will pertain to an undesirable environmental occurrence; the “observed value” of the prediction will then represent a relatively extreme event. However, because it traverses the Pareto front, PEST actually assesses the credibility of all values of the prediction between the user-depicted worst case and the value of the prediction yielded by the (hopefully minimum-error-variance) parameter set that is deemed to calibrate the model.

The hypothesized prediction should be introduced to a PEST control file in which parameters are set to their previously calibrated values. When adding the hypothesized prediction to the “observation data” section of this PEST control file, one should assign it to an observation group that contains no other observations. The prediction should be assigned a weight that is sufficiently high for the contribution of this single-member observation group to the overall objective function to be commensurate with that of all other groups when the model is run on the basis of the calibrated parameter set. Thus, if PEST were rerun after introduction of this prediction to the calibration process, it would have a reasonable chance of assigning values to parameters that would allow this prediction to be respected.

Pareto Control Variables

As described by Doherty (2010b), when run in Pareto mode, PEST traverses the Pareto front by assigning ever-increasing weights to a user-specified observation group. When undertaking Pareto-based hypothesis testing, the initial weight factor assigned to the observation group whose sole member is the hypothesized prediction should be zero (PEST variable `PARETO_WTFAC_START`). The final weight factor (`PARETO_WTFAC_FIN`) should be 1.0; however a modeler may review this setting on an ensuing PEST run if the hypothesized prediction is not achieved or is achieved too readily. The number of weight-factor increments should be set according to the computing power to which the modeler has access. Ideally, it should be set to a high value (anything between 40 and 100), because the use of a high number of increments generally promulgates close PEST adherence to the Pareto front during its traversal. However, acceptable results can often be obtained with only 10 weight-factor increments.

The modeler must also decide how many PEST iterations should be devoted to each weight-factor increment (PEST variable `NUM_ITER_GEN`). A value of 2 or 3 is suitable on most occasions. In most hypothesis-testing contexts, the number of iterations to devote to the final weight factor (PEST variable `NUM_ITER_FIN`) should be the same as that devoted to other weight factors (that is, `NUM_ITER_GEN`), whereas the number of iterations devoted to the initial weight factor (variable `NUM_ITER_START`) should be zero.

Model-run efficiency can be gained if PEST is instructed to cease its traversal of the Pareto front if a threshold value is reached for the hypothesized prediction. This functionality can be implemented through use of the `ALT_TERM` Pareto control variable.

Mapping the Pareto Front

When run in Pareto mode, PEST writes a Pareto objective function (**.pod*) file that is available for inspection at all stages of the Pareto-front traversal process. Data columns within this file can be pasted into plotting software in order to provide the

user with a current view of the Pareto front as it is mapped. At the end of the Pareto run, the binary Pareto parameter data (**.ppd*) file contains parameters and model outputs pertaining to all parameter sets that PEST has encountered because it has attempted to traverse the Pareto front. Data can be extracted from this file by using the `PPD2ASC` and `PPD2PAR` utilities listed in appendix 1 and described in Doherty (2010b).

Multiple Recalibration

This section finishes with a brief description of a simple but effective means by which parameter and predictive uncertainty can be at least partially explored.

In many instances of practical model calibration, only a fraction of the many values that must be assigned to properties and boundary conditions used by a model are adjusted. Where calibration is performed by means of methodologies based on overdetermined parameter estimation, precalibration parameter simplification, lumping, and wholesale fixing are necessary to ensure well-posedness of the inverse problem of model calibration. Where model calibration is based on regularized inversion, such wholesale parameter simplification and eradication is no longer necessary. Nevertheless, such simplification is often done to some extent for convenience, either to reduce the numerical burden of calculating derivatives with respect to many parameters or to account for model-output numerical granularity, which may accompany small changes made to certain of its parameters and/or boundary conditions for the purpose of calculating finite-difference derivatives with respect to these parameters.

We pointed out earlier in this document that a parameter can be fixed during the calibration process yet declared adjustable for the purpose of linear uncertainty analysis. The potential for error that is accrued through fixing a parameter or boundary condition (or a suite of parameters or boundary conditions) at uncertain value(s) can then be assessed by using the `PREDVAR4` and `PREDUNC4` utilities. If a modeler does not wish to do linear uncertainty analysis in this fashion, a simple but effective means of assessing the consequences of assigning an erroneous value to a particular unadjusted parameter or boundary condition is to recalibrate the model with the parameter or boundary condition fixed at an alternative value. This process can be repeated (while varying the same or different parameters or boundary conditions) as many times as necessary. When using the model for environmental management, a prediction may then be made by using parameter fields emanating from all of these separate calibration processes. Although this strategy cannot provide a basis for quantitative analysis of the uncertainty of that prediction, it can provide a basis for assessment of the importance or otherwise of one or a number of assumptions that underpins the model’s construction, at the same time allowing a qualitative exploration of the uncertainty associated with one or multiple predictions to be made.

Uses and Limitations of Model Uncertainty Estimates

A summary of uncertainty methodologies available through the PEST suite has been presented, together with a brief description of the theoretical underpinnings of these methodologies and of the software through which they can be implemented. Many of the methodologies discussed in this document can be implemented in conjunction with highly parameterized models. They therefore adhere to the principle that, to the extent that a prediction depends on property detail that is beyond the capacity of the calibration process to constrain, that prediction will owe much of its uncertainty to variability/heterogeneity of property fields for that characterization. Further, variability constraints must be supplied by the modeler's soft-knowledge, for these constraints cannot be obtained from the calibration dataset. It follows that for some predictions, calibration can inform and lower predictive uncertainty while in other cases it will not. A rigorous method for assessing postcalibration predictive uncertainty should therefore recognize and accommodate the weaknesses of the calibration process. It should inform the modeler where it is reasonable to expect reduction of predictive uncertainty below its precalibration level and where uncertainty must be maintained at its precalibration level because calibration using the current dataset cannot be expected to reduce it.

Both the linear and nonlinear methods described herein for assessment of parameter and predictive uncertainty in the highly parameterized context have utility. Central to both of these methods are two stochastic distributions. The first of these, represented by the $C(\mathbf{p})$ matrix herein, describes the innate variability of parameters. This must be supplied by the modeler based on their expert knowledge of the system. The second of these distributions, represented by the covariance matrix $C(\varepsilon)$, characterizes the noise associated with measurements of system state; traditional uncertainty based on overdetermined problems considers only this term.

A problem in undertaking model-parameter and predictive uncertainty analysis is that neither $C(\mathbf{p})$ nor $C(\varepsilon)$ can be known with a high degree of accuracy. To the extent that a modeler assumes a level of uncertainty regarding properties at a site, the $C(\mathbf{p})$ matrix expresses that assumption. If the level of hydraulic-property variability is greater than assumed, predictive uncertainty may be underestimated; this applies especially to transport predictions that are sensitive to spatial continuity of relatively high or low properties within a model domain. Similar problems pertain to selection of an appropriate $C(\varepsilon)$ matrix. In most cases, model outputs that are assumed to be equivalent to field observations of a system are, in fact, accompanied by a large degree of structural noise, in addition to measurement error. This structural noise arises from the imperfect nature of model simulation algorithms, use of a coarse grid or time discretization, approximations involved in the assignment of boundary conditions, use of erroneous inputs of quantities such as recharge and rainfall, and the

implicit and/or explicit regularization required for obtaining an estimated parameter field with a level of simplification that guarantees its uniqueness. Such structural noise invariably shows a high degree of spatial and/or temporal correlation. Failure to recognize this correlation can lead to the conclusion that there is more information in the calibration dataset than there actually is or that can be transferred to the model through values assigned to its parameters. This can lead to underestimation of the degree to which predictions made by a calibrated model may be wrong.

The uncertainty associated with user-supplied estimates of $C(\mathbf{p})$ and $C(\varepsilon)$ should ideally be incorporated into the analysis of model predictive uncertainty. Where parameter estimation takes place in the overdetermined calibration context, this occurs through calculation of the reference variance (Draper and Smith, 1998) from the value of the minimized objective function. In the underdetermined context, Moore and Doherty (2006) demonstrate that the magnitude of $C(\mathbf{p})$ can be traded off against that of $C(\varepsilon)$, depending on whether a modeler feels more comfortable with a high degree of parameter heterogeneity within a calibrated parameter field or with a high degree of model-to-measurement misfit. Unfortunately, it is unlikely that uncertainty analysis will achieve a level of quantification that complements that associated with other aspects of model use. As a result, whereas the methods for assessing model parameter and predictive uncertainty presented herein provide a powerful mechanism for assessing the credibility of model predictions, rigorous quantification of confidence limits around values of predicted model outputs will remain a worthy, but unreachable, goal.

Despite the fact that predictive confidence limits can only be approximate, assessment of postcalibration model predictive variability still has value. For example, an assessment of the uncertainty associated with a model prediction facilitates use of that prediction in the decisionmaking process. Moreover, based on the premise that the best data to gather are those that are most effective in decreasing the uncertainty of one or a number of key model predictions, the development of optimal and cost-efficient strategies for data acquisition is best based on an assessment of current predictive uncertainty together with an assessment of the extent to which this uncertainty can be reduced. Central to any decisionmaking process is an assessment of risk. Such an assessment is impossible without some assessment of predictive uncertainty. If analysis reveals that certain unwanted environmental occurrences are possible given the current state of system knowledge, then the mechanisms through which such occurrences may eventuate can often be exposed through model-based uncertainty analysis. This, in turn, can lead to optimal design of a monitoring network. Finally, knowledge of the contributions made by different facets of a modeled system to current levels of predictive uncertainty can further a modeler's understanding of the strengths and weaknesses of that model, as well as ways in which it may be improved in order to better serve the decisionmaking process.

Routine analysis of model predictive uncertainty will hopefully lead to a better recognition of what is, and is not, possible for models to achieve. This insight, in turn, may lead to better-informed decisions on whether the approach to modeling taken at a particular study site should be complex and expensive or simple and inexpensive. If underlying information pertaining to critical model predictions is weak, that information may be just as efficiently extracted from the current environmental dataset by using a simple model as a complex one. In many cases, both simple and complex models can be assigned parameter sets that lead to predictions of minimum error variance. However, what a complex model can achieve that a simpler model cannot achieve is quantification of the magnitude of that variance, as well as an indication of those aspects of the system that are contributing most to its current magnitude. If such information is important in its own right and/or is important for designing a monitoring network, a strategy for future data acquisition, or for quantification of risk, then use of a complex model is not optional. However, if predictive uncertainty is already taken into account through implementation of representative engineering safety factors, then use of a simple model may suffice. It is possible therefore, that routine use of complex models in exploration of model predictive uncertainty may lead to greater use of simple models, as modelers gain enough understanding of the predictive strengths and weaknesses of the modeling process that can then support “the courage to be simple.”

References

- Aster, R.C., Borchers, B., and Thurber, C.H., 2005, *Parameter estimation and inverse problems*: Amsterdam, Elsevier Academic Press, 301 p.
- Beven, K., 1993, Prophecy, reality and uncertainty in distributed hydrological modeling. *Advances in Water Resources*, 16, 41-51.
- Belsley, D.A.; Kuh, E., and Welsch, R.E., 1980, *Regression diagnostics—Identifying influential data and sources of collinearity*: New York, Wiley, 292 p.
- Christensen, S., and Cooley, R.L., 1999, Evaluation of prediction intervals for expressing uncertainties in groundwater flow model predictions: *Water Resources Research*, v. 35, no. 9, p. 2627–2639.
- Christensen, S., and Doherty, J., 2008, Predictive error dependencies when using pilot points and singular value decomposition in groundwater model calibration. *Advances in Water Resources*, 31(4), 674-700.
- Cook, R.D., and Weisberg, S., 1982, Residuals and influence in regression: New York, Chapman and Hall, v. 18 of *Monographs on Statistics and Applied Probability*, 230 p.
- Cooley, R.L., 2004, A theory for modeling ground-water flow in heterogeneous media: U.S. Geological Survey Professional Paper 1679, 220 p.
- Cooley, R.L., and Christensen, S., 2006, Bias and uncertainty in regression-calibrated models of groundwater flow in heterogeneous media: *Advances in Water Resources*, v. 29, no. 5, p. 639–656.
- Dausman, A.M.; Doherty, J., Langevin, C.D., and Sukop, M.C., 2010, Quantifying data worth toward reducing predictive uncertainty: *Ground Water*, v. 48, no. 5, p. 729–740.
- Deutsch, C.V., and Journel, A.G., 1998, *GSLIB—Geostatistical software library and user’s guide* (2d ed.): New York, Oxford University Press, 369 p.
- Doherty, J., 2003, Ground water model calibration using pilot points and regularization: *Ground Water*, v. 41, no. 2, p. 170–177.
- Doherty, J., 2008. *Groundwater data utilities*: Brisbane, Australia, Watermark Numerical Computing.
- Doherty, J., 2009. *Model-independent Markov chain Monte Carlo analysis*: Brisbane, Australia, Watermark Numerical Computing.
- Doherty, J., 2010a, PEST, Model-independent parameter estimation—User manual (5th ed., with slight additions): Brisbane, Australia, Watermark Numerical Computing.
- Doherty, J., 2010b, Addendum to the PEST manual: Brisbane, Australia, Watermark Numerical Computing.
- Doherty, J., Fienen, M.F., and Hunt, R.J., 2010, Approaches to highly parameterized inversion: Pilot-point theory, guidelines, and research directions: U.S. Geological Survey Scientific Investigations Report 2010–5168, 36 p.
- Doherty, J., and Hunt, R.J., 2009a, Two statistics for evaluating parameter identifiability and error reduction: *Journal of Hydrology*, v. 366, no. 1–4, p. 119–127, doi: 10.1016/j.jhydrol.2008.12.018.
- Doherty, J., and Hunt, R.J., 2009b, Response to comment on “Two statistics for evaluating parameter identifiability and error reduction”: *Journal of Hydrology*, v. 380, no. 3–4, p. 489–496, doi: 10.1016/j.jhydrol.2009.10.012
- Doherty, J., and Hunt, R.J., 2010, Approaches to highly parameterized inversion: A guide to using PEST for ground-water-model calibration: U.S. Geological Survey Scientific Investigations Report 2010–5169, 59 p.
- Doherty, J., and Welter, D.E., 2010, A short exploration of structural noise: *Water Resources Research*, v. 46, W05525, doi:10.1029/2009WR008377.
- Draper, N.R., and Smith, H., 1998, *Applied regression analysis* (3d ed.): New York, John Wiley and Sons, 706 p.

- Fienen, M.N., Muffels, C.T., and Hunt, R.J., 2009, On constraining pilot point calibration with regularization in PEST. *Ground Water*, v. 47, no. 6, p. 835-844. doi: 10.1111/j.1745-6584.2009.00579.x
- 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 Availability Pilot Project: U.S. Geological Survey Scientific Investigations Report 2010–5159, 44 p.
- Gallagher, M., and Doherty, J., 2007a, Parameter estimation and uncertainty analysis for a watershed model: *Environmental Modelling and Software*, v. 22, no. 7, p. 1000–1020.
- Gallagher, M., and Doherty, J., 2007b, Predictive error analysis for a water resource management model: *Journal of Hydrology*, v. 334, no. 3–4, p. 513–533.
- Hadi, A.S., 1992, A new measure of overall potential influence in linear regression: *Computational Statistics and Data Analysis*, v. 14, p. 1–27.
- Hunt, R.J.; Doherty, J., and Tonkin, M.J., 2007, Are models too simple? Arguments for increased parameterization: *Ground Water*, v. 45, no. 3, p. 254–262, doi: 10.1111/j.1745-6584.2007.00316.x.
- Hunt, R.J., and Doherty, J., 2006, A strategy of constructing models to minimize prediction uncertainty, in *MODFLOW and More 2006—Managing Ground Water Systems*, Proceedings of the 7th International Conference of the International Ground Water Modeling Center: Golden, Colo., Colorado School of Mines, p. 56–60.
- Hunt, R.J., Feinstein, D.T., Pint, C.D., and Anderson, M.P., 2006, The importance of diverse data types to calibrate a watershed model of the Trout Lake Basin, northern Wisconsin, USA: *Journal of Hydrology*, v. 321, no. 1–4, p. 286–296.
- Hunt, R.J., and Welter, D.E., 2010, Taking account of “unknown unknowns”. Editorial for *Ground Water* 48(4): 477, doi: 10.1111/j.1745-6584.2010.00681.x
- Matott, L.S., 2005, OSTRICH, an optimization software tool—Documentation and user’s guide: State University of New York at Buffalo, Department of Civil, Structural and Environmental Engineering, available at <http://www.groundwater.buffalo.edu/software/Ostrich/OstrichMain.html>.
- Moore, C., and Doherty, J., 2005, The role of the calibration process in reducing model predictive error: *Water Resources Research*, v. 41, no. 5, W05050.
- Moore, C., and Doherty, J., 2006, The cost of uniqueness in groundwater model calibration: *Advances in Water Resources*, v. 29, no. 4, p. 605–623.
- Moore, C., Wöhling, T., and Doherty, J., 2010, Efficient regularization and uncertainty analysis using a global optimization methodology: *Water Resources Research*, v. 46, W08527, doi:10.1029/2009WR008627.
- Paige, C.C., and Saunders, M.A., 1982a, LSQR—An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, v. 8, no. 1, p. 43–71.
- Paige, C.C., and Saunders, M.A., 1982b, Algorithm 583—LSQR—Sparse linear equations and least squares problems: *ACM Transactions on Mathematical Software*, v. 8, no. 2, p. 195–209.
- Poeter, E.P.; Hill, M.C.; Banta, E.R.; Mehl, S., and Christensen, S., 2005, UCODE_2005 and six other computer codes for universal sensitivity analysis, calibration, and uncertainty evaluation: U.S. Geological Survey Techniques and Methods 6–A11, 283 p.
- Tonkin, M.J., and Doherty, J., 2005, A hybrid regularized inversion methodology for highly parameterized environmental models: *Water Resources Research*, v. 41, no. 10, W10412, doi:10.1029/2005WR003995.
- Tonkin, M.J., and Doherty, J., 2009, Calibration-constrained Monte-Carlo analysis of highly parameterised models using subspace techniques: *Water Resources Research*, v. 45, no. 12, W00B10, doi:10.1029/2007WR006678.
- Tonkin, M.J., Doherty, J., and Moore, C., 2007, Efficient nonlinear predictive error variance for highly parameterized models. *Water Resources Research*, v. 43, no. 7, W07429, doi:10.1029/2006WR005348.
- Vecchia, A.V., and Cooley, R.L., 1987, Simultaneous confidence and prediction intervals for nonlinear regression models with application to a groundwater flow model: *Water Resources Research*, v. 23, no. 7, p. 1237–1250.
- Yager, R.M., 1998, Detecting influential observations in nonlinear regression modeling of groundwater flow: *Water Resources Research*, v. 34, no. 7, p. 1623–1633.

Appendixes 1–4

Appendix 1. Basic PEST Input

Structure of the PEST Control File

This appendix supplies a short description of all PEST variables. First, a list of all of these variables is provided, with each located in its proper place within the PEST Control File (variables enclosed in brackets are optional). This listing is followed by a series of tables that describe the role of each variable.

```

pcf
* control data
RSTFLE PESTMODE
NPAR NOBS NPARGP NPRIOR NOBSGP [MAXCOMPDIM]
NTPLFLE NINSFLE PRECIS DPOINT [NUMCOM] [JACFILE] [MESSFILE]
RLAMBD1 RLAMFAC PHIRATSUF PHIREDLAM NUMLAM [JACUPDATE]
RELPARMAX FACPARMAX FACORIG [IBOUNDSTICK UPVECBEND]
PHIREDSWH [NOPTSWITCH] [SPLITSWH] [DOAUI] [DOSENREUSE]
NOPTMAX PHIREdstp NPHISTP NPHINORED RELPARSTP NRELPAR [PHISTOPHRESH] [LASTRUN] [PHIABANDON]
ICOV ICOR IEIG [IRES] [JCSAVEITN] [REISAVEITN]
* automatic user intervention
MAXAUI AUISTARTOPT NOAUIPHIRAT AUIRESTITN
AUISENSRAT AUIHOLDMAXCHG AUINUMFREE
AUIPHIRATSUF AUIPHIRATACCEPT NAUINOACCEPT
* singular value decomposition
SVDMODE
MAXSING EIGHTHRESH
EIGWRITE
* lsqr
LSQRMODE
LSQR_ATOL LSQR_BTOL LSQR_CONLIM LSQR_ITNLIM
LSQRWRITE
* svd assist
BASEPESTFILE
BASEJACFILE
SVDA_MULBPA SVDA_SCALADJ SVDA_EXTSUPER SVDA_SUPDERCALC
* sensitivity reuse
SENRELTRESH SENMAXREUSE
SENALLCALCINT SENPREDWEIGHT SENPIEXCLUDE
* parameter groups
PARGPNAME INCTYP DERINC DERINCLB FORCEN DERINCMUL DERMTHD
[SPLITTHRESH SPLITRELDIFF SPLITACTION]
(one such line for each of NPARGP parameter groups)
* parameter data
PARNAME PARTRANS PARCHGLIM PARVAL1 PARLBNB PARUBND PARGP SCALE OFFSET DERCOM
(one such line for each of NPAR parameters)
PARNAME PARTIED
(one such line for each tied parameter)
* observation groups
OBSNAME [GTARG] [COVFLE]
(one such line for each of NOBSGP observation group)
* observation data
OBSNAME OBSVAL WEIGHT OBSNAME
(one such line for each of NOBS observations)
* derivatives command lineDERCOMLINE

```

```

EXTDERFLE
* model command line
COMLINE
(one such line for each of NUMCOM command lines)
* model input/output
TEMPFLE INFLE
(one such line for each of NTPLFLE template files)
INSFLE OUTFLE
(one such line for each of NINSLFE instruction files)
* prior information
PILBL PIFAC * PARNME + PIFAC * log(PARNME) ... = PIVAL WEIGHT OBGNME
(one such line for each of NPRIOR articles of prior information)
* predictive analysis
NPREDMAXMIN [PREDNOISE]
PDO PD1 PD2
ABSPREDLAM RELPREDLAM INITSCHFAC MULSCHFAC NSEARCH
ABSPREDSWH RELPREDSWH
NPREDNORED ABSPREDSTP RELPREDSTP NPREDSTP
* regularisation
PHIMLIM PHIMACCEPT [FRACPHIM] [MEMSAVE]
WFINIT WFMIN WFMAX [LINREG] [REGCONTINUE]
WFFAC WFTOL IREGADJ [NOPTREGADJ REGWEIGHTRAT [REGSINGTHRESH]]
* pareto
PARETO_OBSGROUP
PARETO_WTFAC_START PARETO_WTFAC_FIN NUM_WTFAC_INC
NUM_ITER_START NUM_ITER_GEN NUM_ITER_FIN
ALT_TERM
OBS_TERM ABOVE_OR_BELOW OBS_THRESH NUM_ITER_THRESH (only if ALT_TERM is non-zero)
NOBS_REPORT
OBS_REPORT_1 OBS_REPORT_2 OBS_REPORT_3.. (NOBS_REPORT items)

```

Figure 1–1. Structure of the PEST Control File.

The following tables include a column labelled “usage,” which contains an index whose value is between 1 and 3. An index value of 3 indicates that the variable is likely to vary in value from PEST Control File to PEST Control File, this reflecting either the different nature of different parameter estimation problems, or the fact that, as a control variable, it is one that often requires “tuning” to a particular calibration problem. On the other hand, a usage index value of 1 indicates that the variable rarely requires alteration from the value suggested in PEST documentation. A usage value of 2 indicates potential variability that is between these two extremes

Variables in the “control data” section of the PEST Control File.

Variable	Type	Values	Usage	Description
RSTFLE	text	“restart” or “norestart”	1	Instructs PEST whether to write restart data.
PESTMODE	text	“estimation”, “prediction”, “regularization”, “pareto”	3	PEST’s mode of operation.
NPAR	integer	greater than 0	3	Number of parameters.
NOBS	integer	greater than 0	3	Number of observations.
NPARGP	integer	greater than 0	3	Number of parameter groups.
NPRIOR	integer	0 or greater	3	Number of prior-information equations.
NOBSGP	integer	greater than 0	3	Number of observation groups.
MAXCOMPDIM	integer	optional; 0 or greater	1	Number of elements in compressed Jacobian matrix.
NTPLFLE	integer	greater than 0	3	Number of template files.
NINSFLE	integer	greater than 0	3	Number of instruction files.
PRECIS	text	“single” or “double”	1	Format for writing parameter values to model input files.
DPOINT	text	“point” or “nopoint”	1	Omit decimal point in parameter values if possible.
NUMCOM	integer	optional; greater than 0	1	Number of command lines used to run model.
JACFILE	integer	optional; 0 or 1	1	Indicates whether model provides external derivatives file.
MESSFILE	integer	optional; 0 or 1	1	Indicates whether PEST should write PEST-to-model message file.
RLAMBDA1	real	0 or greater	2	Initial Marquardt Lambda.
RLAMFAC	real	positive or negative, but not 0	2	Dictates Marquardt Lambda adjustment process.
PHIRATSUF	real	between 0 and 1	1	Fractional objective function sufficient for end of current iteration.
PHIREDLAM	real	between 0 and 1	1	Termination criterion for Marquardt Lambda search.
NUMLAM	integer	1 or greater	2	Maximum number of Marquardt Lambdas to test.
JACUPDATE	integer	optional; 0 or greater	2	Activation of Broyden’s Jacobian update procedure.
RELPARMAX	real	greater than 0	2	Parameter relative change limit.
FACPARMAX	real	greater than 1	2	Parameter factor change limit.
IBOUNDSTICK	integer	optional; 0 or greater	1	Instructs PEST not to compute derivatives for parameter at its bounds.
UPVECBEND	integer	optional; 0 or 1	1	Instructs PEST to bend parameter upgrade vector if parameter hits its bounds.
PHIREDSWH	real	between 0 and 1	1	Sets objective function change for introduction of central derivatives.
NOPTSWITCH	integer	optional; 1 or greater	1	Iteration before which PEST will not switch to central derivatives computation.
SPLITSWH	real	optional; 0 or greater	1	The factor by which the objective function rises to invoke split slope derivatives analysis until end of run.
DOAUI	text	“aui,” “auid,” or “noaui”	2	Instructs PEST to implement automatic user intervention.
DOSENREUSE	text	“senreuse” or “nosenreuse”	1	Instructs PEST to reuse parameter sensitivities
NOPTMAX	integer	-2, -1, 0, or any number greater than 0	3	Number of optimization iterations.

Variables in the “control data” section of the PEST Control File.—Continued

Variable	Type	Values	Usage	Description
PHIREDSTP	real	greater than 0	2	Relative objective function reduction triggering termination.
NPHISTP	integer	greater than 0	2	Number of successive iterations over which PHIREDSTP applies.
NPHINORED	integer	greater than 0	2	Number of iterations since last drop in objective function to trigger termination.
RELPARSTP	real	greater than 0	2	Maximum relative parameter change triggering termination.
NRELPAR	integer	greater than 0	2	Number of successive iterations over which RELPARSTP applies.
PHISTOPTHRESH	real	optional; 0 or greater	1	Objective function threshold triggering termination.
LASTRUN	integer	optional; 0 or 1	1	Instructs PEST to undertake (or not) final model run with best parameters.
PHIABANDON	real or text	optional	1	Objective function value at which to abandon optimization process or filename containing abandonment schedule.
ICOV	integer	0 or 1	1	Record covariance matrix in matrix file.
ICOR	integer	0 or 1	1	Record correlation-coefficient matrix in matrix file
IEIG	integer	0 or 1	1	Record eigenvectors in matrix file.
IRES	integer	0 or 1	1	Record resolution data.
JCOSAVEITN	text	“jcosaveitn” or “nojcosaveitn”	1	Write current Jacobian matrix to iteration-specific *.jco file at the end of every optimization iteration.
REISAVEITN	text	“reisaveitn” or “noreisaveitn”	1	Store best-fit residuals to iteration-specific residuals file at end of every optimization iteration.

Variables in the optional “automatic user intervention” section of the PEST Control File.

Variable	Type	Values	Usage	Description
MAXAUI	integer	0 or greater	1	Maximum number of AUI iterations per optimization iteration.
AUISTARTOPT	integer	1 or greater	1	Optimization iteration at which to begin AUI.
NOAUIPHIRAT	real	between 0 and 1	1	Relative objective function reduction threshold triggering AUI.
AUIRESTITN	integer	0 or greater, but not 1	1	AUI rest interval expressed in optimization iterations.
AUISENSRAT	real	greater than 1	1	Composite parameter sensitivity ratio triggering AUI.
AUIHOLDMAXCHG	integer	0 or 1	1	Instructs PEST to target parameters that change most when deciding which parameters to hold.
AUINUMFREE	integer	greater than 0	1	Cease AUI when only AUINUMFREE parameters are unheld.
AUIPHIRATSUF	real	between 0 and 1	1	Relative objective function improvement for termination of AUI.
AUIPHIRATACCEPT	real	between 0 and 1	1	Relative objective function reduction threshold for acceptance of AUI-calculated parameters.
NAUINOACCEPT	integer	greater than 0	1	Number of iterations since acceptance of parameter change for termination of AUI.

Variables in the optional “singular value decomposition” section of the PEST Control File.

Variable	Type	Values	Usage	Description
SVDMODE	integer	0 or 1	3	Activates truncated singular value decomposition for solution of inverse problem.
MAXSING	integer	greater than 0	3	Number of singular values at which truncation occurs.
EIGTHRESH	real	0 or greater, but less than 1	2	Eigenvalue ratio threshold for truncation.
EIGWRITE	integer	0 or 1	1	Determines content of SVD output file.

Variables in the optional “LSQR” section of the PEST Control File.

Variable	Type	Values	Usage	Description
LSQRMODE	integer	0 or 1	1	Activates LSQR solution of inverse problem.
LSQR_ATOL	real	0 or greater	1	LSQR algorithm <i>atol</i> variable.
LSQR_BTOL	real	0 or greater	1	LSQR algorithm <i>btol</i> variable.
LSQR_CONLIM	real	0 or greater	1	LSQR algorithm <i>conlim</i> variable.
LSQR_ITNLIM	integer	greater than 0	1	LSQR algorithm <i>itnlim</i> variable.
LSQR_WRITE	integer	0 or 1	1	Instructs PEST to write LSQR file.

Variables in the optional “SVD-Assist” section of the PEST Control File.

Variable	Type	Values	Usage	Description
BASEPESTFILE	text	a filename	3	Name of base PEST Control File.
BASEJACFILE	text	a filename	3	Name of base PEST Jacobian matrix file.
SVDA_MULBPA	integer	0 or 1	2	Instructs PEST to record multiple BPA files.
SVDA_SCALADJ	integer	-4 to 4	1	Sets type of parameter scaling undertaken in superparameter definition.
SVDA_EXTSUPER	integer	0, 1, 2, -2, 3	1	Sets means by which superparameters are calculated.
SVDA_SUPDERCALC	integer	0 or 1	1	Instructs PEST to compute superparameter sensitivities from base parameter sensitivities.

Variables in the optional “sensitivity reuse” section of the PEST Control File.

Variable	Type	Values	Usage	Description
SENRELTHRESH	real	0 to 1	1	Relative parameter sensitivity below which sensitivity reuse is activated for a parameter.
SENMAXREUSE	integer	integer other than 0	1	Maximum number of reused sensitivities per iteration.
SENALLCALCINT	integer	greater than 1	1	Iteration interval at which all sensitivities recalculated.
SENPREDEWEIGHT	real	any number	1	Weight to assign to prediction in computation of composite parameter sensitivities to determine sensitivity reuse.
SENPIEXCLUDE	test	“yes” or “no”	1	Include or exclude prior information when computing composite parameter sensitivities to determine sensitivity re-use.

Variables required for each parameter group in the “parameter groups” section of the PEST Control File.

Variable	Type	Values	Usage	Description
PARGPNME	text	12 characters or less	3	Parameter group name
INCTYP	text	“relative,” “absolute,” “rel_to_max”	2	Method by which parameter increments are calculated.
DERINC	real	greater than 0	2	Absolute or relative parameter increment.
DERINCLB	real	0 or greater	3	Absolute lower bound of relative parameter increment.
FORCEN	text	“switch,” “always_2,” “always_3”	1	Determines whether central derivatives calculation is done.
DERINCMUL	real	greater than 0	1	Derivative increment multiplier when undertaking central derivatives calculation.
DERMTHD	text	“parabolic,” “outside_pts,” “best_fit”	1	Method of central derivatives calculation.
SPLITTHRESH	real	greater than 0 (or 0 to deactivate)	1	Slope threshold for split slope analysis.
SPLITRELDIFF	real	greater than 0	1	Relative slope difference threshold for action.
SPLITACTION	text	text	1	“smaller,” “0” or “previous.”

Variables required for each parameter in the “parameter data” section of the PEST Control File.

Variable	Type	Values	Usage	Description
PARNME	text	12 characters or less	3	Parameter name.
PARTRANS	text	“log,” “none,” “fixed,” “tied”	3	Parameter transformation.
PARCHGLIM	text	“relative” or “factor”	3	Type of parameter change limit.
PARVAL1	real	any real number	3	Initial parameter value.
PARLBND	real	less than or equal to PARVAL1	3	Parameter lower bound.
PARUBND	real	greater than or equal to PARVAL1	3	Parameter upper bound.
PARGP	text	12 characters or less	3	Parameter group name.
SCALE	real	any number other than 0	2	Multiplication factor for parameter.
OFFSET	real	any number	2	Number to add to parameter.
DERCOM	integer	0 or greater	1	Model command line used in computing parameter increments.
PARTIED	text	12 characters or less	3	The name of the parameter to which another parameter is tied.

Variables required for each observation group in the “observation groups” section of the PEST Control File.

Variable	Type	Values	Usage	Description
OBGNME	text	12 characters or less	3	Observation group name.
GTARG	real	positive	1	Group-specific target measurement objective function.
COVFILE	text	a filename	2	Optional covariance matrix file associated with group.

Variables required for each observation in the “observation data” section of the PEST Control File.

Variable	Type	Values	Usage	Description
OBSNME	text	20 characters or less	3	Observation name.
OBSVAL	real	any number	3	Measured value of observation.
WEIGHT	real	0 or greater	3	Observation weight.
OBGNME	text	12 characters or less	3	Observation group to which observation assigned.

Variables in the optional “derivatives command line” section of the PEST Control File.

Variable	Type	Values	Usage	Description
DERCOMLINE	text	system command	1	Command to run model for derivatives calculation.
EXTDERFLE	text	a filename	1	Name of external derivatives file.

Variables in the “model command line” section of the PEST Control File.

Variable	Type	Values	Usage	Description
COMLINE	text	system command	3	Command to run model.

Variables in the “model input/output” section of the PEST Control File.

Variable	Type	Values	Usage	Description
TEMPFLE	text	a filename	3	Template file.
INFLE	text	a filename	3	Model input file.
INSFLE	text	a filename	3	Instruction file.
OUTFLE	text	a filename	3	Model output file.

Variables in the “prior information” section of the PEST Control File.

Variable	Type	Values	Usage	Description
PILBL	text	20 characters or less	3	Name of prior-information equation.
PIFAC	text	real number other than 0	3	Parameter value factor.
PARNME	text	12 characters or less	3	Parameter name.
PIVAL	real	any number	3	“Observed value” of prior information.
WEIGHT	real	0 or greater	3	Prior-information weight.
OBGNME	text	12 characters or less	3	Observation group name.

Variables in the optional “predictive analysis” section of the PEST Control File.

Variable	Type	Values	Usage	Description
NPREDMAXMIN	integer	-1 or 1	3	Maximize or minimize prediction.
PREDNOISE	integer	0 or 1	2	Instructs PEST to include predictive noise in prediction.
PD0	real	greater than 0	3	Target objective function.
PD1	real	greater than PD0	3	Acceptable objective function.
PD2	real	greater than PD1	3	Objective function at which Marquardt Lambda testing procedure is altered as prediction is maximized/minimized.
ABSPREDLAM	real	0 or greater	2	Absolute prediction change to terminate Marquardt Lambda testing.
RELPREDLAM	real	0 or greater	2	Relative prediction change to terminate Marquardt Lambda testing.
INITSCHFAC	real	greater than 0	2	Initial line search factor.
MULSCHFAC	real	greater than 1	2	Factor by which line search factors are increased along line.
NSEARCH	integer	greater than 0	2	Maximum number of model runs in line search.
ABSPREDSWH	real	0 or greater	1	Absolute prediction change at which to use central derivatives calculation.
RELPREDSWH	real	0 or greater	1	Relative prediction change at which to use central derivatives calculation.
NPREDNORED	integer	1 or greater	1	Iterations since prediction raised/lowered at which termination is triggered.
ABSPREDSTP	real	0 or greater	1	Absolute prediction change at which to trigger termination.
RELPRESTP	real	0 or greater	1	Relative prediction change at which to trigger termination.
NPREDSTP	integer	2 or greater	1	Number of iterations over which ABSPREDSTP and RELPRESTP apply.

Variables in the optional “regularization” section of the PEST Control File.

Variable	Type	Values	Usage	Description
PHIMLIM	real	greater than 0	3	Target measurement objective function.
PHIMACCEPT	real	greater than PHIMLIM	3	Acceptable measurement objective function.
FRACPHIM	real	optional; 0 or greater, but less than 1	2	Set target measurement objective function at this fraction of current measurement objective function.
MEMSAVE	text	“memsave” or “nomemsave”	1	Activate conservation of memory at cost of execution speed and quantity of model output.
WFINIT	real	greater than 0	1	Initial regularization weight factor.
WFMIN	real	greater than 0	1	Minimum regularization weight factor.
WFMAX	real	greater than WFMIN	1	Maximum regularization weight factor.
LINREG	text	“linreg” or “nonlinreg”	1	Informs PEST that all regularization constraints are linear.
REGCONTINUE	text	“continue” or “nocontinue”	2	Instructs PEST to continue minimizing regularization objective function even if measurement objective function less than PHIMLIM.
WFFAC	real	greater than 1	1	Regularization weight factor adjustment factor.
WFTOL	real	greater than 0	1	Convergence criterion for regularization weight factor.
IREGADJ	integer	0, 1, 2, 3, 4 or 5	2	Instructs PEST to perform interregularization group weight factor adjustment, or to compute new relative weights for regularization observations and prior-information equations.
NOPTREGADJ	integer	1 or greater	2	The optimization iteration interval for re-calculation of regularization weights if IREGADJ is 4 or 5.
REGWEIGHTRAT	real	absolute value of 1 or greater	2	The ratio of highest to lowest regularization weight; spread is logarithmic with null space projection if set negative.
REGSINGTHRESH	real	less than 1 and greater than 0	1	Singular value of $\mathbf{X}^t\mathbf{Q}\mathbf{X}$ (as factor of highest singular value) at which use of higher regularization weights begins if IREGADJ is set to 5.

Variables in the optional “Pareto” section of the PEST Control File.

Variable	Type	Values	Usage	Description
PARETO_OBSGROUP	text	12 characters or less	3	name of observation group whose weights are subject to multiplication by a variable weight factor
PARETO_WTFAC_START	real	zero or greater	1	initial weight factor for user-specified observation group
PARETO_WTFAC_FIN	real	greater than PARETO_WTFAC_START	2	final weight factor for user-specified observation group
NUM_WTFAC_INT	integer	greater than zero	2	number of weight factor increments to employ in traversing Pareto front
NUM_ITER_START	integer	zero or greater	1	number of optimisation iterations to employ when using initial weight factor
NUM_ITER_GEN	integer	greater than zero	1	number of optimization iterations to employ when using any weight factor other than PARETO_WTFAC_START or PARETO_WTFAC_FIN
NUM_ITER_FIN	integer	zero or greater	1	number of optimization iterations to employ when using final weight factor
ALT_TERM	integer	zero or one	3	set to one in order to activate PEST termination determined by value of a specified model output
OBS_TERM	text	20 characters or less	3	the name of an observation cited in the “observation data” section of the PEST control file whose value will be monitored for possible PEST run termination
ABOVE_OR_BELOW	text	“above” or “below”	3	determines whether the monitored model output must be above or below the threshold to precipitate run termination
OBS_THRESH	real	any number	3	value that monitored model output must exceed or undercut to precipitate model run termination
ITER_THRESH	integer	zero or greater	3	the number of optimization iterations for which the model output threshold must be exceeded or undercut to precipitate run termination
NOBS_REPORT	integer	0 or greater	3	number of model outputs whose values to report

Files used by PEST

The following tables list files that are read and written by PEST. Many of these possess the same filename base as the PEST Control File, this being designated as *case* in the tables below.

Files read by PEST.

File name	File type	Purpose
<i>case.pst</i>	PEST Control File	Provides problem dimensions, names of files for communication with a model, and values for all PEST control variables.
Arbitrary, commonly *.tpl	Template file	Provides means through which PEST writes current parameter values to a model input file.
Arbitrary, commonly *.ins	Instruction file	Provides means through which PEST reads outputs of interest from model output files.
<i>case.rm</i> f	Run management file	Provides Parallel PEST with information needed to communicate with slaves.
<i>case.hld</i>	Parameter hold file	Supplies details of manual intervention when holding individual parameters, or groups of parameters, at current values.

Files written by PEST.

File name	File type	Purpose
<i>case.rec</i>	Run record file	Contains details of progress of parameter-estimation process.
<i>case.cnd</i>	Condition number file	Contains continuous record of inverse-problem condition numbers.
<i>case.mtt</i>	Matrix file	Contains interim covariance, correlation coefficient, and eigenvector matrices.
<i>case.sen</i>	Parameter sensitivity file	Contains continuous record of composite parameter sensitivities.
<i>case.seo</i>	Observation sensitivity file	Records composite observation sensitivities.
<i>case.res</i>	Residuals file	Contains residuals and associated information recorded in tabular format.
<i>case.rei</i>	Interim residuals file	Contains residuals and associated information recorded in tabular format. This file is rewritten during every optimization iteration.
<i>pest.mmf</i>	Message file	Optionally recorded by PEST before every model run, contains the reason for carrying out the run and the parameter values that it employs.
<i>case.svd</i>	SVD file	Written only if PEST employs truncated SVD for solution of inverse problem; contains eigenvalues and, optionally, eigenvectors of XtQX matrix.
<i>case.lsqr</i>	LSQR file	Records information written by LSQR solver.
<i>case.jco</i>	Jacobian matrix file	Binary file containing Jacobian matrix pertaining to best parameters achieved so far.
<i>case.par</i>	Parameter value file	Records best parameter values achieved so far in parameter-estimation process.
<i>case.par.N</i>	Parameter value file	Records parameters achieved during each iteration when PEST is run in Pareto mode.
<i>case.pod</i>	Pareto objective function file	Record of objective functions encountered when traversing Pareto front.
<i>case.ppd</i>	Pareto parameter data file	Binary file recording parameters and corresponding model outputs encountered during traversal of the Pareto front.
<i>basecase.bpa</i>	Best parameter file	Contains best base parameters achieved so far; the filename base is the same as that of the base PEST Control File.
<i>case.rsd</i>	Resolution data file	Binary file written by PEST whenever it does any kind of regularized inversion. It contains data from which the resolution and “G” matrices can be computed by the RESPROC utility.
<i>case.rst, case.jac, case.jst</i>	Restart files	Contain information (written in binary form) that PEST uses in restarting a previously incomplete PEST run.
<i>case.rmr</i>	Run management record file	Lists history of communications between Parallel PEST and its slaves.
<i>pest.rdy, param.rdy, observ.rdy, pslave.fin, p###.###</i>	Semaphore files	Used by Parallel PEST to communicate with its slaves.

Appendix 2. 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.
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 error analysis; also computes parameter identifiability and relative error variance and uncertainty 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 or regularization.
PREDERR	Computes the error variance of a prediction whose sensitivities are available after a calibration exercise involving any form or 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 done by using truncated SVD; also finds the minimum of the predictive error variance curve.
PREDVAR1A	As for PREDVAR1, but undertakes SVD on $Q^T/2X$ rather than X^TQX .
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 effect on 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.

Nonlinear error analysis.—Continued

Program	Purpose
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 MonteCarlo 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
JATEST	Undertakes serial or parallel model runs to test the integrity of finite-difference-calculated derivatives.
POSTJATEST	JATEST 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.
MULJCOSEN	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 3. 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 model 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 parameter 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.
SMPCHK	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.”
MKMHOBBS	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.

Appendix 4. Background and Theory of PEST Uncertainty Analyses

The theory of Doherty (2010a, 2010b) that underpins PEST's exploration of model parameter and predictive uncertainty is now presented. Much of the theory is based on the assumption of a linear relation between model parameters and model outputs; where appropriate, however, it is extended to accommodate the more usual nonlinear relation between these two sets of quantities.

General

Let the matrix \mathbf{X} represent the action of a model on its parameters \mathbf{p} through which it calculates a set of outputs for which field measurements \mathbf{h} are available for use in the calibration process. Thus, \mathbf{h} represents the calibration dataset which can, without loss of generality, include direct measurements of system properties constituting the elements of \mathbf{p} . Let ε represent the noise associated with this dataset. Thus,

$$\mathbf{h} = \mathbf{X}\mathbf{p} + \varepsilon \quad (\text{A4.1})$$

For ease of notation it is assumed that parameter values of zero lead to model outputs of zero.

Let it be assumed that the model is parameterized to a level of "appropriate complexity." In the present context, this means that parameterization density is such that no errors to model outputs under predictive conditions are incurred through parameter simplification. Thus, all parameterization detail that is salient to model predictions is incorporated into the model parameterization scheme that is encapsulated in \mathbf{p} . Ideally, the same should apply to model outputs under calibration conditions. In practice, this assumption is often violated, and the "structural noise" so incurred is lumped with measurement noise ε . In the discussion that follows it will be assumed that the noise associated with \mathbf{h} , whatever its source, is characterized by a known covariance matrix $C(\varepsilon)$.

Because the model is parameterized to a level of "appropriate complexity" as defined above, it is unlikely that unique estimates for all elements of \mathbf{p} can be obtained on the basis of the calibration dataset \mathbf{h} . It is therefore assumed that the inverse problem of model calibration is ill posed.

Let the covariance matrix $C(\mathbf{p})$ denote the precalibration uncertainty associated with parameters \mathbf{p} . This matrix may have geostatistical origins. More often than not, however, it will be the outcome of subjective deliberations made by those with site expertise. Non-zero diagonal elements denote the fact that exact parameter values are unknown. However, boundedness of diagonal elements denotes the fact that some knowledge does exist of not-unlikely values for these properties. The presence of non-zero off-diagonal $C(\mathbf{p})$ elements may allow further representation of site knowledge, this reflecting the fact that parameters of similar type at the

same location, or of the same type at neighboring locations, exhibit joint variabilities (that is, are statistically correlated).

Let s (a scalar) represent a prediction required of the model. Let the sensitivity of s to all parameters be encapsulated in the vector \mathbf{y} . Thus, ignoring offsets once again,

$$\mathbf{s} = \mathbf{y}^T \mathbf{p} \quad (\text{A4.2})$$

If \mathbf{p} is uncertain, then so is s . Uncertainty of s can be computed from that of \mathbf{p} by using the standard equation for propagation of variance:

$$\sigma_s^2 = \mathbf{y}^T C(\mathbf{p}) \mathbf{y} \quad (\text{A4.3})$$

Equation (A4.3) expresses the precalibration variance (square of standard deviation) of the prediction s . For a nonlinear model, it could be computed through generating many realizations of \mathbf{p} on the basis of $C(\mathbf{p})$, running the model using each such realization to calculate a corresponding s , and then building a frequency histogram of s .

Parameter and Predictive Uncertainty

Use of equation A4.3 to calculate predictive uncertainty takes no account of the fact that parameter sets that do not allow the model to replicate historical measurements of system state should have their probabilities reduced in comparison with those that do. This notion is, of course, expressed formally by Bayes' equation:

$$P(\mathbf{p} | \mathbf{h}) = \frac{P(\mathbf{h} | \mathbf{p})P(\mathbf{p})}{\int P(\mathbf{h} | \mathbf{p})P(\mathbf{p})d\mathbf{p}} \quad (\text{A4.4})$$

Where a model is linear, and where model-parameter variability is characterized by a multi-Gaussian distribution, use of Bayes' equation to express calibration-constrained variability is accomplished in the manner now described.

Let \mathbf{x} be a vector of random variables with covariance matrix $C(\mathbf{x})$. Let it be partitioned into two subvectors \mathbf{x}_1 and \mathbf{x}_2 such that

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \quad (\text{A4.5})$$

Let $C(\mathbf{x})$, the covariance matrix of \mathbf{x} , be represented as

$$C(\mathbf{x}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \quad (\text{A4.6})$$

Now suppose that the elements of \mathbf{x}_2 become known. Then C'_{11} , the covariance matrix of \mathbf{x}_1 conditional on knowing \mathbf{x}_2 , is computed as follows (Koch, 1988):

$$C'_{11} = C_{11} - C_{12}C_{22}^{-1}C_{21} \quad (\text{A4.7})$$

To apply this concept in the modeling context, first equations and are combined to yield

$$\begin{bmatrix} s \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} \mathbf{y}^T & \mathbf{0} \\ \mathbf{X} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \varepsilon \end{bmatrix} \quad (\text{A4.8})$$

Then, using standard matrix relations for propagation of covariance,

$$\begin{aligned} \mathbf{C} \begin{pmatrix} s \\ \mathbf{h} \end{pmatrix} &= \begin{bmatrix} \mathbf{y}^T & \mathbf{0} \\ \mathbf{X} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{C}(\mathbf{p}) & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(\varepsilon) \end{bmatrix} \begin{bmatrix} \mathbf{y} & \mathbf{X}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{y}^T \mathbf{C}(\mathbf{p}) \mathbf{y} & \mathbf{y}^T \mathbf{C}(\mathbf{p}) \mathbf{X}^T \\ \mathbf{X} \mathbf{C}(\mathbf{p}) \mathbf{y} & \mathbf{X} \mathbf{C}(\mathbf{p}) \mathbf{X}^T + \mathbf{C}(\varepsilon) \end{bmatrix} \end{aligned} \quad (\text{A4.9})$$

Application of then leads to

$$\sigma_s^2 = \mathbf{y}^T \mathbf{C}(\mathbf{p}) \mathbf{y} - \mathbf{y}^T \mathbf{C}(\mathbf{p}) \mathbf{X}^T [\mathbf{X} \mathbf{C}(\mathbf{p}) \mathbf{X}^T + \mathbf{C}(\varepsilon)]^{-1} \mathbf{X} \mathbf{C}(\mathbf{p}) \mathbf{y} \quad (\text{A4.10})$$

Through use of appropriate matrix identities given by Athans and Schewpe (1965), this can also be written as

$$\sigma_s^2 = \mathbf{y}^T [\mathbf{X}^T \mathbf{C}^{-1}(\varepsilon) \mathbf{X} + \mathbf{C}^{-1}(\mathbf{p})]^{-1} \mathbf{y} \quad (\text{A4.11})$$

Equation (A4.10) expresses the postcalibration (that is, posterior) uncertainty of the prediction s whose sensitivity to parameters is encapsulated in the vector \mathbf{y} , conditional upon parameters being constrained by the necessity for the model to replicate within a margin of error set by $\mathbf{C}(\varepsilon)$ the field observations \mathbf{h} with model outputs whose sensitivities to parameters are encapsulated in the matrix \mathbf{X} . The following features of equation (A4.10) are noteworthy:

1. Neither the value of the prediction s , the values of parameters \mathbf{p} , nor the values of observations \mathbf{h} feature in these equations. Only sensitivities and covariance matrices are featured in them.
2. The first term of equation (A4.10) is the prior variance of the prediction s ; the second term is the reduction in this variance accrued through conditioning on field data \mathbf{h} .
3. If the vector \mathbf{y} possesses only a single non-zero element of 1.0, then equations (A4.10) and (A4.11) provide the posterior variance of the selected parameter.
4. Equations (A4.10) and (A4.11) can be modified to provide the posterior covariance matrix of model parameters; the \mathbf{y} vector in these equations is simply replaced by the m -dimensional identity matrix \mathbf{I} , where m is the number of elements composing \mathbf{p} .

5. By reconfiguring the elements of \mathbf{X} , equations (A4.10) and (A4.11) can be adapted to include direct measurements of system properties (these constituting individual elements of \mathbf{p}) in the observation dataset.
6. Because of the size of the matrix which requires inversion in each case, equation (A4.10) is better used where observation numbers are small and parameter numbers are large; equation (A4.11) is better used in the opposite situation.

Figure A4.1 shows a probability contour for two random scalar variables x_1 and x_2 encapsulated in a two-dimensional vector \mathbf{x} . Also shown is the marginal probability distribution of x_1 as well as the probability distribution of x_1 conditional upon acquisition of perfect knowledge of x_2 . The fact that the latter is (a) narrower than the former and (b) recentered correspond to the precalibration marginal probability distribution of x_1 is immediately evident from this figure.

Parameter and Predictive Error

Calibration

The above discussion of model predictive uncertainty has avoided use of the word “calibration.” It could be argued that the very concept of “calibration” should be avoided in environmental modeling, for in the face of the high degree of parameter and predictive uncertainty that attends most modeling exercises, what justification can there be for using a single parameter set to make predictions? In contrast, conceptually at least, a prediction should be made by using many different parameter sets, all of which are plausible (as assessed on the basis of the $\mathbf{C}(\mathbf{p})$ covariance matrix of innate parameter variability) and all of which provide an acceptable replication of historical system behavior through providing an adequate fit with the observation dataset \mathbf{h} (with “adequacy” assessed using the $\mathbf{C}(\varepsilon)$ covariance matrix of measurement noise).

Justification for use of a single parameter set can only be based on the premise that this particular parameter set can be shown to possess one or more desirable properties. Given the uncertainty associated with values of parameters inferred through the calibration process, and of predictions that depend on them, the property of minimum error variance would be a logical property to seek for the parameter set which is deemed to calibrate the model. It must be noted however, that “minimum error variance” does not mean “low error variance.” As Moore and Doherty (2005, 2006) point out, predictions made by means of a calibrated model can be considerably in error, despite the use of sophisticated mathematical regularization techniques that can indeed bestow on estimated parameter fields desirable properties such as these.

Notwithstanding its limitations, the “calibrated model” forms the basis of much environmental management. Indeed,

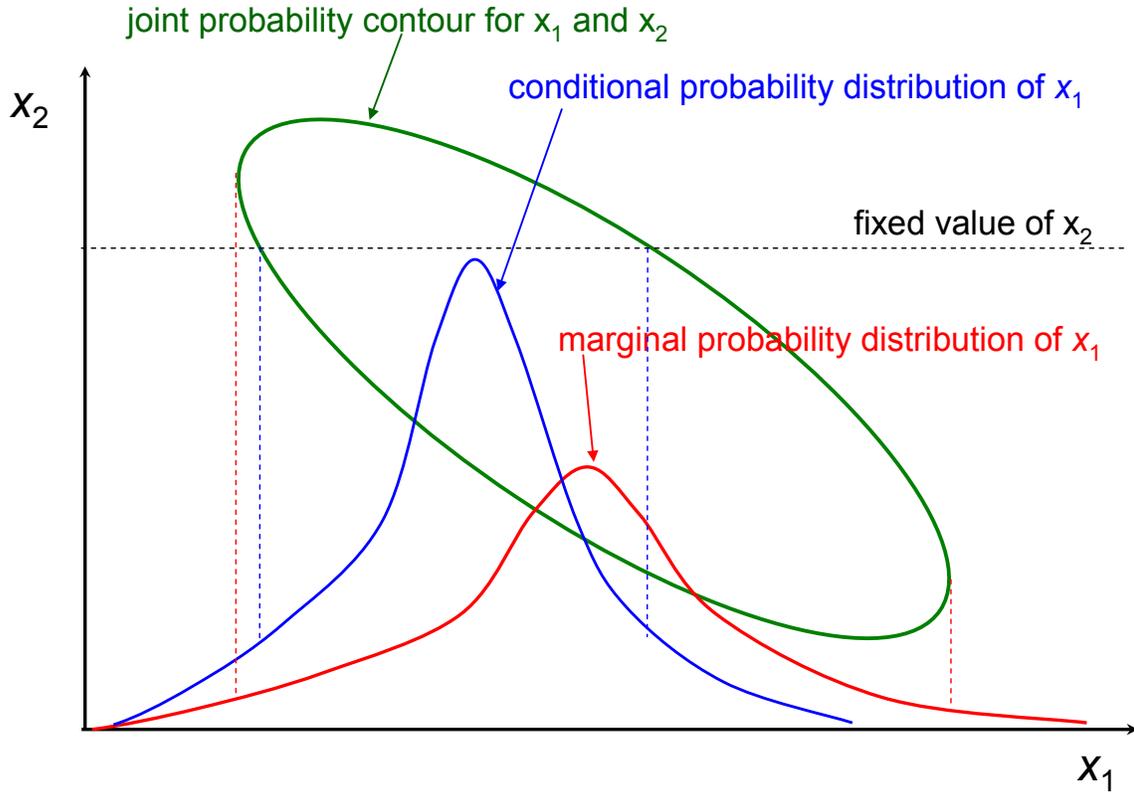


Figure A4.1. Marginal and conditional probability distributions of a random variable x_1 that is correlated with another random variable x_2 .

where large numbers of parameters must be conditioned by large numbers of observations, a strictly Bayesian approach to calculation of predictive uncertainty can be computationally impractical. In contrast, the attainment of parameter uniqueness by use of various mathematical regularization devices (for the example the “SVD-assist” methodology described by Tonkin and Doherty (2005) and supported by PEST), requires a relatively light numerical burden. Furthermore, parameter fields so obtained can be shown to possess properties that do not depart greatly from that of minimum error variance.

In model-use contexts such as this, the task of quantifying parameter and predictive uncertainty must be replaced by that of quantifying model-parameter and predictive propensity for error, or “error variance.” Inevitably, error variance will be larger than the variance of posterior parameter or predictive uncertainty, because its calculation must accommodate the fact that the attainment of parameter uniqueness through a process of regularized inversion may not necessarily result in a minimum-error-variance parameter field, even though it may strive to achieve this goal.

Suppose that the calibrated parameter field \mathbf{p} is computed from observations \mathbf{h} by use of the equation

$$\mathbf{p} = \mathbf{G}\mathbf{h} \quad (\text{A4.12})$$

The exact nature of \mathbf{G} depends on the regularization method used in the calibration process. For example, where solution of the (ill-posed) problem of model calibration is obtained through truncated singular value decomposition (SVD), \mathbf{G} takes the form

$$\mathbf{G} = \mathbf{V}_1^T \mathbf{S}_1^{-1} \mathbf{U}^T \mathbf{Q}^{1/2} \mathbf{h} \quad (\text{A4.13})$$

where \mathbf{Q} is a user-specified measurement weight matrix, and SVD of $\mathbf{Q}^{1/2}\mathbf{X}$ leads to

$$\mathbf{Q}^{1/2}\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (\text{A4.14})$$

\mathbf{U} and \mathbf{V} in equation (A4.14) are orthonormal matrices whose columns span the range space and domain of \mathbf{X} , respectively, and \mathbf{S} is a diagonal matrix of singular values, these being ranked from highest to lowest down its diagonal. If \mathbf{S} is partitioned into \mathbf{S}_1 and \mathbf{S}_2 such that zero and near-zero singular values of \mathbf{S} are assigned to \mathbf{S}_2 while the remainder are assigned to \mathbf{S}_1 , equation (A4.14) can be rewritten as

$$\mathbf{Q}^{1/2}\mathbf{X} = \mathbf{U}[\mathbf{S}_1 \quad \mathbf{S}_2] \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \quad (\text{A4.15})$$

this defining the \mathbf{V}_1 matrix that appears in (A4.13), the orthogonal columns of which span the “calibration solution space.”

Where Tikhonov regularization is used for solution of the ill-posed inverse problem of model calibration, \mathbf{G} becomes

$$\mathbf{G} = (\mathbf{X}^T \mathbf{Q} \mathbf{X} + \beta^2 \mathbf{T}^T \mathbf{Q}_r \mathbf{T})^{-1} \mathbf{X}^T \mathbf{Q} \mathbf{h} \quad (\text{A4.16})$$

where \mathbf{T} expresses Tikhonov constraints on parameter values, \mathbf{Q}_r is a weight matrix associated with these constraints, and β^2 is a variable that specifies the strength with which regularization constraints are applied.

Where parameter estimation takes place through the SVD-assist process implemented in PEST, \mathbf{G} takes the form

$$\mathbf{G} = \mathbf{V}_1 (\mathbf{Y}^T \mathbf{Q} \mathbf{Y} + \beta^2 \mathbf{V}_1^T \mathbf{T}^T \mathbf{Q}_r \mathbf{T} \mathbf{V}_1)^{-1} \mathbf{X}^T \mathbf{Q} \mathbf{h} \quad (\text{A4.17})$$

where \mathbf{V}_1 is obtained from SVD of the $\mathbf{Q}^{1/2} \mathbf{X}$ weighted model sensitivity matrix calculated on the basis of initial parameter estimates, and the \mathbf{Y} matrix encapsulates sensitivities of model outputs to superparameters estimated through the SVD-assisted parameter estimation process.

Where regularization is implemented through user pre-definition of a parsimonious parameter set \mathbf{q} , the \mathbf{G} matrix is computed as

$$\mathbf{G} = \mathbf{L} (\mathbf{Y}^T \mathbf{Q} \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{h} \quad (\text{A4.18})$$

where parameters \mathbf{p} used by the model at the cell or element level are computed from their lumped counterparts, \mathbf{q} , as

$$\mathbf{p} = \mathbf{L} \mathbf{q} \quad (\text{A4.19})$$

Where the parsimonious parameter set \mathbf{q} is composed of zones of piecewise constancy, \mathbf{L} is simply a “selection matrix” that assigns a parameter value to a model cell or element on the basis of the zone in which it lies. \mathbf{Y} in equation (A4.18) comprises sensitivities of model outputs to the lumped parameters used for the purpose of model calibration.

The Resolution Matrix

If equation (A4.1) is substituted into (A4.12), we obtain

$$\mathbf{p} = \mathbf{G} \mathbf{X} \mathbf{p} + \mathbf{G} \boldsymbol{\varepsilon} \quad (\text{A4.20})$$

If measurement noise is zero, this becomes

$$\mathbf{p} = \mathbf{G} \mathbf{X} \mathbf{p} + \mathbf{R} \mathbf{p} \quad (\text{A4.21})$$

where \mathbf{R} is the so-called resolution matrix. This matrix describes the relation between estimated parameters \mathbf{p} and their real-world counterparts \mathbf{p} . If the inverse problem is well

posed, \mathbf{R} is the identity matrix \mathbf{I} . However, where it is not, \mathbf{R} is a rank-deficient matrix that describes the simplification process through which parameter uniqueness has been attained. It should also be noted that obtaining a resolution matrix \mathbf{R} that is equivalent to the identity matrix \mathbf{I} , indicates *only* that the estimated parameter values \mathbf{p} are unique: it does *not* indicate that the parameters have been estimated accurately for, as will be discussed below, accurate estimation of parameters may be eroded through the presence of noise in the calibration dataset on which basis they are estimated.

Where calibration is effected through truncated SVD, \mathbf{R} assumes a particularly simple form, namely

$$\mathbf{R} = \mathbf{V}_1 \mathbf{V}_1^T \quad (\text{A4.22})$$

Equation states that the inferred parameter set \mathbf{p} is the vector projection of the actual (unknown) parameter set \mathbf{p} onto the subspace of parameter space defined by the columns of \mathbf{V}_1 . As has been stated, these span the so-called calibration solution space. This is illustrated in Figure A4.2.

The following features of the resolution matrix are of interest:

1. Each row of \mathbf{R} defines the averaging or integration process through which a single element p_i of \mathbf{p} is related to real-world parameters \mathbf{p} . In most real-world calibration contexts, \mathbf{R} is not strongly diagonally dominant—if it is diagonally dominant at all. Each p_i is thus partly a reflection of its corresponding p_i (that is, the i^{th} element of \mathbf{p}), and partly a reflection of many other real-world properties (that is, elements of \mathbf{p}), these possibly including properties of very different types. This is an unavoidable consequence of the quest for uniqueness that is implicit in the notion of model calibration.
2. The magnitude of the i^{th} diagonal element of \mathbf{R} is the direction cosine between a vector pointing in the direction of the i^{th} parameter and its projection into the calibration solution space. This is equivalent to the “parameter identifiability” of Doherty and Hunt (2009a,b).
3. As equations (A4.18) and (A4.21) imply, even where regularization is implemented manually through pre-calibration parameter lumping, a resolution matrix must inevitably accompany the quest for uniqueness that underpins the process of model calibration. In most cases however, the resolution matrix achieved through this form of regularization will be suboptimal, in that its diagonal elements will possess lower values, and its off-diagonal elements higher values, than those achieved through use of mathematical regularization. See Moore and Doherty (2005) for a more complete discussion of this issue.

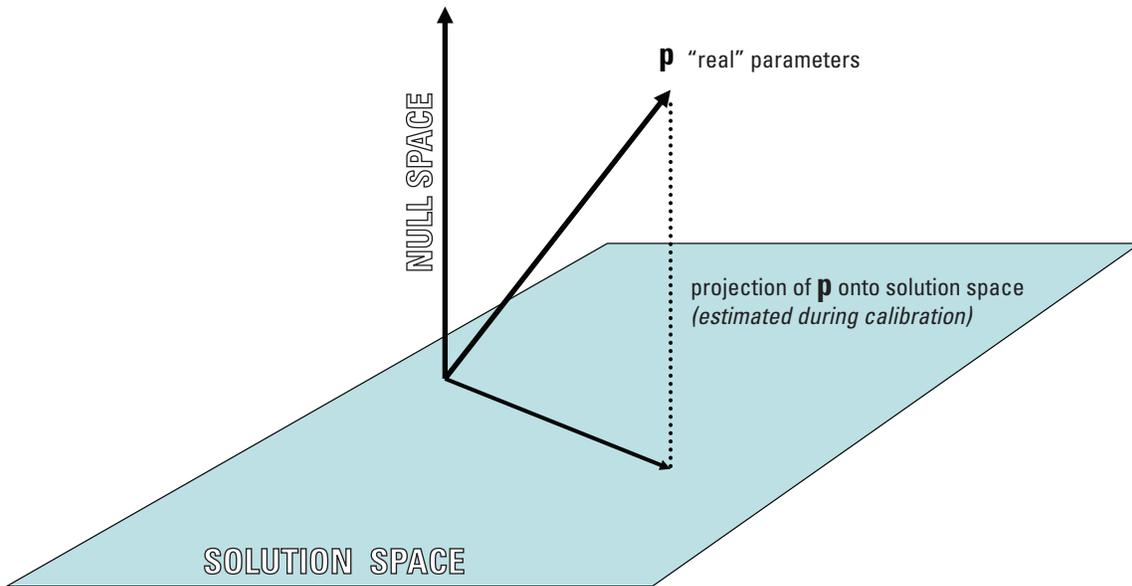


Figure A4.2. Relation between real-world and estimated parameters where model calibration is achieved through truncated SVD.

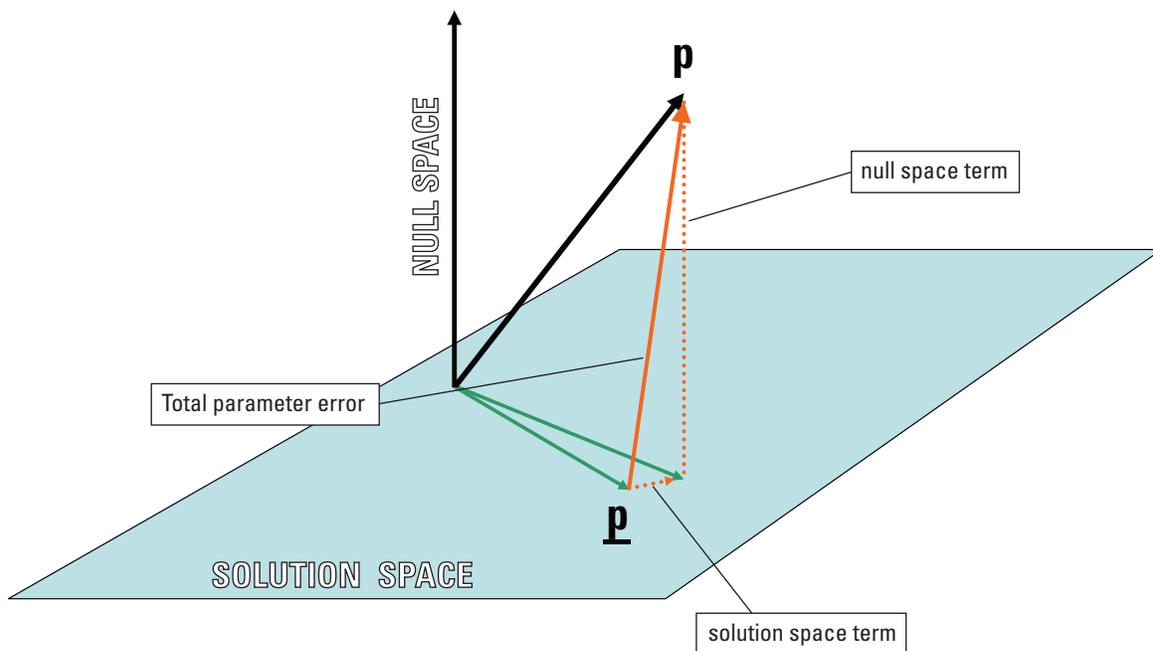


Figure A4.3. Components of postcalibration parameter error.

Parameter Error

From equation the error in parameters that are estimated through the calibration process is formulated as

$$\underline{\mathbf{p}} - \mathbf{p} = -(\mathbf{I} - \mathbf{GX})\mathbf{p} + \mathbf{G}\varepsilon \quad (\text{A4.23})$$

That is,

$$\underline{\mathbf{p}} - \mathbf{p} = -(\mathbf{I} - \mathbf{R})\mathbf{p} + \mathbf{G}\varepsilon \quad (\text{A4.24})$$

It is apparent from equations (A4.23) and (A4.24) that there are two contributors to postcalibration parameter error. The first term on the right side of equation (A4.24) is the so-called “null-space term,” whereas the second is the “solution-space term.” Where parameter estimation is implemented by use of truncated SVD, these two contributions to total parameter error are orthogonal, as illustrated in figure A4.3. As Moore and Doherty (2005) explain, the first term arises from the necessity to simplify when seeking a unique parameter field that is deemed to calibrate a model. The second term expresses the contribution of measurement noise to parameter error.

As the true parameter set \mathbf{p} of equation (A4.24) is not known, the actual parameter error ($\underline{\mathbf{p}} - \mathbf{p}$) is not known. However the penchant for parameter error can be expressed in probabilistic terms. From (A4.24), using standard relations for propagation of variance, it follows that

$$\mathbf{C}(\underline{\mathbf{p}} - \mathbf{p}) = (\mathbf{I} - \mathbf{R})\mathbf{C}(\mathbf{p})(\mathbf{I} - \mathbf{R})^T + \mathbf{GC}(\varepsilon)\mathbf{G}^T \quad (\text{A4.25})$$

where $\mathbf{C}(\underline{\mathbf{p}} - \mathbf{p})$ is the covariance matrix of parameter error. Prior to calibration, the second term of (A4.25) is zero; \mathbf{R} is notionally $\mathbf{0}$ under these circumstances as well. Thus, $\mathbf{C}(\underline{\mathbf{p}} - \mathbf{p})$ becomes $\mathbf{C}(\mathbf{p})$, as it should.

Predictive Error

Equation (A4.2) expresses, in a linear sense, the relation between a prediction s and parameters \mathbf{p} . A prediction made by a calibrated model, however, is calculated as

$$s = \mathbf{y}^T \mathbf{p} \quad (\text{A4.26})$$

with the result that predictive error is inherited from parameter error. Predictive error is calculated as

$$\underline{s} - s = \mathbf{y}^T (\underline{\mathbf{p}} - \mathbf{p}) \quad (\text{A4.27})$$

Predictive error variance is thus calculable as

$$\sigma_{\underline{s}-s}^2 = \mathbf{y}^T \mathbf{C}(\underline{\mathbf{p}} - \mathbf{p}) \mathbf{y} \quad (\text{A4.28})$$

That is,

$$\sigma_{\underline{s}-s}^2 = \mathbf{y}^T (\mathbf{I} - \mathbf{R})\mathbf{C}(\mathbf{p})(\mathbf{I} - \mathbf{R})^T \mathbf{y} + \mathbf{y}^T \mathbf{GC}(\varepsilon)\mathbf{G}^T \mathbf{y} \quad (\text{A4.29})$$

As for parameter error, the first term of equation (A4.29) quantifies the contribution made to predictive error by the need to simplify in order to achieve parameter uniqueness. The second term expresses the contribution made to predictive error by the fact that, in calculating values for the simplified parameter field estimated through the calibration process, the parameter estimates are based on a dataset that is contaminated by measurement noise.

For illustrative purposes, suppose that

$$\mathbf{C}(\mathbf{p}) = \sigma_p^2 \mathbf{I} \quad (\text{A4.30})$$

and that

$$\mathbf{C}(\varepsilon) = \sigma_\varepsilon^2 \mathbf{I} \quad (\text{A4.31})$$

Let it be further assumed that solution for the calibrated parameter field $\underline{\mathbf{p}}$ takes place by means of truncated SVD, so that \mathbf{G} of equation (A4.12) is given by (A4.13). For simplicity, let it be further assumed that $\mathbf{Q} = \mathbf{I}$ and hence that the weights assigned to observations are all 1.0. Equation (A4.29) then becomes

$$\sigma_{\underline{s}-s}^2 = \sigma_p^2 \mathbf{y}^T \mathbf{V}_2 \mathbf{V}_2^T \mathbf{y} + \sigma_\varepsilon^2 \mathbf{y}^T \mathbf{V}_1 \mathbf{S}^{-2} \mathbf{V}_1^T \mathbf{y} \quad (\text{A4.29})$$

where \mathbf{V}_2 contains orthonormal vectors that span the calibration null space while \mathbf{V}_1 contains orthonormal vectors that span the orthogonal complement of this, the calibration solution space. Separation of parameter space into these two subspaces depends on where partitioning of \mathbf{S} takes place to form \mathbf{S}_1 and \mathbf{S}_2 , the latter containing zero and near-zero singular values of $\mathbf{Q}^{1/2}\mathbf{X}$. If there is no truncation at all (and hence no calibration), the last term of equation (A4.32) is zero, so that precalibration predictive error variance becomes

$$\sigma_{\underline{s}-s}^2 = \sigma_p^2 \mathbf{y}^T \mathbf{y} \quad (\text{A4.33})$$

which is the same as precalibration predictive uncertainty. More generally, it becomes

$$\sigma_{\underline{s}-s}^2 = \sigma_p^2 \mathbf{y}^T \mathbf{C}(\mathbf{p}) \mathbf{y} \quad (\text{A4.34})$$

which is the same as equation (A4.3).

Suppose that truncation is now used to implement model calibration. Suppose further that truncation proceeds gradually to include an increasing number of columns in \mathbf{V}_1 and, consequently, a decreasing number of columns in \mathbf{V}_2 . As this process goes forward, the first term of A4.29 falls monotonically

while the second term rises monotonically. Eventually, as singular values within S_1 become very small through attempting to achieve a solution space with too many dimensions (a phenomenon that is often referred to as “overfitting”), the second term dominates the first and eventually rises to infinity as singular values fall to zero. Overall, as the truncation point increases from zero singular values to a high number of singular values, the sum of these two terms (that is the total error variance $\sigma_{\underline{s}-s}^2$) falls and then rises again. Ideally, for a particular prediction of interest, truncation should take place where the error variance of that prediction is minimized. Hopefully, this minimized predictive error variance should approximate the conditional predictive uncertainty variance computed through equations (A4.10) and (A4.11); however, this is not a foregone conclusion, and the result will depend on the nature of the \mathbf{G} matrix through which calibrated model parameters are computed from field measurements.

Regularization-Induced Structural Noise

From equation (A4.1):

$$\mathbf{h} = \mathbf{X}\mathbf{p} + \varepsilon = \mathbf{X}\mathbf{p} - \mathbf{X}\mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{R}\mathbf{p} + \varepsilon$$

so that

$$\mathbf{h} = \mathbf{X}\mathbf{R}\mathbf{p} + \mathbf{X}(\mathbf{I} - \mathbf{R})\mathbf{p} + \varepsilon \quad (\text{A4.35})$$

As discussed above, it is $\mathbf{R}\mathbf{p}$ that is estimated through the calibration process. Though comparing (A4.35) with (A4.1), it is apparent that the “structural noise” η induced through undertaking the parameter simplification encapsulated in \mathbf{R} that is necessary for obtaining a unique solution to the inverse problem of model calibration is

$$\eta = \mathbf{X}(\mathbf{I} - \mathbf{R})\mathbf{p} \quad (\text{A4.36})$$

Its stochastic character can then be described with the covariance matrix

$$\mathbf{C}(\eta) = \mathbf{X}(\mathbf{I} - \mathbf{R})\mathbf{C}(\mathbf{p})(\mathbf{I} - \mathbf{R})^T \mathbf{X}^T \quad (\text{A4.37})$$

Predictive Uncertainty Analysis— Underdetermined Systems

Parameter and Predictive Error

With \mathbf{R} equal to \mathbf{I} , as occurs when the inverse problem of model calibration is solved by means of overdetermined parameter estimation, equation (A4.25) becomes

$$\mathbf{C}(\underline{\mathbf{p}} - \mathbf{p})\mathbf{G}\mathbf{C}(\varepsilon)\mathbf{G}^T \quad (\text{A4.38})$$

Where calibration is implemented through minimization of a least-squares objective function,

$$\mathbf{G} = (\mathbf{X}^T \mathbf{Q} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Q} \quad (\text{A4.39})$$

so that (A4.38) becomes

$$\mathbf{C}(\underline{\mathbf{p}} - \mathbf{p}) = (\mathbf{X}^T \mathbf{Q} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Q} \mathbf{C}(\varepsilon) \mathbf{Q}^T \mathbf{X} (\mathbf{X}^T \mathbf{Q} \mathbf{X})^{-1} \quad (\text{A4.40})$$

If \mathbf{Q} is chosen such that

$$\mathbf{Q} = \sigma_r^2 \mathbf{C}^{-1}(\varepsilon) \quad (\text{A4.41})$$

equation (a4.40) becomes

$$\mathbf{C}(\underline{\mathbf{p}} - \mathbf{p}) = \sigma_r^2 (\mathbf{X}^T \mathbf{Q} \mathbf{X})^{-1} \quad (\text{A4.42})$$

The reference variance, σ_r^2 , is generally estimated through the calibration process as

$$\sigma_r^2 = \frac{\Phi}{(n - m)} \quad (\text{A4.43})$$

where Φ is the minimized objective function, calculated as

$$\Phi = (\mathbf{h} - \mathbf{X}\mathbf{p})^T \mathbf{Q} (\mathbf{h} - \mathbf{X}\mathbf{p}) \quad (\text{A4.43})$$

and n is the number of elements in \mathbf{h} while m is the number of elements in \mathbf{p} .

Where parameter-averaging-induced structural noise is included in $\mathbf{C}(\varepsilon)$, and where \mathbf{Q} is calculated from the resultant $\mathbf{C}(\varepsilon)$ through (A4.41) (see Cooley; 2004; and Cooley and Christensen, 2006), the parameters estimated by using the \mathbf{G} defined in (A4.39) on the basis of the \mathbf{Q} defined in (A4.41) are the best linear unbiased estimates of the “true” \mathbf{p} obtained through user-specified areal averaging of heterogeneous model domain properties; see Cooley (2004) for full details.

Once the covariance matrix of parameter error has been computed by using (A4.42), the error variance of any prediction can be computed using equation (A4.28). Use of equation (A4.28), however, neglects “predictive noise” induced by parameter lumping/simplification. Cooley (2004) and Cooley and Christensen (2006) show how to accommodate this term.

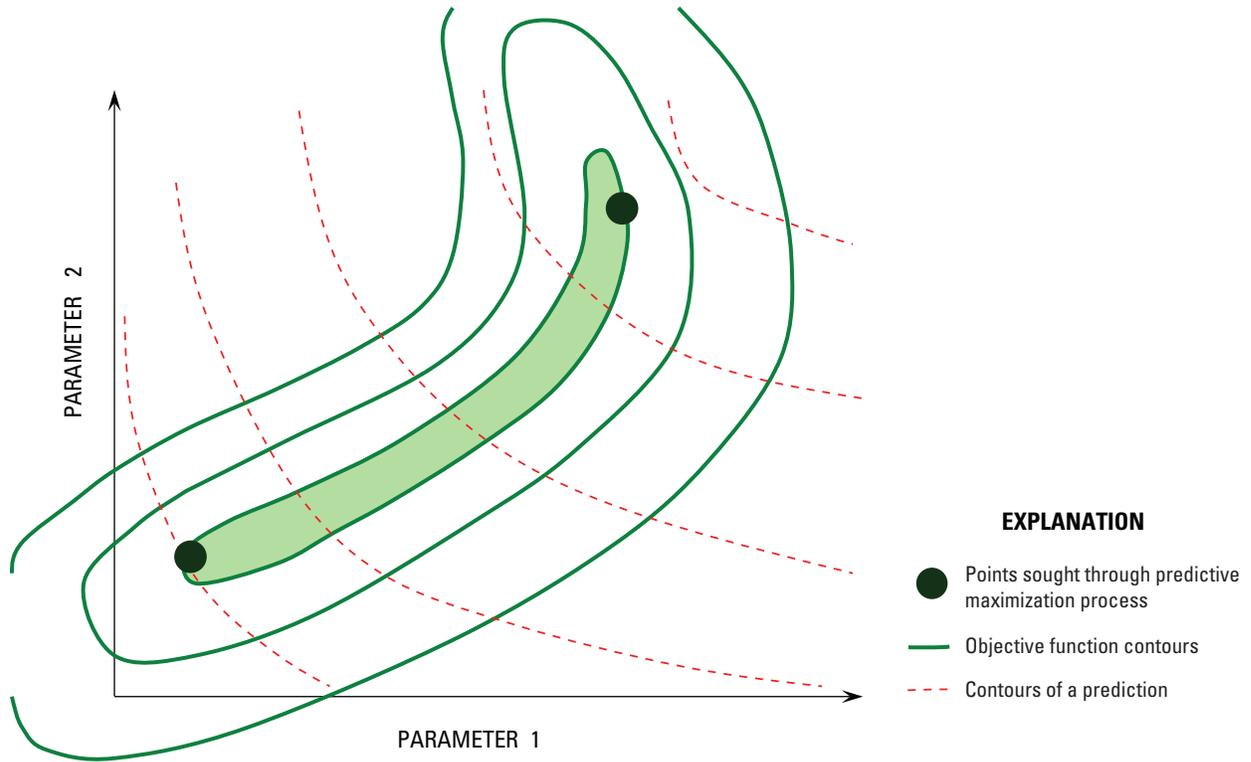


Figure A4.4. Schematic description of calibration-constrained predictive maximization/minimization.

Nonlinear Analysis

Vecchia and Cooley (1987) and Christensen and Cooley (1999) show how postcalibration predictive uncertainty analysis can be posed as a constrained maximization/minimization problem in which a prediction is maximized or minimized subject to the constraint that the objective function rises no higher than a user-specified value; this value is normally specified to be slightly higher than the minimum value of the objective function achieved during a previous overdetermined model calibration exercise. Cooley (2004) and Cooley and Christensen (2006) extend this methodology by including parameter-simplification-induced structural noise in the analysis.

The principle that underlies this method is illustrated in figure A4.6 for a two-parameter system. In this figure, the axes depict optimized parameters that correspond to the minimum of the objective function. The solid lines depict objective function contours; the value of each contour defines the objective function for which parameters become unlikely at a certain confidence level. Each contour thus defines the constraint to which parameters are subject as a prediction of interest is maximized or minimized in order to define its postcalibration variability at the same level of confidence. The dashed contour lines of figure A4.4 depict the dependence of a prediction on the two parameters. The constrained maximization/minimization process through

which the postcalibration uncertainty of this prediction is explored attempts to find the two points marked by circles on the constraining objective-function contour. These points define parameter sets for which the prediction of interest is as high or as low as it can be, while maintaining respect for the constraints imposed by the calibration process.

Let Φ_0 define the objective function value corresponding to a certain postcalibration parameter confidence level. Vecchia and Cooley (1987) show that parameters \mathbf{p} corresponding to the two points depicted in figure A4.6 for which the prediction of interest is maximized/minimized subject to the constraint that the objective function rises no higher than Φ_0 can be calculated as

$$\mathbf{p} = (\mathbf{X}^T \mathbf{Q} \mathbf{X})^{-1} \left\{ \mathbf{X}^T \mathbf{Q} \mathbf{h} - \frac{\mathbf{y}}{2\lambda} \right\} \quad (\text{A4.45})$$

where

$$\left(\frac{1}{2\lambda} \right)^2 = \pm \frac{\Phi_0 - \mathbf{h}^T \mathbf{Q} \mathbf{h} + \mathbf{h}^T \mathbf{Q} \mathbf{X} (\mathbf{X}^T \mathbf{Q} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Q} \mathbf{h}}{\mathbf{y}^T (\mathbf{X}^T \mathbf{Q} \mathbf{X})^{-1} \mathbf{y}} \quad (\text{A4.46})$$

In these equations, \mathbf{y} is the sensitivity of the prediction to parameters used by the model; maximization or minimization is undertaken through selection of the “+” or “-” sign in

equation (A4.46). Equations (A4.45) and (A4.46) are solved for maximized/minimized parameters by using an iterative procedure very similar to that used for solution of the parameter set \mathbf{p} corresponding to the objective function minimum Φ_{\min} .

The value chosen for Φ_0 depends on whether simultaneous or individual confidence limits are sought. For a linear model, use of the latter provides the same predictive confidence intervals as those obtained through use of equations (A4.42) and (A4.28); simultaneous confidence intervals, however, are somewhat broader. See Christensen and Cooley (1999) for further details. Values of Φ_0 for $1-\alpha$ simultaneous and individual confidence intervals respectively are given by

$$\Phi_0 = \Phi_{\min} \left[\frac{m}{n-m} F_{\alpha} (m, n-m) + 1 \right] \quad (\text{A4.47})$$

and

$$\Phi_0 = \Phi_{\min} \left[\frac{t_{\alpha/2}^2 (n-m)}{(n-m)} + 1 \right] \quad (\text{A4.48})$$

where $F(m, n-m)$ depicts the F distribution with $(m, n-m)$ degrees of freedom, $t(m-n)$ signifies a t distribution with $(n-m)$ degrees of freedom, n is the number of observations, and m is the number of parameters.

Where account is taken of predictive noise, equation (A4.46) becomes

$$\left(\frac{1}{2\lambda} \right)^2 = \pm \frac{\Phi_0 - \mathbf{h}^T \mathbf{Q} \mathbf{h} + \mathbf{h}^T \mathbf{Q} \mathbf{X} (\mathbf{X}^T \mathbf{Q} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Q} \mathbf{h}}{\mathbf{y}^T (\mathbf{X}^T \mathbf{Q} \mathbf{X})^{-1} \mathbf{y} + w_e^{-2}} \quad (\text{A4.49})$$

where w_e is the weight assigned to predictive noise and e is the predictive error itself. The actual predictive error obtained through the predictive maximization/minimization process is calculated as

$$e = \frac{-w_e^{-2}}{2\lambda} \quad (\text{A4.50})$$

predictive noise is taken into account, equation (A4.47) becomes

$$\Phi_0 = \Phi_{\min} \left[\frac{m+1}{n-m} F_{\alpha} (m+1, n-m) + 1 \right] \quad (\text{A4.51})$$

while equation (A4.48) remains unchanged. See Doherty (2010b) for further details.

Predictive Uncertainty Analysis—Highly Parameterized Systems

As discussed in the body of this document, the PEST suite provides two methods through which nonlinear uncertainty analysis can be performed in the highly parameterized context. The theory which underpins these methods is now briefly described.

Constrained Maximization/Minimization

This method is described in Tonkin and others (2007). Once a model has been calibrated, predictive maximization/minimization takes place in a similar manner to that discussed above based on equations (A4.45), (A4.46), and (A4.49). However, the maximization/minimization problem is formulated in terms of parameter *differences* from their calibrated values rather than in terms of actual parameter values, while the effect that these parameter differences have on model outputs is also expressed in terms of *differences* between those outputs and those that correspond to the calibrated parameter set. As predictive maximization/minimization takes place, differences between current parameter values and their calibrated counterparts are projected onto the calibration null space, thus forming the vector $\mathbf{p} - \mathbf{p}$, where \mathbf{p} constitutes the calibrated parameter set. From these differences a “null-space objective function” is formed, this being computed as

$$\Phi_n = (\mathbf{p} - \mathbf{p})^T C_n^{-1} (\mathbf{p} - \mathbf{p}) \quad (\text{A4.52})$$

In this equation, $C_n(\mathbf{p} - \mathbf{p})$, the covariance matrix of null-space projected parameter departures from their calibrated values, is computed as

$$C_n(\mathbf{p} - \mathbf{p}) = (\mathbf{I} - \mathbf{R}) C(\mathbf{p}) (\mathbf{I} - \mathbf{R})^T \quad (\text{A4.53})$$

where, as usual, $C(\mathbf{p})$ is supplied by the modeler, and the resolution matrix \mathbf{R} is computed through the antecedent highly parameterized inversion process. At the same time, a solution-space objective function is calculated as

$$\Phi_s = (\mathbf{h} - \mathbf{X} \mathbf{p})^T \mathbf{Q} (\mathbf{h} - \mathbf{X} \mathbf{p}) \quad (\text{A4.54})$$

where

$$\mathbf{h} = \mathbf{X} \mathbf{p} \quad (\text{A4.55})$$

When $\mathbf{p} = \underline{\mathbf{p}}$ (that is, when the model employs the calibrated parameter field) the total objective function, calculated as

$$\Phi_t = \Phi_n + \Phi_s \quad (\text{A4.56})$$

is zero. A prediction of interest is maximized or minimized subject to the constraint that Φ_t rises no higher than a pre-defined value Φ_0 . On the assumption that \mathbf{p} and ε are normally distributed, the confidence level associated with a chosen value of Φ_0 is the same as that of the normal variate whose value is the square root of this. For example, setting Φ_0 to 9.0 allows definition of the 99.7-percent two-sided confidence interval through implementation of predictive maximization followed by predictive minimization.

Null-Space Monte Carlo

As described by Tonkin and Doherty (2009) for the constrained maximization/minimization process described above, prior to implementation of null-space Monte Carlo analysis, it is assumed that a model has been calibrated, this resulting in computation of a parameter field $\underline{\mathbf{p}}$. Stochastic parameter fields are then generated by using an appropriate $\mathbf{C}(\underline{\mathbf{p}})$ matrix. In each case the calibrated parameter field is then subtracted from the stochastically generated parameter field. The difference between the two (that is, $\mathbf{p} - \underline{\mathbf{p}}$) is then projected onto the calibration null space; this ensures that this difference is in accordance with a $\mathbf{C}(\underline{\mathbf{p}} - \underline{\mathbf{p}})$ covariance calculated as

$$\mathbf{C}(\underline{\mathbf{p}} - \underline{\mathbf{p}}) = \mathbf{V}_2 \mathbf{V}_2^T \mathbf{C}(\underline{\mathbf{p}}) \mathbf{V}_2 \mathbf{V}_2^T \quad (\text{A4.57})$$

where the vectors composing the columns of the \mathbf{V}_2 matrix span the calibration null space, this being obtained through truncated SVD of $\mathbf{Q}^{1/2} \mathbf{X}$ as described by equation (A4.15). For regularized inversion based on truncated SVD,

$$(\mathbf{I} - \mathbf{R}) = \mathbf{V}_2 \mathbf{V}_2^T = \mathbf{I} - \mathbf{V}_1 \mathbf{V}_1^T \quad (\text{A4.58})$$

so that equation (A4.57) describes parameter variability arising from the first term of equation (A4.25).

Next, the solution-space component of the stochastically generated parameter field is replaced by $\underline{\mathbf{p}}$, the parameter field arising from the previous regularized-inversion exercise. In most circumstances, this replacement will result in a slightly decalibrated model, this being a consequence of model nonlinearity and the fact that the cutoff between calibration solution and null spaces is not sharp and does not equate with the location of zero-valued singular values.

Recalibration of the model is then effected by adjusting only solution-space-parameter eigencomponents, these being the coefficients of vectors composing the columns of the \mathbf{V}_1 matrix of equation A4.14. Adjustment of these solution-space parameters continues until the objective function falls below

a user-specified level. Although this level could be chosen on the basis of formulas such as equation A4.43, in practice the selection of an appropriate objective function threshold is often somewhat subjective because of the following factors:

1. Selection of a higher objective-function threshold can reduce the computational effort required to achieve it.
2. In most modeling circumstances, “measurement noise,” as encapsulated in $\mathbf{C}(\varepsilon)$, is dominated by structural noise of unknown stochastic character: as such, a theoretically correct objective function threshold is difficult, if not impossible, to determine.
3. In most modeling circumstances, parameter error is dominated by the null-space term of equation A4.23; hence, errors incurred in computation of the solution-space contribution to overall parameter and predictive error degrade its assessment to only a small degree.

References Cited

- Athans, A., and Schweppe, F.C., 1965, Gradient matrices and matrix calculations: Lexington Mass., Massachusetts Institute of Technology, Lincoln Laboratory, Technical Note 1965-53, 34 p.
- Christensen, S., and Cooley, R.L., 1999, Evaluation of prediction intervals for expressing uncertainties in groundwater flow model predictions: *Water Resources Research*, v. 35, no. 9, p. 2627–2639.
- Cooley, R.L., 2004, A theory for modeling ground-water flow in heterogeneous media: U.S. Geological Survey Professional Paper 1679, 220 p.
- Cooley, R.L., and Christensen, S., 2006, Bias and uncertainty in regression-calibrated models of groundwater flow in heterogeneous media: *Advances in Water Resources*, v. 29, no. 5, p. 639–656.
- Doherty, J., 2010a, PEST, Model-independent parameter estimation—User manual (5th ed., with slight additions): Brisbane, Australia, Watermark Numerical Computing.
- Doherty, J., 2010b, Addendum to the PEST manual: Brisbane, Australia, Watermark Numerical Computing.
- Doherty, J., and Hunt, R.J., 2009a, Two statistics for evaluating parameter identifiability and error reduction: *Journal of Hydrology*, v. 366, no. 1–4, p. 119–127, doi: 10.1016/j.jhydrol.2008.12.018.
- Doherty, J., and Hunt, R.J., 2009b, Response to comment on “Two statistics for evaluating parameter identifiability and error reduction”: *Journal of Hydrology*, v. 380, no. 3–4, p. 489–496, doi: 10.1016/j.jhydrol.2009.10.012

- Koch, K.R., 1988, Parameter estimation and hypothesis testing in linear models: Berlin, Springer-Verlag, 378 p.
- Moore, C., and Doherty, J., 2005, The role of the calibration process in reducing model predictive error: *Water Resources Research*, v. 41, no. 5, W05050.
- Moore, C., and Doherty, J., 2006, The cost of uniqueness in groundwater model calibration: *Advances in Water Resources*, v. 29, no. 4, p. 605–623.
- Tonkin, M.J., and Doherty, J., 2005, A hybrid regularized inversion methodology for highly parameterized environmental models: *Water Resources Research*, v. 41, no. 10, W10412, doi:10.1029/2005WR003995.
- Tonkin, M.J., and Doherty, J., 2009, Calibration-constrained Monte-Carlo analysis of highly parameterized models using subspace techniques: *Water Resources Research*, v. 45, no. 12, W00B10, doi:10.1029/2007WR006678.
- Tonkin, M.J., Doherty, J., and Moore, C., 2007, Efficient nonlinear predictive error variance for highly parameterized models. *Water Resources Research*, v. 43, no. 7, W07429, doi:10.1029/2006WR005348.
- Vecchia, A.V., and Cooley, R.L., 1987, Simultaneous confidence and prediction intervals for nonlinear regression models with application to a groundwater flow model: *Water Resources Research*, v. 23, no. 7, p. 1237–1250.

