

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

# **Approaches to Highly Parameterized Inversion: A Guide to Using PEST for Groundwater-Model Calibration**



Scientific Investigations Report 2010–5169



# **Approaches to Highly Parameterized Inversion: A Guide to Using PEST for Groundwater-Model Calibration**

By John E. Doherty and Randall J. Hunt

Scientific Investigations Report 2010–5169

**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: 2010

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., and Hunt, R.J., 2010, Approaches to highly parameterized inversion—A guide to using PEST for groundwater-model calibration: U.S. Geological Survey Scientific Investigations Report 2010–5169, 59 p.

# Contents

Abstract .....	1
Introduction .....	1
Purpose and Scope .....	2
Regularized Inversion .....	2
Mathematical Regularization.....	3
Tikhonov Regularization .....	3
Subspace Regularization.....	4
A “Singularly Valuable Decomposition” —Benefits of SVD for Model Calibration.....	4
SVD-Assist and the Hybrid of Tikhonov and SVD .....	6
Before Running PEST: Model Parameterization .....	6
Parameterization Philosophy.....	6
Spatial Parameterization .....	7
Zones of Piecewise Constancy .....	7
Pilot Points .....	7
Conceptual Overview of Pilot-Point Use (excerpted from Doherty, Fienen, and Hunt, 2010).....	8
Pilot Points in Conjunction with Zones.....	9
Other Parameter Types.....	9
Initial Parameter Values .....	10
Tikhonov Regularization Strategies .....	10
Preferred-Value Regularization.....	11
Preferred-Difference Regularization.....	11
Before Running PEST: Observations Used in Inversion Process .....	12
Formulation of an Objective Function.....	12
Objective-Function Components.....	12
Data of Different Types .....	13
Special Considerations for Concentration Data .....	13
Temporal Head Differences.....	13
Vertical Interaquifer Head Differences .....	13
Joint Steady-State/Transient Calibration.....	14
Declustering .....	14
Digital Filtering .....	14
“Intuitive” and Other Soft Data.....	14
Temporal and Spatial Interpolation.....	14
A Note on Model Validation .....	15
Before Running PEST: Preparing the Run Files.....	15
Control Data.....	15
Memory Conservation.....	15
The Marquardt Lambda.....	15
Broyden’s Jacobian Update.....	16
Number of Optimization Iterations.....	16
Automatic User Intervention.....	16

Solution Mechanism .....	16
Parameter Groups .....	17
Derivatives Calculation .....	17
Regularization Considerations.....	17
Parameter Data .....	18
Initial Parameter Values .....	18
Parameter Transformation .....	18
Scale and Offset.....	18
Observation Groups.....	18
Objective-Function Contributions .....	18
Weight Matrices .....	18
Regularization Groups.....	18
Observation Data .....	19
Model Command Line .....	19
Model Input/Output .....	19
Prior Information.....	19
Regularization.....	20
Target Measurement Objective Function.....	20
Regularization Objective Function .....	20
Interregularization Weights Adjustment.....	20
SVD-Assist.....	21
Number of Superparameters.....	21
Tikhonov Regularization .....	21
Precalibration Sensitivities .....	21
SVDAPREP Responses .....	22
Enhancing SVD-Assist by Also Including SVD .....	22
Parallel Processing .....	23
Marquardt Lambda.....	23
Model-Run Repetition .....	23
Just Before Running PEST: Checking .....	24
Running PEST .....	24
Stopping PEST.....	24
Restarting PEST .....	24
Pausing PEST .....	24
Monitoring PEST Performance.....	25
Classical Calibration of Sparsely Parameterized Models.....	25
Well-Posedness.....	25
Identifying Troublesome Parameters .....	25
Accommodating Troublesome Parameters.....	25
Identifying Bad Derivatives.....	26
Accommodating Bad Derivatives.....	26
Phi Gradient Zero .....	28
Regularized Inversion of Highly Parameterized Problems .....	28
Signs of Regularization Failure .....	28
Rectifying Problems in Regularized Inversion: Tikhonov Regularization .....	29

Rectifying Problems in Regularized Inversion: Subspace Regularization .....	29
Final Model Run .....	29
Automatic Final Model Run .....	29
Manual Final Model Run.....	29
Evaluation of Results .....	30
Level of Fit.....	30
Poorer Than Expected Fit.....	30
Too Good a Fit .....	31
Parameter Values .....	31
Unreasonable Parameter Values.....	31
Reduction of the Level of Fit .....	31
Local Aberrations in Parameter Fields.....	31
Multiple Parameter Fields .....	32
Calibration as Hypothesis Testing .....	32
Other Issues.....	33
Calibration Postprocessing.....	33
Evaluating Derivatives Used in the Calibration Process.....	33
Integrity of Finite-Difference Derivatives .....	33
Manipulation of Jacobian Matrix Files .....	33
Global Optimizers.....	33
Summary of Guidelines.....	34
References.....	35
Appendix 1. Basic PEST Input.....	41
Structure of the PEST Control File .....	41
Files used by PEST.....	48
Appendix 2. PEST Utilities.....	50
Appendix 3. Groundwater Data Utilities .....	54
Reference Cited .....	57
Appendix 4. Singular Value Decomposition Theory .....	58
References Cited .....	58

## Figures

1. An example of JACTEST-calculated model outputs (y-axis) resulting from small sequential parameter perturbation (x-axis). ....27



# Approaches to Highly Parameterized Inversion: A Guide to Using PEST for Groundwater-Model Calibration

By John E. Doherty<sup>1,2</sup> and Randall J. Hunt<sup>3</sup>

## Abstract

Highly parameterized groundwater models can create calibration difficulties. Regularized inversion—the combined use of large numbers of parameters with mathematical approaches for stable parameter estimation—is becoming a common approach to address these difficulties and enhance the transfer of information contained in field measurements to parameters used to model that system. Though commonly used in other industries, regularized inversion is somewhat imperfectly understood in the groundwater field. There is concern that this unfamiliarity can lead to underuse, and misuse, of the methodology. This document is constructed to facilitate the appropriate use of regularized inversion for calibrating highly parameterized groundwater models. The presentation is directed at an intermediate- to advanced-level modeler, and it focuses on the PEST software suite—a frequently used tool for highly parameterized model calibration and one that is widely supported by commercial graphical user interfaces. A brief overview of the regularized inversion approach is provided, and techniques for mathematical regularization offered by PEST are outlined, including Tikhonov, subspace, and hybrid schemes. Guidelines for applying regularized inversion techniques are presented after a logical progression of steps for building suitable PEST input. The discussion starts with use of pilot points as a parameterization device and processing/grouping observations to form multicomponent objective functions. A description of potential parameter solution methodologies and resources available through the PEST software and its supporting utility programs follows. Directing the parameter-estimation process through PEST control variables is then discussed, including guidance for monitoring and optimizing the performance of PEST. Comprehensive listings of PEST control variables, and of the roles performed by PEST utility support programs, are presented in the appendixes.

<sup>1</sup> Watermark Numerical Computing, Brisbane, Australia

<sup>2</sup> National Centre for Groundwater Research and Training, Flinders University, Adelaide SA, Australia.

<sup>3</sup> U.S. Geological Survey.

## Introduction

Highly parameterized groundwater models are characterized by having more parameters than can be estimated uniquely on the basis of a given calibration dataset—in some cases having more parameters than observations in the calibration dataset. Such models, which almost always lack a unique parameter-estimation solution, are commonly referred to as “ill posed.” Ill-posed models require an approach to model calibration and uncertainty different from the traditional methods typically used with well-posed models. Hunt and others (2007) define traditional model calibration as those for which subjective precalibration parameter reduction is used to obtain a tractable (well-posed or overdetermined) parameter-estimation problem. Regularized inversion has been suggested as one means of obtaining a unique calibration from the fundamentally nonunique, highly parameterized family of calibrated models. “Regularization” simply refers to approaches that make ill-posed problems mathematically tractable; “inversion” refers to the automated parameter-estimation operations that use measurements of the system state to constrain model input parameters (Hunt and others, 2007).

Regularized inversion problems are most commonly addressed by use of the Parameter ESTimation code PEST (Doherty, 2010a). PEST is an open-source, public-domain software suite that allows model-independent parameter estimation and parameter/predictive-uncertainty analysis. It is accompanied by two supplementary open-source and public-domain software suites for calibration of groundwater and surface-water models (Doherty, 2007, 2008). This software, together with extensive documentation, can be downloaded from <http://www.pesthomepage.org/>.

The optimal number of parameters needed for a representative model is often not clear, and in many ways model complexity is ultimately determined by the objectives of that model (Hunt and Zheng, 1999; Hunt and others, 2007). However, many benefits can be gained from taking a highly parameterized approach to calibration of that model regardless of the level of complexity that is selected (Doherty, 2003; Hunt and others, 2007; Doherty and Hunt, 2010).

The foremost benefit is that regularized inversion interjects greater parameter flexibility into all stages of calibration than that offered by precalibration parameter-simplification (or *oversimplification*) calibration strategies such as a priori sparse zonation. This flexibility helps the modeler extract information contained in a calibration dataset during the calibration process, whereas regularization algorithms allow the modeler to control the degree of parameter variation. Indeed, high numbers of parameters used in calibration can collapse to relatively homogeneous optimal parameter fields (as described by, for example, Muffels, 2008; and Fienen, Hunt, and others, 2009). Thus, the twin ideals of parsimony—simple as possible but not simpler—are fully met. Finally, the regularized-inversion approach is advantageous because it makes available sophisticated estimates of parameter and predictive uncertainty (Moore and Doherty, 2005, 2006).

## Purpose and Scope

This document is intended for intermediate to advanced-level groundwater modelers who are familiar with classical parameter-estimation approaches, such as those discussed by Hill and Tiedeman (2007), as well as the implementation of classical overdetermined parameter-estimation approaches in PEST, as described by Doherty (2010a). The purpose of this document is to provide

1. a brief overview of highly parameterized inversion and the mathematical regularization that is necessary to achieve a tractable solution to the ill-posed problem of calibrating highly parameterized models,
2. an intermediate and advanced description of PEST usage in implementing highly parameterized parameter estimation for groundwater-model calibration, and
3. an overview of the roles played by PEST utility support programs in implementing pilot-point-based parameterization and in calibration preprocessing and postprocessing.

The PEST software suite has already been 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 operations; intermediate users can read through a logical progression of typical issues faced during calibration of highly parameterized groundwater models—a progression framed in terms of PEST input and output a modeler is likely to encounter. Appendixes are included to facilitate the relation of PEST variables and concepts used in the report body to the broader PEST framework, terminology, and definitions of Doherty (2010a,b). Descriptions provided

herein are necessarily brief, and mathematical foundations are referenced rather than derived, in order to focus on appropriate application rather than already published theoretical underpinnings of regularized inversion. Thus, this document is intended to be an application-focused companion to the full scope of PEST described in the detailed explanations of Doherty (2010a,b) and theory cited by references included therein.

PEST and its utility software are supported by several popular commercial graphical-user interfaces. Through these interfaces, many of the methodologies discussed in this document are readily deployed, with many implementation details concealed from the user. However, some knowledge of the mathematical and philosophical underpinnings of regularized inversion is helpful for successful and efficient use of this methodology, even when its application is made relatively simple. For modelers who are comfortable working at the command-line level, calibration preprocessing and postprocessing functionality provided by PEST utility support software offers customized model-calibration capabilities not available through commercial modeling user interfaces. Some utility programs provided with PEST are mentioned in the body of this document; all are listed in the appendixes. Calibration tools can be further augmented with purpose-specific utility programs written by the modelers themselves. This document is also confined to model calibration. A companion document discusses parameter and predictive uncertainty analysis in the highly parameterized context (Doherty, Hunt, and Tonkin, 2010).

## Regularized Inversion

Similar to classical parameter estimation of overdetermined problems, regularized-inversion approaches are grounded on principles of least-squares minimization (for example, Draper and Smith, 1998), where a best fit is defined by the minimization of the weighted squared difference between measured and simulated observations. In both methods, the computer code automatically varies model inputs, runs the model(s), and evaluates model output to determine the quality of fit. In both methods, parameters estimated through the calibration process are accompanied by error, which consists primarily of two sources. The first is that a model can simulate only a simplified form of a complex natural world; for example, the simulated aquifer has a hydraulic-conductivity distribution that is a simplified version of the complex actual distribution of spatially varied hydraulic properties. The second is that observations used to constrain estimates of parameter values contain measurement noise. When the model is used to simulate future system behavior, its predictions contain inherent artifacts that result from both types of error (Moore and Doherty, 2005, 2006; Hunt and Doherty, 2006).

Appropriate simplification of real-world complexity is an indispensable part of model conceptualization and calibration. Traditionally, parameter simplification is done before

calibration by delineating what is hoped to be a parameter set that is simplified enough for its values to be uniquely estimable, but hopefully not so oversimplified that it fails to capture salient aspects of the system. As the parameter-estimation process progresses, and after it has reached completion, much of the subsequent analysis is evaluating whether that precalibration simplification may have been too strong or too weak. If either is found to be true, reparameterization of the model must take place, and the calibration process must then be repeated.

Calibration as implemented through regularized inversion is founded on a different approach. Parameter simplification necessary for achieving a unique solution to the inverse problem of model calibration is done through mathematical means, as part of the calibration process itself. Thus, the modeler is not required to define a simplified parameter set at the start of the calibration process. Indeed, as discussed by Hunt and others (2007), the modeler ideally provides parameterization detail in the calibration process that is commensurate with hydraulic-property heterogeneity expected within the model domain, or at least at a level of detail to which predictions of interest may be sensitive. Although including flexibility gained from use of many parameters, the properly formulated regularized-inversion process yields an optimal parameter field that expresses only as much complexity as can be supported by the calibration dataset. Heterogeneity expressed in this optimal parameter field arises at locations, and in manners, that are warranted by the data. The information content of the calibration dataset does not therefore need to be placed into simplification schemes or zones that are predefined by the modeler. It can be shown that model predictions made on the basis of such parameter fields approach minimum error variance in the statistical sense (Moore and Doherty, 2005). Furthermore, the potential for wrongness in these predictions can be properly quantified. Inasmuch as a prediction may depend on parameterization detail that cannot be represented in a calibrated model, that detail (which is suppressed during the calibration process) can be formally addressed when the uncertainty of the prediction is explored.

## Mathematical Regularization

Using more parameters than can be constrained uniquely by observations results in formulation of an ill-posed inverse problem; numerical solution of that problem must include the use of one or more regularization mechanisms to stabilize the numerical solution process and identify a unique solution. Although regularization in the broadest sense can include the use of mechanisms to translate subsets of node-by-node grid parameterization to the parameter-estimation process (such as pilot points), mathematical regularization as discussed here is reduced into two broad categories: Tikhonov regularization and subspace regularization. A third hybrid category—a combination of these two—is also available and discussed herein.

## Tikhonov Regularization

Integrity of the calibration process requires that intuitive knowledge and geological expertise be incorporated into the calibration process, together with information of historical measurements of system state. Tikhonov regularization (Tikhonov, 1963a, 1963b; Tikhonov and Arsenin, 1977) provides a vehicle for formally incorporating this “soft” information into the calibration process by augmenting the measurement objective function with a regularization objective function that captures the parameters’ deviation from the user-specified preferred condition (see Doherty, 2003, p. 171–173). Minimization of this combined objective function is a means for determining a unique solution to the inverse problem that balances the model’s fit to the observed data and adherence to the soft knowledge of the system. The regularization objective function supplements the calibration observed dataset through a suite of special pseudo-observations, each pertaining to a preferred condition for one or more parameters employed by the model. Collectively, these constitute a suite of fallback values for parameters, or for relations between parameters, in the event little or no information pertaining to those parameters resides in the observations of the calibration dataset. Where the information content of a calibration dataset is insufficient for unique estimation of certain parameters, or combinations of parameters, the fallback value prevails.

Apart from providing a default condition for parameters and relations between parameters, Tikhonov regularization also constrains the manner in which heterogeneity supported by the calibration dataset emerges in the estimated parameter field. If properly formulated, Tikhonov constraints can promote and govern geologically realistic departures from background parameter fields. Without such constraints, fields that result in a good fit with the calibration dataset may nonetheless be considered suboptimal because of geologically unrealistic parameter values. Indeed, much of the art of formulating appropriate Tikhonov constraints for a particular parameter-estimation problem is directed at obtaining a good fit with geologically reasonable parameter values.

As implemented in PEST, Tikhonov regularization is controlled by a variable that prevents the achievement of model-to-measurement fit that is too good given the level of noise associated with the calibration dataset. As is further discussed later, the modeler supplies a “target measurement objective function” that sets a limit on how good a fit the calibration process is allowed to achieve. PEST adjusts the strength with which Tikhonov constraints are applied as the lever through which respect for this target is maintained, relaxing Tikhonov constraints to achieve a tighter fit, and strengthening these constraints if a looser fit is required. This topic is covered in depth by Doherty (2003) and Fienen, Muffles, and Hunt (2009).

Although use of Tikhonov regularization normally results in parameter fields that are geologically realistic, numerical instability of the calibration process can occur as the fit between model outcomes and field measurements is explored.

This instability arises from mathematical difficulties associated with strong application of default geological conditions in areas where data are limited simultaneous with weaker application of those conditions where data are plentiful. This problem can be partly overcome through use of subspace-enhanced Tikhonov regularization capabilities (Doherty, Fienen, and Hunt, 2010) provided with PEST, through which differential weighting is applied to individual Tikhonov constraints where calibration information is unavailable for the parameters to which the constraints apply. In addition, subspace regularization can also be used in conjunction with Tikhonov regularization to maintain numerical stability.

## Subspace Regularization

In contrast to Tikhonov regularization, which adds information to the calibration process in order to achieve numerical stability, subspace methods achieve numerical stability through subtracting parameters, and/or parameter combinations, from the calibration process (Aster and others, 2005). As a result of the subtraction, the calibration process is no longer required to estimate either individual parameters or combinations of correlated parameters that are inestimable on the basis of the calibration dataset. These combinations are automatically determined through singular value decomposition (SVD) of the weighted Jacobian matrix (see Moore and Doherty, 2005; Tonkin and Doherty, 2005; and Appendix 4).

The Jacobian matrix consists of the sensitivities of all specified model outputs to all adjustable model parameters; each column of the Jacobian matrix contains the sensitivity of all model outputs for a single adjustable parameter. Individual parameters, or combinations of parameters, that are deemed to be estimable on the basis of the calibration dataset constitute the “calibration solution space.” Those parameters/parameter combinations that are deemed to be inestimable (these constituting the “calibration null space”) retain their initial values. It is thus important that initial parameter values be reasonable given what is known about the pre-calibration

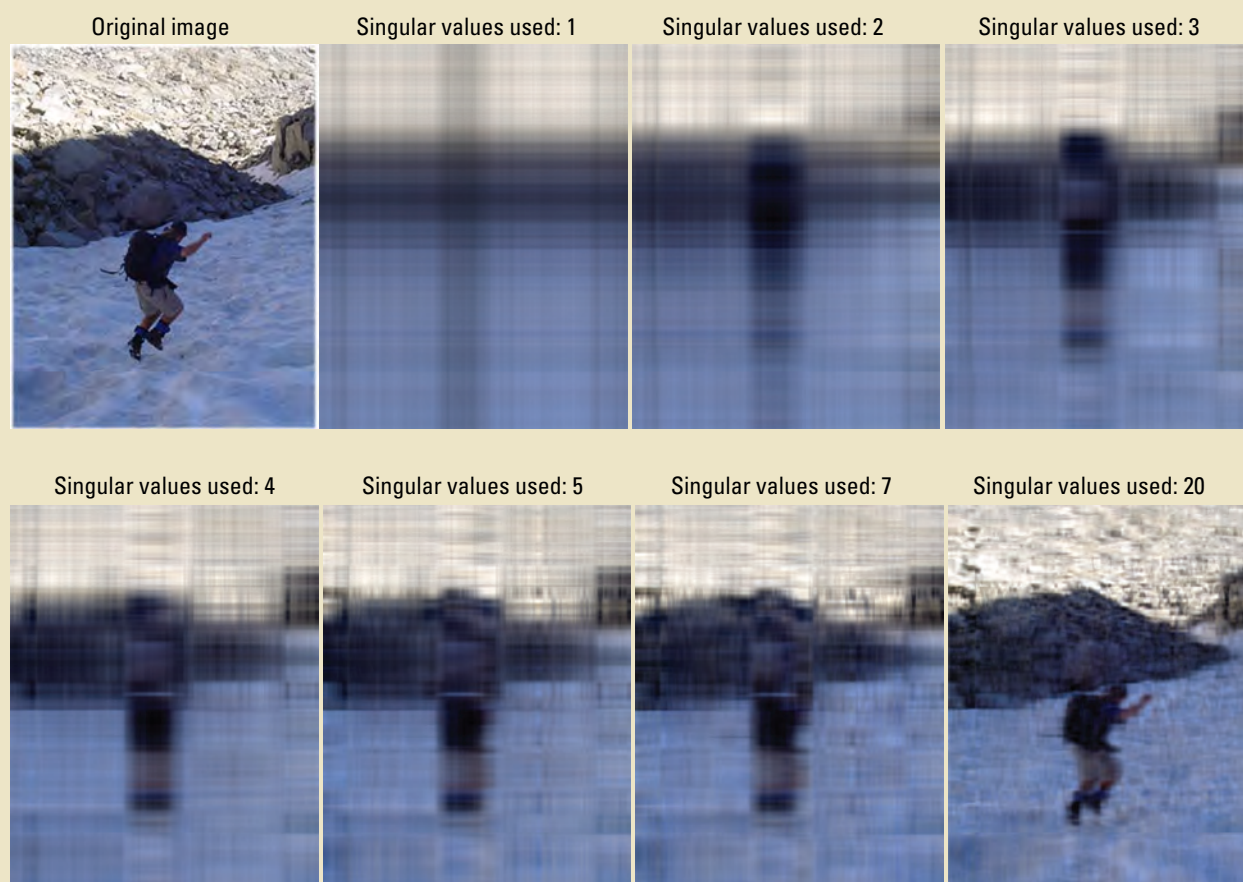
## A “Singularly Valuable Decomposition”<sup>1</sup> — Benefits of SVD for Model Calibration

When large numbers of parameters are added to a model, some can be expected to be insensitive and others highly correlated with other parameters. As a result, even though a parameter *may* be estimable (therefore worth including in the calibration process), it doesn’t mean that it actually *is* estimable. What is needed is an intelligent calibration tool—one that detects what can and cannot be inferred from the calibration dataset and then estimates what it can and leaves out what it can’t—all automatically, without user intervention. Singular value decomposition (SVD) is such a tool.

SVD is a way of processing matrices into a smaller set of linear approximations that represent the underlying structure of the matrix; thus, it is called a “subspace” method. It is used widely in other industries for such tasks as image processing (fig. B1–1)—for example, as commonly experienced in the sequentially updated resolution of images displayed by the software Google Earth. In this use, SVD allows a user to get useful information from an even somewhat blurry image earlier rather than waiting for the entire image to download. In the context of groundwater-model calibration, rather than solving the problem in a space defined by the total number of base parameters and observations in the model, SVD describes a reduced representation of parameter and observation space that shows their relationship to each other in the context of a specific calibration dataset. On the basis of the weighted Jacobian matrix, SVD defines a reduced set of axes for parameter and observation space where certain combinations of observations are uniquely informative of certain combinations of parameters; these combinations define the new reduced set of axes that span each space. Similar to the image-processing example, the subspace represents a more blurry view of the subsurface than exists in the natural world, but a view that defines where combinations of informative observations run out, thereby leaving combinations of parameters inestimable.

What are inestimable combinations? In some cases they are individual parameters that are insensitive and thus have no effect on model-generated counterparts to observations. In other cases they are parameter groups that can be varied in combination with each other in ratios that allow their effects on these model outputs to offset and cancel each other out. The calibration dataset cannot inform these parameters individually. Collectively, these two define the “calibration null space.” SVD-based parameter estimation reformulates the inverse problem by truncating the singular values carried in the parameter estimation processes so that estimation of these parameter combinations is not even attempted. Their initial values (either individually or as combinations) are then retained. Parameter combinations that are not confounded by insensitivity or correlation comprise the complimentary “calibration solution space.” Because this space is defined specifically by using parameter combinations that are estimable, solution of the inverse problem is unique and unconditionally stable.

<sup>1</sup> Kalman, Dan, 1996, A singularly valuable decomposition—The SVD of a matrix: College Mathematics Journal, v. 27, no. 1, p. 2–23.



**Figure B1-1.** An example of singular value decomposition of a photograph image. When the matrix is perfectly known (as is the case with pixels in the original image), the highest resolution for a given number of singular values can be shown visually. For reference, the image with 20 singular values represents less than 10 percent of information contained in the original image in the upper left, yet it contains enough information that the subject matter can be easily identified. Although information of groundwater systems is not as well known as that in this image, a similar concept applies: if too few singular values are selected, a needlessly coarse and blurry representation of the groundwater system results. Moreover, when the information content of the calibration dataset is increased, a larger number of data-supported singular values can be included, resulting in a sharper “picture” of the groundwater system. (Image from and SVD processing by Michael N. Fienen, USGS.)

geological setting. This requirement is in contrast to traditional parameter estimation where estimated parameter values are theoretically independent of their initial values.

Demarcation between the calibration solution and null spaces is achieved through singular value truncation specified by the user. Parameter combinations (also called eigenvectors) associated with singular values that are greater than a certain threshold are assigned to the calibration solution space, whereas those associated with singular values that are smaller than this threshold are assigned to the calibration null space. In practical applications, if too many combinations of parameters are estimated, the problem will still be numerically unstable; if too few parameters are estimated, the model fit may be unnecessarily poor, and predictive error may be larger than that for an optimally parameterized model. Moreover, the SVD approach can be ruthless in its search for a best fit, resulting in calibrated parameter fields that often lack the aesthetic appeal of those produced by Tikhonov regularization.

PEST offers the option of using the LSQR algorithm of Paige and Saunders (1982) as a replacement for SVD. LSQR allows faster definition of solution-space eigenvectors than does SVD in calibration problems with large numbers of parameters (greater than about 2,500); however, their definition is not quite as exact as that provided by SVD (Muffels, 2008), and information needed for uncertainty quantification, such as the resolution matrix, is not calculated.

## SVD-Assist and the Hybrid of Tikhonov and SVD

Although SVD can provide stable and unique model calibration, it does not alleviate the high computational burden incurred by the use of many parameters; that is, the full Jacobian matrix (calculated by perturbing each base parameter) is still calculated each time the parameters are updated. Nor are parameter fields as aesthetically pleasing or geologically reasonable as results obtained from Tikhonov calibration (where reasonableness is built into the regularization process through use of a preferred parameter condition). Two approaches have been developed for PEST to overcome these difficulties.

Tonkin and Doherty (2005) describe the “SVD-Assist” scheme that is implemented in PEST whereby definition of the calibration solution and null subspaces takes place just once on the basis of the Jacobian matrix calculated at initial parameter values. Before the calibration process starts, a set of “superparameters” is defined by sensitivities calculated from the full set of native or base parameter values, thereby reducing the full parameter space to a subset of the full set of base parameters (Tonkin and Doherty, 2005). These combinations of parameters are then estimated as if they were ordinary parameters; whenever derivatives are calculated for the purpose of refining and improving parameter estimates, these are taken with respect to superparameters

rather than individual base parameters. Each iteration of the revised parameter-estimation process then requires a Jacobian matrix calculated by using only as many model runs as there are estimable parameter combinations. Because this number of combinations is normally considerably less than the total number of parameters used by the model, a large computational savings is achieved.

If superparameters are few enough, their values can be estimated by using traditional calibration methods for well-posed inverse problems. If not, Tikhonov regularization (with default conditions applied to base parameters) can be included in a “hybrid” SVD-Assist/Tikhonov parameter-estimation process. Large reductions in run times are achieved because the number of runs needed in each iteration of the parameter-estimation process is related to the number of superparameters. Simultaneous application of Tikhonov-regularization constraints allows the user to interject soft knowledge of the system into the parameter estimation process and thus rein in the pursuit of a best fit to a suitably chosen target measurement objective function. Because of the complimentary increase in speed and likelihood of obtaining geologically realistic parameter fields, the hybrid SVD-Assist/Tikhonov approach is the most efficient, numerically stable, and geologically reasonable means of highly parameterized groundwater-model calibration. However, for highly nonlinear models, the subdivision of parameter space into solution and null subspaces based on initial parameter values may not be applicable for optimized parameter values. In practice, this obstacle is normally overcome by estimating more superparameters than are required for formulation of a well-posed inverse problem and applying Tikhonov regularization or SVD on the superparameters to maintain stability. Also, if necessary, superparameters can be redefined partway through a parameter-estimation process following recomputation of a base-parameter Jacobian matrix.

## Before Running PEST: Model Parameterization

Regularized inversion can be employed for estimating any type of parameter employed by a model. However, certain parameterization schemes and types of parameters are more able to exploit the benefits of regularized inversion than are others. In order to decide how best to use the regularized inversion approach, some understanding of the underlying parameterization concepts is useful.

### Parameterization Philosophy

Those who are new to regularized inversion are required to adopt a different philosophy of parameterization than that behind traditional calibration methods (Hunt and others, 2007). Rather than requiring the modeler to simplify the parameters a priori and subjectively before calibration,

regularized inversion allows the modeler to carry forward any parameter that is of potential use for calibration and prediction. If parameters are properly defined, Tikhonov constraints are properly formulated, and/or the solution subspace is restricted to a small enough number of dimensions, a minimum-error variance solution to the inverse problem of model calibration can be obtained irrespective of the number of parameters employed. Indeed, Doherty, Fienen, and Hunt (2010) show that achievement of a minimum-error variance parameter field is more likely to be compromised by the use of too few parameters than by the use of too many. A regularized-inversion philosophy to parameterization, then, can be summarized as “if in doubt, include it.”

One of the attractions of highly parameterized model calibration is that a modeler is relieved of the responsibility of deciding which parameters to include and which to exclude from the parameter-estimation process, and/or which parameters to combine, in order to reduce the number of parameters requiring estimation and thereby achieve a well-posed inverse problem. Parameters, or parameter combinations, that are inestimable will simply adhere to their initial values or to soft-knowledge default values specified by the modeler (which should be the same) unless the calibration dataset dictates otherwise. In principle, model parameters often not considered for estimation in traditional calibration contexts (such as those pertaining to boundary conditions and/or sources/sinks of water) could also be included in the parameter-estimation process. Although this extension to nontraditional parameters may, or may not, prove beneficial in some calibration contexts, it could be valuable for postcalibration uncertainty analysis if a modeler is unsure of these parameters' values and if one or more critical model predictions may be sensitive to them.

The current practical limit to the total number of parameters that can be employed in the parameter-estimation process is around 5,000. It results from the following factors:

1. If parameters are large in number, each individual parameter may consequentially be of diminished sensitivity. This diminished sensitivity may erode the precision with which derivatives of model outputs with respect to individual parameters can be computed by using finite-parameter differences.
2. If parameter sensitivities are computed by using finite-parameter differences, many model runs are required to fill the Jacobian matrix. Even where the SVD-Assist method is employed for solution of the inverse problem, sensitivities of model outputs with respect to all base model parameters must be computed at least once at the start of the parameter-estimation process so that superparameters can be defined.
3. Memory requirements can overwhelm resources when many parameters are employed in conjunction with a large calibration dataset.
4. Where there are many observations and many parameters ( $>2,500$ ), singular value decomposition of a large Jacobian matrix may require an inordinate amount of computing time. LSQR techniques employed in PEST can mitigate this restriction, however.

## Spatial Parameterization

Spatial parameterization of a model domain may use zones of piecewise constancy, pilot points, or a combination of these, with or without the concomitant use of other parameterization devices.

### Zones of Piecewise Constancy

Zones of piecewise constancy have a long history in traditional parameter estimation as a means for simplifying the natural-world complexity in the model domain. Such an approach can also be used in the regularized-inversion context, where they are commonly chosen to coincide with mapped geological units (thus allowing more geological units to be represented in the parameter-estimation process than would otherwise be possible). Or, they may be used in areas that are mapped as geologically homogeneous but in which head, concentration, and/or other historical measurements of system state suggest the presence of intraformational property heterogeneity. They are probably less suited for use in the latter role, however, because they constitute a cumbersome mechanism for representing continuous spatial variation of hydraulic properties when compared to the other parameterization methods described below.

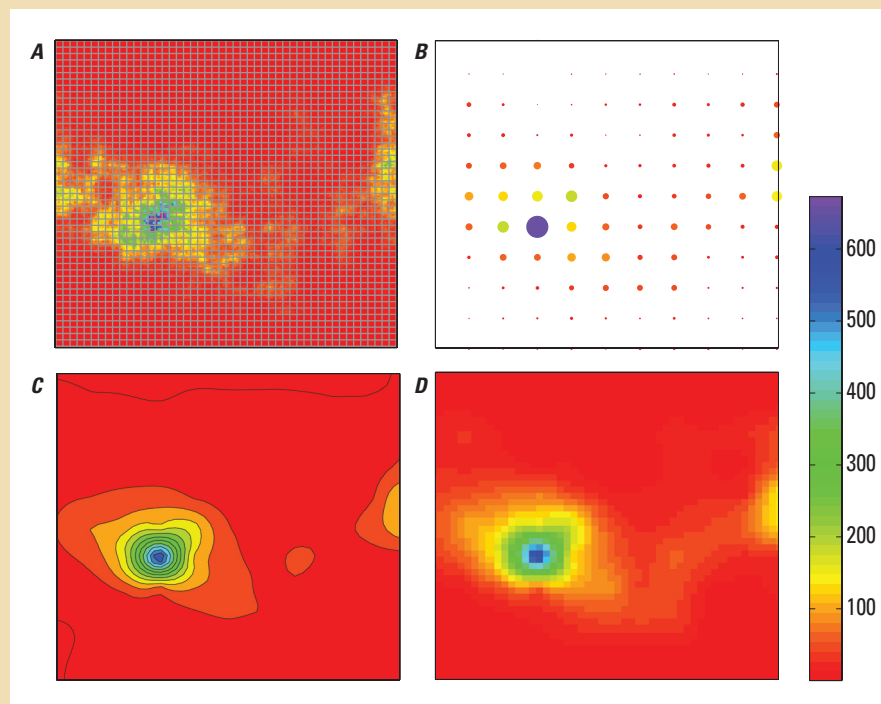
### Pilot Points

Model parameterization by use of pilot points is discussed by de Marsily and others (1984), Doherty (2003), Alcolea and others (2006, 2008), Christensen and Doherty (2008), Doherty, Fienen, and Hunt (2010), and references cited therein. Briefly, parameter values are estimated at a number of discrete locations (pilot points) distributed throughout the model domain; cell-by-cell parameterization then takes place through spatial interpolation from the pilot points to the model grid or mesh. Hydraulic properties ascribed to the pilot points are estimated through the model-calibration process are then automatically interpolated to the rest of the model domain. Currently, the only spatial interpolation device supported by the PEST Groundwater Data Utilities suite is kriging; however, Doherty, Fienen, and Hunt (2010) suggest that a minimum-error variance solution to the inverse problem of model calibration may be better attained through use of orthogonal-interpolation functions.

## Conceptual Overview of Pilot-Point Use (excerpted from Doherty, Fienen, and Hunt, 2010)

The general goal of pilot points is to provide a middle ground between cell-by-cell variability and reduction to a few homogeneous zones for characterizing natural-world heterogeneity in groundwater models. Figure B2–1 depicts a schematic representation of the pilot-point implementation. In Figure B2–1A, a heterogeneous field is depicted overlain by a model grid. This illustrates that, even at the model-cell scale, the representation of heterogeneity requires simplification. In figure B2–1B, a network of pilot points is shown in which the size of the circle is proportional to the parameter value and the color represents the value on the same color scale as in Figure B2–1A. The general pattern of variability in the true field is visible in this image, but the resolution is much coarser than reality. Figure B2–1C shows the pilot-point values interpolated onto a very fine grid and illustrates that much of the true heterogeneity can be reconstructed from a subset of sampled values provided that appropriate interpolation is performed. Figure B2–1D shows the interpolated version of the pilot-point values in Figure B2–1B on the model-cell grid scale, which represents the version of reality that the model would actually see.

In reality, rather than directly sampling the true field as in this illustration, the pilot points are surrogates for the real parameter field estimated from observations in the calibration dataset and are therefore likely to include some error not depicted on this figure. However, the schematic representation depicts the best possible representation of the real field given the displayed density of pilot points.



**Figure B2–1:** Conceptual overview of representing complex hydrogeological conditions through use of pilot points. Panel a) shows the inherent property value overlain by the model grid in gray. Panel b) is a representation of the true property values by a grid of pilot points in which symbol size indicates value. Panel c) shows an interpolated representation of panel b) on an arbitrarily fine grid scale. Panel d) shows the value from the pilot points interpolated to the computational-grid scale. Interpolation in all cases was done by using ordinary kriging. The same color scale applies to all four panels.

Pilot-point emplacement can be regular or irregular, allowing the user to increase pilot-point density where data density is high and to decrease it where data density is low. This maximizes the ability of a given number of pilot-point parameters (this number normally being set by available computer resources) to respond to the information content of a given calibration dataset. Some groundwater modeling graphical-user interfaces support both automatic and manual pilot-point emplacement. A user can select pilot-point locations by clicking on those locations and/or by dragging pilot points to new locations. Any mapping software that supports a digitizing option can also be used to designate pilot-point locations.

Pilot points can be employed to represent any spatially variable property: hydraulic conductivity, specific yield, porosity, and so on. Software provided with the PEST Groundwater Data Utilities suite presently supports only two-dimensional spatial interpolation from pilot points to a model grid or mesh. However, functionality is available within this utility suite for vertical interpolation among pilot-point arrays located at various levels within a multilayer hydrostratigraphic unit to intermediate layers within that unit (see the PARM3D utility).

An immutable set of rules for pilot-point emplacement does not exist. However, the following suggestions, based on a mathematical analysis of pilot-point parameterization suggested by Doherty, Fienen, and Hunt (2010), are salient:

1. Place pilot points so as to avoid large gaps or “outpost” locations. Often a uniform grid of pilot points can be used to ensure some minimal level of coverage of the model domain, which is then augmented with additional pilot points assigned in areas of interest.
2. Place pilot points used to estimate horizontal hydraulic conductivity between head-observation wells along the direction of groundwater gradient.
3. In addition, place pilot points at wells where pumping tests have been done so that these hydraulic-property estimates can serve as initial and/or preferred parameter values.
4. Place pilot points used to estimate storage parameters at the locations where temporal water-level variations are included in the calibration dataset.
5. Ensure that pilot points used to estimate hydraulic-conductivity parameters are placed between outflow boundaries and upgradient observation wells.
6. Increase pilot-point density where data density is greater.
7. However, do not place pilot points any closer together than the characteristic length of hydraulic-property heterogeneity expected within the model domain.
8. If pilot-point numbers are limited by computing resources, consider using fewer pilot points for representing vertical hydraulic conductivity in confining or semiconfining units than for representing horizontal conductivity in aquifers.

## Pilot Points in Conjunction with Zones

Pilot points and zones of piecewise constancy are not mutually exclusive. For example, some zones may have many pilot points, and others just one. When a single pilot point is assigned to a zone, the parameter-estimation process substitutes one value for each node contained in that zone, thus making the pilot-point parameter act as a piecewise-constant zone. In the case of many pilot points in a zone, pilot-point-support software provided through the Groundwater Data Utilities suite allows assignment of families of pilot points to different zones. Spatial interpolation from pilot points to the model grid or mesh does not take place across zone boundaries. With appropriate regularization in place, the parameter-estimation process is thus given the opportunity to introduce heterogeneity preferentially at zone boundaries and to then introduce intrazonal heterogeneity if this is supported by the calibration dataset. In the case of one pilot point in a zone, the application of the parameter to the zone is insensitive to the location of the pilot point within the zone.

## Other Parameter Types

Parameters other than those representing two- and three-dimensional spatially variable properties are also readily employed in the regularized-inversion process. The following are some examples of parameter types that have been employed:

1. Conductance of river/stream beds, drains and general-head boundaries, with spatial variability represented by zones of piecewise constancy or through interpolation between pilot points placed along these linear features.
2. Spatially varying multipliers for recharge, with multipliers represented by pilot points.
3. Elevations of general-head boundaries, these being represented by zones of piecewise constancy and/or pilot points.
4. Transport source terms, these being represented by zones of piecewise constancy.
5. Elevation and spread of a freshwater-saltwater interface, represented by pilot points and variables governing concentration spread across the interface—see the ELEV2CONC utility listed in appendix 3.

The model-independent/universal design of PEST allows for virtually unlimited flexibility in definition of “a model.” A model can in fact be composed of a suite of executable programs encapsulated in a batch or script file. For example, an unsaturated-zone model and/or irrigation-management model may compute recharge for the use of a groundwater-flow model. This, in turn, may provide a flow field that is used by a transport model for computation of contaminant movement. Parameters pertaining to any or all of these models can be estimated simultaneously by PEST on the basis of a diverse

set of data pertaining to many different types of measurement of historical system state. As stated previously, considerations of what is estimable, and what is not estimable, on the basis of the current calibration dataset need not limit the design of the parameter-estimation process; mathematical regularization ensures that estimates are provided only for parameters and/or parameter combinations that are estimable given the calibration data available. Moreover, the design of a suitable Tikhonov-regularization strategy will help ensure that the complex parameter field that emerges from the calibration process is geologically reasonable.

## Initial Parameter Values

Implementation of nonlinear-parameter estimation requires that an initial value be provided for each parameter that is adjusted through the calibration process. In traditional, overdetermined parameter-estimation contexts, initial values assigned to parameters often do not adversely affect the parameter estimation. Provided that no local optima exist and the model is not too nonlinear, PEST will find the global minimum of the objective function and optimal parameter set, irrespective of parameter starting values. Nevertheless, the following guidelines may make that process more efficient:

1. Assign initial values to parameters that are within an order of magnitude of those that are expected to be estimated for them through the calibration process.
2. If parameters vary in sensitivity within that range, assign initial values to parameters in the more sensitive area of their reasonable range.

When regularized inversion is used, these guidelines are no longer relevant. If subspace methods are employed in the parameter-estimation process (for example, if this process is implemented through SVD or through SVD-Assist), the initial values supplied for parameters should be their “preferred values” from a geological perspective. This is because, as stated previously, the values assigned to individual parameters, and/or to combinations of parameters, that are found to be inestimable on the basis of the current calibration dataset will not change from the initial values during the parameter-estimation process. Thus, geologically reasonable parameter values specified at the start of the parameter estimation process will ensure the return of geologically reasonable parameter values at the conclusion of the parameter estimation process. If Tikhonov regularization is employed, the preferred condition should ensure that parameters are assigned geologically reasonable values. Thus, regularization constraints encapsulated in the Tikhonov-regularization scheme should be such that these constraints are perfectly met by initial parameter values, this resulting in an initial “regularization objective function” (see below) of zero.

A problem in implementing this strategy is that a modeler may not know, ahead of the parameter-estimation process, what the preferred value of each parameter actually is. This problem can be addressed in the following ways:

1. Initial values can be assigned on the basis of maximum geological plausibility; such values are then, by definition, of minimum statistical precalibration error variance. The minimum error variance status of inestimable parameters, and parameter combinations, is thereby transferred to the postcalibration parameter field.
2. Prior to regularized inversion on a large parameter set, parameters can be tied or grouped on a layer-by-layer (or even broader) basis. This allows estimation of broad-scale system properties through an overdetermined parameter-estimation exercise based on simplifying assumptions such as that of parameter field uniformity. Layerwide (or even modelwide) parameter values arising from this exercise can then be employed as starting values for an ensuing highly parameterized inversion exercise, in which system-property details are estimated. This was the approach taken by Tonkin and Doherty (2005); Fienen, Hunt, and others (2009); and Fienen, Muffles, and Hunt (2009).

## Tikhonov Regularization Strategies

Tikhonov regularization interjects soft knowledge into the parameter-estimation process, and the PEST framework is flexible with regard to how this soft information is applied. Regularization constraints can be supplied through prior-information equations (in which case these constraints must be linear) or as observations (in which case they can be linear or nonlinear).

In implementing Tikhonov regularization, PEST evaluates two criteria simultaneously:

1. the misfit between measured values (such as heads and flows) and their simulated counterparts (quantified through the traditional “measurement-objective function”) and
2. the departure of the current parameter set from its preferred condition as specified through Tikhonov constraints (encapsulated in the “regularization-objective function”).

Quantification of model-to-measurement misfit is an essential component of all parameter-estimation methodologies; quantification of departure from a preferred parameter state is not. In calculating the regularization objective function, PEST applies a global weight multiplier to all regularization constraints, whether these are encapsulated in observations or in prior-information equations. This multiplier is adjusted in order that a user-supplied “target measurement-objective function” (=the PEST Control File variable PHIMLIM) is respected. The target measurement-objective function specifies a level of model-to-measurement misfit that PEST attempts to achieve but not reduce beyond. Its value is set under the premise that any improvement in fit beyond that specified by the user via PHIMLIM is gained only at the cost of “overfitting,” with a consequential deterioration in the plausibility of the

estimated parameter field. This degradation is most commonly expressed by extreme parameter values; where pilot points are used this is expressed as “bullseyes” of extreme parameter values in a field of more uniform parameter values.

Relative weights applied to Tikhonov-regularization constraints can be set by the modeler. Optionally, this relative weight can be overridden by PEST in the course of the parameter-estimation process via the IREGADJ regularization-control variable (“InterREGularization group weights ADJustment” variable, which is specified in the PEST Control File, as described in appendix 1). If IREGADJ is set to a number greater than zero, PEST adjusts the weights applied to individual or grouped Tikhonov constraints in ways that complement data inadequacy. This capability is discussed in more detail below.

An important principle for designing a regularization scheme is that regularization should be pervasive if it is to be effective, thereby providing a fallback value for every parameter and/or combination of parameters that is inestimable on the basis of the current calibration dataset. Because the estimability of every parameter is generally not known before the parameter-estimation process begins, this fallback offers a safeguard against the assignment of aberrant values to parameters that are poorly informed by the calibration dataset.

A description of the many Tikhonov-regularization strategies that could be employed in calibration of a highly parameterized groundwater model is beyond the scope of this document and, even if offered here, would likely be superseded as research on this topic goes forward. Instead, the discussion below is confined to two broad Tikhonov-regularization options that are readily implemented through PEST utility support software: preferred-value regularization and preferred-difference regularization. Each, or both, can be employed within the same calibration process; they can be applied to different parameter types or same parameter type, or even to the same set of parameters.

## Preferred-Value Regularization

In implementing this form of regularization, a prior-information equation is provided for every adjustable parameter. Each such equation assigns that parameter a value deemed to be of minimum error variance for that parameter. Each such prior-information equation can be given an individual weight. Alternatively, a covariance matrix can be employed for groups of such equations—for example, all prior-information equations that pertain to pilot points that represent a property such as hydraulic conductivity within a single model layer. This covariance matrix is often based on a variogram. If spatial correlation implied in the covariance matrix is a reflection of plausible geological variability, this strategy promotes emergence of heterogeneity in a manner that is of maximum geological likelihood.

## Preferred-Difference Regularization

Through this mechanism, preferred values are entered on the basis of differences between parameters. Most commonly, a “preferred-homogeneity” condition is used, where the preferred difference between parameters is set to zero in the prior-information equations that express parameter differences. This approach designates uniformity as the preferred parameter condition. When pilot points are employed as a parameterization device, weights assigned to prior-information equations that express parameter differences of zero can be uniform. Alternatively, they can be calculated according to a variogram that purports to describe spatial variability of the pertinent hydraulic property type within the model domain; greater weights are then ascribed to prior-information equations linking parameters that show a high degree of spatial correlation (taking directional anisotropy into account) than to those that show a smaller degree of spatial correlation (for example, parameters assigned to pilot points located further apart).

Utility software supplied with PEST allows preferred-difference linkages to be implemented both within and between model layers. In the latter case, the preferred value of parameter differences need not be zero. Where parameters are log-transformed during the parameter-estimation process (as many non-negative parameters should be), these differences actually apply to the logs of parameter values and hence provide the parameter-estimation process with a preferred ratio for interlayer parameter values. However, because such a ratio is rarely known or estimated, the use of interlayer preferred-difference regularization is not widespread.

Nevertheless, the issue of interlayer regularization may be important. As stated previously, for Tikhonov regularization to be effective, it must be applied liberally throughout the model domain. PEST utility support software facilitates construction of a series of layer-specific, intralayer preferred-difference regularization schemes; yet, an ill-posed inverse problem can still result if solution nonuniqueness can exist on a layer-by-layer basis, given the information content of the calibration dataset. This problem can be overcome by

1. use of interlayer difference regularization (as stated previously),
2. use of preferred-value regularization instead of (or in addition to) intralayer preferred difference regularization, and/or
3. concomitant use of subspace regularization, through adoption of truncated SVD and/or SVD-Assist for solution of the inverse problem of model calibration.

Of these, the third option is likely to be most easily implemented in most calibration contexts.

## Before Running PEST: Observations Used in Inversion Process

There is no universal prescription for the manner in which observations should be processed and weighted for model calibration. However, a short discussion as it applies particularly to highly parameterized inversion is presented here.

It is often suggested that the weight assigned to each measurement be inversely proportional to the noise associated with that measurement. Ideally, where noise is correlated between measurements, a weight matrix should be employed instead of individual measurement weights, this matrix being proportional to the inverse of the overall covariance matrix of measurement noise as it applies to the correlated set of measurements. Where a parameter estimation problem is well posed, this strategy ensures that estimated parameter values approach those of minimum error variance.

Suspect observations should be given low weights to prevent corruption of parameters estimated through the calibration process. Rigorous pursuit of the above weighting strategy, however, is often not optimal in real-world groundwater modeling practice because of the following factors.

1. Such an approach may result in an unbalanced regression such that large numbers of observations of one type dominate the total objective function.
2. Where regularization is done through mathematical means as part of the parameter-estimation process itself, weighting on the basis purely of measurement noise, and not accounting for an observation's importance for a prediction of specific interest, may degrade the model's ability to make that prediction (Moore and Doherty, 2005; Doherty and Welter, 2010).
3. Model-to-measurement misfit is commonly dominated by structural noise rather than by measurement noise. Structural noise results from a model's inability to simulate real-world processes exactly, as well as from the parameter simplifications that constitute the manual or mathematical regularization necessary to achieve a unique solution to the inverse problem of model calibration. As Cooley (2004), Cooley and Christensen (2006), and Doherty and Welter (2010) demonstrate, this noise shows a high degree of spatial correlation in even a simple groundwater model; Gallagher and Doherty (2007) explain that structural noise shows a high degree of temporal correlation for a surface-water model, with the correlation between similar flow events being greater than that between flows that are in temporal juxtaposition. Unfortunately, except for synthetic cases, the covariance structure of this noise cannot be known.
4. Even if the covariance matrix of structural noise could be determined, its use in the inversion process would be computationally difficult when a large number of observations are featured in the calibration dataset.

Thus, other observation-weighting approaches are often used in highly parameterized models, some of which are described below.

## Formulation of an Objective Function

In most calibration contexts a "multicomponent" objective function is recommended, with each component of this objective function calculated on the basis of different groups of observations or of the same group of observations processed in different ways (for example, Walker and others, 2009). As discussed below, if properly designed, such an approach can extract as much information from a calibration dataset as possible and transfer this information to estimated parameters. Ideally, each such observation grouping should illuminate and constrain the estimation of parameters pertaining to a separate aspect of the system under study. Furthermore, relative weighting between groups should be such that, at the start of the parameter-estimation process at least, contributions by different groups to the overall objective function should be roughly equal so that none of these groups dominates the objective function or is dominated by the contribution to the overall objective function made by other groups. PEST facilitates this process by listing the contribution made to the overall objective function by all user-defined observation groups at the start of every parameter-estimation iteration.

## Objective-Function Components

In this subsection, some suggestions are presented as to how observations can be collected into separate groups, each informative of different aspects of the system under investigation. When employed in the calibration process, weighting *within* each group should be such that less reliable measurements are penalized for their lack of integrity. However, weighting *among* groups should be such that each is visible in the measurement objective function, at least at the start of the calibration process; this ensures that no group is ignored by PEST and that parameters that are informed by each separate group are seen by the parameter-estimation process. An exception to such an approach is the inclusion of model-run information (reported mass balance, number of iterations or dry cells, and so on) that is given zero weight and included simply for reporting purposes rather than for informing the parameter-estimation process. Because such a zero-weight group does not affect parameter estimation, it is not further considered here.

In the examples presented below, each observation group may be composed of raw data (for example, head measurement) or processed data (for example, drawdown calculated by the time-series processor TSPROC). In the case of processed data, identical processing should be applied to both the field observations and their model-generated counterparts so that "apples are compared to apples." Simulated observations

should be temporally and spatially interpolated to the times and locations of pertinent field measurements before they are processed. All data-processing functionality described below is provided by PEST utility support software.

## Data of Different Types

Data of different types should be included in the parameter-estimation process to the extent possible (Hunt and others, 2006). These should be placed into different observation groups to facilitate monitoring the progress of the parameter-estimation process. However, because different observation groups will likely have different populations, and because different measurement types employ different units, it is unlikely that each group will have equal visibility in the initial measurement objective function. Therefore, the user likely will have to intervene to implement an appropriate weighting strategy that either promotes equal visibility or encourages the parameter estimation process to fit aspects of field measurements that are most closely aligned with key predictions required of the model.

## Special Considerations for Concentration Data

Head data alone are expected to provide little information on geological heterogeneity. On the other hand, concentration data, especially where the data pertain to a contaminant source whose location and timing are known, can provide information on hydraulic property heterogeneity of the material through which the contaminant plume has traveled. Both data types should therefore be included in the calibration process, especially if a modeling objective includes design of a remediation system. Heads and concentrations should be assigned to different groups, and intergroup weighting should be such that each group is visible in the initial objective function.

Intragroup weighting of observations with widely ranging values, such as concentration measurements, deserves special attention. Consideration should be given to weighting concentration measurements in inverse proportion to their magnitudes (with some upper limit for these weights). Thus, the outer reaches of a contaminant plume are highly visible in the objective function, these often being informative of local heterogeneity. On the other hand, simultaneous use of the same concentration data with uniform weights applied to all concentration measurements (these being assigned to a different observation group) may promote better estimation of total contaminant mass within an aquifer.

One potentially useful approach for processing concentration observations is assigning an observed concentration that falls below the nondetection threshold a value equal to the nondetection threshold itself. Model-generated counterparts to field-observed concentrations should be subjected to the same process after every model run. This will ensure that differences between modeled and observed concentrations that are

both below the nondetection threshold are seen as zero by the calibration process. Furthermore, this strategy ensures that no discontinuities in derivatives are incurred as concentrations reach the nondetection threshold (as would occur if nondetections were assigned a concentration value of zero).

## Temporal Head Differences

During transient-model calibration, the use of differences between subsequent head measurements, or between each head measurement and a user-specified reference level (perhaps the first measurement from each particular well), will often facilitate better estimation of storage and/or recharge parameters than would result if head values alone were employed in the calibration process. Thus, failure to exactly match heads need not compromise the ability of the calibration process to estimate a set of parameters that captures the system dynamics (for example, seasonal or multiseasonal head differences). The ability of a model to be employed for short- or medium-term aquifer management will be improved as a result.

## Vertical Interaquifer Head Differences

A calibration process that explicitly includes (often small) interlayer head differences as a separate (and visible) component of a multicomponent objective function can also ensure that these differences are seen by the calibration process and that vertical interlayer conductances are better estimated as a result.

Insight into this strategy (similar to that for temporal head differences) can be gained from noting that if  $x$  and  $y$  are two random variables, the variance of their difference is calculated as

$$\sigma_{x-y}^2 = \sigma_x^2 + \sigma_y^2 - 2\sigma_{xy} \quad (1)$$

Where correlation  $\sigma_{xy}$  between two measurements is high (as is often the case for vertically separated head measurements or for successive head measurements in the same well) the variance of the difference can be very small even though the variance of each head measurement may be large; the difference is thus worth fitting, even if individual measurements cannot be fit so well. The difference thus deserves visibility in the objective function and therefore requires a weight that allows it to be visible. In addition, head differences often constrain specific parameter types (such as vertical conductance or storage) even though head values by themselves are not as informative. This difference is therefore relatively easy to fit.

## Joint Steady-State/Transient Calibration

Joint calibration of steady-state and transient MODFLOW models can be done without difficulty as a result of MODFLOW's ability to mix transient and steady-state stress periods in the same simulation. Simultaneous calibration of this kind brings with it the following advantages:

1. Information on conductance parameters contained within time-averaged head measurements employed in the steady-state component of the calibration dataset can directly inform estimation of these parameters.
2. Strong correlation between conductance and storage parameters that result where parameter estimation is done solely on the basis of a transient model is dramatically reduced.
3. Steady-state heads computed by the steady-state model can, under many conditions, be employed as initial heads for the ensuing transient model.

Steady-state heads should be assigned to a different observation group than that employed for transient heads. Transient-head differences should constitute another observation group. Intergroup weighting should be such that each group is visible in the initial objective function.

## Declustering

Where more head measurements are available from some wells than from others, the user should consider increasing weights associated with heads measured in wells that are more sparsely sampled compared to those from which more samples were collected, especially if heads in these wells are very different. This weighting scheme prevents information from the more densely sampled wells from drowning out that from the more sparsely sampled wells purely because of the numerical preponderance of measurements. Similarly, where spatial density of measurement wells is highly variable, the user should consider assigning lower weights to heads from areas of high well density than those assigned to heads measured in solitary wells; the latter may be the sole repository of information on hydraulic conductivity over large parts of a model domain.

## Digital Filtering

Where measured heads show high temporal variability due, for example, to proximity to pumping or recharge centers, digital filtering may be employed to remove this variability before attempting to fit that dataset to its (filtered) modeled counterpart, especially if the timing and magnitude of causative fluctuating stresses are not exactly known.

## "Intuitive" and Other Soft Data

The calibration process is poorer if any pertinent information is withheld from it. In many instances a single "intuitive observation" of long-term system behavior (for example the observation that total base flow is, on average, a certain percentage of total rainfall or that outflow through a certain boundary is roughly equal to a certain value) can make the difference between estimability and inestimability of a certain parameter or certain combination of parameters. On some occasions there may be a reluctance to include such a poorly known observation in the calibration process, because errors that are possibly associated with its value may be transferred to parameters that are estimated on its basis. However, if the outcomes of a model calibration process are to be at least partially assessed on the basis of whether such an observation is respected or not, then the calibration process is better served with the observation in question included in the calibration dataset, albeit with a low weight if its integrity is questionable. In addition, in some cases a modeler may wish to include in the PEST input dataset measurements that are of low integrity (for example, drillers' reports of head) to which weights of zero are assigned. These can then be used for qualitative assessment of the outcomes of the calibration process.

## Temporal and Spatial Interpolation

Before being matched with field data, model outputs must undergo spatial and temporal interpolation to the sites and times at which field measurements were made. For inflow/outflow measurements, spatial averaging is also required (for example, along pertinent stream or river reaches). Functionality for all of these tasks is available through the PEST Groundwater Data Utility suite (see appendix 3).

A modeler should ensure that the interpolation and averaging steps that are a necessary precursor to the matching of model outputs with field measurements do not contribute to structural noise. For example, structural noise may be induced through any of the following processing tasks:

1. temporal interpolation where model time steps are large and stresses have recently changed,
2. spatial interpolation in areas of high potentiometric-surface curvature.
3. spatial interpolation in regions of high concentration gradient, or
4. summation of stream inflows over reaches where conductance is high, cell width is large, and/or reaches are sharply curved.

## A Note on Model Validation

It is sometimes recommended that a model be calibrated against one data type (for example, heads) and “validated” against another (for example, concentrations). Such a recommendation ignores the fact that these different data types contain information pertinent to different aspects of the modeled system; the calibration process is therefore poorer with either data type omitted.

Although the concept of “validation” is outside the scope of this report, it is worth noting that even the most carefully constructed model affords no guarantee of making a correct prediction. Rather, it is a foundation for developing predictions that lie within the realistic margins of uncertainty estimated by a carefully constructed and well-parameterized model. Where data are scarce, uncertainty margins will be wide—an inescapable consequence of data paucity. It follows, therefore, that a model cannot be validated; it can only be invalidated. Furthermore, it can only be invalidated at a certain level of confidence. Thus, the withholding of data from the calibration process for the purpose of “validation” should be done with caution. If some data are indeed withheld, consideration should then be given to including the omitted data in a final calibration exercise before the model is employed to make important predictions, for data previously withheld for the purpose of validation may add another dimension or two to the calibration solution space when returned to the calibration dataset. Any opportunities to inform the calibration process (and thereby decrease the dimensionality of the null space) before making important predictions should be encouraged.

## Before Running PEST: Preparing the Run Files

PEST requires three types of input files. These are as follows:

1. One or more template files used to insert estimated parameter values into model input files. Commonly one template file is used per model input file that contains parameters to estimate.
2. One or more instruction files that instruct PEST how simulated equivalents of observations are read from model output files.
3. A PEST Control File (*\*.pst*) which supplies
  - a. initial values for all parameters included in the calibration process,
  - b. observed values and weights for all members of the calibration dataset,
  - c. regularization constraints, and
  - d. variables that control the operation of all aspects of the (regularized) inversion algorithm implemented by PEST.

A PEST-input dataset can be prepared by using a text editor, because all of its input files are ASCII files. Where parameter and/or observation numbers are large, software support is in many cases available through commercial graphical user interfaces (GUIs). Support for more complex parameter-estimation problems, and/or for custom extensions to GUI-constructed PEST input files, is also available through utility programs provided with PEST and through its Groundwater Data Utilities suite (see appendixes). Custom programming can also be used to supplement PEST utility support programs.

Although automatic PEST Control File construction removes much of the file-construction burden from the user, a number of issues are often still considered when preparing for a PEST run. Thus, discussion is now directed at a subset of variables within the PEST Control File whose settings control the operation of PEST’s inversion algorithm. The discussion herein is necessarily brief and incomplete; greater detail is available in PEST documentation. PEST input variables are described with names given in the PEST documentation. A list of PEST control variables, together with the positions that they occupy in the PEST Control File, can be found in appendix 1. For ease of reference, headers in this section follow those in a PEST Control File. However, not all headers are required in all cases; some alternative headers are also used.

## Control Data

## Memory Conservation

The MAXCOMPDIM variable instructs PEST to use compressed storage for the Jacobian matrix. Where parameter and observation numbers are large, this compression can make the difference between whether a PEST run can or cannot take place, given the computer memory to which it has access. Note, however, that the cost of using compressed Jacobian matrix storage is slower execution speed.

## The Marquardt Lambda

The primary purpose of the Marquardt Lambda capability is to increase the efficiency of a nonlinear parameter-estimation process, particularly in its early stages, through rotating the parameter upgrade vector towards the direction of steepest objective function descent. However, it also performs a secondary role as a de facto regularization device; high values of the Marquardt Lambda can allow an ill-conditioned parameter-estimation process to proceed notwithstanding locally low parameter sensitivities and/or excessive parameter correlation.

In some instances, particularly where finite-difference derivatives are noisy because of numerical granularity of model outputs and/or where a model is highly nonlinear, calculation of trial parameter upgrades on the basis of widely different Marquardt Lambdas can increase the chances of a substantial reduction in the objective function during any one optimization iteration. A negative setting for the RLAMFAC

variable ( $-3$  suggested) allows a much wider range of lambda values to be tested than would be the case if a value-independent multiplier were employed. This should be complemented by an initial lambda value (RLAMBDA1) of between 0.01 and 100 (20 suggested), with the higher values used in cases that are suspected to be numerically troublesome. See Doherty (2010b) for additional description of this option.

## Broyden's Jacobian Update

Each iteration of the parameter-estimation process undertaken by PEST consists of the following two stages:

1. computation of the Jacobian matrix;
2. testing of one or several parameter upgrades calculated on the basis of different Marquardt Lambdas.

Implementation of Broyden's Jacobian update methodology blurs the distinction between these two stages, because information gained on model-output dependency on parameters through the testing of different parameter upgrades is employed to enhance the accuracy of the Jacobian matrix for use in computation of further parameter upgrades with new Marquardt Lambdas.

Setting the JACUPDATE variable to 999 instructs PEST to implement Broyden's Jacobian update methodology on each occasion that a new parameter upgrade is tested, during every iteration of the parameter-estimation process. Experience has demonstrated that, in many parameter-estimation contexts, this approach can lead to greater gains in objective-function reduction during each iteration of the parameter-estimation process than would otherwise be the case.

## Number of Optimization Iterations

The NOPTMAX variable limits the number of optimization iterations that PEST implements before ceasing execution. However, some of its settings have special significance.

If NOPTMAX is set to 0, PEST will run the model once, compute the total objective function and contributions made to it by different observation groups, and cease execution. Information from such a run can assist the user in assigning weights to different observation groups to ensure that they are all visible in the initial objective function. In addition, this setting allows the modeler to check the PEST setup, including the integrity of template and instruction files, the proper functioning of the model batch file and associated utility software used in conjunction with the model, and so on.

Setting NOPTMAX to  $-2$  instructs PEST to compute a Jacobian matrix and to then cease execution. Computation of a Jacobian matrix is required before PEST's SVD-Assist functionality can be implemented, because superparameters are constructed from native model parameters through singular value decomposition of this weighted matrix.

If NOPTMAX is set to  $-1$ , PEST performs in the same manner as if NOPTMAX were set to  $-2$  but does a final model run after filling the Jacobian matrix, this run being based on

initial parameter values. A complete set of statistics pertinent to the current parameter-estimation problem is also computed and recorded in pertinent PEST output files.

## Automatic User Intervention

Setting the DOAUI variable to "au" instructs PEST to implement its "automatic user intervention" functionality. This forces PEST to do a series of reformulations of an ill-posed inverse problem during each iteration of the parameter-estimation process, with insensitive parameters progressively (temporarily) removed from that process until the condition number of the inverse problem is low enough for a substantial improvement in the objective function to be made. Experience has demonstrated that this approach can be effective where

1. parameters are relatively few,
2. model numerical problems degrade finite-difference-based derivatives (and often thereby create local optima in the objective function), and
3. the model is highly nonlinear.

Unfortunately "automatic user intervention" requires many model runs and is not a parallelizable process. Hence, it should not be implemented if model run times are long. See Skahill and Doherty (2006) for more details.

## Solution Mechanism

Unless instructed to implement truncated SVD as a solution device, PEST employs a standard solution mechanism for the "normal equations" that arise from the Gauss-Marquardt-Levenberg parameter-estimation process. The method (taken from the LINPACK subroutine library) is fast and robust, taking advantage of the positive definite status of the normal matrix. Alternatively, truncated SVD or LSQR can be invoked as a solution mechanism. Current experience suggests that LSQR is superior only for very highly parameterized inversion problems, especially those with many observations, where SVD is too slow.

Where truncated SVD is chosen as the solution mechanism, the modeler must provide a truncation setting that sets the number of eigenvectors retained in the solution process (by using the MAXSING variable) or must specify the eigenvalue ratio at which truncation takes place (by using the EIGTHRESH variable). Although either can define the dimensionality of the calibration solution space, setting EIGTHRESH is preferable because it allows the number of retained eigenvectors to vary from iteration to iteration as parameter sensitivities change, while assuring well-posedness of the modified inverse problem. An EIGTHRESH value of between  $10^{-5}$  and  $10^{-7}$  results in stable inversion ( $5 \times 10^{-7}$  is suggested to start). Values on the  $10^{-5}$  end of the range are more effective when the calibration dataset is accompanied by a large amount of measurement noise, or where numerical noise in derivatives calculation is generated by model imperfections.

The number of eigencomponents that are actually retained can then be checked through inspection of the \*.svd file written by PEST during each iteration of the parameter-estimation process. If set on the low end of the suggested range, EIGHTHRESH should be increased if instability is encountered, or if overfitting occurs.

In theory, a Marquardt Lambda value of zero should be used where truncated SVD is employed for solution of the inverse problem. In practice, experience has demonstrated that use of a wide-ranging Marquardt Lambda (implemented by using a negative value for the RLAMFAC variable suggested above) together with a low initial Marquardt Lambda (RLAMDA1 value of 0.01) often works well, especially if there is a possibility for corruption of finite-difference-calculated derivatives as a result of model numerical imperfections.

## Parameter Groups

### Derivatives Calculation

Variables that govern the calculation of derivatives of model outputs with respect to adjustable parameters are assigned to parameter groups, rather than to individual parameters, to obviate storage of too many such control variables, most of which would be the same. A full discussion of finite-difference derivatives calculation is given in PEST documentation. However, the following points are noteworthy:

1. A relative parameter increment of 1 percent of a parameter's current value is suitable for most occasions (set DERINC to 0.01 and INCTYP to "relative").
2. Where a parameter's value is affected by an arbitrary datum (as often applies for elevation and fixed head parameters), an absolute increment may be better.
3. If a parameter's magnitude can approach zero, a relative increment should be supplemented by an appropriate absolute increment lower bound.
4. PEST should be instructed to begin the parameter-estimation process using two-point derivatives and switch to the use of three-point derivatives as the rate of objective function diminution slows (Set FORCEN to "switch"). In some cases it is helpful to allow the three-point derivative to have a larger increment than the two-point derivative (set DERINCMUL to 2.0). Also see the PHIREDSWH, and NOPTSWITCH variables in the "control data" section of the PEST Control File.
5. Where finite-difference derivatives calculation is hampered by model numerical malperformance (fig. 1 below), consider implementing "split slope analysis" (through appropriate choice of the SPLITTHRESH, SPLITRELDIFF and SPLITACTION variables), as this may improve the efficacy of the parameter-estimation process considerably in difficult numerical environments.

6. Use the JACTEST and POSTJACTEST utilities to check the integrity of derivatives if it appears that PEST performance is being adversely affected.

## Regularization Considerations

The easiest way to add Tikhonov regularization to a PEST Control File is the preferred-value regularization via the ADDREG1 utility supplied with PEST. This simply utility adds the pertinent information to the PEST Control File, invokes regularization, and sets the preferred values of each adjustable parameter to its initial value. When adding regularization prior-information equations specifying preferred-value Tikhonov constraints to a PEST input dataset, ADDREG1 assigns these equations to different regularization groups based on the parameter groups to which parameters belong. This allows PEST to assign differential regularization weight factors to these groups, though if IREGADJ is set to zero, no interregularization group weights adjustment is done. The setting which has widest use is IREGADJ = 1, where the intergroup weights adjustment takes place on the basis of total composite observation group sensitivities. If supplied as 2, weight summation is used as a basis for interregularization group weights adjustment. If IREGADJ is set to 3, the modeler can specify that regularization constraints for some parameters be enforced more strongly than for others. IREGADJ values of 1 (recommended), 2, and 3, can enhance the numerical stability of Tikhonov-based inversion. However, most benefit can be gained from this strategy if definition of regularization groups (and hence of the parameter groups from which regularization groups are derived) is such that parameters whose composite sensitivities are very different are collected into different groups. Although this may be difficult to achieve, the following suggestions have been found to work well in practice.

1. Parameters of different types should be assigned to different parameter groups. For example, horizontal and vertical hydraulic conductivity parameters should be grouped separately.
2. Parameters pertaining to different model layers should be assigned to different parameter groups.
3. Pilot point-based parameters and zone-based parameters of the same type should be assigned to different parameter groups.

Note, however, that these guidelines do not apply if IREGADJ is set to 4 or 5 (in which case subspace enhancement of Tikhonov constraints is implemented). These settings are special cases where differential weighting is applied based on the individual estimability of the parameter (Doherty 2010b). These guidelines also do not apply if IREGADJ is set to 0, or omitted from the "regularization" section of the PEST Control File altogether (in which case differential regularization weight factor adjustment does not take place).

## Parameter Data

### Initial Parameter Values

As has already been discussed, where regularization is implemented in solution of the inverse problem of model calibration—whether this be subspace, Tikhonov, or a hybrid of these—parameters should be assigned the most reasonable initial values from a geological perspective. If properly implemented, the regularized inversion process will deviate from these only to the extent required to appropriately fit the calibration dataset.

### Parameter Transformation

A modeler can choose to log-transform or not log-transform a parameter. Ideally, if a parameter's value is restricted to positive numbers, the parameter should be log-transformed during the inversion process because transformation often results in a more linear relation between a parameter and model output. In highly parameterized inversion, there are additional considerations in addition to linearity enhancement. Log transformation tends to equalize parameter sensitivities because derivatives are taken with respect to relative, rather than actual, parameter values. This log transformation implements a de facto scaling of parameters with respect to their innate variability, a situation that is likely to lead to higher likelihood of estimated parameter values in the regularized inversion setting (see Doherty, Fienen, and Hunt, 2010). Hence, provided that parameters cannot take on zero or negative values and provided that it does not degrade problem linearity to too great an extent, consideration should be given to log-transforming all parameters involved in the parameter-estimation process.

### Scale and Offset

Although scaling and offsetting of parameter values can be applied internally by PEST, non-scaled and non-offset parameter values are actually written to model input files. Like log-transformation, an appropriate parameter scaling strategy can be used to normalize sensitivity with respect to innate parameter variability. However, until information on sensitivities becomes available through the parameter-estimation process (composite parameter sensitivities are recorded on the \*.sen file) appropriate SCALE settings may be difficult to determine.

Use of a negative SCALE value allows logarithmic transformation to be applied to a parameter whose domain is entirely negative and, hence, for which logarithmic transformation would otherwise be impossible. The OFFSET variable may be employed to prevent a parameter that is not log transformed from becoming zero (as seen by PEST), even though its value, as seen by the model, can become zero. This can prevent a slowing of the parameter-estimation process incurred by

the imposition of relative change limits on parameter movement as its value moves from zero (and large relative changes are therefore required of it). Relative parameter change limits (implemented as a means of enhancing stability of the nonlinear parameter-estimation process) are imposed through the RELPARMAX control variable.

## Observation Groups

### Objective-Function Contributions

During each iteration of the parameter-estimation process, PEST computes the contribution of each observation group to the overall objective function. Although only the total objective function is used in the nonlinear regression, reporting by observation group helps the modeler ensure that information pertinent to different aspects of a modeled system is extracted from the calibration dataset. Thus, it is often useful to break observations, including sometimes those of the same data type, into different observation groups in the PEST Control File. For example, each well hydrograph could constitute a separate group for transient-model calibration. Although the total objective function is still the primary metric for automatically evaluating the parameter-estimation process, this enhanced reporting allows a modeler to monitor the contribution to the objective function made by observations pertaining to different regions of the model domain, and/or by observations of different types, and/or by the same observation data type processed in different ways.

### Weight Matrices

Weights can be assigned on an observation-by-observation basis in the “observation data” section of the PEST Control File. Alternatively, a covariance matrix (which PEST inverts to form a weight matrix) can be assigned to an entire observation group in the “observation groups” section of the PEST Control File. Though strictly required in order to achieve minimum-error variance estimation of parameter values in the face of spatially and/or temporally correlated structural noise (see Cooley, 2004), an observation covariance matrix is rarely supplied in practice. Nevertheless, a simple covariance matrix may be warranted for some observation types to accommodate correlation of measurement noise (for example, for inflow into neighboring stream reaches separated by a common gaging station whose measurements may be in error).

### Regularization Groups

Observations and prior-information equations can be assigned to regularization groups rather than to observation groups. An observation group is deemed to be a regularization group if its name begins with the letters “regul.” Weights assigned to regularization groups are subject to global

adjustment when PEST is run in “regularization mode.” They can also be subject to interregularization group weights adjustment in accordance with the setting provided for the IREGADJ regularization control variable.

## Observation Data

Measured values for observations making up the calibration dataset, as well as weights assigned to observations, are supplied in the “observation data” section of the PEST Control File. Observations are also assigned to groups within this section.

The following should be noted:

1. The weight assigned to an observation is overridden if a covariance matrix is supplied for the group to which the observation belongs.
2. A weight of zero (unless overridden by a covariance matrix) effectively removes an observation from the parameter-estimation process; the simulated value associated with the zero-weighted observation is nevertheless reported in all pertinent PEST output files. The sensitivity of the associated model output with respect to all adjustable parameters is also computed.

Sensitivities are recorded in the \*.jco file written by PEST. In some instances a zero-weighted “observation” cited in a PEST Control File may actually be a model prediction of interest. The fact that its sensitivities are computed and recorded in the \*.jco file, notwithstanding its zero-weighted status in the calibration process, allows easy implementation of postcalibration uncertainty analysis for that prediction.

## Model Command Line

The “model” as run by PEST can be a discrete executable, or it can be a batch/script file comprised of many executables run in succession. In most cases, the latter “composite model” approach is used. Certain rules should be followed in writing a model batch or script file to ensure optimal PEST performance:

1. Parameter-sensitive information passed between executable programs through ASCII files should be written with maximum numerical precision to those files. This practice enhances the accuracy of finite-difference-based derivatives of model outputs with respect to adjustable parameters.
2. Commands to delete all files that are written by one executable program and read by another with the composite model batch file should be placed at the top of the batch or script file. Thus, if a program fails to run or encounters an error condition, the program that follows it will also encounter an error condition and, most importantly, the

program that writes files that PEST actually reads will also encounter an error condition because an expected input file is missing. This last program thus fails to write the overall model output file(s) that PEST expects (which PEST deletes itself before running the model). PEST will then terminate execution with an appropriate error message. If this precaution is not taken, an executable program that follows a failed program in the model batch/script file may read an old version of its input file instead of a file that should have been overwritten by the previous program. The outcome of such failure will be an incorrect report of zero sensitivity of certain model outputs with respect to certain model parameters.

## Model Input/Output

Where many observations are featured in the parameter-estimation process, manual production of instruction files is tedious and error prone. Programs provided with the Groundwater Data Utilities suite automate the generation of these files in many common parameter-estimation contexts—at the same time as they automate the naming of observations—and write measured values to the PEST Control File. See appendix 3 for further details.

## Prior Information

Prior information can be used in both regularization and nonregularization settings, though the latter is often not considered optimal (see, for example, Hill and Tiedeman, 2007, p. 261). In the regularization setting, it can encapsulate some or all of the Tikhonov constraints required to ensure parameter estimability. In this case, the weights assigned to prior-information equations are subject to global adjustment as PEST evaluates the strength with which regularization constraints must be applied in order to achieve (but not exceed) the level of model-to-measurement fit specified by the PHIMLIM regularization control variable. Where prior-information equations are assigned to different regularization groups, intergroup relative weighting adjustment can be achieved through appropriate setting of the IREGADJ regularization control variable (discussed previously).

Where highly parameterized inversion is done, prior-information equations that encapsulate Tikhonov constraints may number in the hundreds or even thousands. Where PEST input files are written by a graphical-user interface, prior-information equations will have been written by that interface. Alternatively, utility software supplied with PEST, and with the Groundwater Data Utilities suite, can add prior-information equations encapsulating one or a number of different regularization schema, to a PEST Control File.

## Regularization

If PEST is run in “regularization” mode (as specified in the first line of the Control File), the PEST Control File requires a “regularization” section, located at the end of the Control File. This section contains variables that control the use of Tikhonov Regularization in the calibration process. A few of these variables are now discussed.

### Target Measurement Objective Function

The PHIMLIM variable should be set in accordance with the expected level of model-to-measurement misfit given the current model conceptualization (see, for example, Doherty, 2003; and Fienen, Muffles, and Hunt, 2009); at the limit this represents the fit expected given the measurement noise associated with the observations. A companion variable is also required—PHIMACCEPT—which should be set between 1 percent and 5 percent higher than PHIMLIM.

Unfortunately, the expected fit to a calibration dataset is rarely known until the calibration process is complete. Furthermore, most of this misfit is actually due to structural noise—that is, model-generated errors arising from an inability of the model to replicate every nuance of the behavior of the system that it simulates (Moore and Doherty, 2005). In view of this, the following strategy can often be used to establish a suitable setting for PHIMLIM:

1. Make an initial PEST run with PHIMLIM set very low in order to determine the level of fit that PEST is capable of achieving. During this run, consider setting the FRACPHIM variable to 0.1 or 0.2. This supplies a temporary value for PHIMLIM, equal to this fraction of the current measurement objective function, for use at all stages of the parameter-estimation process.
2. If implementing SVD-Assisted parameter estimation, instruct PEST to write a sequence of base parameter value (\*.bpa) files; these record the best parameter values achieved during successive iterations of the parameter-estimation process.
3. Inspect the final calibrated parameter field. If signs of overfitting are apparent (such as unacceptable and unrealistic spatial variability of parameter values), set PHIMLIM to a value that is somewhat higher than the minimum measurement objective function attained through the parameter-estimation process; then repeat the calibration process with this setting.
4. If implementing SVD-Assisted parameter estimation, inspect the best-fit parameter fields (calculated on the basis of successive \*.bpa files) for signs of the onset of parameter field deterioration due to overfitting. The measurement objective function associated with this field gives a suitable lower bound for PHIMLIM.

5. If observation weights correspond to expected measurement error of each observation, set the PHIMLIM value equal to the number of nonzero weighted observations; this value may result in an acceptable balance between best fit and regularization constraints (Fienen, Muffles, and Hunt, 2009).

### Regularization Objective Function

The closer the total measurement objective function is to PHIMLIM, the stronger the regularization preferred conditions are expressed in the parameter-estimation process. Moreover, the closer the regularization objective function is to zero, the greater is the extent to which parameters, or relations between parameters, adhere to their default conditions. In PEST, Tikhonov regularization is implemented as a constrained optimization process (Doherty, 2003) in which the regularization objective function is minimized subject to the constraint that the target measurement objective function rise no higher than PHIMLIM (if a desired PHIMLIM can actually be achieved).

In practice, to reduce the model-run requirements of the regularized-inversion process, PEST ceases execution as soon as the measurement objective function falls below PHIMLIM. However, if the REGCONTINUE control variable is set to “continue,” PEST will continue the constrained optimization process until the regularization objective function is minimized (and the measurement objective function is exactly equal to PHIMLIM). This approach can mitigate the introduction of spurious heterogeneity to an estimated parameter field, but it may add considerably to overall PEST run time.

### Interregularization Weights Adjustment

Experience has demonstrated the importance of the IREGADJ variable in enhancing the numerical stability of highly parameterized inversion in which regularization is wholly or partly implemented through the use of Tikhonov constraints. Use of this variable results in (1) stronger enforcement of Tikhonov constraints on those parameters (and/or parameter combinations) for which information content of the calibration dataset is low and (2) weaker enforcement on those parameters (or parameter combinations) for which it is higher. This strategy ensures that if PEST needs to reduce the overall regularization-weight factor so that it can introduce heterogeneity where appropriate, then PEST’s ability to enforce Tikhonov constraints elsewhere within the model domain where data may be scarce is not compromised.

An IREGADJ setting of 1 (recommended), 2, or 3 allows regularization-weight adjustment to take place on a group-by-group basis. A setting of 4 or 5, which implements subspace enhancement of Tikhonov constraints, allows weights adjustment to take place on a constraint-by-constraint basis. At the time of writing, the suggested setting for subspace enhancement is 4; complementary control variables NOTREGADJ and REGWEIGHTRAT should then be set to 1 and 50 respectively. FRACPHIM should be set to 0.0 when implementing subspace-enhanced Tikhonov regularization.

## SVD-Assist

Because of the high computational efficiencies gained through estimation of superparameters in lieu of native model parameters (Tonkin and Doherty, 2005), SVD-Assist is commonly employed for inversion in the highly parameterized context. Its implementation requires that sensitivities of model outputs with respect to all base model parameters first be computed; this computation takes place on the basis of a traditional PEST dataset (which normally includes Tikhonov regularization) in which the NOPTMAX control variable is set to  $-1$  or  $-2$ . The PEST utility SVDAPREP is then run in order to create a PEST input dataset in which superparameters, rather than base parameters, are estimated. A number of issues can arise during implementation of SVD-Assisted parameter estimation; these issues plus suggestions for using SVDAPREP, are discussed below.

## Number of Superparameters

The SUPCALC utility provided with PEST assists the modeler in choosing an appropriate number of superparameters to estimate in any particular calibration context. However, experience has demonstrated that the number of superparameters suggested by SUPCALC is often too large because its computations fail to account for the contribution made to measurement uncertainty by structural noise of unknown covariance structure.

In practice, irrespective of the basis on which it is chosen, a modeler's choice of the number of superparameters to employ in the SVD-Assisted inversion process must be reviewed as that process proceeds. If that process shows signs of numerical instability (for example, an objective function that fluctuates rather than gradually falls in a monotonic fashion, a Marquardt Lambda that grows rather than diminishes, high condition numbers—greater than about 3,000—recorded in the \*.cnd file, and/or introduction of excessive heterogeneity to estimated parameter fields), then the number of estimated superparameters may need to be reduced. On the other hand, if condition numbers are low, and if a modeler feels that a better fit between model outputs and field data are reasonable, the number of estimated superparameters should be increased.

The following considerations also are salient:

1. The number of superparameters that can be estimated may depend on the computing resources available to a modeler, including the number of processors available over which model runs can be parallelized.
2. If computing resources do not present a barrier to use of a large number of superparameters, it is better to err on the side of too many rather than too few superparameters in order to compensate for model nonlinearity. Concomitant use of Tikhonov regularization, and use of truncated SVD as a superparameter solution mechanism in addition to the SVD-Assisted inversion process, can prevent numerical instability that would otherwise follow from the use of too many superparameters.

## Tikhonov Regularization

Use of Tikhonov regularization as part of the SVD-Assisted parameter-estimation process is not integral to the success of that process, provided that few enough superparameters are estimated for the inverse problem of model calibration to be well posed, or provided that truncated SVD is employed as a solution device for estimation of superparameters. However, use of Tikhonov regularization in the SVD-Assisted parameter-estimation process does bring with it the following advantages:

1. It helps to guarantee reasonableness of estimated parameter values.
2. Use of a properly chosen PHIMLIM target measurement objective function can prevent overfitting of model outputs to field measurements.
3. Simultaneous use of two types of regularization (that is, SVD-Assist and Tikhonov Regularization) affords extra protection against numerical instability. It also allows a modeler to use more superparameters than are uniquely estimable on the basis of a given calibration dataset. This, in turn, allows some superparameters to move in and out of the calibration-solution subspace as the parameter-estimation process progresses, thereby preventing degradation of that process through excessive model nonlinearity.

## Precalibration Sensitivities

As stated previously, before undertaking SVD-Assisted parameter estimation, a modeler must create a PEST input dataset on the basis of base model parameters. The following considerations are pertinent to creation of that PEST input dataset:

1. As for all regularized inversion, the initial values assigned to base parameters should be those of minimum precalibration error variance. In some cases these could be obtained through estimation of lumped/tied parameters on a global, layer-by-layer, or other basis.
2. Though not essential, it is advisable to add Tikhonov regularization to the base PEST Control File before, rather than after, calculation of pre-SVDA base parameter sensitivities. Doing so allows PEST to eliminate most of the computational burden associated with the first iteration of the forthcoming superparameter estimation exercise, because it is able to formulate superparameter sensitivities from base-parameter sensitivities by using information contained in this base-parameter sensitivity matrix, thereby obviating finite-difference computation of superparameter sensitivities.

Computation of sensitivities with respect to base parameters before starting SVD-Assisted parameter estimation can be the most time-consuming part of the entire highly parameterized inversion process. Consequently, if a modeler wishes

to alter some facet of a base parameter PEST Control File, the computational cost of Jacobian matrix recalculation for that file may be so heavy as to be a powerful disincentive to making such a change. The JCO2JCO utility alleviates this burden by computing a Jacobian matrix file (that is, a \*.jco file) for a new PEST Control File from that associated with an existing PEST Control File, provided that the new PEST Control File is altered from the first one in only the following ways:

1. No parameters are added to the existing PEST Control File; however, parameters in the existing PEST Control File can be fixed, tied, or removed from it.
2. No observations are added to the existing PEST Control File; however, observations in the existing pest Control File can be removed or assigned new weights (or observation covariance matrices).
3. Prior information can be added or removed from the existing PEST Control File.

Other utilities provided with PEST can supplement the use of JCO2JCO, thereby allowing the above restrictions to be overcome. See the PEST documentation for details.

## SVDAPREP Responses

The utility SVDAPREP is the easiest way to automatically convert to a SVD-Assist-based parameter-estimation process. When run at the DOS prompt to create a PEST input dataset for SVD-Assisted parameter estimation, SVDAPREP issues a series of questions to which the user must respond. If run through a graphical user interface, these responses must be provided through the associated dialogue box. SVDAPREP offers defaults for most responses, these being acceptable on most occasions. Hence, only two responses to SVDAPREP queries are discussed here.

As described in detail in Doherty (2010a), SVDAPREP offers four options for computing superparameters from base parameters. These are

1. SVD based on  $\mathbf{Q}^{1/2}\mathbf{X}$ ;
2. SVD based on  $\mathbf{X}^T\mathbf{Q}\mathbf{X}$ ;
3. LSQR without orthogonalization;
4. LSQR with orthogonalization.

The first of these options should be chosen unless the number of parameters is very large (exceeds 2,500), in which case the third or fourth (recommended) option should be selected, because LSQR decomposition of the weighted Jacobian matrix is quicker than SVD decomposition of this matrix but is not as accurate. LSQR variables should be set to the following: LSQR\_ATOL, LSQR\_BTOL, LSQR\_CONLIM and LSQR\_ITNLIM to  $1\text{e-}10$ ,  $1\text{e-}10$ ,  $5\text{e}3$  and  $5\text{e}4$  respectively. Selection of the first and second SVD options should lead to essentially identical superparameters. Where parameters outnumber observations, choice of the second option may lead to slightly faster computation of superparameters than choice of the first option. However, the difference in computation time is not great.

A later SVDAPREP option allows the modeler to automatically save interim results related to estimated parameters (\*.bpa files) and the associated model-to-measured residuals (\*.rei file). As described earlier, the optimal tradeoff between best fit and parameter reasonableness is difficult to know at the start of a parameter-estimation run. Being able to examine the interim parameter sets and associated fit as parameter estimation progresses is a valuable diagnostic tool to evaluate this tradeoff. Saving of both these interim results is invoked when the option "br" is returned to the SVDAPREP query: "Write multiple BPA, JCO, REI, none [b/j/r/n] files (<Enter> if "n")?"

## Enhancing SVD-Assist by Also Including SVD

When creating a PEST Control File for SVD-Assisted parameter estimation, SVDAPREP transfers as much information from the base PEST Control File to the new superparameter PEST Control File as is relevant to the latter file, including all parameter-estimation and regularization control variables. Moreover, if PEST is instructed to use truncated SVD as a solution mechanism for native model parameters in the base PEST Control File, it is instructed to use the same solution mechanism for superparameters in the new PEST Control File. Combining these two approaches has utility for calibrating highly parameterized groundwater models because it allows the modeler to carry a higher number of superparameters (thus enhancing calibration flexibility) than might be safely chosen in an SVD-Assist-only run because SVD constrains the number of SVD-Assist superparameters to only those that maintain numerical stability. As stated previously, this can be useful where a model is moderately to highly nonlinear. Estimated eigencomponents of the base parameter Jacobian matrix can then move in and out of the calibration solution space as the parameter-estimation process progresses.

The SVD section of the PEST Control File should be set so that the maximum number of singular values (MAXSING) is equal to the number of superparameters, so that the eigenvalue ratio (set through the EIGTHRESH variable) is used to calculate the number of superparameters that are actually estimated. As described previously, an EIGTHRESH setting between  $10^{-5}$  and  $10^{-7}$  is appropriate for most general SVD problems; this setting is also appropriate for adding SVD to an SVD-Assist PEST Control File. Use of a negative RLMFAC value, which invokes a wide-ranging Marquardt Lambda search, can also enhance the ability of the SVD/SVD-Assisted parameter-estimation process to lower the measurement objective function, especially in contexts where model numerical imperfections may introduce noise to finite-difference-based derivatives.

## Parallel Processing

Because adding parameters is directly related to the parameter estimation computational effort, model-run parallelization can add enormous efficiency to the calibration of highly parameterized models. When used in conjunction with SVD-Assist, hundreds—or even thousands—of parameters can be included in the model-calibration process, even when the model run times are moderate to long. With the increasing prevalence of multiprocessor desktop computers, clustered computing environments, and cloud computing (Luchette and others, 2009; Hunt and others, 2010), parallelization of model runs during the calibration process is likely to become the norm rather than the exception. Where parallelization opportunities are available to modelers through use of multiple personal computers, PEST-based parallelization of model runs is a relatively simple procedure. PEST can parallelize both the undertaking of model runs required to fill the Jacobian matrix, and the testing of parameter upgrades computed on the basis of different Marquardt Lambdas.

Model runs are made by a series of subordinate machines or processors under the control of a PEST master machine that manages the parameter-estimation process. PEST accesses the parallel-computing environment through three mechanisms. The oldest is the parallel PEST (PPEST) approach that uses small message files to handle communications between processors. Only one additional run management file (\*.rmf) is needed to run PPEST; it contains parallelization information such as the working/run directory location associated with each processor. All other aspects of the PEST input are identical to that employed in a serial PEST run on a single machine. Recently, both a message-passing interface (MPI) and TCP/IP approach have been developed for PEST (BeoPEST: Schreuder, 2009; Hunt and others, 2010). With these approaches, a run management file is not required but is optional; when a run management file is not specified, parallel information is specified in the command used to start BeoPEST. See Schreuder (2009) and Doherty (2010c) for detailed instructions on using these advanced parallel-processing capabilities.

## Marquardt Lambda

PEST will attempt to parallelize the testing of parameter upgrades based on different Marquardt Lambdas if the PARLAM variable in the Parallel PEST run management file is set to 1. It will initiate as many model runs as there are processors available, up to a maximum of 20, provided that those runs will be completed in roughly the same time. This can result in some wastage of model runs because the Marquardt Lambda testing procedure is inherently serial; nevertheless, wasted runs are of little significance if processors are otherwise standing idle. If PARLAM is set to a negative number, PEST limits the number of model runs constituting any Marquardt Lambda testing procedure to the absolute value of the number supplied.

A value of  $-5$  is suggested for PARLAM, but a higher setting can be used if many processors are available.

Unless PEST is undertaking SVD-Assisted parameter estimation, if one or more parameters encounter their upper or lower bounds during any iteration of the parameter-estimation process, the Marquardt Lambda parallelization process is temporarily abandoned: the lambda testing procedure needed for a parameter upgrade must then be run as a serial process because PEST sequentially fixes parameters at their bounds. This approach to reducing the dimensionality of the parameter-estimation problem results in enhanced performance when parameters are tightly bounded. However, this artifact limits the efficiency of the parameter-estimation process in a parallel-computing environment because processors are forced to stand idle until the serial lambda-testing procedure is complete, which can negate the advantage of parallel processing when model run times are long.

A PARLAM setting of  $-9999$  overcomes this problem. In this case, PEST abandons its serialization of lambda-based parameter upgrade testing, even if one or more parameters have encountered their bounds. Instead, it computes parameter upgrades on the basis of NUMLAM Marquardt Lambdas (NUMLAM is specified in the “control data” section of the PEST Control File) and parallelizes the search by concurrently testing the specified number of upgrades. Here it is assumed that NUMLAM is equal to, or less than, the number of processors to which Parallel PEST has access; Parallel PEST will reduce it to this number if this is not the case. Furthermore, it only does one round of lambda testing on these processors, accepting whatever set of parameters results in the lowest objective function at the end of that round. It then moves on to the next optimization iteration, where all available processors are reengaged for finite-difference-derivatives computation to fill the Jacobian Matrix. Where many processors are available to compute these derivatives, the need for a few extra optimization iterations to compensate for a diminished efficiency of the Marquardt Lambda testing procedure is usually worth the cost.

## Model-Run Repetition

If the RUNREPEAT variable is set to 1 or is absent from the run management file, Parallel PEST will not accept that a model run has failed in its execution until it has attempted three repeats of that model run. This ensures that temporary network communications problems do not halt the entire parameter-estimation process. Although this setting can help guard against network-induced errors, it can also prove problematical during the testing phase of a new Parallel PEST inversion run, where model failure is more likely to be the source of the problem than network failure. During this testing phase, a modeler would benefit from knowing that a model error has arisen as soon as it occurs, rather than waiting until three unsuccessful attempts to rerun the model are completed. It is therefore recommended that RUNREPEAT be set to zero during the initial testing phase of Parallel PEST setup.

## Just Before Running PEST: Checking

Before running PEST—either for serial parameter estimation on a single processor or in parallel on many processors—the PESTCHEK utility should be used to check the entire PEST input dataset. Only after PESTCHEK returns an error-free response should PEST be run. The TEMPCHEK and INSCHEK utilities also can be employed to restrict checking to individual template and instruction files, respectively. PESTCHEK, however, checks the PEST Control File, together with all template and instruction files cited therein, for correctness, completeness, and consistency.

## Running PEST

PEST can be run through the GUI for the model or from the command line of the terminal (UNIX/Linux/Macintosh) or DOS box (Windows) by typing PEST followed by the name of the PEST Control File. For brevity, the following discussion will assume that PEST is being run on a Windows platform in a DOS box, though the concepts and capabilities are available on the other platforms as well. When run from the DOS box, all pertinent files and utilities must be located in a directory cited in the PATH environment variable or reside in the PEST working directory for PEST to function properly. In the case of parallel processing, having all files in the PEST working directory creates a portable directory that ensures each processor used in the run will have all necessary files regardless of the PATH specified on the local processor. Regardless of how initiated, when running PEST it may become clear that user intervention is required before the normal termination of the PEST run. If this occurs, the following options are available.

## Stopping PEST

Terminating PEST execution by using the Ctl-C command is a brutal way of accomplishing this task. It can result in problematical model behavior on the next occasion that the model is run, because the operating system may believe that certain of its input or output files are still open. Termination by using Ctl-C also results in little or no output useful for diagnosing the cause of underlying problems.

A better way to stop PEST is to open another command box in the PEST working directory and type PSTOP or PSTOPST at the command prompt. PEST will then cease execution on completion of the current model run. If parallel PEST is being used rather than serial PEST, PEST will cease execution immediately on reception of the PSTOP or PSTOPST command. However, all model instances that are currently running will run to completion. If the PSTOP command is employed, the various processors will remain active on completion of the current model run and will thus be able to detect a Parallel PEST restart without additional

user intervention. In contrast, if the PSTOPST command is employed, all processors will terminate on completion of their current model run and will need to be restarted by the user before re-launching parallel PEST.

Irrespective of whether the serial or parallel version of PEST is running, if the PSTOPST command rather than the PSTOP command is used to terminate PEST execution, PEST will calculate parameter and observation statistics and write them to pertinent output files before ceasing execution. In most circumstances it will also make one final model run on the basis of best parameters achieved up until that point. This last run will not occur when SVD-Assisted parameter estimation is being done, as discussed below.

## Restarting PEST

Various switches can be used for restarting a previously stopped PEST run; these switches are invoked at the end of the command line used to start PEST. The “/r” switch restarts PEST at the beginning of the iteration in which its execution was previously halted. The “/j” switch restarts PEST at that point where computation of the Jacobian matrix was last completed; this is the appropriate switch to employ if undertaking manual intervention for temporarily holding troublesome parameters.

The “/d” and “/s” switches accomplish the same role; however, the latter must be used with Parallel PEST, whereas the former must be used with serial PEST. If PEST execution had previously been terminated while it was filling the Jacobian matrix, use of either of these switches will cause PEST to restart at exactly the same model run at which its execution was previously interrupted, thus preserving elements of the Jacobian matrix already calculated.

If started with the “/i” switch, PEST prompts for the name of a Jacobian matrix file (\*.jco file). It will use derivatives from this file, rather than those that it calculates itself, during its first optimization iteration.

## Pausing PEST

PEST execution can be paused by typing PPAUSE in another command-line window which is open to the PEST working directory. Parallel PEST will pause execution immediately; all processors will pause once they have finished their current model run. In contrast, serial PEST will pause execution on completion of its current model run. In both cases, a modeler can then import model input or output files into appropriate visualization or plotting programs for display of interim calibration results. Conflicts in access to model input/output files by the model on the one hand, and by visualization/display software on the other hand, are thereby avoided. A paused PEST run can be restarted by typing PUNPAUSE in the same command line box as that from which the PPAUSE command was issued.

## Monitoring PEST Performance

This section discusses how a modeler can monitor the performance of the parameter-estimation process as PEST runs and how problematical behavior can be recognized and remedied. The first subsection of this section pertains to classical parameter estimation, in which it is assumed that PEST has been provided with a well-posed (sparsely parameterized) inverse problem. It is also assumed that no regularization has been employed in the parameter-estimation process. In this context monitoring of PEST performance focuses on ensuring that the inverse problem is indeed well posed. Although such attention to well-posedness is less critical when regularized inversion is properly implemented, it is nevertheless an important concern in the regularized inversion context because the onset of ill-posedness indicates a breakdown in the user-supplied regularization mechanism. Other problems that can afflict PEST in the classical parameter-estimation setting, such as problems associated with bad derivatives, are also problematic for regularized inversion of highly parameterized models. Hence, much of the discussion in the first of the following subsections is relevant to highly parameterized inversion as well.

### Classical Calibration of Sparsely Parameterized Models

#### Well-Posedness

The following are signs that the parameter estimation problem may not be well posed:

1. The Marquardt Lambda rises rather than falls as the parameter-estimation process progresses.
2. Condition numbers recorded in the “condition number file” (that is, the *\*.cnd* file) are greater than about 3,000.
3. The reported “maximum factor change” or “maximum relative change” undergone by any parameter during a particular optimization iteration (particularly later iterations) is equal to the global parameter factor or relative change limit FACPARMAX or RELPARMAX. Maximum factor and relative parameter changes are written both to the screen and to the run record file at the end of each optimization iteration.
4. If the IEIG variable in the “control data” section of the PEST Control File is set to 1, PEST records eigenvalues and eigenvectors of the current parameter covariance matrix to its “matrix file” (that is, *\*.mtt* file) at the end of every optimization iteration. If PEST reports that it cannot invert the “normal matrix” to compute the current parameter covariance matrix, or if the ratio of highest to lowest eigenvalue of this matrix is greater than about  $10^7$ , the inverse problem is ill conditioned. At best, values for at

least some parameters will be highly uncertain. At worst, PEST’s progress in lowering the objective function will have been compromised.

5. If the ratio of highest to lowest composite parameter sensitivity recorded in the “composite parameter sensitivity file” (that is, *\*.sen* file) is greater than about 100, this too indicates that the current parameter estimation problem may be so ill-conditioned as to hamper PEST’s ability to lower the objective function.

### Identifying Troublesome Parameters

Troublesome parameters are usually those that have little or no sensitivity to observations contained in the calibration dataset and/or those that are highly correlated with other parameters. Sensitivities are listed in the sensitivity (*\*.sen*) file; correlation information is listed in both the run record (*\*.rec*) file and in the matrix (*\*.mtt*) file, updated during every iteration of the parameter-estimation process. If eigenvalues and eigenvectors cannot be computed because of ill-posedness, then parameters identified as having smallest composite sensitivity are good candidates for being deemed “troublesome.” So too are parameters that have been identified as having undergone the greatest relative or factor change during a recent optimization iteration, with the exception of the beginning of the parameter-estimation process when highly estimable parameters are expected to change appreciably.

### Accommodating Troublesome Parameters

Ideally, troublesome parameters should be fixed, or tied to other parameters, and the parameter-estimation process restarted, because better estimates for these parameters can probably be supplied by the modeler than by the parameter-estimation process. Alternatively, the parameter estimation process can be stopped and restarted from the point at which computation of the Jacobian matrix was most recently completed by using the “/j” switch. Upon restarting, PEST can be instructed to temporarily hold troublesome parameters at their current values by using a user-prepared “parameter hold file” (see Doherty, 2010a, for more details).

To some extent, problems caused by troublesome parameters can also be addressed by using PEST’s “automatic user intervention” functionality, which is implemented by using an “*au*” flag in the “control data” section of the PEST Control File. This gives PEST the ability to make substantial progress in lowering the objective function, even where it has been presented with an ill-posed inverse problem by sequentially (and temporarily) removing troublesome parameters from the parameter-estimation process. Although this brute-force approach can often achieve more successful parameter-estimation results, it requires many more model runs than a non-*au* PEST run. Moreover, a theoretically better means of

accommodating ill-posedness is to employ one of the devices for mathematical regularization described previously, for these are more likely to lead to a solution to an ill-posed inverse problem that approaches minimum error variance than that obtained through automatic user intervention. Nevertheless, automatic user intervention can prove useful in allowing PEST to traverse objective-function surfaces that are pitted with local minima (owing to poor model design or numerical malperformance; see Kavetski and others, 2006).

## Identifying Bad Derivatives

Ideally, as the parameter-estimation process progresses, the objective function should fall (often sharply at first if no precalibration parameter adjustment was done) and then gradually flatten out in a monotonic fashion. This should occur until the objective function falls no further, causing PEST to terminate the parameter-estimation process because improvements in the objective function are too small to justify continuing. A sudden interruption in this descent after one or two iterations is an indication that model malperformance is hampering PEST's ability to lower the objective function. Instead of lowering, the objective function may climb or bounce between higher and lower values during subsequent optimization iterations. The objective function may also show dramatic variability as different Marquardt Lambda values are tested during each iteration. Although an ill-posed problem can also show these signs, such phenomena are expected from errors in the finite-difference computed derivatives used to calculate the Jacobian matrix (fig. 1). Bad-derivative artifacts can be identified through ascertaining whether the ratio of highest to lowest eigenvalue of the current parameter covariance matrix (recorded in the \*.mtt file) is not unduly high (for example, less than  $10^7$ ), with the same applying to condition numbers recorded in the \*.cnd file (for example less than 3000). If these ratios are not unduly high, ill-posedness is not likely, and poor PEST performance most likely stems from errors in finite-difference-computed derivatives. The derivatives of user-specified parameters can be formally tested by using the JACTEST utility described below.

## Accommodating Bad Derivatives

Experience has demonstrated that the Gauss-Marquardt-Levenberg method of parameter estimation is somewhat robust in the face of poor model numerical performance. Nevertheless, there will always be an upper limit to which bad derivatives can be accommodated, because a Jacobian (sensitivity) matrix is a fundamental component of the Gauss-Marquardt-Levenberg algorithm on which PEST is based. Where numerical malperformance is such that finite-difference derivatives have little or no utility, then a global optimizer such as CMAES\_P or SCEUA\_P (see below) must be used in place of a sensitivity-based method. However, use of global methodologies comes at a high cost in terms of model runs,

with the model-run burden increasing enormously as the number of estimated parameters increases. Furthermore, global methodologies cannot be used efficiently in conjunction with either subspace or Tikhonov regularization schemes as currently implemented in PEST.

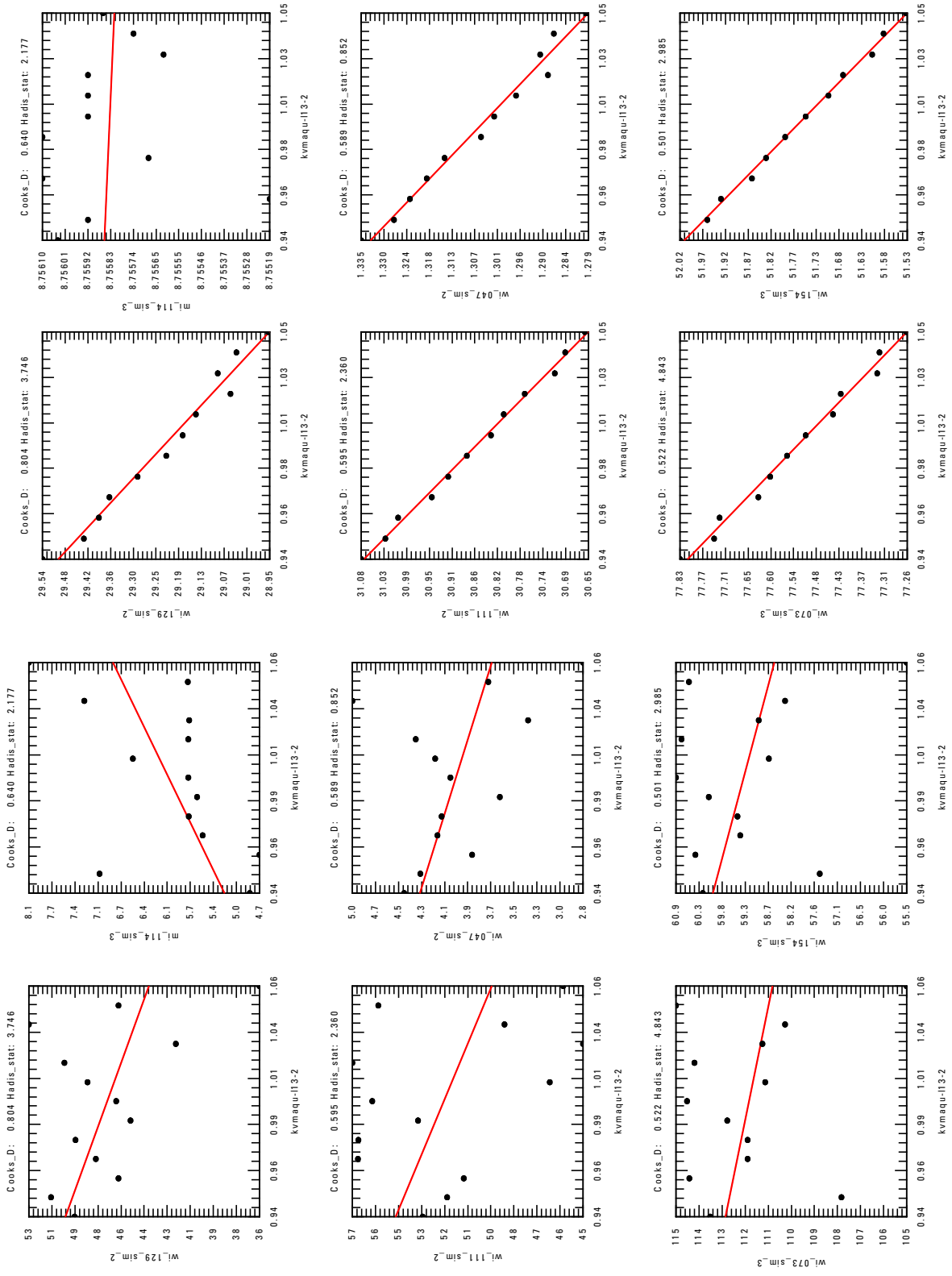
If model numerical malperformance is bad enough to erode the quality of finite-difference-based derivatives but not so bad as to completely undermine their integrity, then the effect of inferior derivatives on the parameter-estimation process can be mitigated to some degree through the use of either or both of two strategies offered by PEST. These strategies are "split slope analysis" and a variant of the automatic user-intervention strategy discussed previously. Both are costly in terms of their model-run requirements but are much less costly than the use of global optimizers.

If requested by the user, split slope analysis is implemented by PEST whenever two model runs, rather than a single model run, are devoted to the calculation of derivatives of model outputs with respect to adjustable parameters by using finite differences. For the first of these runs the parameter is incremented by a small amount; it is then decremented by the same amount for the second model run. For each model output that corresponds to an observation, two slopes are available for comparison. If these are markedly different, then at least one of these slopes is likely to have been corrupted as a result of a bad derivatives calculation associated with model output granularity. Normally, the derivatives are best computed by using the three points (original parameter value and decremented and incremented values); for example, by fitting of a quadratic to these points and obtaining analytical derivatives of the quadratic at current parameter values. However, when split slope analysis indicates potentially bad derivatives the smaller slope is taken as an approximation to local derivatives. Alternatively, a slope of zero can be assigned instead indicating that no change is warranted because none can be trusted; it is the user's choice.

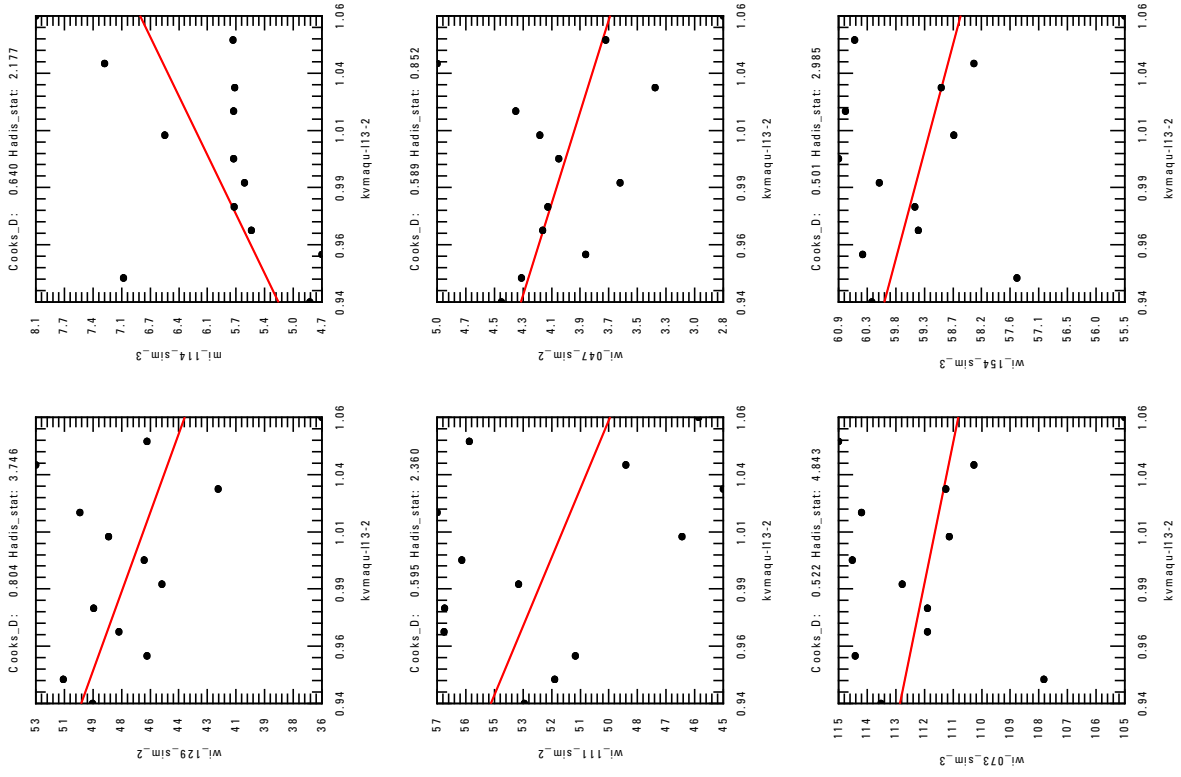
PEST's split slope analysis is implemented according to settings supplied in the "parameter groups" section of the PEST Control File. Three variables (SPLITTHRESH, SPLITRELDIFF, and SPLITACTION) are optional; however, if one of them is supplied for a particular parameter group, then all of them must be supplied for that group. The first two variables are absolute and relative thresholds for invoking the split-slope capability; the last variable specifies the action taken when a slope ratio exceeds a user-supplied threshold. Three actions are available: (1) take the smaller of the two slopes, (2) set the slope equal to zero for the current iteration, and (3) set the slope equal to that calculated in the previous iteration. Doherty (2010b) describes these optional variables in more detail.

The special case of automatic user intervention (au) discussed previously can also be invoked by the user to mitigate the adverse effects of bad derivative calculation. Setting the DOAUI variable in the PEST Control File to "au" alters the operation of the automatic user intervention so that *sensitive*, rather than *insensitive*, parameters are selectively removed from the parameter-estimation process. The use of the au

## b) Tight solver closure



## a) Loose solver closure



**Figure 1.** An example of JACTEST-calculated model outputs (y-axis) resulting from small sequential parameter perturbation (x-axis). In this case bad derivatives result from loose solver-closure criteria (modified from Feinstein and others, 2008).

option involves the assumption that any apparent extraordinary sensitivity is evidence of miscalculation of finite-difference derivatives. For the current iteration, PEST will then sequentially eliminate parameters with unduly high sensitivities from the parameter-estimation process. This temporary removal allows the user to test whether an effective lowering in the objective function can be achieved by removing parameters suspected of having bad derivatives.

## Phi Gradient Zero

If PEST ceases execution with the message “phi gradient zero,” it means that the updated objective function exactly equals the previous objective function. This is an unlikely occurrence, thus indicating that the model is incorrectly set up or that there are issues with the setup of PEST’s communication with the model. Most commonly, PEST has been told to create a model input file (in the “model input/output” section of the PEST Control File) that is not actually called by the model. As a result, although PEST may write to one model input file before each model run, the model actually reads another (non-updated) input file. Given the file does not change from model run to model run, zero sensitivity of all parameters associated with that particular input file results; if this is the only model input file created by PEST, it will report “phi gradient zero.” If there is more than one model input file listed in the “model input/output” section of the PEST Control File and not all are incorrectly named, then a “phi gradient zero” condition will not be reported (because at least some of the parameters will be changed from run to run). However, this condition can still be recognized by the occurrence of zero composite sensitivities for some parameters in the composite parameter sensitivity (\*.sen) file. Zero-valued composite observation sensitivities as recorded in the composite observation sensitivity file (that is, the \*.seo file) also can provide evidence that a model input file has been misnamed in the PEST Control File.

The presence of prior information in a PEST Control File (as is often employed for implementation of Tikhonov regularization) can prevent evidence of parameter insensitivity from being expressed in the \*.sen file in the manner discussed previously, because prior information creates sensitivity for any parameter that it features (this is its role, after all). In many circumstances, problems of this kind can still be recognized. If all adjustable parameters are featured in at least one prior-information equation (as is typical where Tikhonov regularization is employed), then the occurrence of wrongly named model input files can often be recognized through relative *uniformity* of composite parameter sensitivities in place of the higher degree of natural variability that is more common where many parameters are featured in the parameter-estimation process.

In some instances, a “phi gradient zero” outcome can arise where the model run by PEST consists of a series of executable programs encapsulated in a batch file, and one of these programs either fails to run or does not trap a bad result forthcoming from a previous program. In this case, the model input file is not changed because of an error in an intermediate program,

and an older version of the file residing in the run directory is used instead. As discussed previously, this situation can be avoided by deleting intermediate files through which model components cited in a batch process pass information to each other before subsequent components are run. This is achieved by placing deletion commands for all intermediate files at the beginning of the batch file.

## Regularized Inversion of Highly Parameterized Problems

In many respects, the monitoring of PEST’s performance when undertaking regularized inversion, whether regularization is effected through truncated SVD, Tikhonov regularization, SVD-Assist, or any combination of these, is similar to the monitoring of its performance when undertaking classical, overdetermined parameter estimation. In both cases, a primary concern is the detection of signs that the inversion process is failing due to a requirement that too many parameters require estimation based on a calibration dataset of limited information content. However, in the case of regularized inversion (which is specifically designed to accommodate such a situation), this is an indication that the chosen regularization method is inappropriate for the problem, is not as effective as was originally intended, or has failed as the parameter-estimation process has progressed.

When classical parameter estimation fails, the modeler has no choice but to remove/fix/tie parameters that are suspected of being troublesome. Such inflexible measures are not required where parameter estimation is based on regularized inversion; in this case, the remedy lies in adjusting the regularization strategy so that it is able to better accommodate the difficulties encountered in the current parameter estimation context.

## Signs of Regularization Failure

Signs of regularization failure are similar to the signs that appear when classical parameter estimation faces problems due to attempted solution of an ill-posed inverse problem. They include the following:

1. Rising, rather than falling, values of the Marquardt Lambda as the parameter-estimation process progresses.
2. Condition numbers (recorded in the \*.cnd file) that are greater than about 3,000;
3. Parameter estimates that change considerably or abruptly from iteration to iteration.
4. Unrealistic and extreme parameter estimates.
5. A (measurement) objective function trajectory that oscillates from iteration to iteration, rather than falling or flattening.

In relation to the second of the above points, a condition number (\*.cnd) file is not available when singular value decomposition is specified for the solution mechanism. In that case, the condition number is preset to the inverse square root of the user-supplied EIGTHRESH variable, if this variable (rather than the MAXSING variable) is used to determine the singular-value truncation point. However, if the truncation point is determined by the MAXSING variable, condition numbers associated with the current problem can be determined in the following manner:

1. Open the singular value decomposition (\*.svd) file written by PEST; this is updated on every occasion that a parameter upgrade is tested.
2. From the list of singular values provided in that file, compute the ratio of the lowest pre-truncation singular value to that of the highest singular value.
3. The square root of the inverse of this number is the condition number associated with the current inverse problem.

### Rectifying Problems in Regularized Inversion: Tikhonov Regularization

If Tikhonov regularization is being employed, an obvious mechanism for rectifying problem ill-posedness is to increase the target measurement objective function (PHIMLIM) so that regularization constraints are more strongly enforced. This will also reduce the probability of introducing spurious heterogeneity into the estimated parameter field as a result of attempting to achieve too good a fit between model outputs and field data.

In many cases, however, it may appear that the Tikhonov regularization process has failed before a model-to-measurement fit that is worthy of the data (and the model) has been gained. This arises from difficulties involved in applying Tikhonov constraints strongly in some parts of the model domain, where data density is low, while simultaneously releasing their application in other parts of the model domain, where data density is high and indicative of local heterogeneity. This problem can be ameliorated through allowing PEST to vary the strength with which regularization constraints are applied, either on a regularization group by regularization group basis (through setting the IREGADJ control variable to 1, 2 or 3) or through subspace enhancement of the Tikhonov regularization process (through setting the IREGADJ control variable to 4 or 5).

### Rectifying Problems in Regularized Inversion: Subspace Regularization

When undertaking SVD-Assisted parameter estimation, ill-posedness of the inverse problem can be rectified by reducing the number of superparameters for which estimation is being attempted. This can be achieved by rerunning the SVDAPREP utility to build a new superparameter-based PEST

Control File; superparameters of low rank (high parameter number) can also be directly removed from the estimation process by the user by changing the status of a subset of the existing superparameters to “fixed” in the “parameter data” section of the PEST Control File. However, as described previously, using SVD on an SVD-Assisted run can also be used to regularize the inverse problem by flexibly limiting the number of estimated superparameters to those supported by the calibration dataset on an iteration-by-iteration basis. When SVD is employed (whether or not in the context of SVD-Assisted parameter estimation), the number of singular values is decreased and numerical stability is enhanced by increasing the value of the EIGTHRESH control variable.

In a previous section it was suggested that an optimal strategy for calibrating highly parameterized groundwater models is SVD-Assisted model calibration accompanied by the use of Tikhonov regularization, with SVD employed as a solution mechanism for superparameters. In this case, a modeler has the ability to employ any one of the three mechanisms discussed in previous paragraphs to obtain better behavior of the regularized inversion process. Indeed, this solution flexibility constitutes a strong case for using all of these three methods in concert. However, it was also stated previously that use of a higher number of superparameters affords SVD-Assisted parameter estimation protection from adverse effects of model nonlinearity. Hence, when all three of these regularization mechanisms are employed, it is recommended that the Tikhonov regularization component be adjusted first to obtain improvements to a regularized inversion process. If this adjustment does not achieve the desired result, the SVD EIGTHRESH variable should be raised. In the event that problems still persist, the number of estimated superparameters can be manually reduced.

## Final Model Run

### Automatic Final Model Run

Unless PEST is conducting SVD-Assisted parameter estimation, it calls a final model run based on optimized parameters once the parameter-estimation process has reached completion. Thus, when PEST terminates execution, model input files will contain optimized parameters, and model output files will contain model-generated quantities computed on the basis of optimized parameters.

### Manual Final Model Run

When PEST is implementing SVD-Assisted parameter estimation, it cannot make such a final model run because the link between optimized superparameters and corresponding base parameters may have been lost in iterations since the optimized objective function was achieved. The user must therefore do this final model run manually by using the approach below:

1. When PEST does SVD-Assisted parameter estimation, the best-fit base parameter values are recorded in a “base parameter value” file (that is, a \*.bpa file). Note that the filename of a \*.bpa file is derived from the base parameter PEST Control File from which the SVD-Assist dataset was constructed by using SVDAPREP. The PARREP (“PARAmeter REPlace”) utility can be used to build a new base parameter PEST Control File that is identical to the existing one except for the replacement of initial parameter values with optimal values read from the \*.bpa file.
2. The user should set NOPTMAX to zero in the newly created base parameter PEST Control File.
3. The user can then run PEST, based on the new PEST Control File.

For non-SVD-Assisted parameter estimation, best-fit parameter values are recorded in a parameter value file (that is, \*.par file); this has the same filename base as that of the PEST Control File on which the parameter-estimation process is based. The procedure for undertaking a single model run based on optimized parameter values is identical to that described above, except for use of a \*.par file instead of a \*.bpa file.

## Evaluation of Results

The process of model calibration normally involves repeated PEST runs, with the strategy for one particular run often being designed to overcome problems encountered on the previous PEST run. Highly parameterized inversion, as implemented by PEST and its utility support software, offers many options for the design of regularization methodology that is optimal for a particular context. Options include

1. the manner in which a preferred parameter condition is formulated;
2. the manner in which deviations from that condition should arise through the calibration process;
3. the strength with which regularization constraints are enforced, both globally and locally, within the model domain.

All of these options have been discussed in previous sections. Given the availability of these options, it is most likely that formulation of a suitable inverse problem, optimized for use in a particular modeling setting, can be achieved relatively rapidly. It must be pointed out, however, that at the time of writing, the application of regularized inversion to the calibration of groundwater models is still a relatively young endeavor. It is likely, therefore, that further research will result in additional options over time. This will strengthen the use of models in groundwater management by allowing them to make more complete use of both site data and geological expertise. Several metrics can be used to evaluate the outcomes of a highly parameterized model-calibration exercise, and suggested subsequent steps, are now presented.

## Level of Fit

### Poorer Than Expected Fit

A question that must always be asked when assessing the outcomes of a highly parameterized inversion exercise is whether PEST has lowered the objective function to the extent possible for the current conceptual model. Although successful regularized inversion requires reasonable estimated parameters as well as the achievement of a good fit, the degree of fit attained should not be an artifact of numerical malperformance of PEST, or of the hydrologic model, or of inadequacies of the model as a simulator of subsurface processes within a study area, or of failure of the modeler to design a parameterization scheme that is adequately responsive to information contained within the calibration dataset.

If consideration is limited to factors affecting PEST performance, any of the following may explain a disappointing outcome of the parameter-estimation process (in terms of model-to-measurement fit):

1. Too high a setting for the PHIMLIM regularization control variable (if using Tikhonov regularization).
2. Use of too few superparameters if using SVD-Assist.
3. Too high a value supplied for EIGHTHRESH, or too low a value supplied for MAXSING (if employing SVD as a solution device for the inverse problem).
4. Problematical derivatives.
5. Numerical instability encountered through loss of Tikhonov constraints as too good a fit is sought.

To some extent the first and last of these factors are contradictory. However, as has been discussed previously, use of a Tikhonov scheme that fails to allow heterogeneity to emerge where data density is high (and is indicative of such heterogeneity) at the same time as it exercises heavy constraints where data density is low can bestow ill-posedness on an inverse problem before a justifiable fit has been attained and, hence, before all information of relevance to the model parameterization process has been extracted from the calibration dataset.

Means through which a better fit between model outputs and field measurements can be sought on subsequent PEST runs include the following:

1. Lower the PHIMLIM value specified.
2. Introduce greater flexibility to Tikhonov regularization through use of a suitable IREGADJ setting.
3. Use a greater number of superparameters (SVD-Assist).
4. Use a lower EIGHTHRESH value (SVD).
5. Introduce more parameters (for example pilot-point parameters) to areas of the model domain where fits need to be improved.

If, after all of these measures have been taken, the level of model-to-measurement fit is still unsatisfactory, and if poor derivatives as a result of model numerical malperformance can be ruled out, shortcomings in the conceptual model are likely. These shortcomings can occur as a result of its incapacity to simulate important processes because of issues with discretizing space and/or time; alternatively, it can result from misspecification of other aspects of model conceptualization. In the latter case, the model design must be revisited. However, because parameter flexibility is one of the strengths of highly parameterized inversion over classical methods of parameter estimation, failure to find a set of parameters that allows a model to replicate historical system behavior can be more readily attributed to model inadequacies than to parameterization inadequacies. Such insight can enhance early detection of conceptual problems, thus facilitating their early amelioration.

## Too Good a Fit

Use of large numbers of parameters in the calibration process, unaccompanied by a suitable regularization strategy, can readily lead to “overfitting” of model outputs to field measurements such that the fit is unreasonable for the system modeled. Fortunately, it is relatively easy to adjust a regularization scheme to lessen the degree of fit achieved during calibration. Mechanisms for achieving this (for example, raising the value of PHIMLIM) have already been discussed but are revisited here in the context of assessing estimated parameter values.

## Parameter Values

### Unreasonable Parameter Values

In addition to evaluating how well the model simulates observed data, one of the outcomes of a model-calibration process (whether highly parameterized calibration or classical calibration) is an assessment of whether the optimal parameters are reasonable and representative. To make this assessment, an evaluation of the level of noise associated with the calibration dataset is required. Much of this noise is structural noise resulting from simplifications to the real world required by the model, which in turn result in an inability to simulate nuances of behavior to which elements of the calibration dataset pertain. Where parameter estimation is done through classical means, a “reference variance” can be calculated from the minimized objective function  $\sigma_{min}$  as

$$\sigma_r^2 = \sigma_{min}^2 / (n - m)$$

where  $n$  is the number of observations making up the calibration dataset and  $m$  is the number of estimated parameters. On the presumption that weights are inversely proportional to measurement-error standard deviations (or the squares of these, depending on the formulation), the constant of proportionality between expected and achieved fit is thereby estimated.

Unfortunately, a reference variance cannot be computed in the same way in the highly parameterized context, for minimization of the measurement objective function will probably result in overfitting. Moore and Doherty (2006) show that an assessment of the level of noise associated with the calibration dataset cannot be made without a concomitant assessment of the degree of heterogeneity introduced to the estimated parameter field through the calibration process. If an estimated parameter field that shows a high degree of spatial variability is acceptable, then the resulting high level of fit between model outputs and field measurements will result in a low objective function; acceptance of the estimated parameter field therefore implies assumption of a low level of measurement/structural noise associated with the calibration dataset. Conversely, if even a moderate level of spatial parameter variability is considered unacceptable, then model-to-measurement fit must be sacrificed, this being an admission that the reference variance (and hence the amount of noise associated with the calibration dataset) is relatively high.

From this discussion it can be seen that evaluation of the outcomes of highly parameterized inversion must focus to a large extent on the modeler’s evaluation of the parameter field that arises from the calibration process. If a parameter field is judged to be unreasonable, measures discussed in the following subsections can be employed.

## Reduction of the Level of Fit

Parameter fields that adhere too closely to preferred conditions as encapsulated in initial parameter values and/or in Tikhonov-regularization constraints can be estimated through any (or all) of the following means. These will inevitably result in an increase in the measurement objective function realized through the regularized inversion process.

1. Increasing the value of the target measurement objective function PHIMLIM.
2. Setting the REGCONTINUE regularization control variable to “continue.”
3. Decreasing the number of estimated superparameters.
4. Increasing the EIGTHRESH SVD control variable.

## Local Aberrations in Parameter Fields

Highly parameterized inversion may lead to parameter fields that are generally satisfactory but that locally exhibit parametric aberrations such as “bullseyes” around pilot points or extreme parameter values abutting certain model boundaries. In some instances, as Doherty, Fienen, and Hunt (2010) point out, these aberrations are a direct outcome of employing too few parameters in the inversion process; in this case they can be rectified through introducing more parameters to that process—especially in areas where certain observations within the calibration dataset suggest the existence of property heterogeneity.

In other contexts, spurious values for estimated parameters may suggest to a modeler that property heterogeneity does indeed exist within a model domain; however, the modeler may not approve of the manner in which this heterogeneity is expressed in the calibrated parameter field. For example, a laterally continuous conductive fault at 45 degrees to the direction of ambient groundwater flow may be represented in a pilot-point-based estimated parameter field as a string of local “bullseyes” of extreme pilot-point values that reflects the local effects on the flow system due to this feature. However, inclusion of such a fault explicitly in the calibrated parameter field may not have been possible (at least in initial PEST runs), because a modeler may not have been aware of its existence or location. In a case such as this, the use of highly parameterized inversion may have served the calibration process by simply alerting the modeler to the existence of the conductive feature, though the information content of the calibration dataset may have been insufficient to do much more than this. Skilful interpretation of the estimated parameter field may then suggest to the modeler a modified parameterization scheme that expresses inferred heterogeneity in a more meaningful geological context. For example, a modified scheme might employ an anisotropic covariance matrix in conjunction with preferred value Tikhonov constraints, thus promulgating continuity of emergent heterogeneity in a direction aligned with the expected fault—an additional constraint now considered geologically plausible.

## Multiple Parameter Fields

Because an “ideal” level of model-to-measurement misfit cannot be prescribed in most parameter-estimation contexts, and because Tikhonov regularization cannot ensure that inferred heterogeneity is perfectly reflective of geological conditions at all sites, a modeler will inevitably obtain a variety of parameter fields that can all be deemed to “calibrate” a model after multiple PEST runs have been completed. If advice provided in previous sections is followed, these parameter fields will likely be broadly similar. However, they will differ in their representation of heterogeneity that is, on the one hand, substantial enough to be supported by the calibration dataset (and not therefore deserving of relegation to the calibration null space) but on the other hand is not a completely well-defined feature of the model parameter field. This situation cannot be easily avoided because it is a problem with the calibration process itself rather than with the use of mathematical regularization as a calibration tool. Indeed, if regularization were done manually by predefining zones of piecewise constancy, a modeler would still be faced with choices pertaining to how heterogeneity should best be handled. However, the choices would be limited to variations in the geometry of a piecewise-constant field specified by the modeler, one that is usually less geologically representative than those achieved through highly parameterized inversion.

The fact that highly parameterized inversion can often offer a modeler a choice between a variety of reasonable parameter fields (that can then be ranked according to their geological appeal) should be seen as a strength of the method rather than a

weakness. In selecting a preferred parameter field, however, it should be noted that the fields emerging from the calibration process can only be interpreted as simplified versions of the natural world. The selection of one of these fields over another as that which is deemed to “calibrate” the model should be based on considerations of what is most geologically likely or representative (minimum error variance in the statistical sense), and thus leading to predictions of minimum potential wrongness.

Inasmuch as a model prediction depends on details of the modeled system, much of the potential wrongness associated with most model predictions arises from the fact that an optimal calibrated parameter field can be only as complex as the data can support and, in most cases, is far less complex than the natural world (Moore and Doherty 2005, 2006). The fact that various different simplified parameter fields are all compatible with a single calibration dataset demonstrates the inherent nonuniqueness of the inverse problem of model calibration. It also indicates that the degree of nonuniqueness would be greater if the possible parameter fields used to make the prediction included additional geologically realistic complexity not supported by the current calibration data. Such an encompassing estimate of predictive uncertainty can be calculated by using the null space Monte Carlo method (Tonkin and Doherty, 2009; Doherty, Hunt, and Tonkin, 2010). In this analysis, the uncertainty associated with any prediction that depends on realistic parameter detail that cannot be discerned from the calibration data will be higher than that calculated by selection of any one simplified parameter field. Compared to this, differences in prediction uncertainty resulting from the selection of different, but unrealistically simplified, parameter fields are likely to be small (Tonkin and Doherty, 2009).

## Calibration as Hypothesis Testing

The process of model calibration can be viewed as application of the principles of scientific analysis. Scientific analysis advances our understanding of natural systems through a methodology based on hypothesis testing. In the groundwater-modeling context, a current hypothesis regarding the workings of a natural system is encapsulated in the conceptual underpinnings of a numerical model. If the calibration process allows an acceptable fit to be achieved between outputs of that model and field measurements, and if the parameter field that gives rise to that fit is reasonable, then the hypothesis cannot be rejected. It cannot be supposed, however, that the hypothesis represents truth, because other hypotheses may also satisfy these conditions.

Viewed in this manner, the appearance of unreasonable parameter values in a calibrated parameter field, and/or a failure to obtain a good fit between model outputs and field measurements (except perhaps at the cost of unreasonable parameter values) should be viewed as a step along the path of scientific inquiry into the environmental system under investigation. Hopefully, regularized inversion quickens the journey because it can quickly and robustly attain the best

parameter field possible given a conceptual model, even if the problem is ill posed. Moreover, regularized inversion allows the modeler to retain higher levels of parameter flexibility and thus is likely to further scientific inquiry more efficiently than classical parameter estimation techniques that presuppose necessarily simplistic parameter structures in order to achieve a well-posed inverse problem.

## Other Issues

### Calibration Postprocessing

Utilities available through the PEST suite include several postprocessing functions that can add value to the outcomes of the parameter-estimation process (for example, see Doherty, Hunt, and Tonkin, 2010). In fact, some of these tasks can be done before calibration is begun by using only precalibration parameter sensitivities calculated, for example, from a PEST run with NOPTMAX set to  $-1$  or  $-2$ . Useful postprocessing tasks include the following:

1. Tabulation of postcalibration statistics pertinent to one or multiple individual parameters following classical parameter estimation (see the EIGPROC utility).
2. Tabulation of observation influence statistics (Yager, 1998; Hunt and others, 2006) following classical parameter estimation (see the INFSTAT utility).
3. Computation of parameter identifiability and relative parameter error reduction following highly parameterized inversion by using the methodology of Doherty and Hunt (2009) (see the IDENTPAR and GENLINPRED utilities).
4. Calculation of base parameter components of each estimated superparameter (see the PCLC2MAT utility).
5. Calculation of the resolution matrix through which the relation between estimated and real-world parameters is delineated (see the RESPROC and RESWRIT utilities).

Linear and nonlinear parameter and predictive uncertainty analysis is also easily implemented as an adjunct to highly parameterized inversion. Included in such analyses are the ability to compute contributions by different parameter types to current predictive error variance (for example, Hunt and Doherty, 2006), as well as an ability to assess the efficacy of different data-acquisition strategies for reducing current predictive uncertainty (for example, Fienen and others, 2010). Recognizing the existence of geologic detail that cannot be represented in a calibrated model (and hence lies in the calibration null space) is fundamental to the integrity of parameter estimation and predictive uncertainty analysis (Moore and Doherty, 2005). Hence, it is difficult to do such analyses with integrity as an adjunct to classical parameter estimation (Doherty and Hunt, 2010).

## Evaluating Derivatives Used in the Calibration Process

### Integrity of Finite-Difference Derivatives

Implementation of model calibration through highly parameterized inversion often rests on an ability to obtain derivatives of model outputs with respect to adjustable parameters. Where a model can compute these derivatives itself, PEST can make use of these through its external derivatives functionality. In other cases, derivatives must be calculated by using finite parameter differences. Where model outputs suffer from numerical granularity due to solver convergence difficulties, use of adaptive time-stepping schemes, the occurrence of dry cells, or other numerical imperfections, errors will be incurred in finite-difference derivatives calculation as one large number is subtracted from another to compute the (often small) difference between the two.

Means through which problems arising from poor model derivatives can be detected as the parameter-estimation process progresses are discussed in a previous section. However, these means provide only an indication that derivatives may be problematical; they furnish no proof. Furthermore, they shed no light on the magnitude or details of the problem. This information is available through use of the JACTEST and POST-JACTEST utilities. After reading a PEST input dataset, the former program supervises the undertaking of multiple model runs (with an option to parallelize those runs) in which one parameter is sequentially and incrementally varied by the same amount as it is varied by PEST in computing finite-difference derivatives. Information resulting from this process can then be analysed by using POSTJACTEST.

### Manipulation of Jacobian Matrix Files

The contents of a binary Jacobian matrix file (that is, a \*.jco file) written by PEST can be rewritten in ASCII format for user inspection through use of the JACWRIT utility. Other utilities allow individual rows and columns to be extracted from this file for user inspection. See appendix 2 for details. The JCO2JCO utility converts a \*.jco file pertaining to an existing PEST Control File to a new PEST Control File that has removed or transformed parameters and/or observations. However, no new parameters or observations can be added if this utility is to be used. Other utilities allow construction of a composite \*.jco file from separate Jacobian submatrix files and reordering of existing \*.jco files to form new ones. Again, see appendix 2 for details.

## Global Optimizers

If a model's numerical behavior is so poor that use of finite-difference-based derivatives becomes impossible, a modeler is left with no option but to employ a so-called global optimizer to effect computer-based reduction of the calibration

objective function. Two of these global optimizers are provided with the PEST suite. The first (CMAES\_P) encapsulates the covariance matrix adaptation evolution scheme described by Hansen and Ostermeier (2001) and Hansen and others (2003); the second (SCEUA\_P) implements the Shuffled-Complex Evolution algorithm described by Duan and others (1993). Either of these programs can be used interchangeably with PEST because they both read their input data from a PEST Control File, and they both interact with a model through the model's own input/output files through the agency of template and instruction files, as does PEST. Both can undertake model runs in parallel. Experience to date suggests that the former of these programs is more efficient than the latter, especially if more than about 10 parameters are being estimated.

## Summary of Guidelines

This guide suggests approaches for applying PEST and its utility software to the problem of groundwater-model calibration. PEST is still under development. Moreover, no guideline is expected to universally hold, and all guidelines are candidates for future improvement. These issues notwithstanding, a general approach was presented that has utility for many groundwater-model-calibration problems.

**Approach for Parameterization:** Regularized inversion is founded on the use of many parameters that are then constrained by historical measurements of system state together with soft knowledge encapsulated in mathematical regularization. If a model is simple or designed for quick exploration or for processing of a small amount of data, then a classical approach to calibration based on traditional parameter estimation may be acceptable. But if the simulated system is complex, with much time and resources invested in data collection, then a more encompassing regularized inversion approach to model calibration can ensure that information extracted from those data is maximized. Moreover, an approach based on regularized inversion can often be easier to implement than the classical approach because the modeler is not required to simplify the natural world a priori. As a general guide, use as many parameters as time and computing resources allow; but for current computing resources expect an upper practical limit of about 5,000.

**Approach for Observations:** Formulation of a multi-component objective function is essential for ensuring that all observations are seen by the parameter-estimation process. Place observations that are informative of different aspects of a model's parameterization into different observation groups. Ensure that contributions made by different groups to the overall objective function at the start of the calibration process are approximately equivalent. If necessary, process the one calibration dataset in different ways to highlight different components of an overall objective function that are informative of different parameter types or to better reflect what the modeler holds to be important for evaluating model fit. For example, place inter-layer head differences in one observation group and temporal head differences in another observation group.

**Approach for Interjecting Soft Knowledge via Tikhonov Regularization:** Because it is intended to interject the modeler's soft knowledge of a site into the parameter-estimation process, the level and type of Tikhonov regularization is expected to be as variable as the knowledge and whims of modelers themselves. The easiest way to add pervasive regularization to a PEST Control File is through the ADDREG1 utility; this automatically sets a preferred-value constraint for each parameter equal to the initial value of the parameter specified in the PEST Control File. This approach to regularization is more effective if parameters of different types are assigned to different parameter groups, because different types will then be assigned to different regularization groups. This, in turn, allows the modeler to balance regularization weights in accordance with information (or absence of information) in the calibration dataset as it pertains to different parameters through the IREGADJ variable in the "regularization" section of the PEST control file. ADDREG1 sets the IREGADJ regularization control variable to 1 (recommended), allowing PEST to adjust the regularization weight of each group so as to better complement the level of information corresponding to each parameter type contained within the calibration dataset. If Tikhonov constraints are still not being applied optimally, setting IREGADJ to 4 is suggested to implement subspace-enhanced regularization. For early calibration runs, set the target measurement objective function (PHIMLIM) low, but use a FRACPHIM setting of 0.1 to ensure that Tikhonov constraints are still active. On later PEST runs, set PHIMLIM to a value that prevents too good a fit from incurring unrealistic parameter values. If observation weights are inversely proportional to the level of expected measurement noise, a PHIMLIM setting equal to the number of non-zero-weighted observations may provide a defensible tradeoff between attaining an acceptable level of model-to-measurement fit and adherence to the preferred parameter condition.

**PEST Control Variables:** Set RLAMBDA1 to 20 and RLAMFAC to -3. These settings allow rapid variation of the Marquardt Lambda in searching for optimal parameter improvement.

**Solution Mechanism:** Use singular value decomposition (SVD) and/or SVD-Assist (see below). Set MAXSING to the number of adjustable parameters. Set EIGHTHRESH to  $5e-7$ . Alternatively, if you have more than about 2,500 parameters, use LSQR for calculation of parameter upgrades. Set LSQR\_ATOL, LSQR\_BTOL, LSQR\_CONLIM and LSQR\_ITNLIM to  $1e-10$ ,  $1e-10$ ,  $5e3$  and  $5e4$ , respectively.

**Parameter Settings:** Most highly parameterized problems benefit from use of a log transform on the estimated parameters. However, parameters that may become zero or negative cannot be log transformed. If untransformed parameters are excessively sensitive, adjust their SCALE to ensure that their sensitivities are of the approximate magnitude as that of other parameters (as reported in the \*.sen file generated by PEST). Because highly parameterized models are characterized by insensitive and correlated parameters, the initial values specified for all estimated parameters have greater importance

than in classical parameter estimation. Ensure that initial parameter values are geologically reasonable and plausible for the site area.

**Variables Governing Derivatives Calculation:** In the “parameter groups” section of the PEST Control File, set DERINC to 0.01 and INCTYP to “relative.” However, ensure that an absolute lower bound is used for parameters that can become very small. Set FORCEN to “switch” and DERINCMUL to 2.0.

**Checking and Initially Running PEST:** Run the utility PESTCHEK on the PEST Control File before attempting to run PEST because this will do a comprehensive check of the integrity and consistency of many of the components of the PEST run. Fix any errors or unwanted warnings. Run PEST in a NOPTMAX=0 mode to obtain one forward model run that allows a full evaluation of PEST-to-model linkages and the model batch file. Use the reported objective function to evaluate the contributions of different observation groups to the overall objective function, and consider reweighting to obtain a more balanced starting objective function. When the initial objective function is appropriately balanced, run PEST using NOPTMAX=-1 to obtain information on parameter sensitivity and correlation, as well as obtaining the Jacobian matrix needed for singular value decomposition. Because each parameter will require at least one model run, the NOPTMAX=-1 run will benefit from parallel processing if the computation capacity is available, either by using parallel PEST (PPEST, Doherty, 2010a) or BeoPEST (Schreuder, 2009).

**SVD-Assist:** Use SVD alone if sufficient computational resources are available for direct estimation of all adjustable parameters; use SVD-Assist if the high number of parameters is too computationally demanding. If using SVD-Assist, use as many superparameters as you have computing resources for, and add a singular value decomposition section to the PEST Control file to invoke SVD as the solution mechanism for SVD-Assist superparameters (see settings in “Solution Mechanism” section of this summary). This strategy ensures that if one is estimating more superparameters than is justifiable on the basis of the observation dataset used to calibrate the model, the singular value decomposition solution mechanism will maintain numerical stability and will restrict estimation to the number of superparameters that are actually estimable. At the same time, specifying more superparameters than might be supportable grants the flexibility to move superparameters in and out of the solution space from iteration to iteration as supported by the field data, thus better addressing possible model nonlinearity. Use the SVDAPREP utility to create the SVD-Assist PEST Control File. Before running the utility (and before calculating the Jacobian matrix using the NOPTMAX = -1 or -2 option), employ Tikhonov regularization of base parameters and add a “singular value decomposition” section to the base parameter PEST Control File. Respond to all pertinent SVDAPREP prompts with defaults. Two possible exceptions are as follows: select “SVD on  $Q^{(1/2)}X$ ” (option 1, default) for the solution type, unless using more than 2,500 base parameters; if using more than 2,500 base parameters,

choose option 4 (“LSQR with orthogonalisation”). In addition, the tradeoff of fit between observed and simulated values and parameter reasonableness is often most easily assessed if intermediate best parameter (\*.bpa) and residual (\*.rei) files are written during the SVD-Assist run. Choose these options in place of the default (“none”) by typing “br” at the appropriate prompt.

**Postcalibration Analyses:** Regularized inversion supplies a quantitative and comprehensive framework to perform useful ancillary post-calibration analyses. These can provide insights into the parameter-estimation process, the conceptual model tested through the calibration process, and the strengths and weaknesses of the calibration dataset. Utilities which perform highly parameterized post-calibration analyses are included in the PEST suite of programs. The INFSTAT utility can be used to assess what observations are influencing which parameters. IDENTPAR extends traditional sensitivity analysis by directly incorporating parameter correlation into an assessment of parameter identifiability. JACTEST allows inspection of the parameter-to-observation derivatives that drive the parameter-estimation process. Finally, several postcalibration utilities can be used to formally investigate model parameter predictive uncertainty, as well as the worth of future data collection and network design for reducing this uncertainty. Approaches for applying these methods to highly parameterized problems are discussed in detail in Doherty, Hunt, and Tonkin (2010).

## References

- Alcolea, A., Carrera, J., and Medina, A., 2006, Inversion of heterogeneous parabolic-type equations using the pilot points method: International Journal for Numerical Methods in Fluids, v. 51, no. 9–10, p. 963–980.
- Alcolea, A., Carrera, J., and Medina, A., 2008, Regularized pilot points method for reproducing the effect of small scale variability—Application to simulations of contaminant transport: Journal of Hydrology, v. 335, no. 1–4, p. 76–90.
- Aster, R.C., Borchers, B., and Thurber, C.H., 2005, Parameter estimation and inverse problems: Amsterdam, Elsevier Academic Press, 301 p.
- 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, v. 31, no. 4, p. 674–700.
- 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.

- de Marsily, G., Lavedan, C., Boucher, M., and Fasanino, G., 1984, Interpretation of interference tests in a well field using geostatistical techniques to fit the permeability distribution in a reservoir model, *in* Verly, G., David, M., Journel, A.G., and Marechal, A., eds., *Geostatistics for natural resources characterization*: Norwell, Mass., D. Reidel, NATO Advanced Study Institute, ser. C., v. 122, p. 831–849.
- Doherty, J., 2003, Ground water model calibration using pilot points and regularization: *Ground Water*, v. 41, no. 2, p. 170–177.
- Doherty, J., 2007, PEST surface water modeling utilities: Brisbane, Australia, Watermark Numerical Computing.
- Doherty, J., 2008, Groundwater data utilities: 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., 2010c, BeoPEST for Windows: 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., 2009, 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., 2010, 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., 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.
- Doherty, J., and Welter, D.E., 2010, A short exploration of structural noise: *Water Resources Research*, 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.
- Duan, Q.Y., Gupta, V.K., and Sorooshian, S., 1993, A shuffled complex evolution approach for effective and efficient global minimization: *Journal of Optimization Theory and its Applications*, v. 76, no. 3, p. 501–521.
- Feinstein, D.T., Hunt, R.J., and Reeves, H.W., 2008, Calibrating a big model—Strategies and limitations, *in* MODFLOW and More 2008—Ground water and public policy, Proceedings of the 9th International Conference of the International Ground Water Modeling Center: Golden, Colo., Colorado School of Mines, p. 430–434.
- 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.
- Fienen, M., Hunt, R., Krabbenhoft, D., and Clemo, T., 2009, Obtaining parsimonious hydraulic conductivity fields using head and transport observations—A Bayesian geostatistical parameter estimation approach: *Water Resources Research*, v. 45, W08405, doi:10.1029/2008WR007431.
- Fienen, M.N., Muffles, 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.
- Gallagher, M.R., and Doherty, J., 2007, Parameter interdependence and uncertainty induced by lumping in a hydrologic model: *Water Resources Research*, v. 43, no. 5, W05421, doi:10.1029/2006WR005347.
- Hansen, N., Müller, S.D., and Koumoutsakos, P., 2003, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES): *Evolutionary Computation*, v. 11, no. 1, p. 1–18.
- Hansen, N., and Ostermeier, A., 2001, Completely derandomized self-adaptation in evolution strategies: *Evolutionary Computation*, v. 9, no. 2, p. 159–195.
- Hill, M.C., and Tiedeman, C.R., 2007, *Effective groundwater model calibration—with analysis of data, sensitivities, predictions, and uncertainty*: Hoboken, N.J., Wiley-Interscience, 455 p.
- Hunt, R.J., and Doherty, J., 2006, A strategy of constructing models to minimize prediction uncertainty, *in* MODFLOW and More 2008—Ground water and public policy, Proceedings of the 9th International Conference of the International Ground Water Modeling Center: Golden, Colo., Colorado School of Mines, p. 56–60.
- 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.

- 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., Luchette, J., Schreuder, W.A., Rumbaugh, J.O., Doherty, J., Tonkin, M.J., and Rumbaugh, D.B., 2010, Using a cloud to replenish parched groundwater modeling efforts: *Rapid Communication for Ground Water*, v. 48, no. 3, p. 360–365; doi:10.1111/j.1745–6584.2010.00699.x.
- Hunt, R., and Zheng, C., 1999, Debating complexity in modeling; EOS, *Transactions American Geophysical Union*, v. 80, no. 3, p. 29.
- 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.
- Kavetski, D., Kuczera, G., and Franks, S.W., 2006, Calibration of conceptual hydrological models revisited. 1. Overcoming numerical artefacts: *Journal of Hydrology*, v. 320, no. 1–2, p. 173–186.
- Luchette, J.; Nelson, G.K.; McLane, C.F., III; and Cekan, L.I., 2009, Unlimited virtual computing capacity using the cloud for automated parameter estimation, in *Proceedings of the 1st PEST Conference*, November 2–4, 2009, Potomac, Md. [http://www.lulu.com/product/download/pest-conference-2009-proceedings/6503195?productTrackingContext=center\\_search\\_results](http://www.lulu.com/product/download/pest-conference-2009-proceedings/6503195?productTrackingContext=center_search_results)
- Moore, C., and Doherty, J., 2005, The role of the calibration process in reducing model predictive error: *Water Resources Research*, v. 41, no. 5, W05020.
- 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.
- Muffels, C.T., 2008, Application of the LSQR algorithm to the calibration of a regional groundwater flow model—Trout Lake Basin, Vilas County, Wisconsin: University of Wisconsin-Madison, Department of Geology and Geophysics, M.S. thesis, 106 p.
- Paige, C.C., and Saunders, M.A., 1982, LSQR—An algorithm for sparse linear equations and sparse least squares: *ACM Transactions on Mathematical Software*, v. 8, no. 1, p. 43–71.
- Schreuder, W.A., 2009, Running BeoPEST, in *Proceedings of the 1st PEST Conference*, November 2–4, 2009, Potomac, Md. [http://www.lulu.com/product/download/pest-conference-2009-proceedings/6503195?productTrackingContext=center\\_search\\_results](http://www.lulu.com/product/download/pest-conference-2009-proceedings/6503195?productTrackingContext=center_search_results)
- Skahill, B.E., and Doherty, J., 2006, Efficient accommodation of local minima in watershed model calibration: *Journal of Hydrology*, v. 329, no. 1–2, p. 122–139.
- Tikhonov, A.N., 1963a, Solution of incorrectly formulated problems and the regularization method: *Soviet Mathematics Doklady*, v. 4, p. 1035–1038.
- Tikhonov, A.N., 1963b, Regularization of incorrectly posed problems: *Soviet Mathematics Doklady*, v. 4, p. 1624–1637.
- Tikhonov, A.N., and Arsenin, V.Y., 1977, *Solutions of ill-posed problems*: New York, Halstead Press-Wiley, 258 p.
- Tonkin, M.J., and Doherty, J., 2005, A hybrid regularized inversion methodology for highly parameterized models: *Water Resources Research*, v. 41, no. 10, W10412, doi:10.1029/2005WR003995.
- Tonkin, M., and Doherty, J., 2009, Calibration-constrained Monte Carlo analysis of highly parameterized models using subspace techniques: *Water Resources Research* v. 45, W00B10, doi:10.1029/2007WR006678.
- Walker, J.F., Hunt, R.J., Doherty, J., and Hay, L., 2009, Processing time-series data to calibrate a surface-water model in small headwater watersheds, in *Proceedings of the 1st PEST Conference*, November 2–4, 2009, Potomac, Md. [http://www.lulu.com/product/download/pest-conference-2009-proceedings/6503195?productTrackingContext=center\\_search\\_results](http://www.lulu.com/product/download/pest-conference-2009-proceedings/6503195?productTrackingContext=center_search_results)
- 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]
RLAMBDA1 RLAMFAC PHIRATSUF PHIREDLAM NUMLAM [JACUPDATE]
RELPARMAX FACPARMAX FACORIG [IBOUNDSTICK UPVECBEND]
PHIREDSWH [NOPTSWITCH] [SPLITSWH] [DOAUI] [DOSENREUSE]
NOPTMAX PHIREdstp NPHISTP NPHINORED RELPARSTP NRELPAR [PHISTOPTHRESH] [LASTRUN] [PHIABANDON]
ICOV ICOR IEIG [IRES] [JCOSAVEITN] [REISAVEITN]
* automatic user intervention
MAXAUI AUISTARTOPT NOAUIPHIRAT AUIRESTITN
AUISENSRAT AUIHOLDMAXCHG AUINUMFREE
AUIPHIRATSUF AUIPHIRATACCEPT NAUINOACCEPT
* singular value decomposition
SVDMODE
MAXSING EIGTHRESH
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
SENRELTHRESH 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 line
DERCOMLINE
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 * PARNAME + PIFAC * log(PARNAME) ... = PIVAL WEIGHT OBGNAME
(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]]

```

**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”	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.
NINSLFE	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.

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

Variable	Type	Values	Usage	Description
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	“au,” “aud,” or “noau”	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.
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 atol variable.
LSQR_BTOL	real	0 or greater	1	LSQR algorithm btol variable.
LSQR_CONLIM	real	0 or greater	1	LSQR algorithm conlim variable.
LSQR_ITNLIM	integer	greater than 0	1	LSQR algorithm itnlim 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
OBSNME	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.
OBSNME	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.
RELPREDSTP	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 RELPREDSTP 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.

**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
<u>case</u> .pst	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.
<u>case</u> .rmf	Run management file	Provides Parallel PEST with information needed to communicate with slaves.
<u>case</u> .hld	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 $X^T Q X$ 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>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</i> , <i>case.jac</i> , <i>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</i> , <i>param.rdy</i> , <i>observ.rdy</i> , <i>pslave.fin</i> , <i>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 and uncertainty 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 $\mathbf{Q}^{1/2}\mathbf{X}$ rather than $\mathbf{X}^T\mathbf{Q}\mathbf{X}$ .
PREDVAR4	Computes contribution made to the error variance of a prediction by different parameters and/or groups of parameters.
PREDVAR5	Computes observation worth through its 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.
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.

**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
MAT2SRF	Writes a matrix in SURFER grid format.
MATADD	Performs matrix addition.

**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
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.

**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. Singular Value Decomposition Theory

This appendix presents a short discussion of singular value decomposition of the Jacobian matrix and of associated suggestions for use of observation data in the calibration process.

Let  $\mathbf{X}$  represent the Jacobian matrix of sensitivities of model outputs used in the calibration process to parameters estimated through that process. Through singular value decomposition,  $\mathbf{X}$  can be represented as

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

where the columns of  $\mathbf{U}$  are orthogonal unit vectors that span the range space of  $\mathbf{X}$ , the columns of  $\mathbf{V}$  are orthogonal unit vectors that span the domain of  $\mathbf{X}$  (that is, parameter space), and  $\mathbf{S}$  is a diagonal matrix of singular values. Let  $\mathbf{S}$  be now partitioned into two submatrices, one of which ( $\mathbf{S}_1$ ) contains zero and near-zero-valued singular values while the other ( $\mathbf{S}_2$ ) contains significantly non-zero singular values. Let  $\mathbf{U}$  and  $\mathbf{V}$  be similarly partitioned, so that:

$$\mathbf{X} = [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{bmatrix} \mathbf{S}_1 & 0 \\ 0 & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \approx \mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_1^T \quad (\text{A4.2})$$

Let  $\mathbf{h}$  represent a calibration dataset,  $\mathbf{p}$  a set of model parameters, and  $\boldsymbol{\varepsilon}$  a vector of model/measurement noise. Then,

$$\mathbf{h} = \mathbf{X}\mathbf{p} + \boldsymbol{\varepsilon} \quad (\text{A4.3})$$

If noise is ignored for the sake of simplicity, and equation (A4.2) is introduced to (A4.3), this becomes

$$\mathbf{S}^{-1}\mathbf{U}_1^T\mathbf{h} = \mathbf{V}_1^T\mathbf{p} \quad (\text{A4.4})$$

In this equation,  $\mathbf{U}_1^T\mathbf{h}$  is a vector, each element of which is the projection of observations constituting the calibration dataset onto one of the orthogonal unit vectors spanning the range space of the model (that is, of  $\mathbf{X}$ ).  $\mathbf{V}_1^T\mathbf{p}$  on the right-hand side of equation (A4.4) also is a vector. Each element of this vector is the projection of real-world parameters  $\mathbf{p}$  onto an orthogonal axis of parameter space comprising one of the columns of  $\mathbf{V}_1$ . Each of these columns defines a “pattern” or “basis function” in parameter space, this pattern involving more and more detail as the singular value with which it is associated decreases in value. However, truncation of singular values beyond  $\mathbf{S}_1$  ensures that there is an upper limit to this detail, this leading to an estimated parameter field that is simpler than the complex hydraulic-property field  $\mathbf{p}$  that exists in reality.

Because  $\mathbf{S}_1^{-1}$  is a diagonal matrix, the elements  $\mathbf{U}_1^T\mathbf{h}$  are related to those of  $\mathbf{V}_1^T\mathbf{p}$  on a one-to-one basis. That is, each estimable parameter projection that constitutes an element of  $\mathbf{V}_1^T\mathbf{p}$  is informed entirely by a single corresponding observation

projection defined by the respective element of  $\mathbf{U}_1^T\mathbf{h}$ . To state this in another way, each linear combination of observations specified by  $\mathbf{U}_1^T\mathbf{h}$  is uniquely and solely informative of a single combination of real-world hydraulic properties described by  $\mathbf{V}_1^T\mathbf{p}$ . Certainly this information is contaminated by measurement noise, and thus these estimates of parameter combinations furnished by the calibration process will be in error.

Equation A4.4 demonstrates that the one observation dataset can be processed in different ways to provide information on separate aspects of (broad-scale) model parameterization. In simple calibration contexts, this is an obvious notion. For example, it is well known that calibration to head data alone results in non-unique conjunctive estimates of recharge and conductivity parameters; see, for example, page 35 in Hunt (1987) and Haitjema (2006). The addition of one flux measurement to a calibration dataset can break this non-uniqueness; see, for example, Poeter and Hill (1997). By placing head measurements into one observation group and the single flow measurement into another observation group, and by providing weights to measurements comprising these respective groups according to a schedule which ensures that neither group making up the objective function (irrespective of the number of elements comprising each group or of the size of the numbers used to represent each measurement type), estimates for conductivity and recharge parameters forthcoming from the calibration process will be as good as can be obtained under the circumstances.

Examples of where the same dataset, processed in different ways, can lead to formulation of a multicomponent objective function in which each component is informative of different aspects of model parameterization include the following:

10. The use of temporal and vertical head differences as sources of information on groundwater storage and vertical conductance parameters, respectively.
11. In the surface-water-modeling context, the use of accumulated monthly flow volumes and digitally filter extracted base flows as sources of information on gross basin evapotranspiration and shallow lumped interflow/base-flow conductance respectively (see Doherty and Johnston, 2003; and Doherty, 2007, for further details).

## References Cited

- Doherty, J., 2007, PEST surface water modeling utilities: Brisbane, Australia, Watermark Numerical Computing.
- Doherty, J., and Johnston, J.M., 2003, Methodologies for calibration and predictive analysis of a watershed model: *Journal of the American Water Resources Association*, v. 39, no. 2, p. 251–265.

- Haitjema, H.M., 2006, The role of hand calculations in ground water flow modeling: *Ground Water* v. 44, no. 6, p. 786–791.
- Hunt, R.J., 1987, A survey of the hydrogeochemistry of the Leopold Memorial Reserve, Baraboo, Wisconsin: University of Wisconsin-Madison, Department of Geology and Geophysics, M.S. Thesis, 170 p.
- Poeter, E.P., and Hill, M.C., 1997, Inverse models—A necessary next step in ground-water modeling: *Ground Water* v. 35, no. 2, p. 250–260.





