A Synthetic Case

Linear Analysis

A GMDSI tutorial

by Rui Hugman and John Doherty

gmdsi
Groundwater Modelling Decision Support Initiative

BHP Flinders University | National Centre for Groundwater Research and Training | RioTinto

PUBLISHED BY

The National Centre for Groundwater Research and Training
C/O Flinders University
GPO Box 2100
Adelaide SA 5001
+61 8 8201 2193

DISCLAIMER

PREFACE

The Groundwater Modelling Decision Support Initiative (GMDSI) is an industry-funded and industry-aligned project focused on improving the role that groundwater modelling plays in supporting environmental management and decision-making.

Over the life of the project, GMDSI will produce a suite of tutorials. These are intended to assist modellers in setting up and using model-partner software in ways that support the decision-support imperatives of data assimilation and uncertainty quantification. Not only will they focus on software usage details. They will also suggest ways in which the ideas behind the software which they demonstrate can be put into practice in everyday, real-world modelling.

GMDSI tutorials are designed to be modular and independent of each other. Each tutorial addresses its own specific modelling topic. Hence there is no need to work through them in a pre-ordained sequence. That being said, they also complement each other. Many employ variations of the same synthetic case and are based on the same simulator (MODFLOW 6). Utility software from the PEST suite is used extensively to assist in model parameterization, objective function definition and general PEST/PEST++ setup. Some tutorials focus on the use of PEST and PEST++, while others focus on ancillary issues such as introducing transient recharge to a groundwater model and visualization of a model's grid, parameterization, and calculated states.

The authors of GMDSI tutorials do not claim that the workflows and methodologies that are described in these tutorials comprise the best approach to decision-support modelling. Their desire is to introduce modellers, and those who are interested in modelling, to concepts and tools that can improve the role that simulation plays in decision-support. Meanwhile, the workflows attempt to demonstrate the innovative and practical use of widely available, public domain and commonly used software in ways that do not require extensive modelling experience nor an extensive modelling skillset. However, users who are adept at programming can readily extend the workflows for more creative deployment in their own modelling contexts.

We thank and acknowledge our collaborators, and GMDSI project funders, for making these tutorials possible.

Rui Hugman

John Doherty

CONTENTS

1. INTRODUCTION

This document is part of series of tutorials which demonstrate workflows for parameter estimation and uncertainty analysis with PEST/PEST++. These are not the only (or necessarily the best) workflows; their purpose is to take the reader through the basics of how to accomplish common tasks. The present document is a tutorial on how to perform common linear analysis tasks using utilities from the PEST suite. Linear analysis is applied to a model which has been setup and calibrated in a previous GMDSI tutorial.

The PEST roadmaps are a useful complement to this series of tutorials. Going through **Roadmap 12: Linear Analysis** prior to commencing this tutorial is recommended. The roadmap provides a brief discussion of the concepts and theory, as well as an outline of steps required to undertake linear analysis. The tutorial focuses on practical implementation of concepts discussed in the Roadmap. The abovementioned GMDSI tutorial on model calibration provides details of the model and PEST configuration on which the current tutorial is based. A brief summary is provided in Chapter 1.1.

This tutorial assumes that you are at least partially familiar with PEST and the PEST software suite. Although completion of the previous tutorial is not a requirement, this is recommended if you are not familiar with PEST.

Completed versions of all files generated during this tutorial can be found in the folder named *completed*. These are useful if you need to troubleshoot your own files, or if you wish to jump into the tutorial at a specific point. That being said, it is recommended to at least read through the parts that you do not complete yourself.

Software Executables

A number of programs are used throughout this tutorial. To make completion of this tutorial easier, executable versions of these programs have been placed in relevant folders; these are **.exe* files. This is generally not recommended practice, for data files and executable files should be kept separate!

Preferably, executable files (i.e. programs) should be located in a folder that is cited in your computer's PATH environment variable. Doing this allows them to be run from a command prompt that is open to any other folder without including the full path to these executables in the command to run them.

1.1 Background

The case being modelled is entirely synthetic. It is the authors' experience that building synthetic models that look like "the real thing" never really works well. However the synthetic nature of a didactic model does not detract from the lessons that it teaches. Some design decisions have been taken to highlight the application of specific tools or techniques at the expense of sometimes making the case "un-realistic". So please bear with us, dear reader, if sometimes things look a bit weird!

Our imaginary site looks something like Figure 1. Several streams traverse the area, flowing from west to east. These streams are perennial. Stream flow is maintained by groundwater. Let us imagine that these streams are one of the last remaining habitats of a particular species of frog which needs an aquatic environment all year round. Let us accept that it is in humanity's best interest to keep this species of frog alive.

The nearby town of Makebelievesville needs to expand its water supply through groundwater abstraction. Land access and existing infrastructure limit possible sites for a wellfield. Of these, the preferred site is close to one of the streams; see the red star in Figure 1. Proposed abstraction rates are a constant 2,000 m^3/d. Proximity of groundwater extraction to the stream raises concerns that

streamflow may be depleted. After extensive research, ecologists have determined that the frog species requires a minimum stream flow rate of 6,500 m^3/d to survive. If proposed abstraction reduces groundwater discharge to the streams to less than this threshold “a bad thing” happens (and humanity will suffer).

A numerical model was developed to assess whether proposed abstraction will reduce groundwater discharge to the streams below the critical threshold of 6,500 m^3/d under long-term average conditions (i.e. steady state). This model simulates two steady state stress periods. The first represents historical conditions. The second accounts for the proposed additional abstraction.

Model parameters were calibrated against hydraulic head and stream flow measurements. Values of recharge rate, hydraulic conductivities in each model layer, and conductances of boundary conditions were parameterized using pilot points. Preferred-value Tikhonov regularisation was implemented. Covariance matrices were used instead of weights in prior information equations that specified preferred parameter values, in order to “spread out” emergent hydraulic property heterogeneity.

The model prediction of interest is simulated groundwater discharge into the upper stream reach during the second stress period. This observation is named *p-trib_1* in the PEST control file. The calibrated parameter set results in a predicted discharge of 6,899 m^3/d to the stream once additional abstraction is in place.

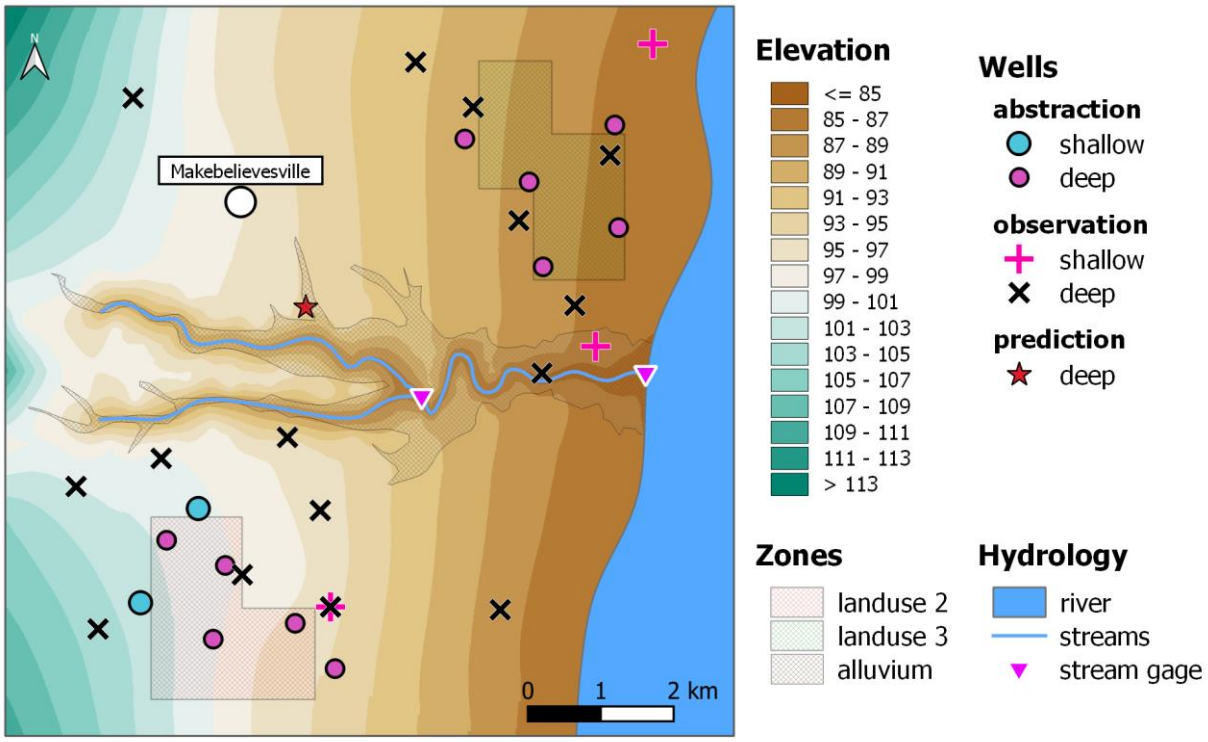

Figure 1 – Layout of the synthetic case and main features of interest.

2. LINEAR ANALYSIS

Linear uncertainty analysis is also known as “first order second moment” (or “FOSM”) analysis. It provides approximate mathematical characterisation of prior predictive probability distributions, and of posterior parameter and predictive probability distributions. It has other uses as well. It can be used to demonstrate how the history-matching process bestows worth on data. It can also be deployed to track the flow of information from field measurements of system state to parameters, and ultimately from parameters to model predictions. It does all of these things by implementing Bayes equation under the following assumptions.

- The prior probability distribution of parameters is multiGaussian.
- “Measurement noise” (including structural noise) is also characterized by a Gaussian distribution.
- The relationships between model outputs that correspond to measurements of system state and parameters employed by a model can be approximated by the action of a matrix on a vector.
- Model outputs that correspond to predictions of management interest can be calculated using another matrix that acts on model parameters.

Ideally linear analysis is undertaken after a model has been calibrated. However, if a model is truly linear (which it never is), the outcomes of linear analysis are independent of parameter values and can therefore, in theory, be applied with the user-supplied prior mean parameter values.

If calibration has been undertaken, then minimum-error variance (i.e. calibrated) parameter values should be assigned to parameters as their initial parameters in the “parameter data” section of the PEST control file on which linear analysis is based. The Jacobian matrix should be calculated using these parameters. And, if the uncertainty of a prediction is going to be examined, then the model output that pertains to this prediction must be included as an “observation” in the PEST input dataset; sensitivities of this model output to model parameters will therefore appear in the Jacobian matrix.

2.1 Getting Ready

Three files are needed prior to undertaking linear analysis with the PEST suite. These are as follows:

1) PEST control file in which:
   a) calibrated parameters are initial values (this is easily achieved using the PARREP utility);
   b) all regularisation is removed (this can be done using the SUBREG1 utility); and
   c) observation weights are equal to the inverse of the standard deviation of measurement noise (accomplished using PWTADJ2).
2) A Jacobian matrix (JCO file) calculated for calibrated parameters (obtained by running PEST with NOPTMAX set to -2 in the PEST control file, or by using the JCO2JCO utility with a pre-existing JCO file);
3) A “parameter uncertainty file” in which prior parameter uncertainties are recorded (prepared by the user).

PEST Control File

A PEST input dataset and model files have been provided for you in the tutorial folder. As previously mentioned, this dataset and model were developed during the GMDSI tutorial on model calibration. All PEST input files are housed in the *pest-linunc* folder. All model files are housed in the *runmodel* folder. (The latter include MODFLOW 6 inputs and outputs, as well as files required by the PLPROC and OLPROC model pre/postprocessors).

The *pest-linunc* folder contains a PEST control file named *cov-par6.pst*, as well as associated PEST output files. This control file already contains calibrated values for parameters as their initial values in its "parameter data" section (as you may recall if you completed the previous GMDSI tutorial).

1 Open *cov-par6.pst* in your text editor of choice and inspect it.

2 This control file already contains calibrated parameter values as initial parameter values. These were inserted using the PARREP utility (see the GMDSI tutorial on model calibration).

3 The model prediction is already included as an observation in the PEST input dataset. Our model prediction of interest is groundwater inflow to the upper reaches of the streams during the second stress period. In the PEST control file this "observation" is named *p-trib_1*. If you inspect the bottom of the "observation data" section of this file, you will see this observation. Note that it has a weight of zero; hence it has no role in parameter estimation

4 If you inspect the "control data" section of the PEST control file, you will note that the NOPTMAX variable is set to -2; the value of this variable is shown in **bold** in the text box below. This instructs PEST to calculate and record the Jacobian matrix and then cease execution. This has already been accomplished for you. The Jacobian matrix is stored in a JCO file (i.e. a Jacobian matrix file) named *cov-par6.jco*.

```
* control data
restart regularisation
    2205      23       7    2205      12
3  5    single  point
 10.0  -3.0  0.3  0.03  10  lamforgive uptestmin=20
 30.0  30.0  0.001
0.1
-2  0.005  4  4  0.005  4
0  0  0 parsaveitn reisaveitn
```

5 Although you do not need to do this for the purpose of this tutorial, you may if you wish, re-calculate the Jacobian matrix. To do so, simply run PEST_HP using the *cov-par6.pst* control file. (See the GMDSI tutorial on model calibration for instructions on how to run PEST_HP). If you do this, PEST will run the model 2206 times (the number of parameters plus one). The more agents you use, the faster will this process be.

PEST linear analysis utilities all assume that weights listed in the "observation data" section of a PEST control file are equal to the inverse of the standard deviation of "noise" (i.e. potential error) associated with field measurements. The utility PWTADJ2 writes a new PEST control file in which this is the case; it back-calculates "noise" from the level of fit achieved with a calibration dataset.

6 Open a command line in the *pest-linunc* folder. Run PWTADJ2 to write a new PEST control file named *cov-par6-wtadj2.pst* based on the original *cov-par6.pst*. As the calibration process did not have trouble with any of the observation groups, the same adjustment factor will be applied to all groups; see documentation of PWTADJ2 in Part 2 of the PEST manual for more discussion on this option. To accomplish this, type the following command and then press <enter>:

```
pwtadj2 cov-par6.pst cov-par6-pwtadj2.pst ng
```

7 You should see the following text appear on your screen:

```
 - reading PEST control file cov-par6.pst for first time...

 - reading PEST run record file cov-par6.rec...
 - file cov-par6.rec read ok.

 - re-reading file cov-par6.pst and writing file cov-par6-pwtadj2.pst...
```

```
 - file cov-par6.pst read ok.
 - file cov-par6-pwtadj2.pst written ok.
```

8 Inspect the *pest-linunc* folder. You should see a new file named *cov-par6-wtadj2.pst*. If you inspect it in a text editor, you will note that it is almost identical to *cov-par6.pst*; however observation weights in the "observation data" section have been altered, and NOPTMAX has been set to 50.

The provided PEST control file is configured to run in "regularisation" mode; it includes a set of prior information equations and a "regularisation" section; so too, therefore, does *cov-par6-wtadj2.pst*. When undertaking linear analysis, the PEST control file must be configured to run in "estimation" mode, and all prior information must be removed. This can be accomplished in a single step using the SUBREG1 utility.

9 Type the following command into the command line and then press <enter>:

```
subreg1 cov-par6-wtadj2.pst linunc.pst
```

10 You should see the following text appear on your screen;

```
 - file cov-par6-wtadj2.pst read ok.
 - file linunc.pst written ok.
```

11 You should see a new file named *linunc.pst* in your folder. If you inspect it, you will see that PEST is instructed to run in "estimation" mode, and that the "regularisation" and "prior information" sections have been removed.

Jacobian Matrix File

At this point we have a clean PEST control file, ready for undertaking linear analysis (*linunc.pst*). We also have a Jacobian matrix file (*cov-par6.jco*), calculated using a PEST control file with calibrated parameters as initial values, and with the prediction as an observation (*cov-par6.pst*).

However, the PEST control file which was used to calculate the Jacobian differs from the PEST control file which we are about to use for linear analysis. To undertake linear analysis, we need a Jacobian matrix file which corresponds to the *linunc.pst* PEST control file. There are two ways in which this can be accomplished: (1) setting NOPTMAX in *linunc.pst* to -2 and re-calculating the Jacobian matrix; or (2) using the JCO2JCO utility to calculate a new JCO file for the altered PEST control file, based on the original PEST control file. Depending on the alterations that have been done to a PEST control file, it is not always possible to do the latter. Fortunately, in the present case it is.

12 Type the following into the command line and then press <enter>:

```
jco2jco cov-par6.pst linunc.pst
```

13 You should see the following text appear on your screen:

```
- file cov-par6.pst read ok.
- file cov-par6.jco read ok.
- file linunc.pst read ok.
- file linunc.jco written ok.
```

14 If you inspect your folder, you should see a new JCO file named *linunc.jco*.

Prior Parameter Uncertainty File

Lastly, we need a file which characterizes prior parameter uncertainty. Part 2 of the PEST manual provides specifications for a parameter uncertainty file. By convention, this type of file has an extension of **.unc*. The current document will only discuss those aspects of its construction that are pertinent to this tutorial. Many utilities of the PEST suite require that the user provide a file that specifies parameter uncertainties. These can be prior or posterior uncertainties but are usually the former. The utilities described in the current tutorial obtain information which they require to build a

prior parameter covariance matrix from such an uncertainty file. This matrix is often referred to as a C(**k**) matrix in PEST parlance; C() signifies "covariance matrix" while **k** (a vector) signifies parameters.

A parameter uncertainty file is a text file in which the uncertainties of parameters (or groups of parameters) are characterised. Utilities of the PEST suite which read an uncertainty file must also read a PEST control file which specifies parameters and observations which define an uncertainty analysis problem. All parameters listed in the PEST control file must be referenced in the uncertainty file. Note, however, that the uncertainty file can include more parameters than those that are featured in the PEST control file.

Parameter uncertainty can be characterized in two different ways in a parameter uncertainty file:

1) The standard deviation for each individual parameter can be listed;
2) A covariance matrix can be supplied for a collection of parameters.

A single parameter uncertainty file can include both of these options. However, only one option can be used for each individual parameter. The uncertainty file is divided into blocks; each block implements one of the options described above for a set of parameters. Each block begins with a START line and ends with an END line, as will be demonstrated shortly.

In the current tutorial, all prior parameter uncertainties are characterised with covariance matrices. All of the parameters used in the inversion process are associated with pilot points; prior parameter correlation is an outcome of spatial proximity of respective pilot points. Prior covariance matrices for these pilot point parameters have previously been prepared using the PPCOV* family of utilities; other GMDSI tutorials detail how to apply these utilities to prepare covariance matrix files. The covariance matrix files have been provided for you in the *runmodel\preproc* folder; these files all have an extension of *.*mat*. These are the same covariance which were used for regularisation during the calibration process (see the GMDSI calibration tutorial).

15 Inspect the *runmodel\preproc* folder. You should see six files with the extension *.*mat*. Each corresponds to a parameter group in the *linunc.pst* PEST control file.

16 Create a new empty text file in the *pest-linunc* folder. Name this file *param.unc* while opening it in your text editor of choice. We will now proceed to populate this file with COVARIANCE_MATRIX blocks for each of the *.*mat* files that characterize the prior uncertainties of a set of parameters.

17 We shall begin with the block for parameters which belong to the *kx1* parameter group. All blocks in an uncertainty file must begin with a "START *blocktype*" line and end with an "END *blocktype*" line (where *blocktype* is either "COVARIANCE_MATRIX" or "STANDARD_DEVIATION". Start by typing the following text into *param.unc* to create a new, empty COVARIANCE_MATRIX block:

```
START COVARIANCE_MATRIX

END COVARIANCE_MATRIX
```

18 Now we will proceed to fill the COVARIANCE_MATRIX block. The first line of the block must reference the keyword *file*; this must be followed by the name of a covariance matrix file. If the file is not in the same folder as the PEST control file (as in our case), then the file name must include the file path. The covariance matrix file that pertains to parameters in the *kx1* group is named *cov1.mat*; it is located in the *runmodel\preproc* folder. Update the COVARIANCE_MATRIX block as shown below with the relative path to *cov1.mat*:

```
START COVARIANCE_MATRIX
file "..\runmodel\preproc\cov1.mat"
END COVARIANCE_MATRIX
```

19 Next we need to provide information on which parameters in the PEST control file are associated with this covariance matrix. There are a number of ways in which this can be done. PPCOV can be instructed to provide parameter names within the file itself. Alternatively, parameter-to-covariance matrix linkages can be provided through the parameter uncertainty file itself. There are two ways to do this. They are (1) providing the name of an external file with a list of parameters that are linked to a particular parameter matrix file, or (2) using the *first_parameter* and *last_parameter* keywords. We will use the second option as this is easy to implement; however this only works if parameters in the "parameter data" section of the PEST control file are listed in the same order as that in which they are represented in the covariance matrix file. Care must be taken when constructing covariance matrix files to ensure that this is the case. This will often happen naturally as a consequence of the way in which these files are constructed; they normally inherit their order from a user-prepared pilot points file.

20 Update the COVARIANCE_MATRIX block as shown below with the *first_parameter* and *last_parameter* keywords. Note that parameter *kx1pp001* is the first parameter of the *kx1* parameter group listed in the "parameter data" section of the *linunc.pst* PEST control file. Likewise, *kx1pp523* is the last of the same group. If you inspect *cov1.mat* with a text editor, you should see that it is comprised of the same number of rows and columns (523) as the number of parameters between *kx1pp001* and *kx1pp523*.

```
START COVARIANCE_MATRIX
file "..\runmodel\preproc\cov1.mat"
first_parameter kx1pp001
last_parameter kx1pp523
END COVARIANCE_MATRIX
```

Optionally, a covariance matrix can be multiplied by a factor supplied with a *variance_multiplier* keyword. This is useful in cases where the same covariance matrix file is used for different sets of parameters characterised by different variances. In our case, each set of parameters is characterised by a different covariance matrix file. However, all the supplied covariance matrix files were originally constructed with a variance (sill) of 1.0. This value does not reflect the prior variances of our parameters. Thus, we need to correct it using the *variance_multiplier* keyword.

21 Update the COVARIANCE_MATRIX block as shown below by introducing a *variance_multiplier* of 0.25. All elements of the matrix are multiplied by this number. Diagonal elements therefore become 0.25. The standard deviations of respective parameters therefore become 0.5 (the square root of 0.25).

```
START COVARIANCE_MATRIX
file "..\runmodel\preproc\cov1.mat"
first_parameter kx1pp001
last_parameter kx1pp523
variance_multiplier 0.25
END COVARIANCE_MATRIX
```

A Short Mathematical Detour

You may be wondering "*how did you come up with 0.25?*". Good question.

First, recall that parameters of the *kx1* parameter group are all log-transformed (in fact, all parameters in the current PEST control file are log-transformed). Therefore, this variance actually pertains to the variance of the log-transformed parameters (log to base 10).

Now you may be asking "*so what?*". Fair enough.

Say, for example, that we believe that parameters within the *kx1* parameter group can vary across two orders of magnitude. (It is not unusual for hydraulic conductivities to exhibit this level of variability). This means that parameter values span two orders of magnitude. The span of log-transformed parameters is therefore 2. (Span can also be referred to as "range"; here we use the term span to avoid confusion with the concept of range in a variogram.)

From the expected parameter span (our prior belief) we can use a simple rule of thumb to estimate parameter standard deviations, under the assumption that parameter values follow a normal distribution. To obtain the standard deviation of a parameter, divide its span by four.

So, if the span of log-transformed parameters is 2, then the standard deviation of log-transformed parameter is approximately 0.5. Variance is obtained by squaring standard deviation ($0.5^2 = 0.25$).

The same approach can be adopted for parameters that are not log-transformed, as long as you have an estimated span of the parameter values. (Often the span of values that a parameter may take can be equated to the difference between its upper and lower bounds.)

Back to the Prior Parameter Uncertainty File

22 OK. So, we have one COVARIANCE_MATRIX block configured in the *param.unc* uncertainty file. Blocks for each of the remaining parameter groups still need to be added.

23 In *param.unc*, create a COVARIANCE_MATRIX block for each parameter group; assign the pertinent covariance matrix file to that block. Parameter groups *kx2, kx3, strc, ghb1, ghb3* and *rch* are associated with covariance matrix files *cov2.mat, cov3.mat, cov_stream.mat, cov_ghb.mat, cov_ghb.mat* and *cov_rch.mat* respectively. (Note that the same covariance matrix file is used for *ghb1* and *ghb3*).

24 With the exception of the block that is associated with the *rch* parameter group*, variance_multiplier* values can all be set to 0.25. In the COVARIANCE_MATRIX block that pertains to the *rch* parameters, set the *variance_multiplier* to 0.03.

25 Once finished, we are ready to undertake linear analysis. Your *param.unc* file should now look like this:

```
# Kx layer 1
START COVARIANCE_MATRIX
file "..\runmodel\preproc\cov1.mat"
first_parameter kx1pp001
last_parameter kx1pp523
variance_multiplier 0.25
END COVARIANCE_MATRIX

# Kx layer 2
START COVARIANCE_MATRIX
file "..\runmodel\preproc\cov2.mat"
first_parameter kx2pp001
last_parameter kx2pp523
variance_multiplier 0.25
END COVARIANCE_MATRIX

# Kx layer 3
START COVARIANCE_MATRIX
file "..\runmodel\preproc\cov3.mat"
first_parameter kx3pp001
last_parameter kx3pp523
variance_multiplier 0.25
END COVARIANCE_MATRIX

# streams
START COVARIANCE_MATRIX
file "..\runmodel\preproc\cov_stream.mat"
first_parameter drncpp01
last_parameter drncpp31
variance_multiplier 0.25
END COVARIANCE_MATRIX

# GHB layer1
START COVARIANCE_MATRIX
file "..\runmodel\preproc\cov_ghb.mat"
first_parameter ghbc1pp01
last_parameter ghbc1pp41
variance_multiplier 0.25
END COVARIANCE_MATRIX

# GHB layer3
START COVARIANCE_MATRIX
file "..\runmodel\preproc\cov_ghb.mat"
first_parameter ghbc3pp01
last_parameter ghbc3pp41
variance_multiplier 0.25
END COVARIANCE_MATRIX

# recharge rate
START COVARIANCE_MATRIX
file "..\runmodel\preproc\cov_rch.mat"
first_parameter rchpp001
last_parameter rchpp523
variance_multiplier 0.03
END COVARIANCE_MATRIX
```

2.2 Running Linear Analysis Utilities

A suite of utility programs whose names begin with “PREDUNC” are included in the PEST suite. (“PREDUNC” stands for “PREDictive UNCertainty”.) A complementary suite of utilities have names that begin with “PREDVAR”. These explore parameter and predictive *error* rather than *uncertainty.* In general, programs from the PREDVAR suite are not as useful as those from the PREDUNC suite. Programs for both these suites are described in detail in Part 2 of the PEST manual.

All of the programs which belong to these suites can be run individually to accomplish specific linear analysis tasks. Alternatively, the GENLINPRED or GENLINPRED_ABBREV utilities can be used to undertake a default set of tasks. GENLINPRED and GENLINPRED_ABBREV run utilities from the PREDNUC and PREDVAR suites in the background, automating file-preparation and keyboard response tasks that normally accompany their use.

The current tutorial demonstrates the use of GENLINPRED_ABBREV. Its intention is to introduce you to these tools and provide some insights on how to interpret their outputs. Unlike GENLINPRED, GENLINPRED_ABBREV does not run utilities from the PREDVAR suite of PEST utilities that assess parameter and predictive error; however it does make use of the PREDVAR1A utility to assess the dependence of predictive error on singular values that emerge from singular value decomposition of the Jacobian matrix. As another abbreviation, it focusses on parameter groups rather than individual parameters when assessing contributions to predictive uncertainty, and on observation groups rather than individual observations when assessing data worth; this can reduce analysis time enormously. More bespoke analyses can be undertaken using individual members of the PREDUNC suite.

GENLINPRED_ABBREV

26 Open a command line window in the *pest-linunc* folder (or use a window that is already open to this folder) and run GENLINPRED_ABBREV by typing the following command, and then pressing <enter>.

```
i64genlinpred_abbrev
```

27 You will be prompted to provide the name of a PEST control file; this forms the basis for GENLINPRED_ABBREV’s linear analysis tasks. Provide the name of the PEST control file which we built in Chapter 2.1.

```
Enter name of PEST control file: linunc.pst
```

28 You should see the following text appear on your screen. Respond to the prompt with the letter ***u***. This informs GENLINPRED_ABBREV that you wish to provide an uncertainty file in which prior parameter uncertainties are listed. Alternatively, if you were to respond with ***b***, GENLINPRED_ABBREV would calculate prior parameter uncertainties from parameter bounds supplied in the PEST control file.

```
 - reading PEST control file linunc.pst....
 - file linunc.pst read ok.

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

29 Next GENLINPRED_ABBREV asks you to provide the name of the prior parameter uncertainty file. Respond to the prompt with the name of the uncertainty file produced in Chapter 2.1.

```
Enter name of parameter uncertainty file: param.unc
```

30 Next you are asked whether observation weights are the inverse of measurement noise. As we adjusted the weights using PWTADJ2 (see Chapter 2.1) in order to achieve this very purpose, respond to this prompt with ***y*** or simply press <enter>.

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

31 At the next prompt, simply press <enter>. This instructs GENLINPRED_ABBREV to record the outcomes of its analyses in a file named *genlinpred_abbrev.out*. Alternatively, you could provide an output file name of your choice.

```
Enter name for output file <Enter> if genlinpred_abbrev.out:<enter>
```

32 Next, you are asked if you wish to perform comprehensive analysis of a specific prediction or parameter. We wish to perform an analysis of the prediction *p-trib_1*; so respond with ***y.***

```
Perform comprehensive analysis of a prediction/param? [y/n] <Enter> if "y": y
```

33 Now GENLINPRED_ABBREV asks for the name of the prediction. In our case this is an observation name that is featured in the PEST input dataset. Respond to the prompt accordingly.

```
Enter name of prediction/parameter to analyze: p-trib_1
```

34 GENLINPRED_ABBREV's final prompt requests the name of a file that contains sensitivities of this prediction to all parameters. Simply provide the name of the Jacobian matrix file produced in Section 2.1.

```
Enter file to read its sensitivities ["p" if a parameter]: linunc.jco
```

35 GENLINPRED_ABBREV now goes to work. As it does so, it records its progress on the screen. It may take a few minutes to complete all of its analyses. (The more parameter and/or observation groups that are featured in the PEST control file, the longer does this process take). When it has finished you should see the following screen printout; this indicates that everything went as intended:

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

 SUPCALC has recommended the use of 20 solution space dimensions
 for computation of parameter identifiability and relative
 error reduction.

 - running IDENTPAR to compute parameter identifiabilities...
 - program IDENTPAR run ok.

 - running PREDUNC7 to compute rel param uncert variance reductions...
 - program PREDUNC7 run ok.

 - running JROW2VEC to extract predictive sensitivities...
 - program JROW2VEC run ok.

 - running PREDVAR1A to obtain pred. error variance sing. val. dependence...
 - program PREDVAR1A run ok.

 - running PREDUNC1 to compute predictive uncertainty...
 - program PREDUNC1 run ok.

 - running PREDUNC4 to compute param contribs to pred uncert variance...
 - program PREDUNC4 run ok.

 - running PREDUNC5 to compute observ worth through subtraction...
 - program PREDUNC5 run ok.

 - running PREDUNC5 to compute observ worth through addition...
 - program PREDUNC5 run ok.
```

```
- see file genlinpred_abbrev.out for complete GENLINPRED output.
```

36 Inspect the *pest-linunc* folder. You should see a new file named *genlinpred_abbrev.out*. The outcomes of linear analysis tasks that were managed by GENLINPRED_ABBREV are collected in this file.

37 Open *genlinpred_abbrev.out* in a text editor to inspect it.

38 The first section of this file begins with the text "*ANALYSIS DETAILS*". This provides a summary of the linear analysis tasks which GENLINPRED_ABBREV undertook. Here you can review the responses which you provided to its prompts in Section 2.2.

39 If you scroll through *genlinpred_abbrev.out* you will see several sections, each with outcomes of a specific linear analysis task. Each section usually references the specific utility from the PREDUNC or PREDVAR suite that was used to carry out the task. The following Sections discuss these outputs.

All files generated up to this point can be found in the folder named *completed*.

2.3 Relative Parameter Uncertainty Variance Reduction

Calibration (hopefully) reduces the uncertainties of (at least some) model parameters. If these parameters are spatially distributed (as are pilot point parameters), it can be instructive to display parameter uncertainty in a spatial setting. Sometimes it is useful to compare prior and posterior parameter uncertainties, or to calculate ratios of these uncertainties. These ratios can then be mapped.

GENLINPRED_ABBREV calculates the "relative parameter uncertainty variance reduction" (RPUVR) of each parameter. This is an interesting statistic; it shows the information content of the calibration dataset as it pertains to different parameters, and to different groups of parameters. ("Information" is what reduces uncertainty.) RPUVR is calculated as:

$$\mathrm{RPUVR} = 1 - \left(\frac{posterior\ parameter\ uncertainty\ variance}{prior\ parameter\ uncertainty\ variance}\right)$$

As is evident, this ratio varies between 0.0 and 1.0. Larger values indicate greater reduction in parameter uncertainty *in comparison to prior parameter uncertainty*. Thus, a high ratio may still be accompanied by a high level of parameter uncertainty; however this uncertainty will be much lower than it was prior to calibration. Thus RPUVR provides a qualitative representation of where history-matching has been most effective in reducing parameter uncertainty.

As stated above, GENLINPRED_ABBREV provides RPUVR as one of its outputs. The current tutorial will demonstrate how to transfer these values to model input files for plotting. Note, however, that GENLINPRED_ABBREV does not provide the actual values of prior and posterior parameter uncertainties. The PREDUNC7 utility can be used to calculate these. Once evaluated, they can then be used for plotting purposes, and/or to calculate other calibration-informative statistics if you wish. See the **PEST Roadmap 12: Linear Analysis** for further details.

40 With *genlinpred_abbrev.out* open in a text editor, find the line that contains the following text; it should be on line 50:

```
Outcomes of parameter identifiability and error/uncertainty variance reduction
analysis
```

41 Beneath this line you should see three columns of values; see the first few lines in the text box below. The first column lists parameter names, the second column lists parameter identifiabilities, while the third lists parameter RPUVRs. It is the contents of this third column that we are interested in plotting.

```
    Parameter              Identifiability        Rel_uncert_var_redn
    ---------              ---------------        -------------------
    drncpp01                3.5913889E-02          1.4474500E-02
    drncpp02                1.1033870E-02          1.3719200E-02
    drncpp03                1.8694715E-03          1.0290200E-02
(…)
```

If you wish, you can copy/paste these values into any spreadsheet software for processing and plotting. Figure 2 shows one way to plot these data. Here histograms of RPUVR are plotted for each parameter group. The y-axis is normalized to show the proportion of parameters in each histogram bin. As you can see, calibration has done very little to reduce uncertainty of conductance (*drnc*, *ghbc1* and *ghbc3*) and recharge (*rch*) parameters. Hydraulic conductivity parameters (*kx1*, *kcx2* and *kx3*) have incurred the largest relative reductions in uncertainty variance; the largest information gain is in layer 3, where a greater proportion of parameters exhibit a larger value of RPUVR.

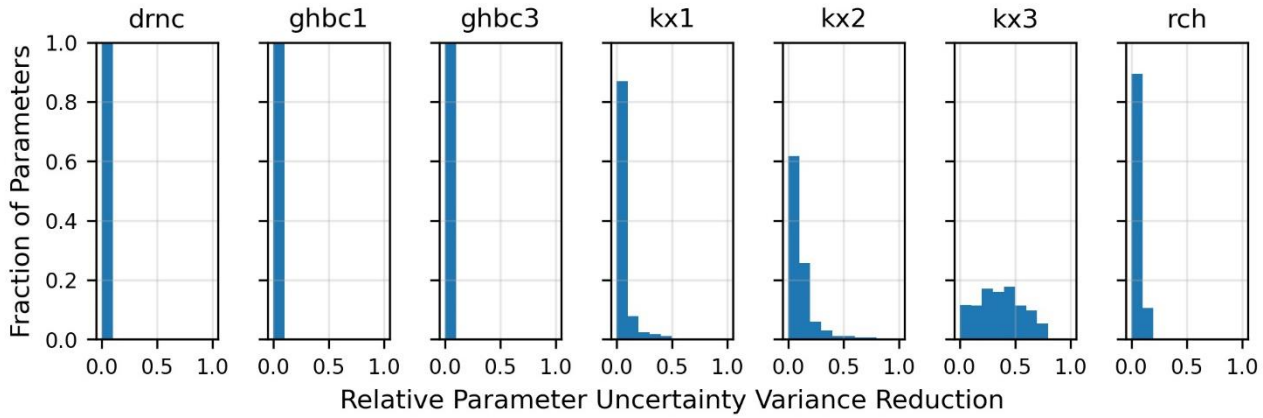

Figure 2 – Histograms of relative parameter uncertainty variance reduction per parameter group.

While Figure 2 provides an overview of RPUVR per parameter group, it does not provide information on its spatial distribution. A simple way to display this ratio spatially is to assign these values as parameter values to a model. They can then be displayed using a model interface such as a GUI or FloPy. Alternatively, using tools available in the PEST Groundwater Utilities suite (see the GMDSI tutorial on model visualisation), they can be transferred to GIS or 3D visualization platforms. The process begins by using PARREP and PEST as follows:

42 In the *pest-linunc* folder you should have a file named *cov-par6.par*. This is a "parameter value file". Make a copy of this file and name it *reluncred.par*.

43 Open *reluncred.par* in a text editor. The first line of this file contains the text "*single point*". From the second line onwards, there are four columns of values. These list parameter names, parameter values, scale and offset respectively. The first few rows of this file look like this:

```
  single point
      drncpp01     99.98309999999999          1.000000          0.000000
      drncpp02     99.99010000000000          1.000000          0.000000
      drncpp03     100.0015000000000          1.000000          0.000000
(…)
```

44 Replace values in the second column of *reluncred.par* (the parameter values) with values listed in the *Rel_uncert_var_redn* column of file *genlinpred_abbrev.out.* The first few rows of *reluncred.par* should now look something like this. (Exact formatting and spacing does not matter, as long as the first row is "single point", and as long as this row is followed by four columns separated by whitespace.)

```
single point
      drncpp01     0.0144745     1.0     0
      drncpp02     0.0137192     1.0     0
      drncpp03     0.0102902     1.0     0
(…)
```

45 Now that we have a parameter value file constructed with RPUVR values as parameter values, we can use PARREP to prepare a PEST control file in which RPUVR values are initial parameter values in the "parameter data" section of this file.

46 Open a command line window in the *pest-linunc* folder (if you have not already done so). Type the following command, and then press <enter>.

```
parrep reluncred.par linunc.pst linunc-red.pst 0
```

47 You should see the following text appear on your screen:

```
Reading parameter value file reluncred.par ----->
Data for 2205 parameters read from file reluncred.par.

Reading file linunc.pst and writing file linunc-red.pst ----->
File linunc-red.pst written ok.
```

48 Inspect the *pest-linunc* folder. You should see a new PEST control file named *linunc-red.pst*. If you inspect this file with a text editor, you will see that it has NOPTMAX set to zero and that initial parameter values are equal to parameter RPUVRs.

49 Run PEST_HP with *linunc-red.pst*, using one agent. (See the GMDSI tutorial on model calibration if you are unsure how to accomplish this). PEST will write the model input files and then attempt to run the model. Do not worry if the model fails. Our only concern is that PEST wrote "hydraulic property" values based on "parameter" values to model input files.

50 Inspect the agent's *runmodel* folder. If you inspect the MODFLOW 6 input files in the *runmodel\model* folder, you will find that parameter-informed model input values now correspond to RPUVR values obtained from *genlinpred_abbrev.out.*

51 Make a copy of the agent's *runmodel* folder. Name the copied folder *runmodel-relparuncred*.

52 Parameter-informed model inputs (which are now values of RPUVR) can now be plotted using your visualisation device of choice. The plots shown in Figure 3 were generated using FloPy and other Python libraries. Similar plots can be produced using most GUI's, as well as commercial plotting and visualization packages, with input files for these packages written by members of the PEST Groundwater Utilities suite.

Figure 3 shows the spatial distribution of RPUVR for hydraulic conductivities, recharge, and stream conductances. As we saw for Figure 2, the greatest relative uncertainty reduction was achieved for hydraulic conductivity parameters, followed by recharge. Stream conductance accrued an almost negligible amount of uncertainty reduction; you can just about see it if you squint very hard. Uncertainty reduction tends to concentrate where there are observation data, especially if there are also local stresses. See, for example, the two orange/red areas for K in layer 3; these coincide with

the locations of pumping wells in this layer. On the other hand, the dampening effects of boundary conditions (streams in this case) can limit uncertainty reduction for parameters in their proximity. Of note is the clear influence of the nested piezometer in reducing the local uncertainty of the hydraulic conductivity of the aquitard (layer 2).

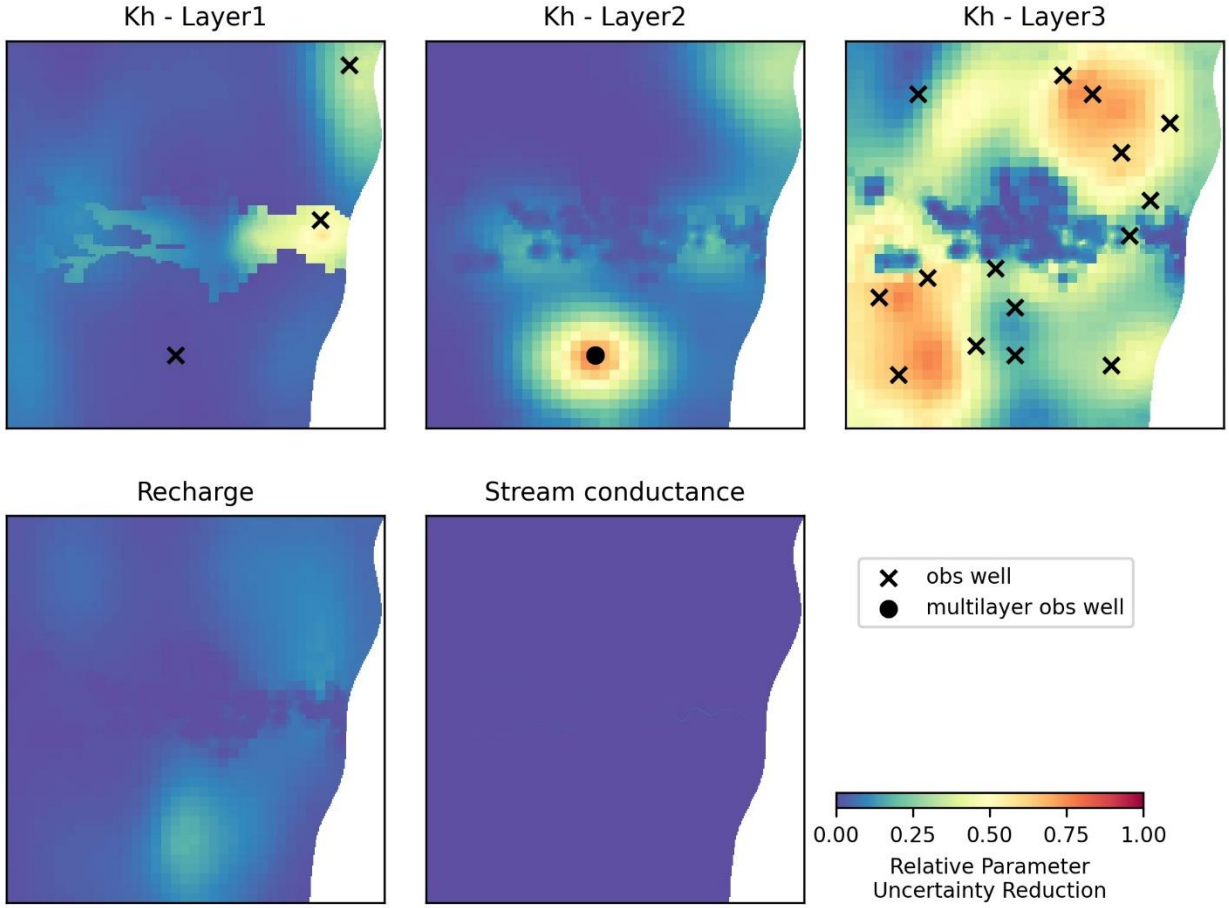

Figure 3 – Spatial distribution of RPUVR for (top, left to right) hydraulic conductivity in layers 1, 2 and 3 and (bottom, left to right) recharge rate and stream conductance. Values for GHB conductance are not shown as they are not very interesting. Black crosses denote piezometers in respective layers. The black circle marks the location of the nested piezometer, with measurements in both layer 1 and layer 3.

It is worth mentioning that these types of plots do not necessarily provide useful quantitative measures of posterior parameter uncertainty. However they provide excellent qualitative indicators of flow of information from a calibration dataset to parameters, and how this flow varies with location.

All files generated up to this point can be found in the folder named *completed*.

2.1 Predictive Uncertainty

Assessment of the uncertainty of a prediction is often the main motivation for linear analysis. PREDUNC1 (managed by GENLINPRED_ABBREV) calculates the prior and posterior uncertainty of a user-specified prediction.

53 With *genlinpred_abbrev.out* open in a text editor, find the line which contains the text "*PREDUNC1 analysis for prediction "p-trib_1""*. (It should be at line 2632.) You should see the following:

```
PREDUNC1 analysis for prediction "p-trib_1" ----->

    Pre-calibration:-
    Total uncertainty variance           =   5304716.
    Total uncertainty standard deviation =   2303.197

    Post-calibration:-
    Total uncertainty variance           =   28920.17
    Total uncertainty standard deviation =   170.0593
```

The above text box lists the variance and standard deviation of the *p-trib_1* prediction; it list both of its prior (pre-calibration) and posterior (post-calibration) uncertainties. As you can see, variance and standard deviation of the posterior uncertainty are much smaller than those of the prior. This indicates that history-matching reduced the uncertainty of this prediction. Now, with the assumption that the simulated value of *p-trib_1* obtained using calibrated model parameters represents the mean of a normal distribution, the range of possible predictive outcomes for this prediction can be determined.

If you inspect the output files which PEST produced when it calibrated the model (for example, *cov-par6.rei*, *cov-par6.res or cov-par6.rec*) you will see that the model's *p-trib_1* prediction when provided with calibrated parameters is 6899.620 m^3/d. For a variable that has a normal distribution, certain proportions of its possible values will fall within certain numbers of standard deviations from the mean. Approximately 68% of possible values lie within 1 standard deviation of the mean, 95% lie within two standard deviations of the mean, while 99% of possible values lie within 3 standard deviations of the mean). Thus, we expect 99% of possible values for the *p-trib_1* prediction to lie roughly within three standard deviations of 6899.620 m^3/d (i.e. within about 3 times 170 m^3/day of this value). Hence groundwater discharge to the stream should lie between 6389.6 m^3/d and 7409.6 m^3/d. Recall from Chapter 1.1 that the "bad thing" which management of our groundwater system seeks to avoid is discharge dropping below 6500 m^3/d. In this case, our posterior prediction uncertainty is too large to state unequivocally that the bad thing will not happen.

Figure 4 shows a nice way to visualize prediction uncertainty. Gaussian distributions are plotted using prior and posterior means and standard deviations. (This plot was generated using readily-available Python packages. Similar plots can be produced using other commonly used software such as MSExcel, Grapher, etc.) As you can see, the posterior uncertainty of the prediction is a lot smaller than its prior uncertainty. However the posterior uncertainty interval still contains the "bad thing" that we seek to avoid. Note that the mean of the prior (12,846 m^3/d) of the *p-trib_1* prediction was obtained using initial parameter values assigned before commencing calibration.

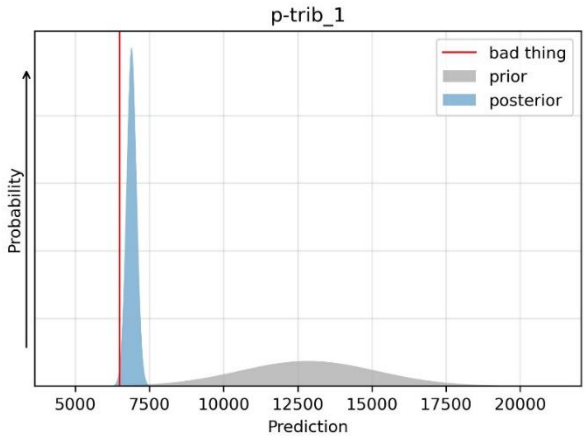

Figure 4 – Diagrammatic representation of the prior and posterior probability distribution of simulated values for the *p-trib_1* prediction Note that because probabilities that are evaluated using linear analysis are only approximate, no values are displayed on the y-axis to avoid misleading the reader.

All files generated up to this point can be found in the folder named *completed*.

2.2 Predictive Error Potential

If the Jacobian matrix (i.e. the matrix of sensitivities of model outputs used in the calibration process to parameters employed by a model) is subjected to singular value decomposition, parameter space can be subdivided into two orthogonal subspaces. One of these subspaces is referred to as the "calibration null space", while its orthogonal complement is referred to as the "calibration solution space". The null space is comprised of combinations of parameters that are completely uninformed by the history-matching dataset. The individual components of each of these parameter combinations are thus completely correlated with each other. Therefore, if they are varied in ratios that are defined by these combinations, they have no effect on model outputs that correspond to field measurements.

In contrast, each of the linear combinations of parameters that collectively comprise the calibration solution space is uniquely estimable on the basis of the calibration dataset. In most groundwater history-matching contexts, the dimensionality of the null space is far greater than that of the solution space. The dimensionality of the latter space can be viewed as the number of individual pieces of information that are accessible to model parameters through the history-matching process. Each such piece of information supports unique estimation of one so-called "super-parameter" (i.e. a linear combination of native parameters).

The PEST SUPCALC utility can be used to estimate the dimensionality of the calibration solution space.

54 With *genlinpred_abbrev.out* open in a text editor, find the line which contains the text "*PREDVAR1 analysis for prediction "p-trib_1"*". (This is line 2667.) You should see the following text. (The first and last three lines of this section of *genlinpred_abbrev.out* are shown here.)

```
  PREDVAR1 analysis for prediction "p-trib_1" ----->

      Sing_val       Null-space       Soln-space     Total
      index          term             term           variance          std_dev
      --------       ----------       ----------     --------          -------
         0             5304716.         0.000000      5304716.         2303.197
         1             5304491.        8.3643490E-04  5304491.         2303.148
         2             4606499.         23.53600      4606523.         2146.281
(…)
      2090          996.4186      1.0000000E+35  1.0000000E+35    3.1622777E+17
      2200          1.402273      1.0000000E+35  1.0000000E+35    3.1622777E+17
      2205          0.000000      1.0000000E+35  1.0000000E+35    3.1622777E+17
```

The values in the columns shown in the text box list the error variance of the *p-trib_1 prediction*. This is decomposed into contributions from its null-space component, its solution-space component. These are listed for increasing numbers of singular values. Figure 5 plots each term against the number of singular values. Both plots in Figure 5 display the same thing; in the right-hand plot the y-axis is log-scaled for visibility.

The number of singular values at which the total predictive error variance is minimized indicates the optimal dimensionality of the solution space. This number will be about the same for all predictions made by the model; however for some predictions the error variance will be reduced more than for others. Usually this number will be far less than the number of parameters featured in a model. Its upper limit is set by the number of observations comprising the calibration dataset. However it will normally be less than this because of repetition of information in different observations, or because of measurement noise associated with those observations. In the cuurent case, the number of number

of uniquely estimable parameters (20) is almost equal to the number of observations (21). This is an outcomes of:

- The small number of observations that comprise the calibration dataset;
- The small amount of noise associated with these observations; and
- The fact that “structural noise” arising from model defects does not contribute to model-to-measurement misfit, as reality is actually defined by the model itself in our synthetic, make-believe world.

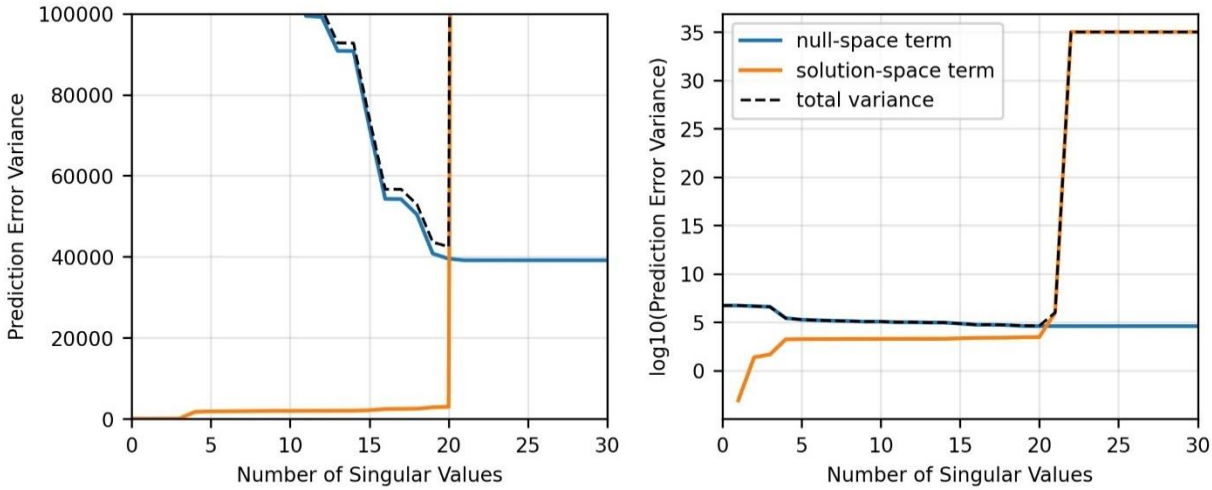

Figure 5 – Predictive error variance vs number of singular values used in history-matching. Both plots display the same thing. The left y-axis is truncated at 1e5, while the right y-axis is log-scaled for visibility. The x-axis is truncated at 30 singular values for visibility.

2.1 Parameter Group Contributions to Prediction Uncertainty

PREDUNC4 calculates the contributions made by groups of parameters to the uncertainty of a prediction. (Note that GENLINPRED can calculate contributions made by individual parameters; however this can be very time-consuming.) These can be informative when considering whether simplifications that are embodied in a model’s design degrade its performance, or whether certain parameters can be omitted from the history-matching process (for example if available computational time is limited).

In the current case for example, GHB boundary conditions were assigned to the western edge of the model domain in order to reduce its lateral extent. These GHBs are a simplified representation of the broader groundwater system which lies on the other side of this boundary. If GHB conductance parameters contribute only minimally to the uncertainties of decision-critical predictions, then this simplification is reasonable for a model which is constructed to make those predictions.

55 With *genlinpred_abbrev.out* open in a text editor, find the line which contains the text “*PREDUNC4 analysis for prediction "p-trib_1"*”. (This is line 2644.) You should see the following:

```
PREDUNC4 analysis for prediction "p-trib_1" ----->

    Contributions to predictive uncertainty variance

    Parameter      Pre-cal            Post-cal
    group          contribution       contribution
    ---------      ------------       ------------
    kx1              857769.0           4879.940
    kx2              81123.00           5346.490
    kx3              4139429.           17804.39
    rch              156297.0           3322.980
    strc             33180.00           322.5300
    ghb1             5375.000           10.43000
    ghb3             31545.00           60.05000
```

The three columns of data listed in the above text box represent contributions to the uncertainty variance of the *p-trib_1* prediction by each parameter group before and after history-matching. These three columns of data can be copied into any spreadsheet or plotting package for visualisation. (Figure 6 was produced using Python-callable plotting functions; note that the y-axis is log-scaled.)

A comparison of pre- and post-calibration values for each parameter group reveals the effectiveness (or otherwise) of calibration in reducing uncertainties for parameters belonging to each group. As you can see from Figure 6, calibration decreased contributions to predictive uncertainty from all parameter groups. (This is good news; history-matching was worth the trouble).

As is discussed above, evaluation of parameter contributions to predictive uncertainty can be useful for assessing whether a particular type of model simplification erodes a model's adequacy for a particular management purpose. In our case, GHBs comprise simplifications of groundwater processes that take place outside of the model domain. Let us take a look at the contribution of GHB conductance parameters in layer 1 (*ghb1*) and layer 3 (*ghb3*) to the uncertainty of our *p-trib_1* prediction. As you can see in Figure 6, even before calibration, both of these parameter groups contributed the least to the uncertainty of this prediction (although not much less than stream conductance). However, after calibration their contribution to predictive uncertainty is even further decreased. Both of these groups contribute at least an order of magnitude less to the uncertainty of the *p_trib_1* prediction than any of the remaining parameter groups. The uncertainty of this prediction is dominated by hydraulic conductivity and recharge parameters. We can therefore conclude that simplifications embodied in the western GHB are not inimical to the model's management purpose.

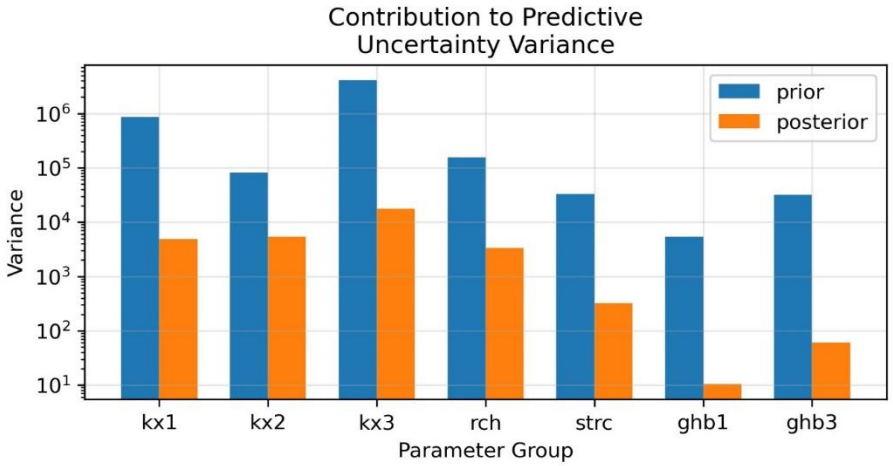

Figure 6 – Comparison of parameter group contribution to the prior and posterior uncertainty of prediction *p-trib_1*.

2.2 Data Worth

The worth of data is measured by their ability to reduce the uncertainties of model predictions that we care about. GENLINPRED_ABBREV evaluates the worth of observations comprising an observation group in two different ways. In the first of these ways, it calculates the posterior uncertainty of a decision-critical prediction under the assumption that this group comprises the entirety of the observation dataset. The worth of this data is indicated by how much the uncertainty of a decision-critical prediction falls relative to its prior uncertainty. In the second of these ways GENLINPRED_ABBREV removes the observation group from the existing calibration dataset. The worth of these data is indicated by how much this causes the posterior uncertainty of the decision-critical prediction to rise. Note that GENLINPRED, in contrast to GENLINPRED_ABBREV, can carry out this same process for individual observations (as well as for observation groups); however this can be numerically time consuming. The PREDUNC5 utility (which is run by both GENLINPRED and GENLINPRED_ABBREV) can undertake more sophisticated calculations of data worth.

56 With *genlinpred_abbrev.out* open in a text editor, find the line which contains the text "*PREDUNC5 observation subtraction analysis for prediction "p-trib_1"*". (This is line 2660.) You should see the following:

```
PREDUNC5 observation subtraction analysis for prediction "p-trib_1" ----->

Increases in predictive uncertainty variance incurred through loss of
observations

      Observation_group        Variance_increase
      -----------------        -----------------
      heads1                     91.59000
      heads3                     20781.40
      streams                    2914662.
      headiff                    48.33000
      pstreams                   0.000000
```

57 The first of these columns list observation groups; the second lists the increase in predictive uncertainty variance incurred by omission of the observation group from the calibration dataset.

58 Just below the previous section, you should see a similar table. This describes the effects of adding each observation group to an otherwise empty calibration dataset. In this case, the two columns list observation groups, and the corresponding decrease in predictive uncertainty variance from its prior uncertainty that is attained if each observation group comprises the entirety of the calibration dataset.

```
PREDUNC5 observation addition analysis for prediction "p-trib_1" ----->

Decreases in pre-calibration predictive uncertainty variance incurred through
addition of observations

      Observation_group        Variance_decrease
      -----------------        -----------------
      heads1                     1231498.
      heads3                     2355850.
      streams                    5253921.
      headiff                    25.00000
      pstreams                   0.000000
```

The contents of the two previous text boxes are plotted in Figure 7. Each of these boxes provides a different measure of the information content of the different observation groups.

The increase in prediction uncertainty accrued through loss of an observation group from the entire calibration dataset (Figure 7, left) is a measure of the uniqueness of the information content of that observation group with respect to the prediction. It represents how much predictive uncertainty increases if a group is removed, assuming that all other observation groups are still part of the history-matching dataset.

The decrease in prediction uncertainty incurred through addition of an observation group to an otherwise empty calibration dataset (Figure 7, right) is a measure of the inherent information content of that observation group.

Figure 7 (left) shows that the *streams* and *head3* observation groups (which pertain to groundwater discharge to streams and heads in layer 3, respectively) contain a lot more unique information that is pertinent to future stream inflow predictions than members of the *head1* or *headdiff* observation groups. This is unsurprising for *head3*, as it contains a lot more observations than any of the others. In contrast, although the *streams* observation group contains only two observations, the information that is contained in these observations is very relevant to the model prediction; this is unsurprising as the prediction pertains to groundwater discharge to the streams. Figure 7 (right) shows that the absolute information content of each observation group with respect to this prediction is similar, with the exception of *headiff*. The latter does very little to reduce the uncertainty of the stream inflow prediction.

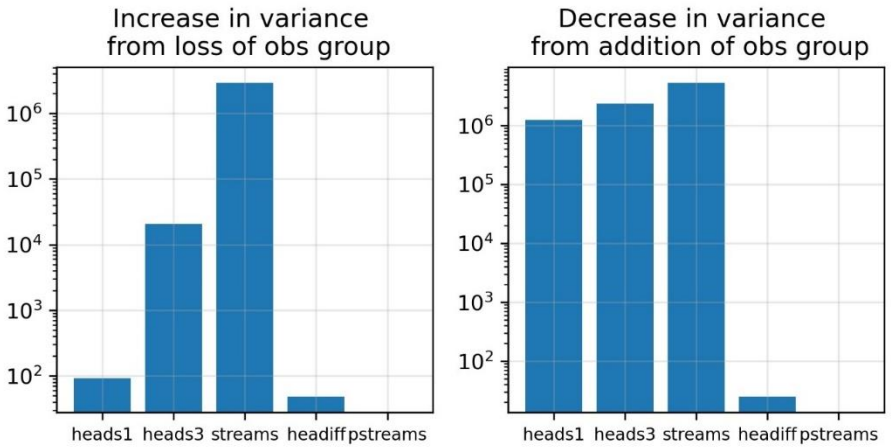

Figure 7 – Contribution to predictive uncertainty from (left) loss and (right) addition of observation groups to the history-matching dataset.

The outcomes of analyses such as those exemplified above provide useful insights into the information content of an observation dataset, and how this information is distributed between its members. These insights can be useful, for example, for prioritization of data collection. It is important to note that these same analyses can be undertaken in order to assess the worth of observations for which measurements do not yet exist. The equations on which linear uncertainty analysis are based do not include terms that represent the actual values of observations; only the sensitivities of model-generated counterparts of observations to model parameters are required. Therefore, linear analysis can be used to assess the ability (or otherwise) of yet-ungathered data to reduce the uncertainty of decision-critical predictions. This topic will be addressed in another GMDSI tutorial.

2.3 Final Words

This tutorial demonstrates how easy-to-use members of the PEST utility suite can provide a wealth of useful information on model parameter and predictive uncertainty. The current tutorial provides only a brief introduction to these utilities; they can accomplish much more than is demonstrated herein. Hopefully this tutorial provides a starting point for their continued use in exploring data worth and the flow of information in real-world decision-support contexts.

Some of the GMDSI Worked Examples demonstrate applications of linear analysis in the real-world. Other GMDSI tutorials explore more granular application of some of these utilities for common tasks. As has been mentioned, the PEST Roadmap 12: Linear Analysis provides an overview of the theory on which they are based, along with some practical advice. Some videos available through the GMDSI Webinar series also discuss linear analysis and provide insights into their real-world applications.

gmdsi

Groundwater Modelling
Decision Support Initiative

gmdsi.org

BHP

Flinders UNIVERSITY

NATIONAL CENTRE FOR GROUNDWATER RESEARCH AND TRAINING

RioTinto