Getting the Most out of PEST

John Doherty

Watermark Numerical Computing

August, 2013

Table of Contents

This Document

The purpose of this document is to provide a quick overview of optimal settings to use with PEST, and of some of the functionality that is available through PEST and its support software.

At the time of writing, documentation for PEST resides in the PEST manual and in the addendum to that manual. The addendum is way too big. Furthermore, items are only roughly grouped according to subject matter. Instead the ordering of material in the addendum is partly chronological as the addendum was supplemented whenever changes were made to PEST. Later this year, or early next year, I will write a new edition of the manual (and dispense with the addendum). In the meantime, the present document is intended to provide some answers to frequently asked questions on PEST usage, and to provide some guidance on how best to use PEST.

This document is brief. Headers should guide you quickly to matters that interest you. Full details and theory are available in the manual and addendum.

PEST Tutorial

A tutorial can be downloaded from the PEST web pages at:

http://www.pesthomepage.org

Two example problems are provided in the tutorial. One is a groundwater model and the other is a surface water model. Aspects of PEST usage that are demonstrated using these examples include the following.

- Traditional inversion;
- Highly parameterized inversion;
- Tikhonov regularization;
- Use of pilot points;
- Singular value decomposition;
- SVD-assist;
- Linear uncertainty analysis;
- Null space Monte Carlo analysis.

The theory behind these methods is also covered in the tutorial document. In addition, considerable discussion is devoted to the use of models in environmental decision support, and the role that calibration and uncertainty analysis should play in this.

List of All PEST Control Variables

See the preface of the addendum to the PEST manual for a list of all PEST control variables, their locations within the PEST control file, and tables of their meanings.

Optimal PEST Settings

Every case is different. The settings which guide PEST usage in different inversion contexts may need to reflect this. However the following course of action will result in stable PEST performance

on most occasions. It will also result in estimates of parameter values that, on many occasions, achieve the goal of a minimum error variance solution to the inverse problem, regardless of the nonuniqueness of that problem. This does not mean that the parameters achieved are right - only that their potential for wrongness has been minimized. This potential can still be very high – a reflection of the nonuniqueness of their estimated values. Their post-calibration uncertainty, and that of predictions which are sensitive to them, can be analysed using linear and nonlinear uncertainty analysis functionality supported by PEST and its utilities; see below.

Here is a general "recipe" for PEST setup.

1. Build a PEST control, template and instruction files in the usual manner. In the PEST control file file:

    - Set initial parameter values at their preferred values based on expert knowledge.

    - Assign parameters of different types to different parameter groups. Make sure that the names of these groups are 6 characters or less in length. (This will allow the ADDREG1 utility to formulate unique names for regularization groups which constrain them – see below.)

    - Assign appropriate bounds to parameters, these reflecting the endpoints of their range of realistic values.

    - Assign observations of different types to different observation groups.

    - Weight observations within each group in a way that provides suitable weighting relatively. For some groups this will mean that all observations are assigned a weight of 1.0. For other groups weights may need to reflect the values of the observations to which they pertain, especially where the latter can vary over many orders of magnitude, as can river flows and contaminant concentrations in groundwater. In such cases higher weights may need to be provided to low-valued observations so that they are visible to PEST.

    - Log transform all parameters whose lower bound is greater than 0.0. (Provide a lower bound that is slightly greater than zero, rather than exactly zero, if necessary.)

    - Set RLAMDA1 to 10.0 and RLAMFAC to -3; the latter guarantees fast movement of the Marquardt lambda.

2. Introduce a "singular value decomposition" section to the PEST control file. Use of singular value decomposition (SVD) ensures that PEST maintains numerical stability, regardless of how ill-posed is the inverse problem. (In its early days, PEST sometimes failed to work where inverse problems were ill-posed.) In the "singular value decomposition" section set:

    - SVDMODE to 1;
    - MAXSING to the number of adjustable parameters;
    - EIGTHRESH to 5×10^{-7};
    - EIGWRITE to 0.

3. However if you are estimating more than about 2500 parameters, introduce an "lsqr" section to the PEST control file instead of a "singular value decomposition" section. In this section set:

    - LSQRMODE to 1;
    - LSQR_ATOL and LSQR_BTOL to 10^{-6};
    - LSQR_CONLIM to 1000;
    - LSQR_ITNLIM to 4 times the number of adjustable parameters;
    - LSQRWRITE to 0.

4. Set NOPTMAX to zero in the "control data" section of the PEST control file and then run PEST. PEST will run the model once and calculate an objective function, together with the contribution made to the total objective function by each observation group. This will allow you to ensure that all is working as it should. It will also provide information needed by the PWTADJ1 utility, which is run next.

5. Run PWTADJ1. This creates a new PEST control file in which weights are adjusted such that the contribution made to the total objective function by each observation group is the same. This prevents the information content of any group from being invisible to the inversion process.

6. Add Tikhonov regularization to the PEST control file by running the ADDREG1 utility. ADDREG1 automatically sets the target measurement objective function (PHIMLIM) to 10^{-10}. This, of course, is way too low; however it is often a good idea to attain the lowest measurement objective function that you can on the initial PEST run. Then, on later runs, set PHIMLIM to a value somewhat higher than the lowest achievable objective function in order to keep estimated parameter values closer to their initial values if you judge that this is a reasonable thing to do. Set PHIMACCEPT about 2% higher than PHIMLIM.

    In adding regularization to the PEST control file ADDREG1 automatically does the following.

    - It provides a prior information equation for each parameter in which the preferred value of that parameter is equated to its initial value.

    - It assigns these prior information equations to regularization groups that inherit their names from the parameter groups to which the parameters belong. (It must add "*regul_*" as a prefix to the name of each parameter group; that is why parameter group names should be 6 characters or less in length.)

    - It sets the IREGADJ variable to 1. This allows PEST to assign different regularization weight factors to different prior information groups.

    - It sets the FRACPHIM variable to 0.1. This instructs PEST to temporarily adopt a target measurement objective function of one tenth the current measurement objective function during all iterations, unless PHIMLIM is greater than this, in which case PHIMLIM prevails.

7. Run PEST.

SVD-Assist

General

Use of SVD-assist decreases PEST execution time dramatically as PEST needs to carry out only as many model runs per iteration as the number of super parameters that it is estimating. This number need only be as large as the dimensionality of the solution space of the inverse problem (or less if computer resources are limited).

How many super parameters should you estimate? The more the better; estimate as many as your computing resources allow. It doesn't matter if you estimate more super parameters than are uniquely estimable. If you follow the "recipe" outlined above in setting up the base PEST control file, the use of singular value decomposition as a solution device is carried over to the super parameter PEST control file. This ensures numerical stability, regardless of the number of super parameters that are cited in the super parameter PEST control file. The extra super parameters ensure more robust behaviour where models are very nonlinear.

SUPCALC

The SUPCALC utility suggests a suitable number of super parameters for use in the current inversion problem. In general, unless your computing resources prohibit it, you should not use fewer super parameters than are suggested by SUPCALC. However feel free to use as many more as you like.

SUPCALC's prompts and some typical responses follow.

```
Enter name of PEST control file: case.pst
Enter expected value of measurement objective function: 310.0

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

Use uncertainty file or bounds to specify parameter variability?  [u/b]: b

Enter name for eigenvector gain/loss output file: nul
```

SUPCALC prompts and typical responses.

Note that if the bounds options is selected in response to the second last of these prompts, then bounds applied to parameters in the "parameter data" section of the PEST control file should reflect their range of credible values. Bounds thus become an expression of your expert knowledge.

Before SVDA

Before setting up an SVD-assisted PEST run (this is done using SVDAPREP – see below) you must first run PEST. For this run NOPTMAX should be set to -1 or -2 (the latter is slightly quicker). PEST will calculate a Jacobian matrix and then cease execution. The Jacobian matrix forms the basis for definition of super parameters.

SVDAPREP

After running PEST with NOPTMAX set to -1 or -2, run SVDAPREP. Prompts and typical responses are shown below.

```
 Enter name of existing PEST control file: case2.pst
 Use pre-defined super-parameter file? [y/n]  (<Enter> if "no"): <Enter>
 For computation of super parameters:-
    if SVD on Q^(1/2)X                - enter 1
    if SVD on XtQX                    - enter 2
    if LSQR without orthogonalisation - enter 3
    if LSQR with orthogonalisation    - enter 4
 Enter your choice (<Enter> if 1): 1
 Enter number of super parameters to estimate: 52
 Enter name for new super pest control file: case1_svda
 Enter offset for super parameters  (<Enter> if 10): <Enter>
 Enter value for RELPARMAX (<Enter> if 0.1): <Enter>
 Write multiple BPA, JCO, REI, none [b/j/r/n] files (<Enter> if "n")? <Enter>
 Parameter scale adjustment [SVDA_SCALADJ] setting (<Enter> if 2): <Enter>
 Make new model batch file silent or verbose?  [s/v] (<Enter> if "s") : <Enter>
 Automatic calc. of 1st itn. super param. derivs?  [y/n] (<Enter> if "y") : <Enter>
```

SVDAPREP prompts and typical responses.

Note the following:

- Choose the first of the four options for computation of super parameters unless the number of base parameters is more than about 2000. In that case considerable time can be saved if you choose option 3 (in which case super parameters are computed using LSQR).

- The above settings are conservative. Sometimes SVD-assisted parameter estimation will proceed at greater speed if you ask for an offset of 20 for super parameters and, at the same time, ask for RELPARMAX to be 0.2. So respond with "20" and "0.2" to the two prompts that follow the prompt in which you supply the name of the new super parameter PEST control file if you wish to gain some speed.

Obtaining Optimized Parameters

If using SVD-assist, best parameters achieved up to any stage of PEST execution are contained in a parameter value file whose extension is "*.bpa*". The filename base is the name of the base parameter PEST control file.

Suppose that the base parameter PEST control file is named *base.pst* and that the super parameter PEST control file is named *super.pst*. After the PEST run is complete, you can obtain a new base parameter PEST control file named *base1.pst* containing optimized parameter values as its initial parameter values using the PARREP utility as follows.

```
parrep base.bpa base.pst base1.pst
```

Set NOPTMAX to 0 in *base1.pst* and run PEST. PEST will run the model just once. Verify that the measurement objective function is as expected. (The regularization objective function will be different from that which was achieved during the previous optimization process as PEST does not optimize regularization weights when undertaking just a single model run.) Model input files now contain optimized parameter values. Model output files contain quantities calculated by the model on the basis of these parameter values.

Very Nonlinear Models

If a model is very nonlinear then optimal super parameter definition may vary with parameter values. In this case you may need to re-compute super parameters after a few iterations of SVD-assisted inversion. The procedure is as follows.

1. Run PEST using SVD-assist as above.

2. After a few iterations have elapsed and it is clear that the objective function is not falling any more, stop PEST execution. (Use PSTOP, or more brutally <Ctl-C>, to achieve this.)

3. Create a new base PEST control file in which previously optimized parameter values are used as initial parameter values; use PARREP to do this as above.

4. Set NOPTMAX to -2 in the new base PEST control file. Then run PEST to create a new Jacobian matrix.

5. Run SVDAPREP once again to build a new super parameter PEST control file based on the new Jacobian matrix and the semi-optimized parameter values.

6. Run PEST on this new super parameter PEST control file to continue the SVD-assisted inversion process.

Inspecting Super Parameters

So-called "super parameters" are the coefficients by which combinations of model parameters are multiplied prior to summation in order to form a new set of parameters which the model then uses. These combinations of parameters are obtained through singular value decomposition of the weighted Jacobian matrix. Each is a unit vector, and each is orthogonal to all other unit vectors which comprise other parameter combinations. Furthermore, each is uniquely and entirely informed by a distinct combination of observations, these being referred to in PEST documentation as "super observations".

You can tabulate the parameter and observation combinations that make up each super parameter and its partnered super observation using the SUPOBSPAR and SUPOBSPAR1 utilities.

Termination Criteria

When using PEST myself I normally adopt conservative settings of 50, 0.005, 4, 4, 0.005 and 4 for the termination criteria NOPTMAX PHIREDSTP NPHISTP NPHINORED RELPARSTP NRELPAR. These settings ensure that a slowing down of the optimization process does not result in its premature termination. However these settings can also result in inordinately long PEST run times. So I mostly end up stopping PEST execution myself when it appears that the objective function is not going to fall much further. This can be done using <Ctl-C> or the PSTOP or PSTOPST commands. In the latter case PEST undertakes a final model run with optimized parameters (unless you are undertaking SVD-assisted inversion). In the first two cases you will need to undertake this model run yourself. This is easily achieved by using the PARREP utility to create a new PEST control file in which initial parameter values are previously optimized parameter values. Set NOPTMAX to 0 in that file.

The optional PHISTOPTHRESH control variable can be used to instruct PEST to cease execution if the objective function falls below a certain value. The PHIMLIM variable has the same role when PEST is run in regularization mode.

Parameter Scaling

Theoretically, parameters should be scaled with respect to their innate variability before being estimated. This guarantees something approaching a minimum error variance solution to the inverse problem. At the same time it guarantees that some parameters (such as recharge rates) are not over-sensitive because they are expressed by low numbers.

Logarithmic transformation of all parameters can achieve this to some extent.

Alternatively, the BOUNDSCALE control variable can be used to perform scaling by parameter prior variability, with the distance between parameter bounds being taken as an indicator of this variability.

Leaving Tracks

At any time during the parameter estimation process, and at the end of the parameter estimation process, best parameters can be found in the "parameter value file". For normal PEST usage this file has the same filename base as the PEST control file, but has an extension of "*.par*". When implementing SVD-assisted parameter estimation, this file has the same filename base as the base parameter PEST control file, but with an extension of "*.bpa*" (for "best parameters").

Sometimes it is useful to inspect best parameters from previous iterations. This may be warranted if, for example, you think that PEST has achieved too good a fit in later iterations and that some parameters have suffered because of this. You may consider that parameters estimated during a previous iteration, where the fit was not so good, may be more realistic.

Use the PARSAVEITN variable in the "control data" section of the PEST control file to tell PEST to record a parameter value file pertaining to every iteration, without these iteration-specific files being over-written by parameter value files from later iterations. However if you are using SVD-assist, this is achieved through use of the SVDA_MULBPA variable. This variable is automatically set if you respond with "b" to the following prompt, which is issued by SVDAPREP when it writes the input dataset for an SVD-assisted PEST run:

```
Write multiple BPA, JCO, REI, none [b/j/r/n] files (<Enter> if "n")?
```

In both cases the iteration number is added as a suffix to the end of the name of the normal parameter value file. So suppose that the normal parameter value file is named *case.par*. Then iteration-specific parameter value files will be named *case.par.1*, *case.par.2* etc. File *case.par* will still, at any stage of the PEST inversion process, and at the end of that process, contain best parameters achieved up until that time. The same protocol applies to the *.bpa* file in SVD-assisted parameter estimation.

Defence against Model Failure

If the model run by PEST fails in its execution, it will not complete the writing of its output files (or will not write them at all). When PEST thinks that the model has finished execution, it look for

these files and, failing to find them or failing to read all of them, ceases execution itself with an appropriate error message. This can be prevented using the LAMFORGIVE and DERFORGIVE variables.

Model failure is most likely to occur during that part of an optimization iteration in which PEST varies the Marquardt lambda in order to test the effectiveness of different parameter values in lowering the objective function. This is where parameters differ the most between successive model runs. With the LAMFORGIVE variable in the PEST control file set to the string "*lamforgive*", PEST construes absence or incompleteness of model output files as an indicator of model malperformance, induced through use of the parameter values that it asked the model to use. It will associate an objective function of 10^{30} with that set of parameter values and will initiate another model run based on an alternative set of parameter values.

If the DERFORGIVE variable is set to "*derforgive*", then PEST can accommodate model run failure during finite-difference derivatives computation. In this case derivatives of all model outputs with respect to the parameter whose variation led to run failure are set to zero for that optimization iteration.

Defence against Bad Model Performance

Detecting Bad Model Performance

PEST calculates derivatives of model outputs with respect to parameters by taking differences between model outputs calculated on the basis of incrementally varied parameter values. Sometimes derivatives calculated in this way do not have integrity. This can happen if a model is numerically unstable, if its solver's convergence criteria have not been set tight enough, or if numbers are not written to its output files with maximum precision. It can also happen if the numerical path to the solution is not repeated from model run to model run, as can happen if a model employs adaptive time-stepping.

The JACTEST utility can be used to explore the integrity of finite-difference-calculated derivatives. First choose a parameter; then instruct JACTEST to calculate all model outputs for a number of different values of that parameter as the latter is varied from its initial value by a succession of increments – the same increments that PEST usees in finite-difference derivatives calculation. The file which JACTEST writes can be imported into a spread sheet where graphs of model output vs. parameter value can readily be drawn. These should be smooth. It does not matter if they are curved; however they should not be jagged (unless a particular model output is insensitive to the varied parameter and varies only by one or two significant figures). The POSTJACTEST utility can be used to order entries in the JACTEST output file so that model outputs with worst derivatives performance are easy to find.

Detecting and Ignoring Suspicious Derivatives

When PEST uses central differencing to calculate derivatives, it first increments a parameter and then decrements it, running the model on each occasion. It uses the corresponding model output values, together with those calculated for the reference parameter value, to evaluate derivatives with respect to the incrementally-varied parameter. PEST can also be asked to compare individual, two-point derivatives corresponding to the individual forward and reverse parameter increments. If these are very different for a particular model output, PEST takes this to mean that one of them is

in error – probably the one with the higher absolute value. It can then be given the option of (a) using only the smaller of the two slopes to calculate a two-point derivative with respect to the varied parameter, (b) re-using the derivative calculated during the previous iteration, or (c) equating the derivative to zero for that parameter. See the SPLITTHRESH, SPLITRELDIFF and SPLITACTION variables, all of which belong to the "parameter groups" section of the PEST control file.

Five Point Derivatives

PEST allows use of four or five parameter values for calculation of derivatives. Naturally this is expensive, as it requires that many more model runs be undertaken than if just forward or central derivatives calculation is undertaken; in fact it requires that four model runs per iteration be undertaken for each parameter for which derivatives of this type are requested. However the extra numerical burden may be worth it where a model behaves badly. Set FORCEN in the "parameter groups" section of the PEST control file to "*switch_5*" or "*always_5*" and DERMTHD to "*minvar*" or "*maxprec*" (preferably "*minvar*").

Jacobian Matrix Files

JCO File Re-use

If you have calculated a Jacobian matrix and then alter a PEST control file, there may be no need to re-compute the Jacobian matrix for the new PEST control file. If you fix or tie some parameters, change the log-transformation status of some parameters, alter the bounds associated with some parameters, remove observations, adjust observation weights, or add or subtract prior information from a PEST control file to create a new PEST control file, the JCO2JCO utility can be used to construct a Jacobian matrix file for the new PEST control file from that of the old one.

Viewing the Jacobian Matrix

The Jacobian matrix file is recorded in binary, compressed format to save space. It is named *case.jco* where *case* is the filename base of the PEST control file. It can be translated to ASCII format using the JACWRIT or JCO2MAT utilities. Rows of the binary-stored Jacobian matrix can be extracted using the JROW2VEC utility. MATCOLEX following JCO2MAT can be used to extract a column. You can undertake singular value decomposition of the Jacobian matrix using the MATSVD utility.

Transferring a JCO file between WINDOWS and UNIX

A JCO file generated by PEST on a UNIX platform may not be readable by PEST (and PEST utilities) running on a WINDOWS platform. The opposite is also the case. To transfer a JCO file between these platforms, do the following:

1. Convert the JCO file to ASCII format using the JCO2MAT utility.
2. Transfer the ASCII file to/from the UNIX platform.
3. Convert the ASCII file to binary format using the MAT2JCO utility.

Building a JCO File from its Parts

If sensitivities for some parameters lie within one JCO file and those for others lie in another JCO file, the two can be combined using the JCOPCAT utility.

First PEST Iteration for Free

If you start PEST with the "/i" switch it will prompt for the name of an existing Jacobian matrix file. This will save it from having to compute the Jacobian matrix itself during its first iteration. There is no need to use this switch if undertaking SVD-assisted inversion however; PEST already gives you the first iteration for free by calculating a super parameter Jacobian matrix from the base parameter Jacobian matrix.

Posterior Uncertainties and Other Statistics

General

When PEST is run in "estimation" mode and when it does not use singular value decomposition to solve the inverse problem, it records a parameter covariance matrix, together with parameter correlation coefficients and other uncertainty data, at the bottom of its run record file. However if any form of regularization is used (either Tikhonov regularization – as when it is run in "regularization" mode - or SVD/LSQR) these items are not recorded.

This is why. If a post-calibration parameter covariance matrix is to be computed using the traditional formula $\sigma^2(\mathbf{J}^t\mathbf{Q}\mathbf{J})^{-1}$ where $\mathbf{J}$ is the Jacobian matrix and σ^2 is the reference variance, then the inverse problem must be well-posed for this matrix to be calculable, and *very* well posed for it to have validity as an indicator of posterior parameter uncertainty. This does not happen often, for most problems that we encounter in environmental modelling embody more parameters than can be estimated uniquely. This is fine if regularization is used in solution of the inverse problem (either Tikhonov, SVD/LSQR, or both); PEST remains numerically stable and, if regularization is properly used, a minimum error variance solution to the inverse problem of model calibration can thereby be attained. However under these circumstances the posterior covariance matrix calculated under the assumption of problem well-posedness is incorrect (if it can be calculated at all).

Where a problem is ill-posed, the prior uncertainties of parameters must be taken into account when computing their posterior uncertainties (i.e. their post-calibration uncertainties). For some parameters, and combinations of parameters, prior uncertainties will be reduced through calibration; for others, the prior uncertainty may not be reduced at all.

PREDUNC7

The PREDUNC7 utility can be used after any PEST run to obtain a posterior parameter covariance matrix. Prompts and typical responses are shown in the box below.

```
Enter name of PEST control file: case.pst
Enter observation reference variance: 13.2

Enter name of parameter uncertainty file: param.unc
Enter name for posterior parameter covariance matrix file: temp.mat

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

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

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

- reading parameter uncertainty file param.unc....
- parameter uncertainty file param.unc read ok.
- inverting observation covariance matrices....
- forming XtC-1(e)X matrix....
- inverting prior C(p) matrix....
- inverting [XtC-1(e)X + C-1(p)] matrix....
- writing file temp.mat...
- file temp.mat written ok.
```

PREDUNC7 prompts and typical responses.

The reference variance (required by the second prompt) is available from the bottom of the PEST run record file. Alternatively it is easily calculated as the (expected) objective function value divided by the (number of observations minus number of parameters). Or, if you have used PWTADJ2 (see below) to formulate a new PEST control file with weights informed by the objective function (and possibly its components), the reference variance is 1.0.

A parameter uncertainty file (required by the third of PREDUNC7's prompts) is discussed next. You may also want to cite the covariance matrix file that PREDUNC7 writes in another parameter uncertainty file that you make yourself so that other PEST utilities can use it for their own processing.

Parameter Uncertainty File

Many of the programs of the PEST suite require this file. It is often used to provide prior parameter uncertainties, these being parameter uncertainties based on expert knowledge. PEST documentation provides its format; see also the tutorial mentioned above.

Assessment of prior uncertainties is sometimes hard. These are supposed to encapsulate your expert knowledge of parameters as it exists before the inversion process. Often you will have already expressed this knowledge through placing bounds on parameters when constructing the PEST control file. An easy way to formulate a list of prior parameter standard deviations (this being one option for filling a parameter uncertainty file) is to divide the difference between these bounds (or their logs for parameters that are log-transformed) by 4.0. This procedure relies on the the implicit assumption that parameters are (log)normally distributed and that their bounds span their 95% confidence intervals.

Sometimes the prior covariance matrix should express prior correlation between parameters. If you are using pilot points as a spatial parameterization device in a groundwater model, these can be calculated from a variogram using the PPCOV utility from the Groundwater Data Utility suite.

Other Posterior Statistics

The PPSTAT utility provides a comprehensive suite of post-calibration statistics, these pertaining to both parameters and observations. See also INFSTAT1.

Linear Predictive Uncertainty Analysis

Through linear uncertainty analysis, the following can be achieved.

- Calculation of the uncertainty of a prediction or parameter;
- Calculation of the contributions that different parameters or parameter groups make to that uncertainty;
- Calculation of the efficacy of different existing or new data in reducing that uncertainty;
- Solution and null space contributions to parameter/predictive uncertainty.

These are evaluated using the PREDUNC* and PREDVAR* suites of programs. Use of these programs can be as complicated as you like. However easy access to some of their functionality can be gained through use of the GENLINPRED utility.

The PREDUNC* suite uses Bayes equation (with certain assumptions) in calculating all of the above quantities. The PREDVAR* suite uses subspace analysis based on singular value decomposition. The former examines posterior uncertainty, whereas the latter examines post-calibration error. If the inverse problem is properly formulated, the two are almost the same (with uncertainty a little lower than error). When using GENLINPRED, ask for "uncertainty" rather than "error" analysis to reduce the run time, except when exploring solution and null space contributions to parameter/predictive error (for these cannot be calculated for uncertainty).

Before doing any kind of uncertainty or error analysis it is often helpful to modify the PEST control file so that weights are informed by objective function components attained during the inversion process. This can be achieved using the PWTADJ2 utility. This alters weights so that the objective function (and, optionally, each of its components) is equal to the number of observations (in the component group). The reference variance is thus approximately 1.0. Linear analysis utilities supplied with the PEST suite use weights as an indication of measurement/structural noise. With weights adjusted in this manner the inverse of each weight can be considered to represent the standard deviation of noise associated with each corresponding observation.