

FePEST in FEFLOW 7.0

User Guide

FePEST is the graphical user interface for running FEFLOW models with the software PEST developed by J. Doherty.



PLEASE NOTE

COPYRIGHT

This document refers to proprietary computer software which is protected by copyright. All rights are reserved. Copying or other reproduction of this manual or the related programs is prohibited without prior written consent of DHI. For details please refer to your 'DHI Software Licence Agreement'.

LIMITED LIABILITY

The liability of DHI is limited as specified in Section III of your 'DHI Software Licence Agreement':

'IN NO EVENT SHALL DHI OR ITS REPRESENTATIVES (AGENTS AND SUPPLIERS) BE LIABLE FOR ANY DAMAGES WHATSOEVER INCLUDING, WITHOUT LIMITATION, SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES OR DAMAGES FOR LOSS OF BUSINESS PROFITS OR SAVINGS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION OR OTHER PECUNIARY LOSS ARISING OUT OF THE USE OF OR THE INABILITY TO USE THIS DHI SOFTWARE PRODUCT, EVEN IF DHI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY TO CLAIMS OF PERSONAL INJURY TO THE EXTENT PERMITTED BY LAW. SOME COUNTRIES OR STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL, SPECIAL, INDIRECT, INCIDENTAL DAMAGES AND, ACCORDINGLY, SOME PORTIONS OF THESE LIMITATIONS MAY NOT APPLY TO YOU. BY YOUR OPENING OF THIS SEALED PACKAGE OR INSTALLING OR USING THE SOFTWARE, YOU HAVE ACCEPTED THAT THE ABOVE LIMITATIONS OR THE MAXIMUM LEGALLY APPLICABLE SUBSET OF THESE LIMITATIONS APPLY TO YOUR PURCHASE OF THIS SOFTWARE.'

PRINTING HISTORY

November 2015



1	Introduction	1
1.1	What Is PEST?	1
1.2	Where to Get PEST?	2
1.3	What Is FePEST?	2
1.4	Where to Start...	2
1.4.1	If You Are New to PEST	2
1.4.2	If You Are an Experienced PEST User	3
1.5	Related Literature	3
1.6	Overall Workflow	4
1.7	Acknowledgements	4
2	Methods and Concepts	7
2.1	GLMA Search Algorithm	7
2.1.1	Derivative Calculation	10
2.1.2	Observation Weights	11
2.2	Pilot-Point Method	12
2.3	Parameter Non-uniqueness	14
2.4	Prior Knowledge	14
2.4.1	Prior Information	15
2.4.2	Tikhonov Regularization	15
2.4.3	Regularization of Pilot-Point Parameters	16
2.5	Subspace Regularization	19
2.5.1	Truncated Singular Value Decomposition	20
2.5.2	SVD-Assist	20
2.5.3	Least Squares (LSQR)	21
2.6	Calibrated-Constrained Monte Carlo Analysis	21
2.6.1	Stochastic Parameter Generation	22
2.6.2	Null-Space Projection	22
2.6.3	Readjustment of Parameters	23
2.6.4	Outcome of Calibration-Constrained Monte Carlo	23
3	Fundamental Setup	25
3.1	FEFLOW and FePEST	25
3.2	Parameter Definitions	26
3.2.1	Assignment method	26
3.2.2	Parameter Settings	28
3.2.3	Parameter Groups	30
3.3	Observation Definitions	30
3.3.1	Definition of Observations	31
3.3.2	Observation settings	33
3.3.3	Observation Groups	35
3.4	Prior Information	35
3.5	Regularization	36
3.5.1	Tikhonov Regularization	36
3.5.2	Subspace Regularization	38
3.5.3	Regularize by Super Parameters: SVD-Assist	39
3.6	Parallelization	39



4	Parameter Estimation: History Matching/Calibration	43
4.1	Required Settings	43
4.1.1	Optimization Control	43
4.1.2	Other Settings	43
4.2	Starting PEST	44
4.2.1	Preflight Checks	44
4.2.2	Running PEST	44
4.2.3	Output during PEST Run	45
4.2.4	Output after PEST Run	47
4.3	Export to FEFLOW	48
5	Predictive Analysis: Best and worst case scenarios	49
5.1	Required Settings	49
5.1.1	Optimization Control	49
5.1.2	Observation and Type	49
5.1.3	Objective Function Limits	50
5.1.4	Other Settings	50
5.2	Starting PEST	50
5.3	Output during Predictive Analysis	50
5.4	Estimation Results	51
6	Pre-calibration Monte Carlo	53
6.1	Criteria for randomly generated parameters	53
6.2	Optimization of random parameter sets	54
6.3	Running FePEST under Pre-calibration Monte Carlo mode	55
6.4	Results of the Pre-calibration Monte Carlo Analysis	56
7	Post-Calibration Monte Carlo	59
7.1	Settings of the Post-Calibration Monte Carlo	59
7.2	Running FePEST under post-calibration Monte Carlo mode	60
7.3	Results of the Post-calibration Monte Carlo Analysis	60
8	Other Tools	63
8.1	Sensitivity Export	63
8.1.1	Required Settings	63
8.1.2	Export of Sensitivities	63
8.2	Model Stability Tests (JACTEST)	64
8.2.1	Using JACTEST	64
8.3	Linear Analysis (GENLINPRED)	65
8.4	Sensitivity Analysis (SENSAN)	67
8.4.1	Definition of Parameter Variability Sets	68
9	Customized PEST Setups	71
9.1	Creating the PEST Setup	71
9.2	Convenience Tools	72
9.2.1	File Structure	72
9.3	Running Customized PEST Setups	74
9.4	Integrating IFM Plug-ins and Third-Party Code	75



10	FePEST Tutorial	77
10.1	Conceptual Model	77
10.2	Field Measurements and Reference Data	77
10.3	FEFLOW Model Preparation	78
10.4	FePEST Initial Configuration	79
10.5	Definition of a New FePEST Problem	80
10.6	Optimization Settings	81
10.7	Parameter Definition	82
10.8	Observations Definition	87
10.9	Parallelization Mode	90
10.10	Estimation Method and Regularization Setup	90
10.10.1	Regularization Setup	91
10.10.2	Settings of SVD-Assist	92
10.11	Running PEST	92
10.12	Postprocessing PEST Results	93
10.12.1	Optimization Results in FePEST	93
10.12.2	Results Postprocessing in FEFLOW	95
10.12.3	Sensitivity Estimation	99
10.12.4	Additional Exercises	100





1 Introduction

This chapter aims at FEFLOW users that may or may not have prior experience with PEST. Various references to the PEST documentation and other texts are provided within the text for further reading.

The methods are described in their functional relation to the overall workflow and their relation to other methods. This initial orientation will help to understand how PEST works, and should motivate the user on the way of becoming proficient in uncertainty modeling.

The focus will be (with some exceptions) on those PEST methods that are actually supported by FePEST. It should be emphasized that PEST provides many more, exciting features, which will become available in future releases of FEFLOW. To use them already now, we encourage you to see the PEST manual for further information.

1.1 What Is PEST?

PEST is a software widely used in environmental modeling to calibrate models, to determine uncertainty associated with parameters and predictions, and for related tasks. Today, PEST is probably the most commonly used software for the calibration of groundwater models.

However, PEST provides much more than calibration. Besides assisting the modeler in classical calibration tasks, it implements methods that address the fact that the outcome of a calibration is not unique, and that the prediction given by a calibrated model is only one out of many possibilities.

Instead of only providing one calibrated model, PEST aims to analyze the spectrum of possible solutions and consequently the uncertainty range associated with parameters and predictions. Of course, these methods (for both traditional calibration and uncertainty analysis) must be learned and understood before they can be successfully applied. Fortunately, good literature is available to do so. See the literature review (section 1.5) for more information.

PEST is model-independent. Any modeling software that reads input and writes output from a file—or can be adapted to do so—can be linked to PEST. The effort to do this depends on the complexity of the file format.

On a more technical level, PEST can be seen as a toolbox of different programs to setup, run, and evaluate the results of a specific task (e.g. calibration). These programs are all started from the command-line prompt, and are configured using command-line parameters and/or configuration files without a graphical user interface. FePEST has been developed to provide more convenient access to PEST functionality when using FEFLOW models—without limiting it.



1.2 Where to Get PEST?

PEST is free software. It is developed by John Doherty (Watermark Numerical Computing) and can be downloaded, including all documentation, from www.pesthomepage.org.

Please also see the FePEST installation instructions for further information.

1.3 What Is FePEST?

FePEST links PEST with a FEFLOW model through a convenient graphical user interface.

With FePEST it is possible to use a range of PEST functionality without the need to learn the syntax of its various command line tools and configuration files, and to manually adapt FEFLOW's model files for usage in PEST. Visual feedback on the optimization progress is provided during and after the model run. This will give the user major savings of time both in terms of learning and productive work.

The range of supported PEST features in this release of FePEST includes:

- Model calibration (including Regularization and pilot point parametrization)
- Subspace methods
- Parallelization (locally and/or using remote servers, incl. file transfer)
- Linear sensitivity analysis
- Predictive analysis (worst-/best case evaluation)

1.4 Where to Start...

1.4.1 If You Are New to PEST

Most PEST features required for model calibration using PEST are accessible through FePEST. The user is in many cases not required to make changes to the PEST files and therefore does not need to learn the syntax of the files and commands of PEST.

However: It is still essential that the methods of PEST are understood to be able to interpret the results of a PEST run correctly! It is suggested that the time saved on understanding PEST's file structure and command line tools is invested in understanding these methods.



1.4.2 If You Are an Experienced PEST User

PEST provides more functionality than FePEST can cover. We have chosen to implement those work flows that will be most relevant for most users (and more will be implemented in the future).

This does not pose a limitation to use FePEST for other methods at all. In fact, it is developed in an "open" way thus that the benefit of a rapid PEST file setup can be used even for unsupported methods. The following aspects will support experienced users to customize their setups:

- The user interface cites the original names of PEST variables and tools. Experienced users will therefore quickly recognise how the user interface relates to the respective PEST functionality.
- The file setup created by FePEST strictly follows the syntax specified in the PEST documentation and is fully accessible. The structure of these files - including several batch-files - are designed to allow adaptations for different purposes.
- Especially if the optimization involves the pilot-point method, experienced users will use FePEST for the otherwise elaborate fundamental work steps like the definition of observations and parameters, and then adapt the resulting PEST setup for the particular purpose.
- Users familiar with the FEFLOW IFM programming interface can use a respective feature in FePEST to allow IFM plug-ins to communicate with PEST directly. This extends the scope of FePEST beyond the predefined observation and parameter types.

1.5 Related Literature

- The following list of literature should help users new to PEST to find a reasonably easy access to the science behind its tools. Initially, PEST might be sought as a tool to accelerate model calibration. PEST has a very high potential to accomplish this task if the underlying concepts are sufficiently understood. Later, more advanced methods might be applied to understand the uncertainties associated with calibrated models and predictions made by them.
- The document **Use of PEST and Some of its Utilities in Model Calibration and Predictive Error Variance Analysis: A Roadmap** provides a first overview.
- For a more in-depth understanding, the PEST tutorial **Methodologies and Software for PEST-Based Model Predictive Uncertainty Analysis** is recommended. It provides a comprehensive introduction to basic and advanced methods, and conveys important knowledge of the concepts behind them.
- The **PEST Users Manual** and the **Addendum to the PEST manual** are the primary and most complete reference to all PEST features and tools. This manual will refer to these documents regularly for further reading.



- Another useful document **Getting the Most out of PEST** describes some general settings and procedures that avoids a major part of typical problems. FePEST uses a major part of these recommendations by default.

See www.pesthomepage.org for additional documentation on PEST.

1.6 Overall Workflow

The structure of this document follows the work flows to perform the supported tasks in FePEST.

After a brief introduction to some of PESTs methods and algorithms (Section 2), the following tasks are explained:

- **Fundamental Problem Setup (Section 3)**
The fundamental problem setup prepares the model to be processed by PEST. These configurations form the basis for all basic and advanced PEST methods. It includes the definition of adjustable parameters, observation and prior knowledge, the choice for subspace regularization methods and the setup of remote servers for parallel computing.
- **Parameter Estimation/History Matching/Calibration (Section 4)**
The history matching process targets the estimation of a parameter set that satisfies both the historical observations and the prior knowledge. After doing the required settings and starting PEST, the feedback of PEST is reviewed and the resulting model is opened in FEFLOW.
- **Predictive Analysis (Section 5)**
For most environmental models it is possible to find more than one calibrated model, with predictions varying at different levels. Finding maximum or minimum possible key predictions among these models is a simplistic approach to identify worst-case or best-case scenarios of well-posed problems that are compatible with the calibration data set.
- **Sensitivity Analysis (Section 8.1)**
FePEST allows to export the sensitivities of the parameters. These can be used to create sensitivity maps in FEFLOW, or be processed with other software or PEST tools.
- **Customized PEST Setups (Section 9)**
Experienced PEST users may want to use PEST methods that go beyond the functionality of FePEST.

1.7 Acknowledgements

The authors like to thank John Doherty of Watermark Numerical Computing for his ongoing support of the development of FePEST and its documentation.

John Doherty is the author of PEST and as such provides regular training courses in calibration and predictive uncertainty analysis of numerical models. His documentation of PEST, as well as various papers and tutorial exer-



cises on this topic are a valuable source of information learning-stage and on-going work with this software.

Parts of the text and several illustrations used in this manual have been derived from his work.

The FEFLOW model shown in Figures 2.6, 2.9 and 2.12 is based on work by J. Doherty and C. Moore.





2 Methods and Concepts

The fundamental methods applied by PEST during the calibration process are briefly described in the following. As already mentioned in Section 1, this description allows the reader to quickly understand their respective role in the overall workflow. The PEST documentation, to which specific references are provided, describes the methods in full detail.

2.1 GLMA Search Algorithm

The central feature of the PEST engine is the GLMA search algorithm, that iteratively optimizes the model parameters to improve its fit to observed data and other objectives.

The fit to the observations is hereby expressed through the **Measurement Objective Function**. In the simplest case, this will be the weighted sum of squares of the residuals between measurement and simulation results:

(2.1)

$$\Phi = \sum_i w_i (h_i^{obs} - h_i^{sim})^2$$

where (h^{obs} denotes an observation (typically from a field measurement), h^{sim} its related simulation result, and w the weight that has been applied to the measurement (observation weights will be discussed in Section 2.1.2).

The equation shown is also the formulation FePEST uses to defines the measurement objective function. The observations h_i are loaded from the FEFLOW model and the weights w_i can be changed by the user within the user interface (Figure 2.1)

Observations					
Definitions:					
Hydraulic head (hea)					
Mass concentration (mas)					
Def.	Observation type	Obs. point	X	Y	Z
mas	Mass concentration	TravelTime	250.00	0.00	0.00
hea	Hydraulic head	well01	65.00	615.00	0.00
hea	Hydraulic head	well02	245.00	615.00	0.00
hea	Hydraulic head	well03	425.00	615.00	0.00
hea	Hydraulic head	well04	65.00	435.00	0.00

Figure 2.1 The objective function is defined through the observation definition in FePEST.

The search algorithm used in PEST is the **Gauss-Levenberg-Marquardt algorithm (GLMA)**. The GLMA changes the model parameters until a minimum objective function value is found. Running PEST, the user will observe two working steps per iteration:

- Derivative calculation: The parameters are changed incrementally. By repeating the model run for each parameter, and observing the resulting changes of observation values, the partial derivative for each pair of parameter and observation can be calculated by finite-difference approximation. These derivatives form the elements of the **Jacobian matrix**. The numerical effort to calculate the Jacobian matrix usually dominates the iteration.
- The parameter values are adjusted aiming to reduce the objective function. The direction and magnitude of the adjustment is expressed by the **parameter upgrade vector**. To identify the optimal direction of this vector, the GLMA uses a combination of two strategies:
- While the objective function shows a predominantly linear behavior, the method of gradient descent is applied. This method determines the parameter upgrade vector from the direction of steepest descent of the objective function. This can often be observed during the initial phase of the optimization.
- Objective-function nonlinearity is addressed via the Gauss-Newton method. This method computes a parameter upgrade vector based on the presumption of a quadratic behavior of the objective function.

The two methods are not mutually exclusive: The GLM algorithm interpolates between them, controlled by a scaling parameter (the **Marquardt-Lambda**).

PEST dynamically updates lambda depending on the progress in reducing the objective function. The current lambda as displayed by FePEST during the PEST run is a good indicator for the current nonlinearity of the objective function.

- high lambda values (e.g., > 10) indicate linear behavior (and predominant use of the gradient descent method).
- small lambda values (e.g., < 2) indicate nonlinear behavior (and predominant use of the Gauss-Newton method).



Figure 2.2 and Figure illustrate the development of the objective function and the Marquardt lambda during a typical PEST optimization. Gradient descent is used in the first iterations, indicated by higher lambda values. When the objective function approaches its (local) minimum, Lambda falls to near zero indicating almost exclusive use of the Gauss-Newton method

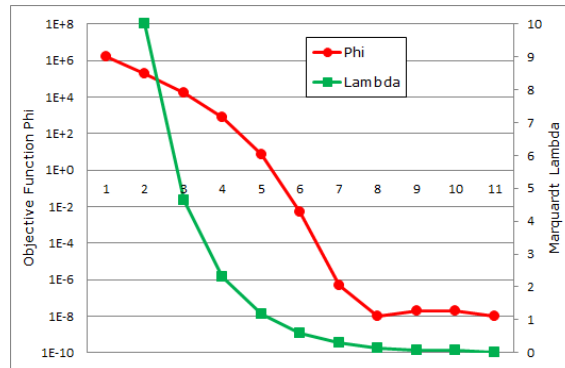


Figure 2.2 Development of the objective function and the Marquardt lambda during a PEST run.

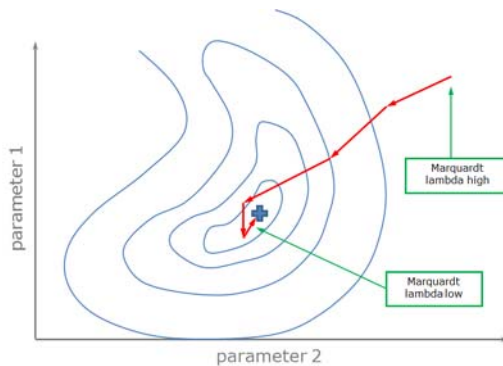


Figure 2.3 Schematic illustration of contours of the objective function and the path of the parameter upgrades vectors, after Doherty

If successful, the GLMA will find a parameter set that constitutes a **local minimum** of the defined objective function. This is an important restriction because multiple local minima might be present, and it is not guaranteed that the one found is also the global minimum.

It is therefore possible that different PEST runs result in different parameter sets if the iteration starts at different initial parameter values. These should therefore be chosen as close as possible to those values that are expected.

The modeler should also critically review the resulting parameter set and the model-to-measurement-misfit (see Figure 2.4). Strong, but also very low (see section Section 2.4) departures indicate potential problems with the optimization.

Further reading: PEST Manual (5th Ed.), Ch. 2.1: The Mathematics of PEST.

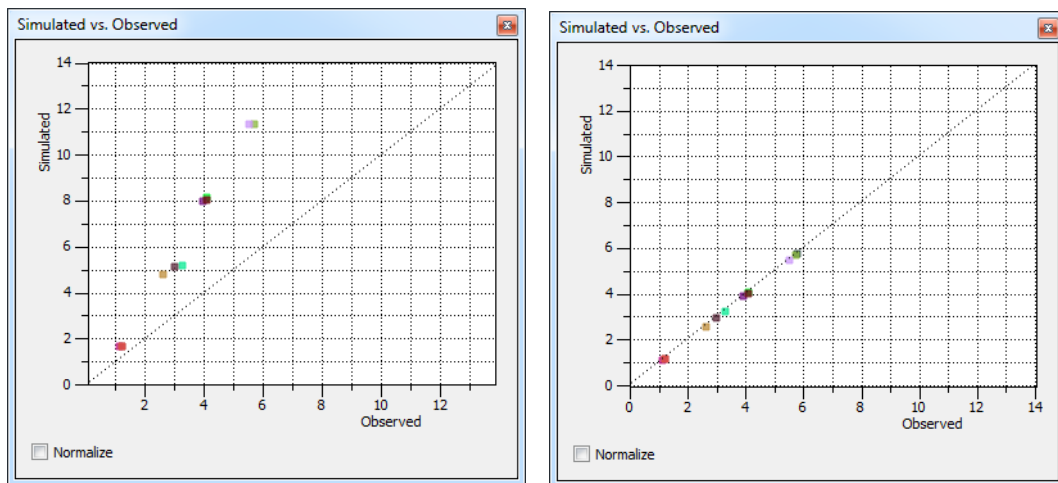


Figure 2.4 Scatter plot of simulated vs. observed data before (top) and after (bottom) optimization.

2.1.1 Derivative Calculation

The calculation of derivatives is a fundamental element of the GLM algorithm.

The derivatives are calculated through numerical differentiation. Each parameter is incrementally changed, and the model is run each time to calculate and record the resulting change of the model observations. The derivative of each parameter \hat{a} observation relationship is then calculated through finite-difference techniques.

Correct calculation of derivatives is of critical importance to the optimization, as failure to it will lead to an unstable optimization procedure and PEST will not be able to lower the objective function.

Model instabilities are a frequent cause of PEST failures!



Model instabilities are a frequent cause of PEST optimization failure!

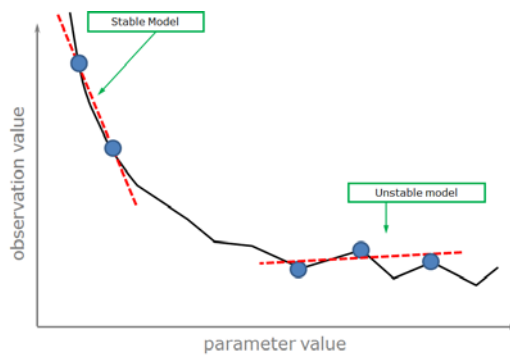


Figure 2.5 Calculation of derivatives during stable (left) and unstable (right) model behavior. If the model is unstable, random noise dominates the physical change of the observation value. PEST tries to compensate with higher-order finite-difference methods.

Instabilities introduce noise to the observation, which is random and not related to the physical-based simulation result. Figure illustrates this effect. If this noise - and not the incremental parameter change - dominates the observation response, the direction of the upgrade vector becomes random itself and the optimization will fail.

Even though certain countermeasures are available (see Section 3.2.3), the modeler should always aim at maintaining maximum stability of the FEFLOW model. With **JACTEST** (Section 8.2) PEST provides a tool to check the integrity of the derivatives calculation for specified parameters.

Further reading: PEST Manual (5th Ed.), Ch. 2.3: The Calculation of Derivatives.

2.1.2 Observation Weights

The weight of an observation controls how strong its residual (the deviation between computed and measured result) contributes to the measurement objective function. A reasonable choice of weights can positively influence the convergence behavior and result of the GLM algorithm.

Different weighting strategies can be applied (alone or in combination), some examples are given in the following.

Weighting by Measurement Noise

A common strategy of adjusting observation weights is applying the inverse of its expected measurement noise as a weight factor. The contribution of less trustworthy observation values to the measurement objective function is reduced, limiting the risk of inaccurate measurements having a negative impact on the optimization and leading to the estimation of parameter values which are thereby in error.



Weighting by Absolute Measurement Value

The absolute values of observations in a PEST optimization can encompass several orders of magnitudes, especially (but not limited to) if observations of different types are involved (e.g., Hydraulic head [m] and Mass-concentrations [mg/l]).

Observations with small values are therefore under-represented in the measurement objective function. Normalizing the values by assigning a weight equal to the inverse of the absolute compensates for this effect and makes sure that the information contained in these values finds appropriate representation in the optimization.

Equalizing Observation-Group Contributions

Observations of different types are assigned to different observation groups. One may also decide to manually assign observations of the same type to different observation groups. Using a spreadsheet or the PEST tool PWTADJ1 observation weights can be adjusted to equalize the total contribution of each observation group to the total objective function at the start of the optimization process. This helps to ensure that the information that is contained in each of these observation groups is used in estimation of model parameters, and not undervalued or overvalued because of too low or too high a contribution to the initial objective function.

De-clustering

Observations can be correlated. Water level measurements at observation wells close to each other are often not independent. It is likely that values and changes at these wells are similar, and the worth of information contributed by one well is diminished because it was already contributed by a different well in its vicinity. The worth of the information provided by each of the wells is therefore lower than a separate measurement at a larger distance. In this case, the weight of correlated observations should be reduced.

Time series are good examples of this principle as well. Daily measurements of the groundwater level might carry the same worth of information as a measurement taken on a monthly basis does. Because the daily measurement has more measurement points, it would be over-represented if the weights of each of these measurement points are not compensated for. This makes it advisable to normalize the weights of observations of time series by the sampling rate of the measurement.

Further reading: PEST Manual (5th Ed.), Ch. 2.1.2: Observation Weights.

2.2 Pilot-Point Method

The pilot-point method defines parameters as a spatially variable distribution.



In the classical calibration approach, it is a common assumption that geologic formations have spatially constant parameter values. In reality, this is rarely true.

Therefore, instead of applying a homogeneous parameter value across a zone, varying values for the parameter are assigned at particular locations (the **pilot points**). Each pilot point represents an adjustable parameter in PEST. An interpolation method then creates a continuous distribution of this parameter. Figure 2.6 illustrates the method.

The resulting large number of parameters adds to the degrees of freedom in the inversion process. This will generally lead to a better fit to the measurement data. At the same time, it will increase the level of nonuniqueness and therefore better reproduce the uncertainty associated with the model predictions.

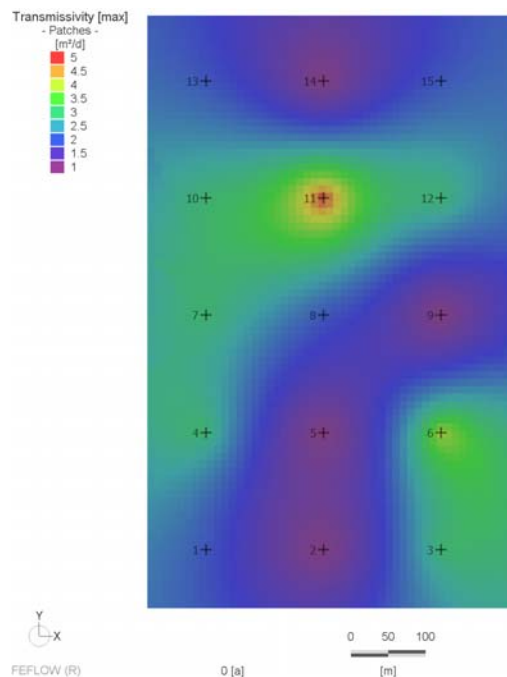


Figure 2.6 Distribution of hydraulic transmissivity, interpolated from a set of 15 pilot points.

Pilot points often lead to lower objective functions and better fits to measured data. However, the modeler should be aware of the risk of **over-fitting** the parameter field and should always check that the solution is plausible in a geological sense, and regularize it if necessary (see Section 2.4.2).

Further reading: PEST Groundwater Data Utilities, Part A, ch. 5: Model Parameterization based on pilot-points.



2.3 Parameter Non-uniqueness

A typical challenge when history-matching (calibrating) an environmental model is the inherent **non-uniqueness** associated to the inverse solution. Usually many different parameter sets exist which are all compatible with the historical observation data.

Observation data is usually sparse and usually not sufficient to uniquely identify more than just a few of the large number of model parameters that can be made adjustable.

This has two consequences:

- Different calibrated parameter sets lead to different predictions. This makes it difficult to use a single model alone for decision-making.
- Some or many of the parameters will be insensitive to observations. The GLMA-based optimization process can become unstable under this condition, leading to long optimization run-times or even failure to optimize.

Regularization techniques can provide a defence against these issues. They restrict the parameter search to identifiable parameters, either by adding additional constraints to the parameters (**Structural Regularization**, **Tikhonov Regularization**) or separating identifiable parameters from non-identifiable parameters (**Subspace Regularization**).

This manual restricts itself to Tikhonov regularization (discussed in Section 2.4.2) and Subspace regularization (discussed in Section 2.5).

See the PEST tutorial *Methodologies and Software for PEST-Based Model Predictive Uncertainty Analysis*, pp. 46 (see the literature review, Section 1.5), for a very good discussion on different regularization techniques.

2.4 Prior Knowledge

Prior knowledge is introduced in the optimization if some knowledge about the estimated range of parameters values exists. (This is also referred to as pre-calibration parameter probability.)

The general procedure can be explained in comparison to the history matching process: In history matching, the departure of computed observations from their measured values is expressed as a function (the **measurement objective function**). Minimizing this function leads to a parameter set that reproduces the historical measurements, hence a calibrated model is found.

When using prior knowledge, the departure of the applied parameter values from parameter values preferred by the modeler is expressed as a second function (the **regularization objective function**). This kind of regularization is therefore a method that introduces knowledge about the plausibility of



parameter values into the calibration process. This knowledge is often subjective, but nevertheless valuable.

PEST implements two principal methods to perform a concurrent optimization on measurement and regularization objective function: Prior Information and Tikhonov Regularization.

2.4.1 Prior Information

Prior information is the simplest way to implement preference for parameter values or to preferred relationships between them (e.g., a preferred ratio between horizontal and vertical hydraulic conductivity). The sum of squares of departures from these equations contribute to the regularization objective function.

Minimization of the total of regularization and measurement objective function leads to a parameter set that reproduces the historical measurements and shows a plausible parameter distribution at the same time.

Further reading: PEST Groundwater Data Utilities (5th Ed.), ch. 2.1.3: The Use of Prior Information in the Parameter Estimation Process.

2.4.2 Tikhonov Regularization

The Tikhonov regularization method as implemented in PEST automatically generates a number of “information” equations, which defines the initial value of each parameter as the preferred value (see Figure 2.7). The user can also make changes to these equations, or set up his/her own additional equations.

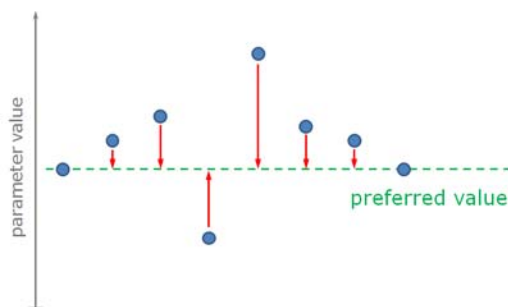


Figure 2.7 Regularization of parameters: departures (red) from preferred parameter values (green) are penalized.

When using Tikhonov regularization the calibration process is formulated as a constrained minimization process as follows “minimize the regularization objective function while ensuring that the measurement objective function is set at the user-specified target”. Figure 2.8 illustrates this approach. If this

user-specified target is not met, then PEST minimizes the measurement objective function. In the meantime it adjusts weights applied to prior information such that they act as Lagrange multipliers in the constrained optimization process. PEST thus determines the appropriate relative weighting between measurements and respect for prior information in accordance with a user's choice of target measurement objective function.

As a result, Tikhonov-Regularization reduces the number of possible parameter sets that constitute a calibrated model by rejecting calibrated models with unrealistic parameter values.

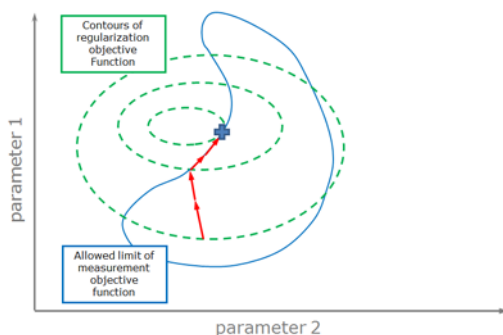


Figure 2.8 Constraint optimization: the regularization objective function (green contours) is minimized while staying within the defined limits (red contour) of the measurement objective function.

Further reading: Methodologies and Software for PEST-Based Model Predictive Uncertainty Analysis: Regularization (p. 46).

2.4.3 Regularization of Pilot-Point Parameters

If parameter fields are defined as a varying distribution using the pilot point method, this will allow a better fit to the observation data during history matching compared to a result obtained using constant parameters. While this is favorable to some extent, the resulting parameter field might look implausible, especially when pilot points are placed at a high density.

Figure 2.9 provides an example. Because there are more pilot points (104, purple crosses) than observations (12, flags), a perfect match between observed and simulated results is obtained.

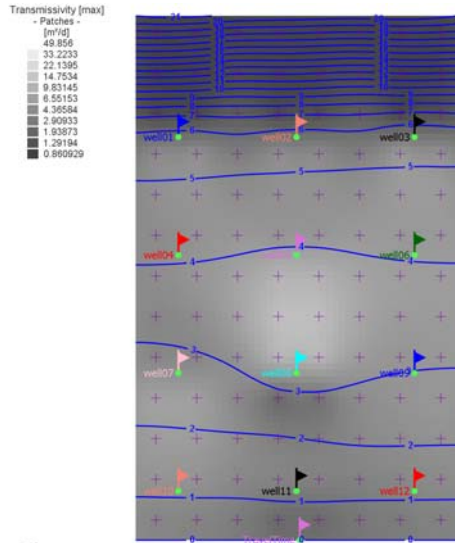


Figure 2.9 An over-fitted parameter field.

The transmissivity field however reveals that this result is flawed nonetheless: The distribution looks somewhat "bumpy", especially around the observations. Even more severe, the transmissivity above the northernmost row of observation points is totally different (lower) from the one in the remaining area.

A parameter distribution like this is unlikely, and accordingly a prediction made with this model has a high potential of wrongness even though it is perfectly aligned with its calibration data. This state is called **overfitting**.

To prevent overfitting, a second objective (next to the measurement objective function) is required, through that plausibility is preferred.

A common approach is to prefer homogeneous distributions of parameters over heterogeneous distributions. If different values are assigned to neighboring pilot points to lower the measurement objective function, this difference will be penalized and will give rise to the regularization objective function (See figure Figure 2.10). As a consequence, the optimization will yield a balanced compromise between calibration fit and homogeneity Finding the right distance within these penalties are applied is important. Differences between closely located points needs to be penalized stronger as the likeliness of parameter differences becomes smaller. Pilot points located far apart (above a certain distance, the **correlation length**) do not need to be penalized at all.

This distance and the strength of correlation are defined through a variogram (Figure 2.11)

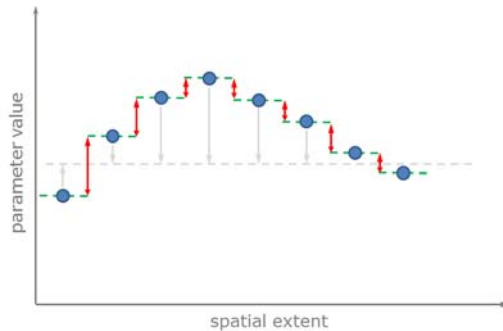


Figure 2.10 Regularization of pilot point parameters: by penalizing differences between pilot points (red), a homogeneous (smooth) distribution is preferred. Initial parameter values are still preferred through prior information (grey).

Thus, within the range of correlation, implausible heterogeneities are suppressed unless they are necessary to meet the targeted value of the objective function. PEST calculates the expected correlation between each two pilot points and creates a **covariance matrix** which is used to impose the correct weights. In summary, the correlation length allows to define a preferred variability of a model property, in addition to the preferred mean value that is provided through the initial parameter value.

Figure 2.12 shows the same model, regularized with a correlation length of 200 m. The transmissivity field is smoother, but still reflects general trends suggested by the observation data. Even though it yields a stronger model-to-measurement misfit, predictions made using this model will have higher confidence.

Further reading: PEST Groundwater Data Utilities, ch. 5.6: Regularization (of pilot points).

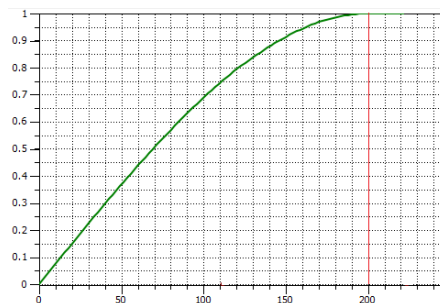


Figure 2.11 A spherical-type variogram with a correlation length (Range) of 200 m.

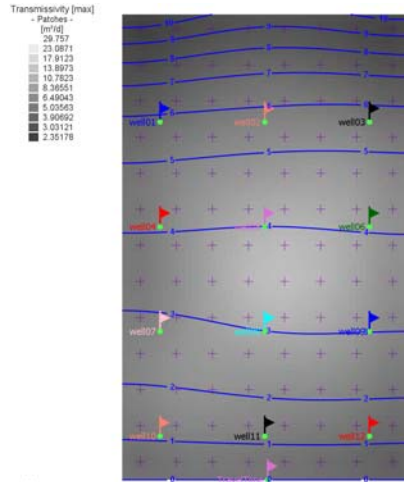


Figure 2.12 Regularization of pilot points parameters leads to a smoother parameter field (compare with Figure 2.9)

2.5 Subspace Regularization

Subspace regularization follows a different approach than Tikhonov regularization.

The fundamental idea of subspace regularization is to separate identifiable parameter components from non-identifiable parameter components in order to exclude the latter one from the parameter search.

The identifiability of a parameter is related to the way and extent it influences existing observation data (if a parameter does not influence any of the existing observations, it cannot be identified).

Parameters (called **base parameters** in the following) are usually neither completely identifiable or non-identifiable. It is however possible to create linear combinations of base parameters for which this is the case. These are called **super parameters**.

The transformed (super-)parameter space is separated into two subspaces: One subspace is comprised of combinations of parameters that have an influence on observations. These combinations of parameters can be uniquely estimated through the history matching process.

The remaining parameter combinations occupy the so-called **null subspace**. These combinations of parameters have no or very small influence on model outputs corresponding to observations; hence estimation of these parameters through history matching is not possible.



The groups are also denoted as sub-spaces of the **parameter space** (the parameter space is the combination of these two orthogonal subspaces, and contains all parameters):

- The group of identifiable parameters is called solution subspace (or often just **solution space**)
- The group of non-identifiable parameters is called null subspace (or often just **null space**)

2.5.1 Truncated Singular Value Decomposition

Singular Value Decomposition (SVD) is the name of the method through which the parameter space is partitioned into the two orthogonal solution and null subspaces.

In most groundwater modeling contexts the solution space is smaller than the null space. The earth is complex, and the information content of most calibration data sets is insufficient to provide unique estimation of the parameters which describe this complexity.

SVD analyzes the **Eigenvectors** of the covariance matrix to identify the super parameters. The **Eigenvalues** - a measure for the post-calibration variability of their associated Eigenvectors - are the criterion to decide if a parameter is associated with the solution space and therefore included in the optimization or not. The ratio of highest to lowest eigenvalue is a measure of the extent to which the inverse problem approaches ill-posedness. If this ratio is more than about $5e-7$ then the problem can be considered to be ill-posed (in which case PEST would fail to optimize).

The **truncated SVD** separates the parameter space into solution and null subspace using this ratio as a criterion, and therefore omits any super parameters that are too insensitive to be uniquely estimated. As a consequence, the inversion of the solution space is always well-posed and a stable optimization is guaranteed (unless flawed by other sources of error, e.g. bad derivative calculation, Section 2.1.1).

FePEST applies truncated SVD with a threshold of $5e-7$ by default in any PEST setup.

Further reading: PEST Manual (5th Ed.), Ch. 8.4: Truncated Singular Value Decomposition.

2.5.2 SVD-Assist

The SVD-A method uses the concept of the SVD to reformulate the inversion problem in a way that it can be solved with much less numerical effort.

The SVD first identifies those combinations of parameters which are uniquely estimable on the basis of the current calibration data set and defines them as



super parameters. SVD-A creates and run a new PEST setup based on super parameters instead of base parameters.

When calculating finite-difference derivatives of model outputs with respect to parameters, it actually calculates these derivatives with respect to the super parameters rather than the native model parameters (base parameters). Hence only as many model runs are required per iteration as there are dimensions in the solution space (i.e. number of super parameters).

This method is of fundamental importance for the pilot point method, as it allows calibration and predictive error analysis of highly parametrized models (e.g., using 1000+ parameters) with reasonable effort.

Further reading: PEST Manual (5th Ed.), Ch. 8.5 SVD-Assist

2.5.3 Least Squares (LSQR)

Least Squares (LSQR) is an alternative to the SVD method for highly-parametrized inversion problems. Experience has shown that its application is useful when more than 2500 parameters are involved.

Further reading: C. C. Paige and M. A. Saunders, LSQR: An algorithm for sparse linear equations and sparse least squares, TOMS 8(1), 43-71 (1982). and C. C. Paige and M. A. Saunders, Algorithm 583; LSQR: Sparse linear equations and least-squares problems, TOMS 8(2), 195-209 (1982).

2.6 Calibrated-Constrained Monte Carlo Analysis

Monte Carlo analysis can be undertaken by FePEST to understand the uncertainty of a specific prediction or a set of prediction. From a groundwater modeller perspective, for example a prediction can be set of hydraulic heads in the model, inflows to a open-cast mine, concentration peak at certain model areas, geothermal storage in the underground, etc.

The clear advantage of Monte Carlo analysis from a decision-maker point of view is the fact that we can learn the uncertainty of any prediction. Contrarily to other methods presented here (i.e. Predictive Analysis PEST mode), where uncertainty is only provided by one prediction at a time.

Predictive uncertainty analysis is accomplished by making predictions of interest with all parameter fields, and by then undertaking stochastic analysis of these predictions (PEST Manual). The outcomes of this analysis include probability distributions (histograms) of individual predictions and/or group of predictions. From here we can learn about mean and variance of our predictions.

A definition of calibration-constrained Monte Carlo analysis stands for identifying multiple (i.e. several) model scenarios, which all of them maintains valid the assumption of calibration, i.e. minimum reduction of the measurement



objective function. All these several model scenarios starts from a stochastic process (Monte Carlo) and after subsequent “adaptations”, they provide both an acceptable fit to the historical observations and reasonable parameter distributions (i.e. regularized field).

The generation of parameter fields which respect calibration constraints is undertaken by combining the SVD-Assist as a mathematical regularization in PEST and the pre-calibration null-space projection of differences between stochastic parameter fields and the “calibrated” model.

2.6.1 Stochastic Parameter Generation

The start point of the analysis requires a model, which is considered to be “calibrated”. Such a stage can be achieved by running FePEST in the standard operation mode (Estimation) with any combination of regularization approaches and prior information.

Based on a Monte Carlo method, random parameters are generated with a mean value equal to the calibrated parameters. The prior information covariance matrix (Regularization -> Tikhonov section in the Problem Settings) and using a mean equal to the calibrated parameter field.

2.6.2 Null-Space Projection

The null-space projection is overtaken by the PEST Utility PNULPAR. The starting point of the null-space projection is a set of randomly-generated parameters (e.g. using PEST Utility RANPAR, see section 6.3) and knowledge of the dimensionality of the problem.

Typically, the value of the calibrated parameters is considered as the mean value for the stochastic generation of the parameters. The number of dimensions of the calibrated solution space is approximately equal to the number of super parameters used by the SVD-Assisted calibration.

During the operation of PNULPAR three task are carried out:

- Calculate the different between each parameter set and the calibrated parameters.
- This difference is subsequently projected onto the calibration null space
- The projected differenced is re-added to the calibrated parameter set.

The random parameter sets modified as described above do not strictly reflect a perfect calibrated model, because non-linearities in the problem. Therefore, adjustment of these parameters is required to validate the calibration constraints. Such modifications take place through the alternations made to the calibration solution space (with an untouched null space).

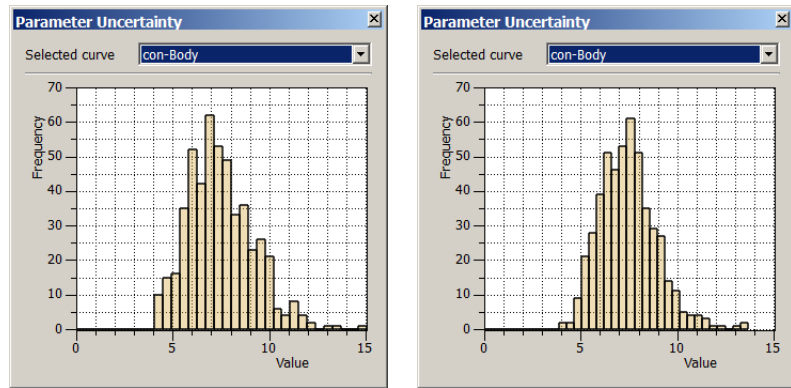


Figure 2.13 Parameter set (Conductivity) without null-space projection (left) and with null-space projection (right).

Figure 2.13 shows an example of randomly-generated parameters without (RANDPAR utility) and with (PNULPAR utility) projection onto the null space.

2.6.3 Readjustment of Parameters

In the previous steps, a set of “almost calibrated” parameters were generated. In order to fully complete the calibration task for each parameter sample, a series of calibration runs with PEST is undertaken. The SVD-Assist regularization in PEST is used here to adjust the parameters rapidly. This methodology will only run the model as many times as the number of super parameters (or dimensionality of the problem). Typically, after two iteration cycles of the PEST optimization, the calibration stage is achieved.

2.6.4 Outcome of Calibration-Constrained Monte Carlo

The final outcome of the calibration-constrained Monte Carlo provides a set model parameters, which all of them calibrate the model similarly. In addition we have tracked the evolution of a prediction (or several ones) and the evolution of the objective function for all these models. These last two can be used to create histograms to understand explore the uncertainty of the modelling task.





3 Fundamental Setup

The fundamental setup prepares the model to be processed by PEST. Its steps are common to most PEST methods.

The fundamental setup comprises the definition of parameters, observations and prior knowledge (if available). A decision on using subspace methods is to be made, and finally, parallel computing may be configured to distribute workload on different computers.

3.1 FEFLOW and FePEST

FePEST requires a FEFLOW model as a starting point for any PEST setup. FePEST asks for the file name of the FEFLOW model when a new optimization project is created.

When setting up a FEFLOW model that is planned to be subject to a PEST optimization, bear in mind that the run-times should be reasonably short and that the model must run stable. It is possible to exchange the FEFLOW model by another one, provided that the new model file contains the same observation points and parameter zone selections (usually a modified version of the original model).

It is also possible to open a model simultaneously in FEFLOW and FePEST. If changes are made and the model is saved within FEFLOW, a reload must be performed within FePEST to inform it about the changes made.



Always perform a reload in FePEST after saving the FEM file in FEFLOW.

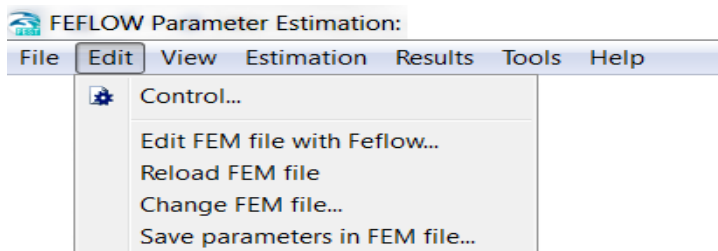


Figure 3.1 The Edit menu allows access to the problem settings dialog (Control option) and allows to open, reload and change the FEM file.

The Edit menu allows to save a FEM file in order to visualize all the parameter values as a results of the interpolation settings (i.e. Kriging settings for each parameter definition). Such as option is very useful to evaluate the interpolation quality before running the optimization problem.



3.2 Parameter Definitions

Choosing the model parameters that PEST can change to minimize the objective function is usually the first part of the setup.

An adjustable parameter can be described by its parameter type (e.g., hydraulic conductivity) and the assignment method (e.g., a certain geological formation).

- The **parameter type** can be any time-constant material property of the FEFLOW model.
- The **assignment method** can be based on zonally constant, pilot points and tied definitions.

The spatial definition of a adjustable parameter (zone or pilot point) can apply any elemental selection that is stored in the FEFLOW model or the entire model domain. FePEST also support the placement of pilot points in 3D.

Note that the choice for the zones already constitutes a kind of regularization (called structural regularization), that has the power to significantly influence the calibration process (in a good or bad way).

If making changes to elemental selections in FEFLOW, remember that you must reload the model after you have saved them in the FEFLOW file.

3.2.1 Assignment method

An optimization problem commonly comprises many parameters sharing identical or similar settings.

To avoid that the user needs to setup these parameters one-by-one, similar parameters are defined through a **parameter definition**. A parameter definition allows central adjustment of default values of its dependent parameters which are created from it. This allows convenient management of large lists of parameters.

Usage of a parameter definition differs depending on the choice of the assignment method. Possible assignment methods are:

Zonally constant:

The user specifies one or multiple zones in which the value of a parameter is to be estimated. A zone is any elemental selection stored in the FEM file. A dependent parameter will be created for each of these zones (see Figure 3.2). During each model call of a PEST run, a constant value will be assigned to the chosen FEFLOW model property in each separate zone.

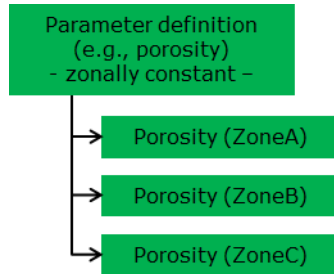


Figure 3.2 Zonal parameter distribution

Interpolate from pilot points:

The user chooses exactly one zone, which can be the entire model domain, a layer or an elemental selection (a separate definition is required for each zone if pilot points are used).

Within the limits of this zone, a cloud of pilot points is either imported from a file or is automatically generated by FePEST. Each of these points represents a single parameter of the PEST setup (see Figure 3.3)

The parameter definition of a pilot-point based parameter includes the settings of the interpolation method that is used to interpolate parameter values between pilot points. Available options are Kriging (2D or 3D depending on the placement of the pilot point) and Radial Basis functions (only 2D).

Note that FePEST will apply the same settings to regularize the pilot-point parameters if the respective option for the Tikhonov Regularization is active.

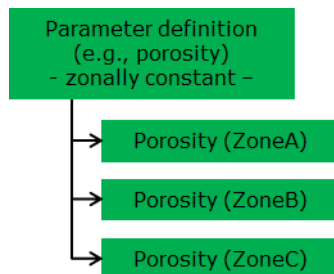


Figure 3.3 Pilot point based parameter definition

Tied to other parameter definition

A parameter definition can be tied to an existing parameter definition. In this case the parameter values within the definition will be changed according to the parent parameter definition, always maintaining the ratio of their initial values. Tied parameter definitions can be associated to zonally-constant or pilot point parent parameter definitions.



A typical application of tied parameters is to maintain the anisotropy ratio between three major conductivity directions. For example, the modeller could consider to calibrate only K_{xx} and then simply update the values of K_{yy} and K_{zz} based on the first one. Thus, K_{yy} and K_{zz} are tied parameter of K_{xx} .

IFM implemented parameter definition

FePEST allows the possibility to adjust parameter(s) provided directly by a FEFLOW plug-in. For example, these IFM-implemented parameters can be FEFLOW parameters not supported directly by the FePEST interface or any kind of user-defined information.

Also see Section 9.4 for information on how to implement parameters through IFM plug-ins.

3.2.2 Parameter Settings

Each parameter definition requires additional configuration of PEST parameters for the optimization problem (Default tab). This information would be used to modify the parameter during the task of the model calibration:

Parameter transformation

- **Logarithmic** transformation often allows faster and quicker reduction of the objective function by linearizing of the system).
- A parameter can be **fixed**, thereby removing it from the optimization.
- The **tied** parameter will change according to the initial ratio in respect to the parent parameter.

Change limit

This setting defines how to determine the maximum offset with that a parameter can be changed within a single iteration. Note that relative change limits are not allowed if the parameter is log-transformed.

Initial value

The parameter value that is applied in the first iteration. The value assigned in the FEFLOW model will be used by default, assuming that this represents the preferred value of the expert modeler.

Bounds

An upper and a lower limit can be defined for the parameters. A good approach is to use a very low and very high value initially that does not impose a restriction to the optimization.



Scale and offset

Before a parameter value is assigned to the model, it is multiplied with Scale and then added to Offset. These settings usually do not need to be changed.

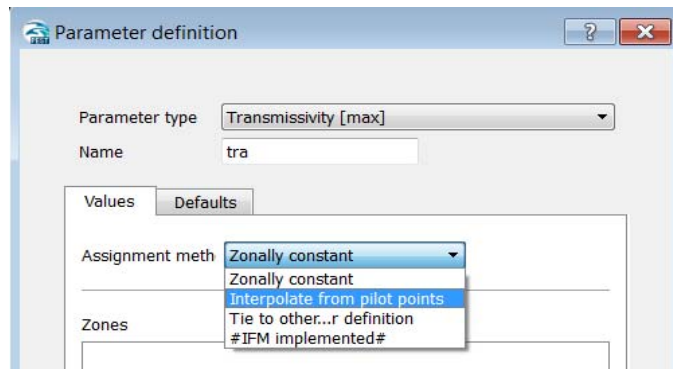


Figure 3.4 Creating a new parameter definition. Choose from Zonally constant, pilot point interpolated parameter distributions, tied parameter definitions or IFM implemented. The Defaults tab defines the standard settings for the dependent parameters.

After all the changes described in the previous paragraphs are done, pressing the OK button applies the new parameter definition. At the same time its dependent parameters are added to the parameter list.

These parameters inherit the default parameters of their parent definition (Figure 3.5). If required, changes to settings of individual parameters can be done (overriding the defaults), including the removal of parameters.

Working with large lists is easier with a spreadsheet program. The Copy/Paste buttons allow a quick transfer to and from other programs using the system clipboard.

Parameters					
Definitions:					
Transmissivity [max] (tra)					
Def.	Parameter type	Zone	Pilot point	Active	N:
tra	Transmissivity [max]		1	yes	tra-1
tra	Transmissivity [max]		2	yes	tra-2
tra	Transmissivity [max]		3	yes	tra-3
tra	Transmissivity [max]		4	yes	tra-4
tra	Transmissivity [max]		5	yes	tra-5
tra	Transmissivity [max]		6	yes	tra-6
tra	Transmissivity [max]		7	yes	tra-7
tra	Transmissivity [max]		8	yes	tra-8
tra	Transmissivity [max]		9	yes	tra-9
tra	Transmissivity [max]		10	yes	tra-1
tra	Transmissivity [max]		11	yes	tra-1

Figure 3.5 After a parameter definition has been created (top), its dependent parameters appear in the list below.

3.2.3 Parameter Groups

The parameter groups allows the configuration of the derivative calculation for the parameters (see section Section 2.1.1. FePEST applies default values that have been tested to work with most FEFLOW models, and often adjustments are not necessary when setting up a PEST model.

If the PEST optimization however fails due to bad derivatives calculations (which cannot be excluded as every model is different), a reconfiguration of these settings might be required. Detailed explanation and literature references to the different settings are given in the FePEST help system.

NOTE: The PEST utility tool JACTEST (Section 8.2) is a utility to check for bad derivative calculation.

By default, FePEST defines one parameter group for each separate parameter definition and assigns all adjustable parameters to this group accordingly:

- The Derivative method is chosen automatically (option “switch”, starting with the more effective 2-point methods, switching to higher order methods if required)
- The Increment size (set to 1.5% by default) can be increased if minor model instability issues are observed. Do not increase this value unnecessary, as high values would violate the linearity assumption of the derivative calculation.

3.3 Observation Definitions

The observations provide the primary (and only) measure that informs the optimization algorithm about the model-to-measurement misfit.



For each observation FePEST requires:

- its type (e.g., hydraulic head, fluid rate budget),
- its location (e.g. an observation coordinates, stored nodal selection),
- the time at that it was recorded (in case of a transient model) and
- the value that was observed in reality.

Because the observation points set in FEFLOW already contain this information, it is possible to import them directly from the FEFLOW model.

By default, all observations are weighted with a weight of unity. These can be adapted by the user (compare Section 2.1.2).

Supported observation types are all observations of system state (e.g., hydraulic head, saturation, mass concentration, temperature and age species) and rate and/or period budget values (fluid, mass and heat) depending on the models problem class. Reference values are constant values if the model is steady state. In a transient model, a time series contains the data of observations at different points in time.

Advanced users may also use IFM plug-ins or third party software for other types of observations.

3.3.1 Definition of Observations

Observations are defined in a similar way as parameters. If observations are of the same type (e.g., observations from the same well field), often similar or identical settings are required.

For convenience, and to avoid many repetitive adaptations of these setting when creating or making changes to observations, each observation depends on an **observation definition**. A definition allows central adjustment of the default values of its dependent observations (see Figure 3.6). This allows a better management of a large number of observations.

Def.	Observation type	Obs. point	X	Y	Z
mas	Mass concentration	TravelTime	250.00	0.00	0.00
hea	Hydraulic head	well01	65.00	615.00	0.00
hea	Hydraulic head	well02	245.00	615.00	0.00
hea	Hydraulic head	well03	425.00	615.00	0.00
hea	Hydraulic head	well04	65.00	435.00	0.00

Figure 3.6 Observation definitions (top) and their dependent parameters (list below).

Observation type

The observation type can be any kind of process variable available in the FEFLOW problem (e.g. hydraulic-head, mass concentration, temperature, etc.). In variable-saturation problems using the Richards' equation in FEFLOW, additionally an observation can be defined based on either saturation or moisture content.

In the case the **Budget-History Charting** option in the FEM file has been activated, an observation definition can be created using the Rate Budget or Period Budget available depending on the problem class (flow, mass, heat and age).

An observation or a group of observations can be also defined based on a **User-Data nodal distribution** available in the FEFLOW problem. For the implementation of this kind of observations, FePEST requires the location of the observation points (coordinates) imported from an external file in the source section.

Advanced users may choose to create additional observations in the FePEST setup, that can be implemented using FEFLOW IFM plug-ins or third-party software (see also section Section 9.4).

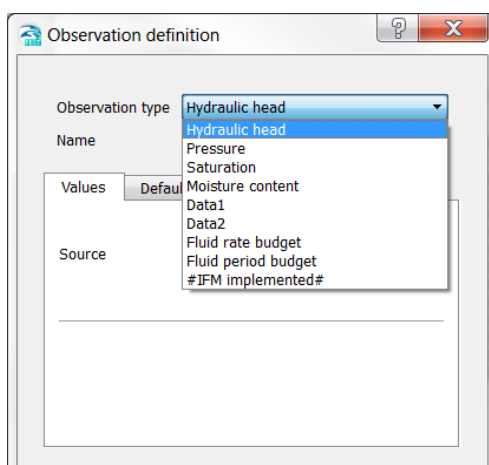


Figure 3.7 Creating a new observation definition

Source

FePEST can automatically import the observation points stored in the FEFLOW model. If the model is transient, it adds multiple observations per observation point, each representing a sample point of the respective time series.



Alternatively, observation data can be imported directly from an external import file. This option is mandatory from observations definitions based on user-data distributions.

3.3.2 Observation settings

In similar manner than the parameter definitions, all the observations in FePEST require additional information for the PEST optimization.

Synchronization of time observations

This option is available only if the FEFLOW file linked to the FePEST problem is transient in the FEFLOW Problem Settings. The time value related to each observation point does not necessary coincide with the automatic time-steps suggested by the Predictor-Corrector scheme in FEFLOW. The user can opt to choose if the observation times should be used for the calculations in the FEFLOW side. If the synchronization is set as false, FePEST will simply make linear interpolation in time between existing time steps to retrieve the observation value.

Weights

Each observation used in the PEST problem requires a weight. A weight of one is applied by default every time a new observation definition is created. However, this value can be overwritten by using any of the three options or combinations between them:

- **Inverse of measured value:** is used as the weight of an specific observation. This option is practical in cases the observations changed several orders of magnitude within the same group (e.g. mass concentration of peak values in respect to the tail breakthrough.)
- **Inverse of measurement noise:** A weight is calculated as the inverse of the confidence interval value defined in the observation point list in the FEM file.
- **Check box:** FePEST applies automatically observation weights under several user-defined criterion such as inverse of measured values, inverse of measurement noise, equalizing groups, declustering spatially and declustering time-series.

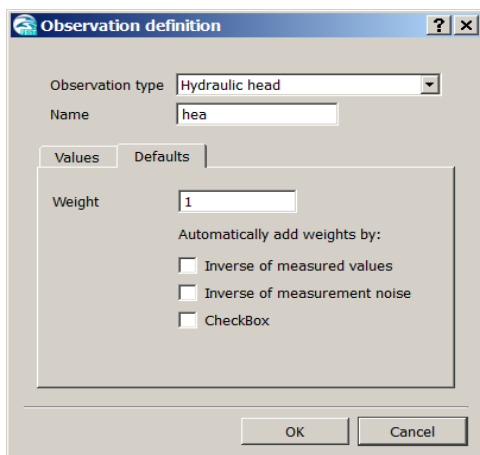


Figure 3.8 Defining observation weights

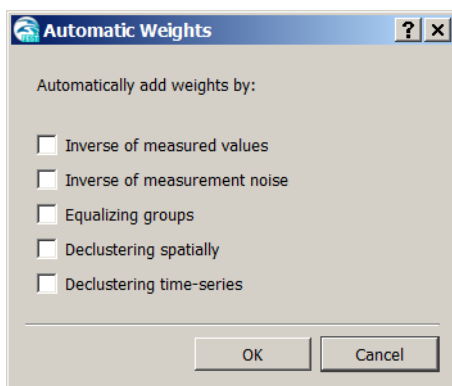


Figure 3.9 Adjustment of automatic weights

It is possible to change settings of particular observations (overriding the defaults). This is required especially when adapting the observation weights to achieve better optimization behaviour and results.

HINT: Working with large lists is easier using a spreadsheet program. The Copy/Paste buttons allow a quick transfer to and from other programs using the system clipboard.



3.3.3 Observation Groups

FePEST automatically assigns observations of different type to different observation groups. This allows PEST to distinguish how strong the different groups contribute to the measurement objective function.

In the same way it is possible to allocate observations of the same type to different groups (e.g., to differentiate between hydraulic head measurements in different aquifers).

Advanced users may specify an observation covariance file to define the weights for these observations.

3.4 Prior Information

Prior information (or prior knowledge) can be described during this step of the PEST setup. In most cases, the prior information provided in the Tikhonov regularization page, will be preferred over the manual setup of equations of this section prior information (see Section 3.5 for details).

The addition of manual prior information follows the standard criteria of the prior information equations defined in PEST. These equations of prior information allow to define preferred values for parameters or preferred relations between parameters. Any deviation from this relationship will contribute to the regularization objective functions. As though there are certain similarities to observations:

Name

A user-defined name to identify the equation.

Weight

Each equation has a weight to allow to control the strength with that it contributes to the objective function (relative to other prior informations and field observations). Individual weights of prior information should reflect the trust that is associated with the underlying assumptions.

Group

It is possible to associate different prior informations to different observation groups, hereby defining more multiple regularization groups. The principle is the same as for the observation groups. New groups are introduced in the section Parameter Groups of the Problem Settings dialog (see Section 3.3).

Equation

A default formula is provided for convenience, which needs to be adapted for the specific purpose by the user. The formula itself has to be written in a special syntax specified in the PEST manual. If a parameter log-transformation is specified in Section 3.2, this has to be considered in the syntax of the equations. Note that FePEST will not check for correct syntax on its own, however

the PESTCHEK feature Tools menu can perform these verification (recommended after each change made to prior information).

3.5 Regularization

The regularization page in the Problem Settings dialog provides the complete set of combination of regularization by prior information (Tikhonov) and mathematical regularization (Singular Value Decomposition, SVD-Assist, or Least Square).

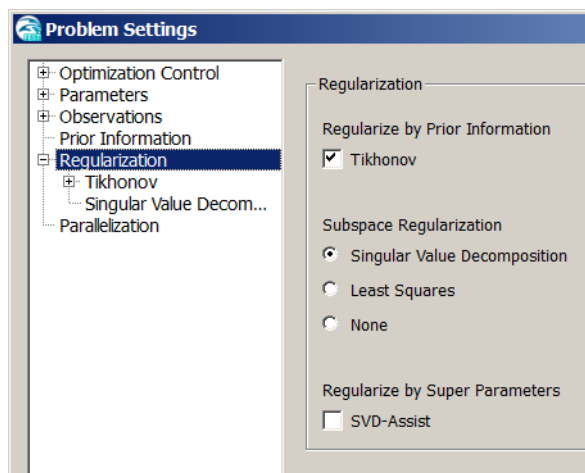


Figure 3.10 Regularization possibilities in FePEST

3.5.1 Tikhonov Regularization

When activated, Tikhonov regularization can implement three different aspects of regularization in the PEST setup:

Automatic regularization mechanisms

FePEST provides three automatic options as regularization definitions for the Tikhonov regularization. Advanced users can provide other regularization definitions modifying either manually or automatically (utility GENREG) the PEST Control File.

Regularization based on initial parameter values

Assuming that the initial parameter value represent the expected values of the expert modeller, equations of prior information are created that will penalize departures of parameter values from initial values.

As a result, calibration adjusted parameters will be close to the values preferred from a modeller's point of view.



Regularization based on preferred differences

This level of regularization suggests a preferred homogeneity in the parameter distributions between adjacent pilot points. PEST creates prior information equations, which basically contains each two adjacent pilot points, and indicate a preferred difference of zero.

Typically, such a regularization approach is intended for zones of the model that belong to the same geological unit.

Regularize pilot point using covariance matrix

Pilot-point type parameter values for each definition are regularized taking into account the expected correlation (i.e., using a covariance matrix). Differences in parameter values of pilot points closer than their correlation length will be penalized. As a result, smoother parameter definitions will be preferred over heterogeneous ones.

The range parameter in the Kriging Settings indicated for each pilot-point parameter definition is used to compute the covariance matrix. The larger the range value is, the larger the number of pilot points influencing the regularization weights.

Objective Function Limits

As described in Section 3.4, the Tikhonov regularization aims at finding a minimum of the regularization objective function while observing user-defined limits for the measurement objective function (for which the model is still considered calibrated).

The following settings determine this limit:

Target measurement objective function (PHIMLIM)

This is the limit for the measurement objective function below which the model is considered calibrated.

The value can be calculated by summing up the (weighted) measurement noise associated with the observations. If this is not possible, a value somewhat higher than the objective function that results from a calibration run without Tikhonov regularization can be chosen.

Acceptable measurement objective function (PHIMACCEPT)

This additional threshold is usually set slightly (5% to 10%) above PHIMLIM. PHIMLIM and PHIMACCEPT define a buffer zone for the measurement objective function value for stability reasons. In this zone, an objective function value is tolerated even though it does not meet the target value. This is necessary as the parameter upgrade vector will often "miss" the exact limit because it relies on a linearity assumption.

Figure 3.11 illustrates how the iteration adapts to these limits.

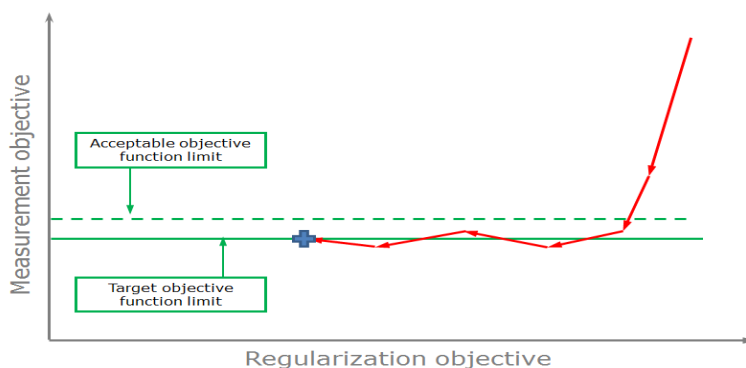


Figure 3.11 Development of the regularization and measurement objective function during Tikhonov-regularized parameter estimation.

Successive reduction of the objective function Limit (FRACPHIM)

This option allows a different strategy to determine the target objective function limits. If set to a non-zero value (allowed values are between 0 and 1), PHIMLIM is determined by multiplying the last achieved value of the measurement objective function with FRACPHIM.

In this way, PHIMLIM decreases from iteration to iteration. It will however never be smaller than the value defined in PHIMLIM.

Optimal values for FRACPHIM are normally in the range 0.1 to 0.3.

FePEST activates this option by default in combination with a very low objective-function value (these are the same defaults as applied by the PEST tool ADDREG1). This improves the well-posedness of the optimization and thereby leads to a more stable behavior of the GLMA optimization. The limits should however be adapted by the user as these settings might not yield the desired plausible parameter values.

See Section 7.3.3 of the PEST user manual for a full discussion of these variables.

Weight Factors and Adjustment

Settings to determine the weight factor can be used for fine-tuning or troubleshooting. Their default values follow general recommendations that are suitable for most applications of PEST.

3.5.2 Subspace Regularization

Subspace regularization methods can improve the stability of ill-posed inverse problems and significantly reduce the modelling efforts required for an optimization.



Singular Value Decomposition (SVD)

It is recommended to use SVD for any PEST setup (unless LSQR has been chosen, which tends to be faster for PEST runs involving more than 2500 parameters). FePEST activates this option by default (see Figure 3.10).

The following parameter settings determine the number of singular values/super parameters considered during the optimization:

- Eigenvalue threshold - all super parameters lower than this value will be truncated. The default threshold value is set to 1e-6. At this value, the optimization is still regarded well-posed, which ensures stability of the optimization.
- Maximum number of singular values/super parameters - this limits the number of singular values/super parameters to a maximum number.

See the PEST users manual (5th Edition), section 8.4.2: Implementation of SVD with PEST for a full discussion of these settings.

3.5.3 Regularize by Super Parameters: SVD-Assist

SVD-Assist is a useful methodology to speed up the calibration of highly parametrized models (often involving the pilot point method).

Either of the following two ways can be used to determine the number of singular values:

- FePEST uses the SUPCALC tool to determine the optimal number of super parameters. This is the default option.
- The number of super parameters can be manually specified. This is particularly useful if multiple CPUs are used in parallel:

Since each iteration requires one model run per super parameter to calculate the Jacobian matrix, optimal speed up is attained if the number of super parameters is set to a multiple of total number of slaves used for parallel computations (See section Section 3.6).

3.6 Parallelization

An inherent issue of calibration and uncertainty analysis, scenario runs and sensitivity analysis is the large number of model calls and the associated computational complexity in terms of each model run-time.

Fortunately, many steps of a PEST run, especially the numerically expensive calculation of the Jacobian matrix, is very suitable for parallel computing. The use of multiple computers (be it a limited number of office PCs, a HPC cluster or cloud-based computers) can reduce the computation time significantly. Parallelization can also improve model run-times significantly on a stand-alone computer.



In the case of highly-parametrized inversion processes, parallel computing may be necessary to finish a computation within the project time frame.

FePEST uses the BeoPEST utility- a network capable version of PEST - for obtaining better run-time efficiency. FePEST also transfers the required model files to the slave computers.

The port denotes the network (IP) port of the local computer (Host) that FePEST uses for communication. If the default settings conflicts with a different application, choose a different port number. Make sure that no router, firewall or anti-virus software blocks the network connection even if no remote computers are involved (localhost only).

The slaves show a list of servers that are used to solve model run jobs during the PEST run (initially empty). Servers can be added/removed to/from the list, and/or the settings of existing entries can be edited.

See also installation instructions for parallel computing.

Host name

The host name specifies the host name or IP-address of the server computer. Enter "localhost" to add the local computer to the list.

Note that FePEST has to be started in server mode on all remote servers (except the local computer) before commencing the PEST run.

No. of Slaves

The number of slaves is the number of models that will be run simultaneously on the computer. In most cases, this will be set to the number of available CPU cores on that computer.

Host name	No. of slaves	Active
localhost	8	yes
Server1	4	yes
192.168.1.42	8	yes

Figure 3.12 The parallelization page contains the slave machines running FePEST in server mode.

FePEST must be started and set to server mode on each of the slave servers.

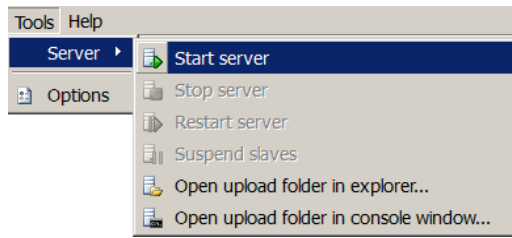


Figure 3.13 When running in server mode, the current computer acts as a slave server that can receive run jobs from another computer.





4 Parameter Estimation: History Matching/Calibration

The terms parameter estimation, history matching and calibration are used synonymously in this document.

History matching of a model requires completion of the fundamental setup as explained in Section 3.

After the fundamental setup has been completed, PEST can be used for history matching. The history matching process targets the estimation of a parameter set that optimally satisfies both, the historical observations and the prior knowledge (if provided). The resulting parameter set is then referred to as a calibrated model.

This section explains the required settings, how the run is commenced and how to interpret the visual feedback during and after the run. Exporting a new FEFLOW model with optimized parameters is the final step.

4.1 Required Settings

4.1.1 Optimization Control

By default, FePEST activated the option for doing history matching and applies the respective PEST operation mode accordingly.

Note that the FePEST operation mode is slightly different than the PEST operation mode: If "Estimation" is set, FePEST sets the "regularization" mode (if the calibration involves Tikhonov regularization) and "estimation" mode (if the calibration does not involve Tikhonov regularization).

A second important option is the NOPTMAX setting in the termination criteria setting. For calibration purposes, this option must be set to "Number of iterations". It assumes a sufficiently high value with the default set to 30.

4.1.2 Other Settings

The default settings are well suited for a range of optimization processes. However, these can be altered for fine-tuning and trouble-shooting. The individual settings will not be explained here, but more information is provided in the FePEST help system and the PEST documentation.

See Section 4.2.2 Control Data of the PEST user manual (5th Edition) for a full discussion of these variables.



4.2 Starting PEST

The history matching process is initiated by sequence of steps, as described below.

4.2.1 Preflight Checks

Before running the model, a check of the PEST setup is recommended. PEST provides several tools (PESTCHEK, TEMPCHEK and INSCHEK) that can be run through the Estimation menu of the FePEST user interface.

These utilities check the most important files of the PEST setup, namely the control file, the template file and the instruction file, for errors and warnings. Errors prevent PEST from running whereas the warnings indicate possible problems and provide suggestions for improving the setup.

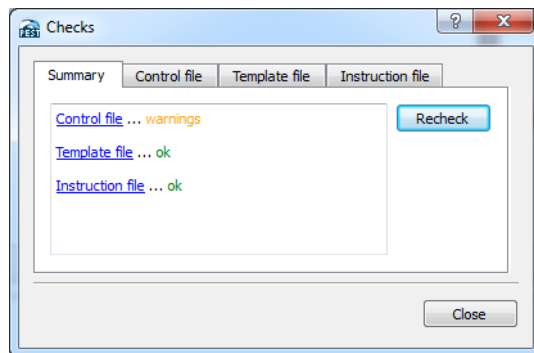


Figure 4.1 The Checks utility in the Estimation menu checks the PEST setup for validity and provides suggestions for improvement.

4.2.2 Running PEST

PEST is started using the Run button in the toolbar. Usually no changes are required for the sequence of steps in the prompted dialog (see Figure 4.2).

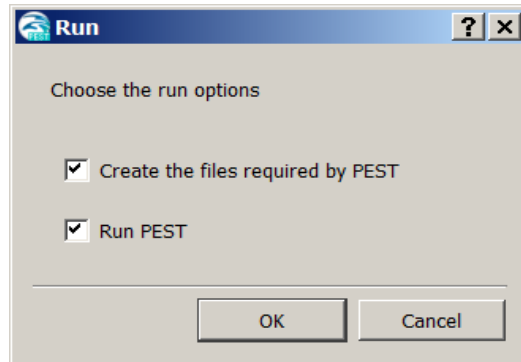


Figure 4.2 The Run dialog. Usually all options will be active.

The sequence entails the following steps:

- **Create the files required by PEST**
FePEST generates the required PEST files according to the settings.
- **Recalculate Jacobian Matrix**
This option is available if the Jacobian matrix is already computed in a previous PEST run (file with extension JCO). Advanced users can choose to deactivate this option to save computational effort.
- **Start PEST**
This calls PEST and starts the history matching process. The FePEST windows display the progress of the PEST iteration.

Experienced users of PEST may choose to generate the PEST files without having to start the PEST run. See also Section 9.

It is also possible to interrupt (pause) or stop the PEST run. A stopped PEST run can be continued at a later point in time, even if FePEST is closed during its execution. This is particularly useful if the optimization is interrupted, for example, due to a computer shut down/ restart etc.

4.2.3 Output during PEST Run

During the PEST run, FePEST reads and analyzes the output of PEST and displays key information in the form of several panels and charts:

Output

The primary output of PEST is shown in this window. Users familiar with the output of PEST will find information (that is omitted in PEST) in the FePEST output here.

Status

The key information on the status of the PEST run is shown in the Status panel.

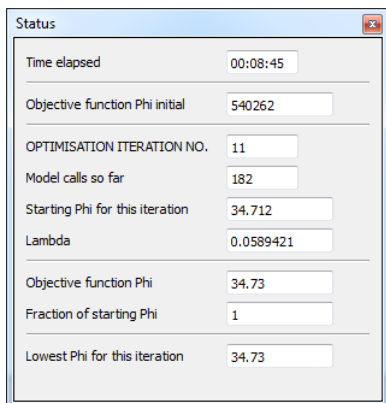


Figure 4.3 The Status panel contains key information about the progress of the optimization.

Objective Function

The objective function and its profiling illustrates the progress of the optimization run and makes it easier to identify problems (if any) during the run. In principle, if everything is done correctly, the function values should be monotonically decreasing.

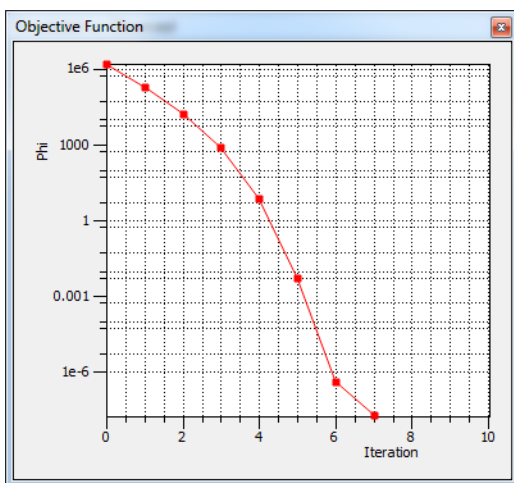


Figure 4.4 The development of the objective function is shown in the correspondent panel.

Simulated vs. Observed

The panel shows a scatter plot comparing the simulated values against (observed) reference values.



Simulated vs. Time

The time-dependent observations in the model are plotted against the measured time series. This panel is not available for steady-state models.

Due to the restricted size of the text panels, these may not contain the complete file. In such a case, click the “View” button to open the corresponding file in an external text editor.

4.2.4 Output after PEST Run

PEST provides additional statistics after the history matching process. FePEST displays these and other key information in additional panels.

- **Run Details**
PEST saves important information about the setup and results of its run into a run record file. This can be viewed in the Run details panel.
- **Parameter Sensitivities**
For all iterations, PEST saves the composite parameter sensitivities in the sensitivities file. These can also be viewed in the Run Details panel. This information can be particularly useful, for example, to identify the hypersensitive parameters. The composite sensitivity of a parameter is a measure of the sensitivity of all model outputs for the associated observations for this parameter. By inference, it is a measure of the information content of the calibration data set with respect to this parameter.
- **Observation Sensitivities**
PEST writes the observation sensitivities of the last iteration to an observation sensitivities file. This is shown in observation sensitivities.
- **Residuals**
This panel lists the measured and simulated value for each model observation, along with their differences (the residual).
The native (non-weighted) residuals enable the identification of well-matched and poorly-matched calibrated observations. The weighted residuals (shown separately) offer a valuable check for an appropriate choice of observation weights.
The content of this tab is identical to the residuals file created by PEST.
- **Covariance and Correlation matrix**
Covariance and Correlation matrices are only available when not using regularization.
These tables show the covariances and correlations between the parameters, respectively. The latter is normalized to values ranging from 0 (no correlation) to 1 (strong correlation). A color gradient facilitates interpreting the values.



- ***Eigenvectors and Eigenvalues***

Eigenvectors and Eigenvalues are only available when running PEST in estimation mode and not using singular value decomposition.

Each observation contains some information for identifying the calibrated parameters. Since the observations and parameters are often correlated, the information provided by a particular observation might overlap with the information from a different observation. Therefore, the number of observations is not necessarily proportional to the combined information provided by these observations.

By doing Principle Component Analysis (PCA) on a covariance matrix, orthogonal combinations of parameters are identified. With PCA, it is also possible to know the extent to which these combinations are informed by the calibration process. These combinations are the eigenvectors of the covariance matrix. A low eigenvalue associated to an eigenvector indicates low post-calibration variability. Conversely, a high eigenvalue implies high post-calibration variability.

The ratio of highest to lowest eigenvalue is a measure of stability for the inverse problem and indicates whether the problem is well-posed. If the value exceeds $5e-7$, the problem is considered to be ill-posed. It is not possible to solve an ill-posed problem using PEST without using regularization methods (namely, Tikhonov or SVD). Solving a problem as such without regularization may lead to numerical instabilities and erratic behaviour of the objective function.

The contents of the text panels can be easily transferred to a spreadsheet program for further processing and visualization.

See section 5.2: The PEST Run Record and section 5.3: Other PEST Output Files of the PEST user manual (5th Edition) for a full discussion of these output files.

4.3 Export to FEFLOW

After PEST has finished its optimization (either by reducing the objective function to an acceptable level or by meeting another termination criterion), FePEST displays a list of resulting parameter values.

For further inspection, one can save the resulting model in a new FEM file and open it using FEFLOW.

The optimized parameter may be used as the initial parameter values for further use of PEST utilities, for example for doing predictive analysis.

To return to this dialog at a later point in time, use the Show Results option in the Estimation menu.



5 Predictive Analysis: Best and worst case scenarios

Predictive Analysis requires the completion of the fundamental setup. It is further recommended to successfully run a history matching (calibration) process before starting predictive analysis. The parameter set found resulting from the history matching process is used as the initial parameter values in the current FePEST setup (in the Show results panel).

It is common to have multiple calibrated models, especially for environmental models. However, the predictions made by using these models may vary significantly despite the fact that they all honour historical data. The prediction attained by a particular model is therefore just one out of the possibly many outcomes.

Predictive analysis is a simple tool in PEST for non-linear model predictive error and uncertainty analysis. It searches for the calibrated model with the maximum or minimum key prediction. This facilitates identification of the worst-case and/or best-case scenarios among the set of calibrated models.

However, the use of PEST in predictive analysis model is restricted to well-posed problems only. This implies that it is applicable only to cases where the variability result from measurement noise. In case of an ill-posed problem, advanced users may choose to use PEST methods such running PEST in Pareto mode.

Further reading: PEST Manual (5th Ed.) Ch. 6: Predictive Analysis

5.1 Required Settings

5.1.1 Optimization Control

The required operation mode in PEST is the Predictive analysis mode; the optimization control section of the Problem Settings dialog contains the respective setting.

5.1.2 Observation and Type

Predictive analysis aims to either maximize or minimize a key prediction while maintaining the objective function below a user-specified value.

The prediction could be any observation (e.g. a specific hydraulic head) that is defined during the fundamental problem setup. Usually, this particular observation assumes zero weight so that it is excluded from the measurement objective.



The Predictive Analysis section of the problem settings dialog prompts the settings for the choice of the prediction whose value must be maximized/minimized, and for the maximum value permitted for the objective function.

5.1.3 Objective Function Limits

The maximum/minimum prediction is sought subject to the constraint of maintaining the calibrated state of the model. It is important to define a maximum threshold for the objective function value. The model is considered to be calibrated if the value of the objective function is less than the maximum threshold. This is done by providing objective function limits. The model is considered to be calibrated only if the value of the objective function is less than the defined target objective function.

A common choice is to choose a value which is 5-10% greater than the value of the objective function obtained in the preceding calibration process. However, it should be noted that the choice can significantly impact the result. If the statistics of measurement noise are known, the limiting target objective function could be set in accordance with theory. See the PEST manual for more details.

See the PEST Manual (5th Ed.), Ch. 6.2.2: PEST Variables used for Predictive Analysis for a full discussion of these variable.

5.1.4 Other Settings

The default settings are chosen such that they are suitable for doing predictive analysis for many cases. The individual settings will not be explained here, but a detailed information is provided in the FePEST help system and the PEST documentation.

5.2 Starting PEST

To start the predictive analysis, follow the same steps as described for the parameter estimation in Section 4.

5.3 Output during Predictive Analysis

The objective function and its profiling illustrates the progress of the optimization run.

If the initial parameter set is not from a calibrated model, PEST runs a calibration process to ensure that the objective function value is less than the defined target objective function.



PEST then starts maximizing/minimizing the model prediction. The objective functions landing between target objective function and acceptable objective function are permitted. Outputs follow after the PEST run is completed.

5.4 Estimation Results

The first line of the parameter list show the maximum or minimum prediction value found. The corresponding parameter set may or may not resemble the calibrated model, and can be exported to FEM file for further investigation.

PEST also generates additional statistical data. These are shown in additional panels after the model run.





6 Pre-calibration Monte Carlo

FePEST provides an option for doing a “pre-calibration” Monte Carlo analysis directly from the Problem Settings dialog (see Figure 6.1). The “pre-calibration” entails the use of Monte Carlo method before calibrating of the FEFLOW model. This is particularly useful to have a first idea of the sensitivities of the model output to certain parameter variation based on the prior knowledge of the modeller (e.g. range of conductivities for a specific unit, etc.). The results of the pre-calibration Monte Carlo analysis can then be used in the calibration process, for example, with a primary focus on certain model parameters.

6.1 Criteria for randomly generated parameters

The Monte Carlo Analysis enables the user to randomly generate set of parameters (zonally-constant or pilot points) based on a (user-defined) criteria:

- **Number of samples:** maximum number of parameter realizations for Monte Carlo analysis.
- **Distribution type:** this can either be a normal distribution or a uniform distribution. The defined distribution type is used for the random generation of the parameter. If the parameter definition uses log-transform, the parameter generation also result in a log-transformed values.
- **Mean method:** the random set of parameter is centred on an user-defined mean value, which could either be the current initial value of the parameter or computed as the midpoint of the bounds of the parameter. This information is provided in the section Parameter Definition of the Problem Settings dialog.
- **Respect ranges:** it is possible for some set of parameters, generated by a fully-random process, to fall outside of the expected parameter ranges (for example, defined for the calibration). The user has the option to set the ranges of the parameter (including tied parameters) accordingly.
- **Uncertainties:** the uncertainty associated to the distribution of the parameters (i.e. how much a parameter can vary according to modeller’s knowledge) can be prescribed by three options:
 - **Bounds:** The standard deviation for the parameter generation is computed from the parameter bounds automatically. This is calculated as the difference between the log-transformation of the parameter bounds divided by 4. The rough estimation is based on the assumption that the parameters are log-normally distributed, and the intervals between respective parameter bounds correspond roughly to 95% pre-calibration parameter confidence intervals. The differences between these bounds then correspond to approximately 4 parameter standard deviation. Alternatively, the standard deviation can be provided by the user in the input field.
 - **Covariance matrix:** the same covariance matrix used for the Tikhonov regularization is used as a measure of uncertainty to compute the random parameters.

- **File:** Experienced users can provide an “uncertainty file (*.unc)” for the standard deviation or covariance matrix of a parameter. The uncertainty files conform to certain PEST definitions. It can also provide measures of uncertainty either for a group of parameters or to specific parameters.
- **Seed number:** By default, the parameters generated using the Monte Carlo Analysis in FePEST are fully random. This implies that the same parameters cannot be obtained between different FePEST problems even when the same distribution type, mean value and uncertainty source is considered. However, identical parameters can be obtained by providing the same seed number across all FePEST problems.



The Monte Carlo option in FePEST (either pre-calibration or postcalibration) automatically generates parameter values to all available and active parameter definitions in the project, i.e. defined in PEST Control file (.pst).*

6.2 Optimization of random parameter sets

After the parameters are randomly generated, the user may decide to run a separate PEST problem for each parameter set by activating the option **Optimize**. In doing so, FePEST automatically creates a new PEST Control file for each set of parameter generated. The PEST run is fully parallelized using the BeoPEST utility, as described in the Parallelization section of the Problem Settings dialog.

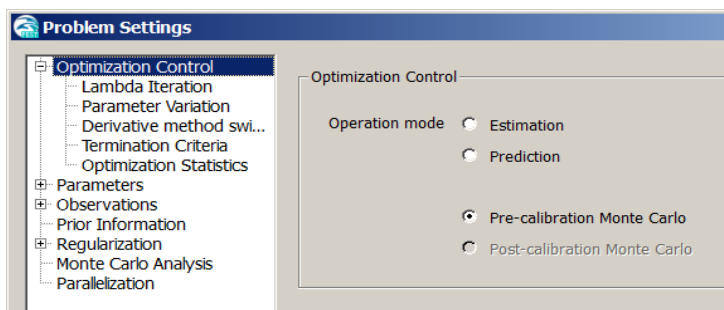


Figure 6.1 Optimization page: Setting pre-calibration Monte Carlo mode.

The optimization option for pre-calibration Monte Carlo analysis allows the user the following (Figure 6.2):

Maximum number of iterations and threshold value

A PEST problem is run separately for each parameter set that is randomly generated using the Monte Carlo analysis. By default, the number of PEST iteration is set to 30 cycles. The PEST optimization will terminate if the maximum number of PEST iterations is reached and/or the threshold objective function (Phi parameter) is achieved.



Objective function

A PEST problem is run separately for each parameter set. No optimization is performed. The aim is to estimate the current objective function for each of the parameter sets without any adjustments. A FEFLOW model is run once per each Monte Carlo realization.

Jacobian and statistics

PEST is run in a special estimation mode to compute the Jacobian matrix and statistics for each parameter set.

Jacobian only

PEST is run in a special estimation mode to compute the Jacobian matrix for each parameter set. Note that each computation of the Jacobian matrix requires a FEFLOW model run.

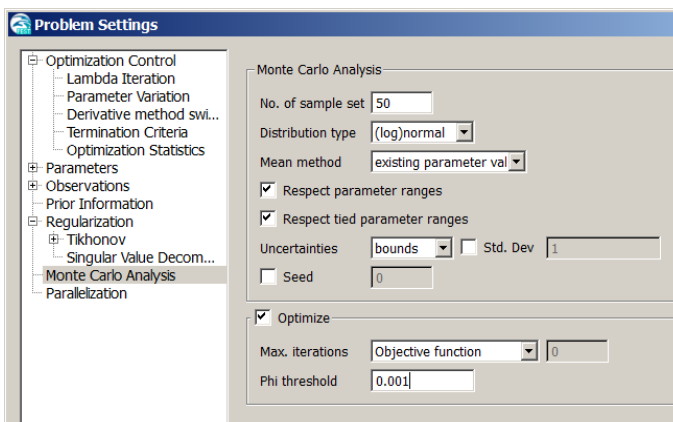


Figure 6.2 Monte Carlo Analysis: Settings and options.

6.3 Running FePEST under Pre-calibration Monte Carlo mode

To run FePEST with the pre-calibration Monte Carlo mode, the same steps are required as for other operation modes (e.g. estimation or prediction). The run dialog is prompted either by clicking the Run button or from the Estimation menu. The dialog prompts the user to create the PEST files for the Monte Carlo analysis and/or to run the problem. (Figure 6.3).

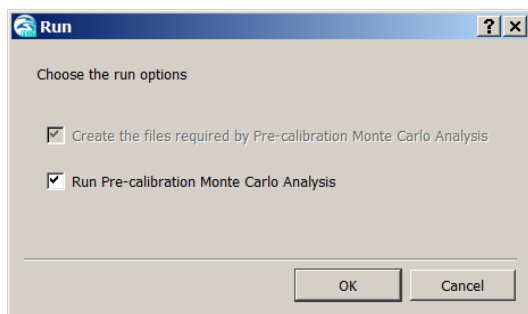


Figure 6.3 Running pre-calibration Monte Carlo analysis in FePEST.

If the option of creating files is enabled, standard PEST files (such as the PLPROC files), regularization files, and several other files (as described below) are created under the pre-calibration Monte Carlo mode:

Parameter file (.par)*

By default the name of this file is composed by *mc_rand* followed by a number, ranging from 1 to the total number of samples for the Monte Carlo analysis. The file contains four columns, namely the parameter name, value, scale and offset. The values are in fact the randomly generated values (using the normal or uniform distribution) for each parameter. The scale and offset are typically set to 1 and 0 respectively.

Monte Carlo PEST file (mc.pst)

This is a PEST control file, which is the same as the project control file (*case.pst*) except that *mc.pst* contains the values of the parameters from each realization. The optimize option in FePEST expects a PEST control file for each parameter realization. Instead of creating separate files, FePEST updates the files after each PEST run.

Parameter summary table (.dat)*

A file under the name *mc_rand_mulpar* is created after executing the PEST utility MULPARTAB. This utility reads all the parameter files contained in the FePEST working directory and creates a file containing the information from all the parameter files.

6.4 Results of the Pre-calibration Monte Carlo Analysis

For generating random parameters, FePEST uses the PEST utility RAND-PAR. After “Pre-calibration Monte Carlo Analysis” is successfully finished, two additional dialogs appear in FePEST, namely “**Parameter Uncertainty**” and “**Observation Uncertainty**”. The first one is always visible whereas the second chart appears only if the user activates the option “**Optimize**” in “Monte



Carlo Analysis” settings. The charts plot a histogram for the parameters and observations, respectively.

A histogram plot for “Parameter Uncertainty” is shown in Figure 6.4. Different parameters can be visualized by changing the parameter name directly in the combo box available. The plots for “Observation Uncertainty” follow the same pattern.

The “Observation Uncertainty” dialog enables the user to see the histograms for each observation defined in the FePEST project (e.g. process variables, rate and period budgets, IFM-implemented observations, etc.).

Moreover, the results of the uncertainty plots can be exported from the “Properties” dialog of the chart in case of further post-processing.

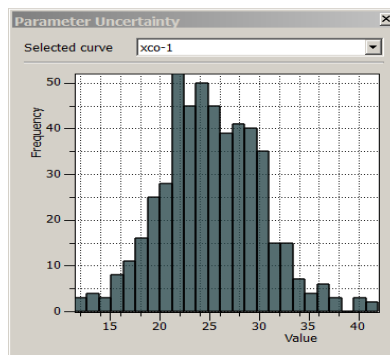


Figure 6.4 Monte Carlo Analysis: Parameter uncertainty chart.

All the different model scenarios created by the “Pre-calibration Monte Carlo Analysis” are available from the menu “**Results - Show Results**” (as shown in Figure 6.5). Each scenario can be saved in a FEFLOW FEM file. This is done by selecting the relevant column followed by clicking the “Save” button.

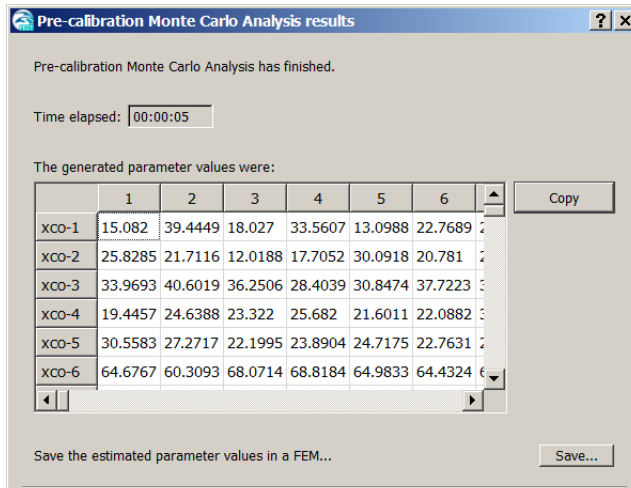


Figure 6.5 Pre-calration Monte Carlo results.



7 Post-Calibration Monte Carlo

The way in which “Post-Calibration Monte Carlo analysis” in FePEST works is very similar to the previously discussed “Pre-calibration” option. However, post-calibration analysis can benefit from pre-calibration results (i.e. Jacobian matrix and super parameters from SVD-Assist) to generate a set of random realizations resulting in an “almost calibrated” set of parameters. The “almost calibrated” implies that the measurement objective function is reached with minimum effort.

7.1 Settings of the Post-Calibration Monte Carlo

The operation mode of “Post-Calibration Monte Carlo” is defined in the **Optimization Control** section of the Problem Settings dialog (as shown in Figure 7.1).

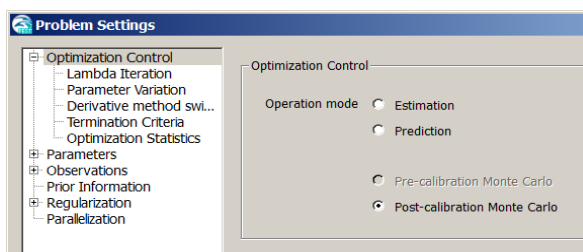


Figure 7.1 Activating post-calibration Monte Carlo operation mode in FePEST.

“Post-calibration Monte Carlo” is similar to the “Pre-calibration Monte Carlo” with the exception of the null-space projection. The Monte Carlo based parameter generation in FePEST, calls the PEST utilities, namely RANDPAR and PNULPAR (one after the other), to create randomly generated parameters that are “almost calibrated”.

The null space projection requires the dimension of the solution space (number of super parameters). This is already known to FePEST as the problem is already calibrated. The user can further select between the “Auto” or “Manual” option to set the dimension of the solution space (Figure 7.2):

- **Auto:** FePEST will run the PEST Utility SUPCALC to determine the number of super parameters in the system and uses this to carry out the null-space projection.
- **Manual:** the user is prompted for the dimension of the solution space.

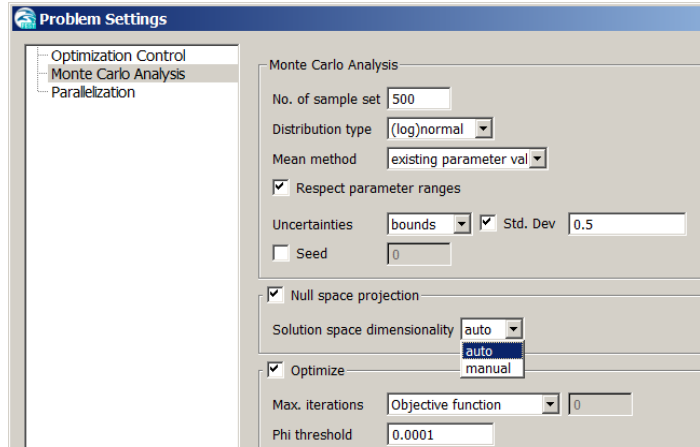


Figure 7.2 Monte Carlo Analysis: Post-Calibration settings.

Similar to “Pre-calibration Monte Carlo”, the post-calibration mode also allows the user optimize the FEFLOW problem for each sample set. If the optimize option is active, PEST will do history matching. This is done relatively quickly as the optimization starts with an “almost calibrated” set of parameters.

7.2 Running FePEST under post-calibration Monte Carlo mode

The button “Run” should be clicked to execute the “post-calibration Monte Carlo” mode. Similar to other modes, it first creates the PEST files. In addition to the RANDPAR output files (*mc_rand*.par*), some new parameter files appear in the working directory under the name *mc_nsp*.par*. These files contain the “almost calibrated” parameters that result from the PNULPAR utility.

7.3 Results of the Post-calibration Monte Carlo Analysis

The “Parameter Uncertainty and Observation Uncertainty” histogram charts are also available for this mode. The “Parameter Uncertainty” dialog plots histograms for each parameter definition. Note that the histograms now relate to the parameters after null-space projection, i.e. “almost calibrated” parameters. Similarly, the “Observation Uncertainty” dialog plots histograms for each observation definition. Figure 7.3 shows an example of these two charts after optimization.

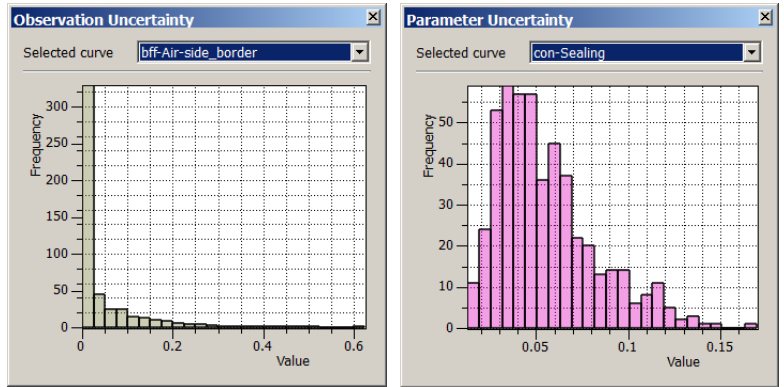


Figure 7.3 Results of post-calibration Monte Carlo analysis: Parameter Uncertainty and Observation Uncertainty charts.

If the optimize option was enabled, a table summarizing the parameter values and the measurement objective function can be accessed from the menu “Results - Show Results”.

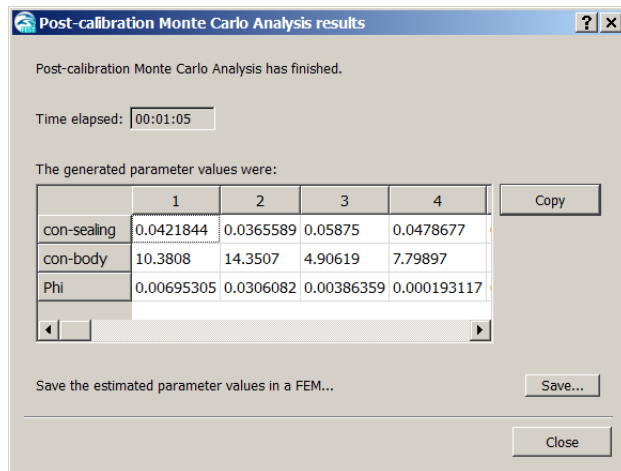


Figure 7.4 Post-calibration Monte Carlo Analysis results.





8 Other Tools

8.1 Sensitivity Export

Linear sensitivity indicates how the parameters influence certain observations. This is expressed in terms of derivatives (saved in a Jacobian matrix) that are calculated during the optimization.

The derivatives (also referred to as sensitivities) are a valuable by-product of the optimization and they can be exported for further investigations.

8.1.1 Required Settings

Once the Jacobian matrix is calculated, the sensitivities are ready for export. This is done whenever optimization is performed (i.e. with any mode - parameter estimation or predictive analysis).

If no optimization is done, the NOPTMAX setting in the Termination criterion settings can be set to "Jacobian only" (-1). With this setting, PEST calculates the Jacobian matrix of the first iteration and terminate without making any changes to the initial parameter values.

Further reading: PEST Manual (5th Ed.) section 2.2.9: Termination Criteria.

8.1.2 Export of Sensitivities

The "Sensitivities" feature in the "Estimation" menu executes the JROW2VEC utility of PEST. This exports the sensitivities of parameters for the selected observations. The resulting table can be

- transferred to a spreadsheet program for further processing (using the "Copy" option), and/or
- exported to an ASCII table (*.dat) file. This is done in conjunction with the spatial maps that relate the parameters to their respective zones or pilot point locations. The sensitivity maps can be created in FEFLOW or GIS software (a FEFLOW example is shown in Figure 8.1).

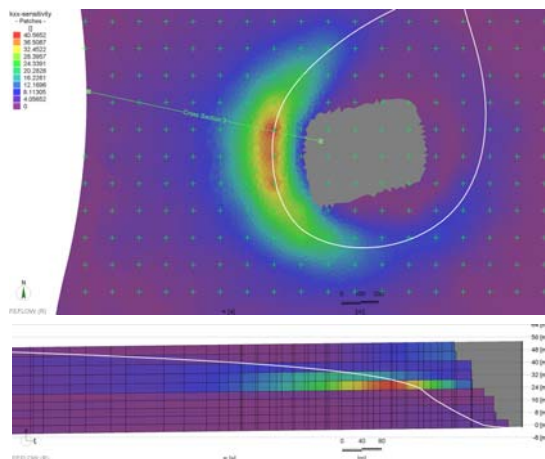


Figure 8.1 Sensitivity maps (plain view of layer 5 and cross-section of indicated line shown.) of hydraulic conductivity in a 3D model. These are created by spatial interpolation of pilot point parameter sensitivity. White line indicates the water table.

8.2 Model Stability Tests (JACTEST)

As discussed in Section 2.1.1, it is important that the numerical model is stable enough to reproduce observation - parameter relationships.

As a measure of trouble-shooting, PEST provides a tool (JACTEST) to check if the model is running sufficiently stable to calculate the Jacobian matrix. This tool can be activated using FePEST with the “Model Stability” option in the “Estimation” menu.

JACTEST runs the model multiple times with incremented/decremented parameter values for derivative calculation and plots them against the selected model outputs (observations).

Further reading: Addendum to the PEST Manual (5th Ed.) section 3.20: JACTEST

8.2.1 Using JACTEST

If PEST fails to reduce the objective function and is suspended due to bad derivatives, it is recommended to stop PEST (if still running). Open the “**Show Results**” dialog and “**Apply**” the latest values as initial parameter values (you may want to save the FPS file now under a different name).

Open the “Model Stability” dialog from the “Estimation” menu. The user is prompted to do the following:



- Specify the parameters to be tested - only the parameters that are suspected to cause instabilities should be chosen here.
- Specify the number of iterations to be performed for each parameter. A value of 4 (or greater) should at least identify the cases where parameter–observation relations are random.

After the calculation has finished, the diagram window shows the resulting observation values. Stability of the model is ensured if the slopes are similar (with slight changes). An example for a stable model is shown in Figure 8.2.

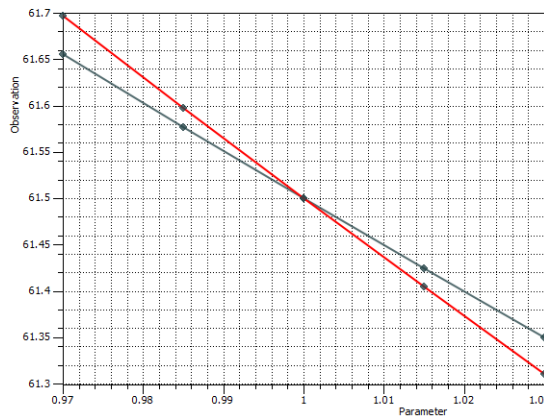


Figure 8.2 JACTEST result indicating stable model behaviour. Two parameters are incremented / decremented four times each by 1.5 % around the initial value, the resulting values for a particular hydraulic head observation are plotted.

However, if the parameter values change randomly (and significantly), noise dominates the observation. It is useful to identify the affected parameter–observation combinations, and open and run the model in FEFLOW to find and solve the instability issue.

8.3 Linear Analysis (GENLINPRED)

The PEST Utility GENLINPRED stands for general linear predictive uncertainty/error analyser. This utility driver sets different utilities to carry out the assessment of the uncertainty and/or error variance of a parameter or prediction. Some of the tasks undertaken by GENLINPRED utility in FePEST are summarized below:

- Estimation of the optimal number of dimensions of the calibration solution and null spaces.



- The parameter identification and relative reduction of the error variance of each parameter. It also computes the relative uncertainty variance reduction of each parameter.
- Calculation of the solution and null space components of the total error variance of a specific FEFLOW prediction at different levels of singular value truncation. It follows with an evaluation of the number of pre-truncation singular values employed in the calibration of a FEFLOW model.
- Estimation of the uncertainties for the pre- and post-calibration stages.
- Contribution to the pre- and post-calibration error variance and/or uncertainty of a nominated prediction (parameter) made by different parameter groups or individual parameters.
- The worth of different observation groups or different individual observations for lowering the post-calibration error variance and/or uncertainty of a nominated prediction (parameter).

The PEST Utility GENLNPRED is available in FePEST from the menu Estimation - Utilities. GENLNPRED dialog requires certain inputs depending on the specific tasks. Further details of the parameters can be found in the Addendum to the PEST Manual.

After GENLNPRED is run, all the results are available in the Linear Analysis tab in FePEST. For a more comfortable visualization, the PEST result file can be directly opened from FePEST by pressing the button "View".



Figure 8.3 Linear Analysis (GENLINPRED).

8.4 Sensitivity Analysis (SENSAN)

Sensitivity analysis of particular model outputs to particular model configurations (i.e. material properties) is a common modelling task. Such a task can also be considered as the initial step before model calibration in order to understand the relevance of the model inputs.

FePEST provides a fast way to estimate sensitivities, relative differences and model outputs from a specific set of model parameters. For this task FePEST uses the PEST Utility SENSAN.

A user-defined parameter list can be imported that can be used to configure several FEFLOW model scenarios. In principle, unlimited number of model runs can be used in SENSAN, but the utility is not parallelized in PEST. Therefore, it is best to choose the maximum number of scenarios such that the computational resources are not exceeded. If the uncertainty of the parameters to the model outputs has to be explored, the Monte Carlo Analysis in

FePEST is recommended as it takes advantage of the parallelization capabilities.

8.4.1 Definition of Parameter Variability Sets

The definition of the variability parameter set can be carried out either manually or automatically (Figure 8.4). The manual addition of model scenarios is included by clicking on the button “New” and then modifying the parameter value in each new row of the parameter variability table. The automatic generation of parameters involves the modification of all active parameter by a specified factor (e.g. 10% of the initial parameter value).

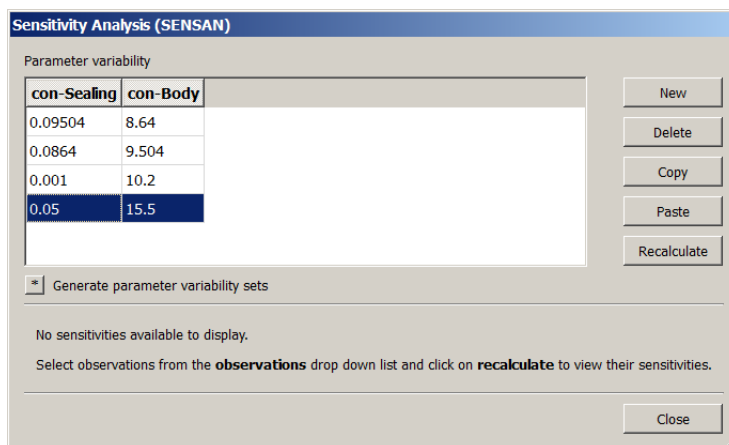


Figure 8.4 Sensitivity Analysis Utility (SENSAN): Parameter variability set.

Once the parameter variability sets are defined, SENSAN utility is executed by pressing the button “Recalculate”. Immediately, the information about SENSAN run and its progress is available in the Output tab in FePEST.

After SENSAN has finished to run all the FEFLOW model scenarios, a summary table of the results from the multiple runs is available in the Sensitivity Analysis dialog. This table includes three main results (Figure 8.5):

- **Absolute output:** includes the different parameter set and model outputs. The model outputs are the standard observation points (i.e. information of the FEFLOW process variables), rate/period budgets, IFM-implemented observations, etc.
- **Relative differences:** contains the estimate of the relative differences (absolute differences) for all the parameter variability sets with respect to the initial parameter value (i.e. problem settings - parameter definitions).
- **Sensitivities:** the sensitivities of all the parameter variability sets with respect to the initial parameter value are listed.

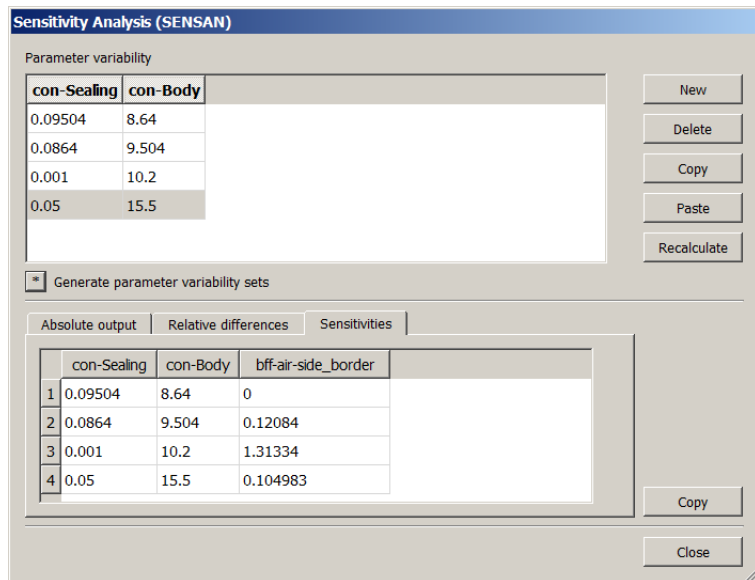


Figure 8.5 Results of Sensitivity Analysis Utility (SENSAN): Absolute values, relative differences and sensitivities.

All the information available in the different section of the Sensitivity Analysis dialog can be copied entirely to any spreadsheet editor or text editor for further post-processing (if needed).





9 Customized PEST Setups

PEST is a model-independent parameter estimator and is easy to integrate with most numerical models. The transparent design of its file setup allows the user a lot of flexibility for customization and implementation of desired functionality.

One of the challenges during FePEST development was to make sure that PEST setup also maintains its flexibility and that it could be used independently of FePEST. FePEST provides a convenient user interface to set up the PEST files, but at the same time, the user is free to continue working with PEST in the traditional way (i.e. by directly editing its configuration files and running its tools from the command line).

This release of FePEST implements those PEST features that are commonly used for model calibration. As it was discussed before, PEST has a lot more to offer.

Experienced users of PEST could potentially benefit from features like Pareto methods or Null-Space Monte-Carlo analysis, or editing certain PEST variables or options that are not accessible through the FePEST GUI. Implementation of own pre- or post-processing code (including third-party software and IFM plug-ins) into the PEST iteration loop might also be desired.

To benefit from both, FePEST and traditional PEST, it is ideal to configure the PEST setup in FePEST and generate the respective files that can be edited (if desired). Finally PEST can be started (in some cases, this can be done again from within FePEST).

9.1 Creating the PEST Setup

The PEST setup files are created when the PEST run is commenced. To edit the files, press the run button, but disable the options for running PEST (and recalculating the Jacobian, if applicable) before pressing the OK button (see Figure 9.1).

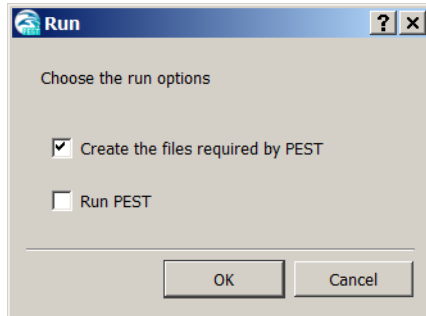


Figure 9.1 Settings in the Run dialog to create the PEST files.

The Checks option will also initiate the creation of the files if these are not present when it is executed.

9.2 Convenience Tools

To make the manual editing of the file setup easier, and to allow quick access to the relevant files and locations, the following features appear in the “Estimation” menu of FePEST after the PEST files have been generated.

- **Checks**
Runs the PEST error check tools PESTCHEK (checks the control file), INSCHECK (checks the instructions file) and TEMPCHEK (checks the template file). Errors and warnings - if raised - will be shown in the dialog.
- **Open work folder in explorer**
Use this option to open the work folder, to operate with files or to open them for editing.
- **Open work folder in console window**
Use this option to open a new command line window, from where PEST (and other) commands can run.
- **Open control file**
Directly opens the PEST control (*.pst) file in the standard editor.
- **Show command**
Shows a command line string with that the PEST run can be initiated. Use this command if you want to run PEST outside the FePEST GUI.

9.2.1 File Structure

Figure 9.2 shows the basic file structure as it is created by FePEST for the simplest case. Additional files may be present depending on the selected methods.



- **PEST script (run_pest.bat)**
This script file is executed when the PEST run is started by the user within FePEST. Besides other commands, the PEST executable is called. The content of this script will vary depending on the settings done in FePEST (e.g., SVDAPREP will be run from this script if SVD-Assist is active).
Change this script to implement additional commands to the overall PEST process.
- **Model script (run_model.bat)**
This script is executed by PEST directly to commence a model call (It is cited as the model command in the PEST control file). The actual FEFLOW execution is cited in here.
Add commands to this file that should be executed with every model run.
- **FePEST input file (<case>.fpi)**
PEST writes parameter values to the FePEST input file (as advised in the respective template file), from where they are read by FePEST. The format is kept very simple, thus that user-specific scripts or plug-ins can access it.
- **FePEST output file (<case>.fpo)**
FePEST writes observation values to the FePEST output file, from where they are read by PEST (as advised in the respective instructions file).
Once again, a very simple format is used thus that user-defined scripts or plug-ins can access this file.
- **PEST control file (<case>.pst)**
This is the primary PEST control file which defines the configuration of PEST. Edit this file to make changes that are not supported by FePEST.
- **SVDA files (super.pst and others)**
If SVD-Assist is used, a second pest setup will appear in the directory after SVDAPREP has been executed by the PEST script.
- **Supportive files**
Other files may appear depending on settings. E.g., several PLPROC files will be generated if the pilot-point method is applied, and a covariance matrix will be created by PPCOV if pilot-points are regularized.
The FePEST help contains a detailed description of these and other files.

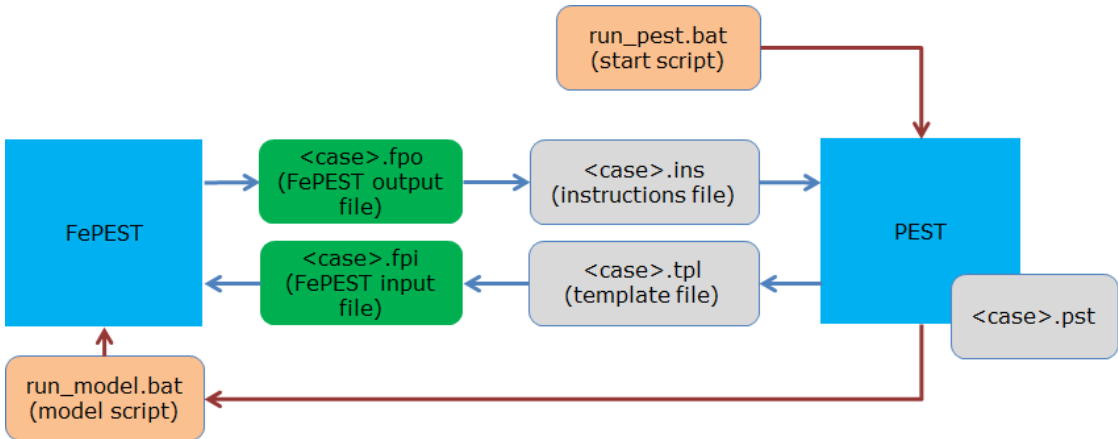


Figure 9.2 PEST file structure created by FePEST. Additional files may be present depending on the used methods (e.g., SVDA or pilot points)

9.3 Running Customized PEST Setups

A customized setup of files can be run in two ways: Either from within or outside of FePEST:

Run from within FePEST

This option is more convenient and should be preferred if applicable. Click on “Run” again, but deactivate the Create Files option this time (otherwise, the files would be overwritten, see Figure 9.3).

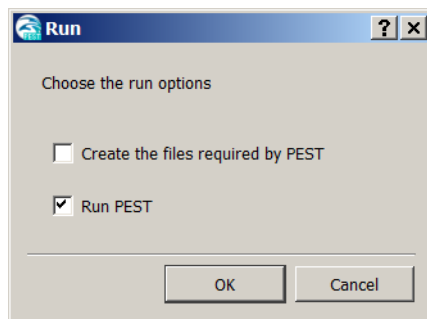


Figure 9.3 Settings in the Run dialog to run PEST without recreating the file setup.

FePEST will commence the PEST run by running the PEST script (run_pest.bat under windows), including all user modifications.

Note that because FePEST parses the primary output of PEST during run-time, FePEST visual feedback might fail partially or completely if the PEST output changes significantly due to the user modifications.



PEST script from command line

The FePEST is part of a superordinate batch run, the PEST script (or its contents) can be incorporated into another script.

Even though visual feedback cannot be provided any more, FePEST (called in its simulation mode) still ensures that parameters and observations are correctly transferred between FEFLOW and PEST.

9.4 Integrating IFM Plug-ins and Third-Party Code

When creating a new observation definition or parameter definition, FePEST offers the type "IFM implemented" to add a specified number of additional parameters to the PEST file setup. These can be directly modified by an IFM plug-in or a third-party code (e.g., a script or another program).

IFM plug-ins

An IFM plug-in must be attached to the FEFLOW model and registered in FEFLOW before the PEST run is started. If parallelization involves remote servers, the plug-in must be installed on all remote machine as well (these will not be transferred automatically).

It opens the fpi-file during run-time of the model to read parameters, and opens the fpo-file to save observations.

Scripts and third-party software

Any program used for additional postprocessing can access the fpi (to read parameters) and fpo file (to save observations) accordingly. To include them in the optimization, they have to be started appropriately from one of the batch files (either the model batch file or the pest run batch file).

For user-defined parameters, consider creating a new parameter group with appropriate settings for the derivative calculation.





10 FePEST Tutorial

For this exercise, a new FEFLOW file **fepest1.fem** is created based on the original demonstration-exercise file **exercise_fri_13.fem**. At this stage, all boundary conditions, material properties and problem settings have already been assigned. The file is located in the folder femdata in the tutorial section.

While a steady-state flow problem is considered here, the same workflow can be followed for other FEFLOW problem classes (including transient simulations) as well. The tutorial covers the four main steps that apply to any FePEST problem:

- Definition of observation points (in FEFLOW and/or FePEST)
- Definition of parameter zones (in FEFLOW)
- Definition of a PEST problem (in FePEST)
- Post-processing of PEST results (in FePEST and FEFLOW).

10.1 Conceptual Model

The material property **Conductivity K_{xx}** is calibrated in this FePEST demo exercise. Conductivity measurements were taken in different locations at the surface level (**Layer 1** in FEFLOW). Initially, the spatial distribution of conductivity in FEFLOW for this slice was computed by means of a linear Akima regionalization method. Using existing information of conductivity measurements, structure parameters of the semivariogram (or variogram) were identified. Methods for variogram fitting are outside of the scope of this chapter.

Two sets of field measurements (**Hydraulic head**) are used for the parameter estimation problem.

Pilot points together with a regularization approach are applied to estimate the final conductivity distribution for **Layer 1**. Simple **Prior knowledge** is used to incorporate existing information of conductivity in the regularization techniques.

10.2 Field Measurements and Reference Data

Seven **Observations Points** are already defined in **fepest1.fem**. Together with four additional observation points to be included later in FePEST, they will be used for the calibration of **K_{xx}** . Further details of observation-point manipulation will be discussed in this chapter.

Figure 10.1 shows a distribution of measured conductivity values. Here, a first overview gives an idea of the wide range of conductivity values in the model domain. A classical approach in this situation is to apply a log transformation to the distribution of **K_{xx}** . From the measurement distribution, a variogram structure can be estimated using geostatistical methods. In our case, the distribution of **$\log(K_{xx})$** is well represented by a spherical variogram with

nugget = 0, sill = 0.58, and range = 4200. This information will be implemented in the FePEST problem.

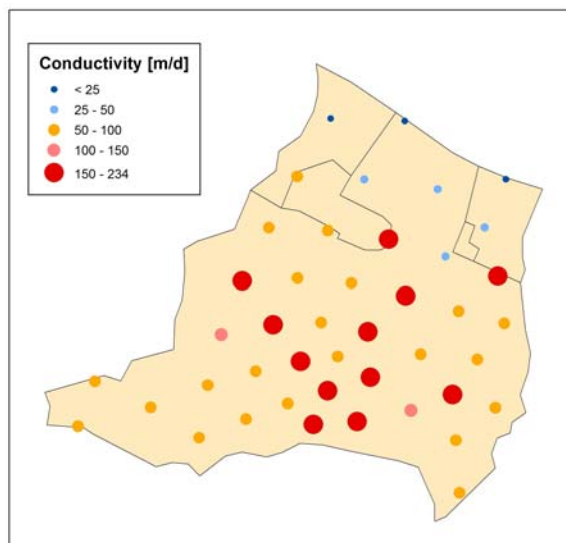





Figure 10.1 Field measurements of conductivity in the Friedrichshagen FEFLOW model domain.

10.3 FEFLOW Model Preparation

If the parameter estimation considers only a specific zone of the FEFLOW model domain (e.g., a layer, slice, or arbitrary subdomain), some preliminary steps are needed in FEFLOW before initiating FePEST.

In this FePEST demo exercise, the conductivity is calibrated on layer 1 only; therefore, an element selection of this area is created first after opening **fep-est1.fem** in FEFLOW 6.2:

- In the active view, go to **Layer 1**.
- In the  **Selection** toolbar, indicate **Elements** as selection target.
- Click on  **Select All**.
- In the  **Entities** panel, store the element selection and rename it as **Zone1**.

This element selection **Zone1** will be available in FePEST as a **Zone** for the parameter estimation.

The **Observation Points** already included in the FEFLOW problem can be verified by opening the **Observation Point Properties** dialog (Figure 10.2).

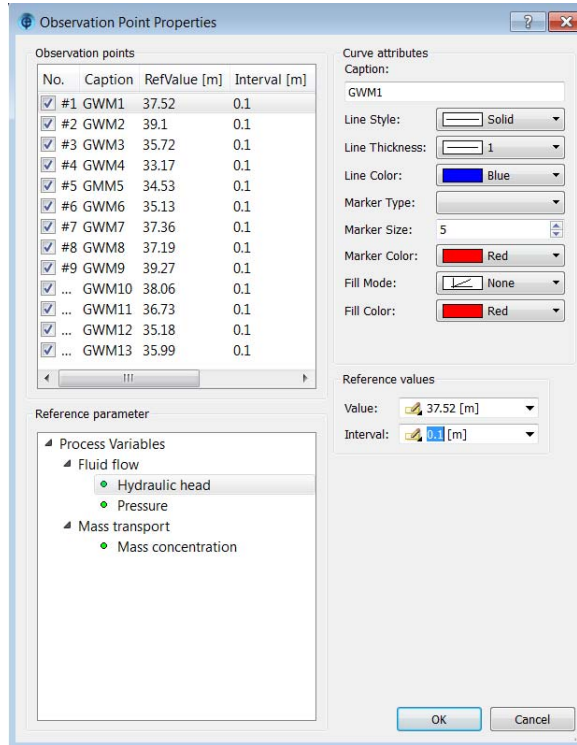


Figure 10.2 Observation Point Properties dialog.

The same dialog is also used to assign measurement time series to observation points.

Click on **OK** to leave the dialog without any changes. In the **Slice** view, navigate through the different slices. Notice that **Observation Points** are situated on slices 1 and 4 of the FEFLOW model while the actual parameter estimation in this example will be limited to the first slice. Observation points to be included for the calibration problem can be still defined later in FePEST.

Save the FEFLOW file as **fepest2.fem** and then close FEFLOW.

10.4 FePEST Initial Configuration

FePEST is the graphical user interface for PEST. Although FePEST provides full communication between FEFLOW 6.2 and PEST (main program and PEST utilities), the user has the option to manually update the PEST program files and thus to control which PEST version is used in FePEST. During the installation of FEFLOW 6.2, the PEST program files are automatically installed as well. Their location is controlled via the menu **Tools > Options** in FePEST (Figure 10.3).

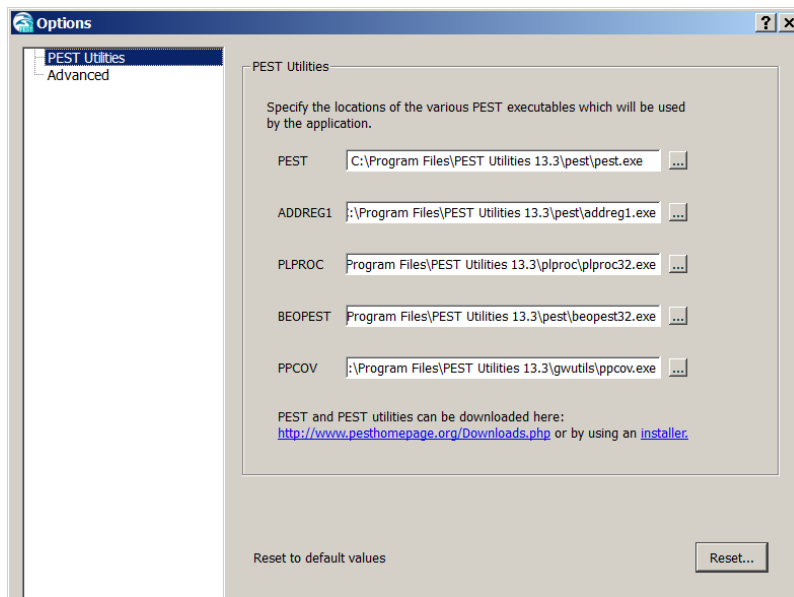


Figure 10.3 Location of main PEST files and PEST utilities.

10.5 Definition of a New FePEST Problem

Open FePEST and create a new problem from the menu **File > New**. A dialog prompts for the location of the FEFLOW fem file that will be used for the FePEST optimization. Select the previously saved file

`\\tutorial\femdata\fepest2.fem`.

Click on **Bu** **OK** to create a new FePEST problem. FePEST now automatically retrieves all important information from the FEFLOW model such as problem class (flow, mass, heat and/or age), material properties, observation points with their corresponding time series (if applicable), stored element selections, etc.

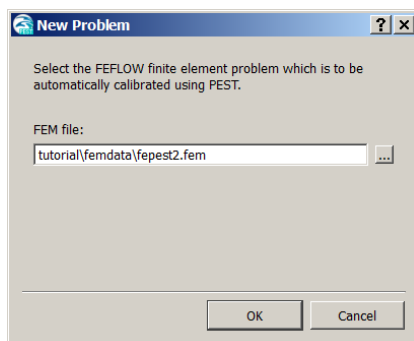


Figure 10.4 Definition of new FePEST problem.

10.6 Optimization Settings

Next, the FePEST **Problem Settings** dialog will prompt for the information needed to define the PEST problem and to create the main PEST files:

- Optimization Control
- Parameters
- Observations
- Prior Knowledge
- Subspace Regularization
- Predictive Analysis
- Parallelization

Experienced PEST users will recognize the traditional code names of the PEST parameters shown in bold letters.

While for most groundwater models, the default values suggested by the PEST manual and indicated in the **Problem Settings** dialog should be appropriate, some modifications may be needed for highly-parameterized FEFLOW models.

In the **Optimization Control** section, the PEST operation mode is indicated. For the case of this exercise, PEST runs in **Estimation** operation mode. This is the default mode in FePEST.

In **Optimization Control > Termination Criteria** (Figure 10.5), the different criteria to terminate the PEST execution are listed. A maximum number of PEST iterations (**NOPTMAX**) of 30 is recommended. It is worthy to notice that a single PEST iteration includes several FEFLOW model runs. The total number of FEFLOW runs depends on the number of parameter to be calibrated and the selected kind of PEST statistics. The most-demanding PEST operation is the computation of the **Jacobian Matrix**.

Other termination criteria are the relative reduction of the objective function (**PHIREDSTP**), the maximum number of consecutive iterations which failed to lower the objective function (**NPHINORED**), the maximum relative parameter change (**RELPARSTP**), and the maximum number of consecutive iterations with minimal parameter change (**NRELPAR**).

The value of **NOPTMAX** in FePEST also defines the type of action to undertake in the current PEST run. In a classical PEST configuration (no FePEST interface), a value of **NOPTMAX** equal to 0 means that PEST runs only once to compute the measurement objective function. A value of -1 computes the **Jacobian Matrix** and statistics, and a value of -2 the **Jacobian Matrix** only. In FePEST these options are controlled via the combo box below **NOPTMAX** in the **Termination Criteria** section.

The type of optimization statistics (covariance matrix, correlation matrix and eigenvalues) for the final PEST run needs to be indicated in the **Optimization Control > Optimization Statistics** section.

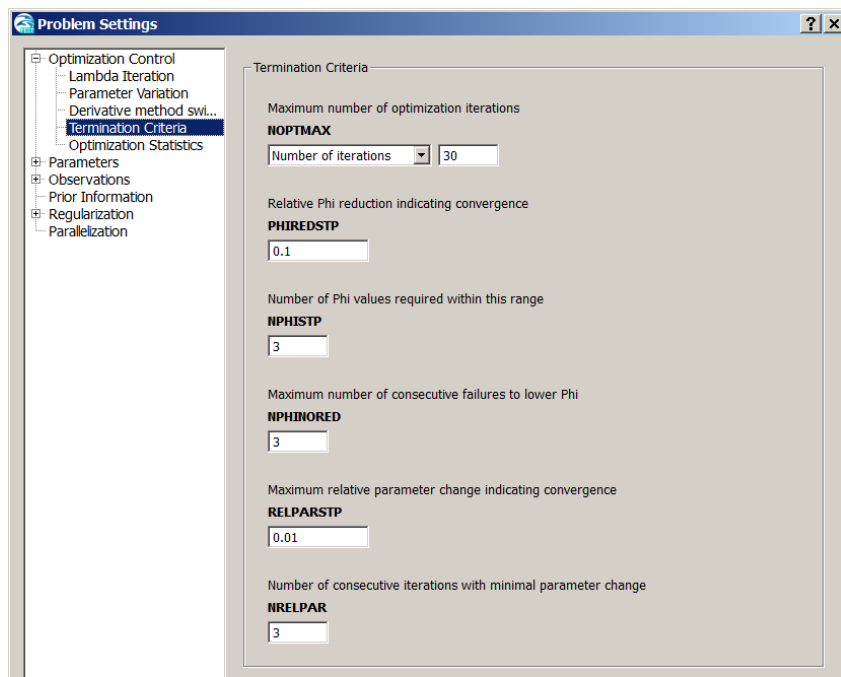


Figure 10.5 Problem Settings dialog: Termination Criteria.

10.7 Parameter Definition

Parameters in a FePEST optimization problem can be defined either as homogeneous zones, or as spatially variable using pilot points. In both cases, material properties (so-called **Parameter definitions**) are specified for



Zones (or for the entire model domain). Typically, zones reflect different geological units with respective FEFLOW material properties.

Zones correspond to stored element sections in a FEFLOW fem file. After opening the file with FePEST, all existing element selections will be listed in the **Zones** section of the **Parameter definition** dialog.

In the **Parameters** section of the **Problem Settings** dialog, click on **Bu New** to create a new parameter definition. The **Parameter definition dialog** (Figure 10.6) shows that the FEFLOW file contains an element selection named **Slice_1**.

In the **Parameter definition** dialog (Figure 10.6), select **Slice_1** as **Zone**, choose **K_{xx}** as **Parameter type** and **Pilot points in 2D, all layers adjustable** as the **assignment method**.

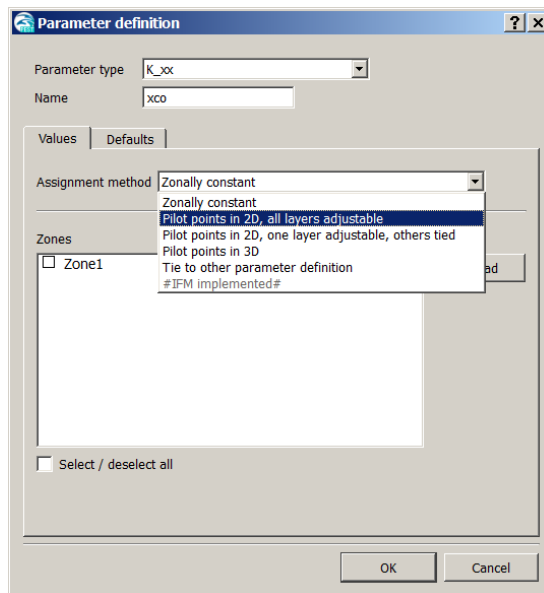


Figure 10.6 Parameter definition dialog.

The assignment method via **#IFM implemented#** is for advanced users with experience in FEFLOW interface programming. A FEFLOW plug-in (previously attached and registered in FEFLOW 6.2) can be made responsible for transferring parameter values to FePEST. This is a powerful option to calibrate FEFLOW parameters not available via the FePEST graphical interface, e.g., boundary conditions.

After defining **Pilot points in 2D, all layers adjustable** as the assignment method, the **Parameter definition** dialog slightly changes.

Now the locations of pilot points need to be specified using a specific geographical position (X, Y, Z), slice or layer number as reference.

FePEST provides a very convenient automatic generator of pilot points. Click on **Bu** **Generate** to open the **Pilot point generation** dialog (Figure 10.8) which will prompt for the number of points desired. For this exercise, enter a value of **123** **60**. Pilot points can be distributed in one of four automatic patterns: **uniform**, **stagger rows**, **stagger columns** and **random**.

Choose the option **uniform** and close the dialog by clicking **Bu** **OK**.

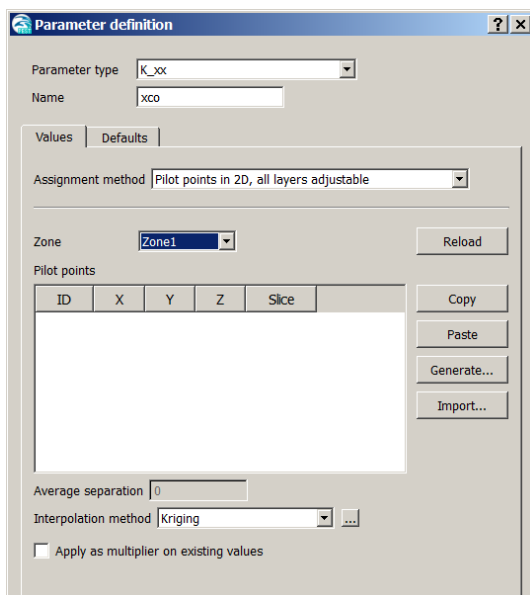


Figure 10.7 Parameter definition dialog in pilot point mode.

Notice that the actual number of pilot points will depend on the geometry of the FEFLOW model domain and pattern specified. Indeed, the final number of pilot points in our case is 61. If the automatic generation was not satisfactory, pilot points can be edited and/or deleted in the **Parameter definition** dialog.

PEST will calibrate parameter values at each individual pilot point. Subsequently, PEST will interpolate (and/or extrapolates) parameter values over the entire **Zone** specified as pilot-point domain. **Kriging** or **Radial basis functions** are available as interpolation methods.

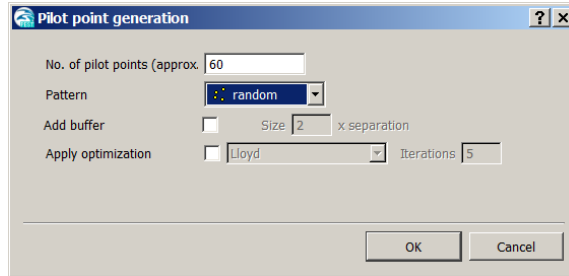


Figure 10.8 Pilot point generation dialog.

For this exercise, the **Kriging** interpolation method is used.

Click on **...** **Interpolation method properties** to assign properties of the experimental variogram. In the **Kriging configuration** dialog (Figure 10.9), four well-known variogram types are available (**Spherical**, **Exponential**, **Gaussian** and **Power**). In Geostatistics, these are called theoretical variograms (or semivariograms) and they represent the spatial structure of a measured property, in particular, how field measurements are correlated depending on sampling distance. Notice that all the variogram properties (**range**, **sill**, **nugget**, etc.) have to be set up in FePEST and the uncertainty of these parameters will also be reflected in the parameter estimation.

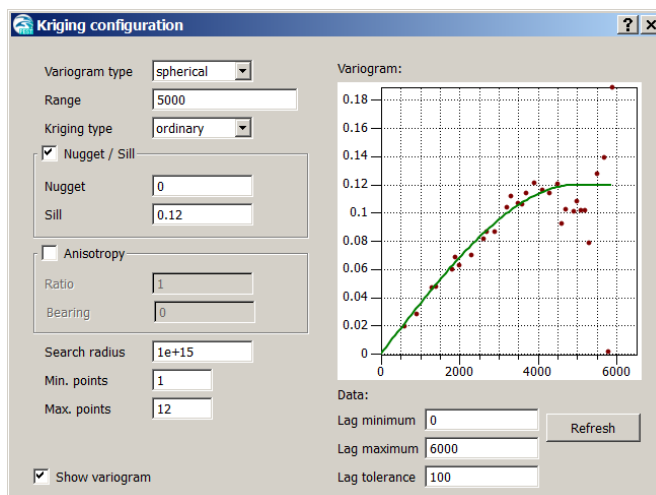



Figure 10.9 Kriging configuration dialog.

After activating **✓ Show variogram** and a click on **Bu Refresh**, FePEST computes the experimental variogram (dots) and theoretical variogram (continuous line) in the chart. FePEST can also compute variograms for different **lag tolerances**. This feature gives a faster overview about the goodness-of-



fit between experimental and theoretical variograms. Chart information can be exported by choosing the  **Properties** option from the context menu.


For this exercise, use the values presented in Figure 10.9. Click on  **OK** to confirm the changes in the **Kriging configuration** dialog.

Next, click on the **Defaults** tab in the **Parameter definition** dialog. Depending on the material property, a parameter log-transformation may improve the PEST optimization result. Parameters such as conductivity and transmissivity are typically log-transformed because they can change several orders of magnitude within a given model domain.

FePEST offers four possibilities to define a PEST parameter (**none**, **log**, **fixed** and **tied**):


- **None:** Parameter calibration is carried out without any transformation of the parameter.
- **Log:** A log-transformation is applied to the parameter before PEST is executed. PEST statistics are presented based on transformed parameters.
- **Fixed:** The parameter is listed within the PEST control file, but is not modified during PEST optimization. Fixed parameters do not influence the objective function.
- **Tied:** The parameter is tied to a so-called “parent” parameter and its value will depend on the current value of the parent. Tied parameters will be updated during the PEST run with the same ratio as given by the initial parameter values.

In this exercise, **Initial value** is set as **current**, i.e., FePEST will automatically retrieve all parameter **K_{xx}** values from the FEFLOW fem file.

In a first PEST run, it is always recommended to provide wide bounds for the parameter to be estimated so that the PEST optimization will not be influenced. Here, modify **Upper Bound** to a value of  **500 m/d**. In further PEST runs, parameter bounds can be adjusted based on parameter tendencies observed from previous runs.

Click on  **OK** to close the dialog.

Since our FEFLOW model contains information in three dimensions, conductivity values for **K_{yy}** and **K_{zz}** are also available in the fem file. However, these two material properties in FEFLOW are related to values of **K_{xx}**, i.e. **K_{yy} = K_{xx}** and **K_{zz} = K_{xx} / 10**. Therefore, PEST and FePEST should also update the spatial distributions of **K_{yy}** and **K_{zz}** according to the modifications of **K_{xx}**. This is accomplished by defining **K_{yy}** and **K_{zz}** as **tied** parameters, with the parent parameter **K_{xx}**.

Click on  **New** to create a new parameter definition. Indicate conductivity **K_{yy}** as parameter type. In the **Assignment method**, choose **Tied to other parameter definition**. Indicate **xco** (Conductivity **K_{xx}**) in the field **Tied to**.



In the **Defaults** tab, modify **Upper Bound** to a value of 10^3 500 m/d. Click on **Bu** **OK** to confirm the changes. FePEST automatically creates 61 pilot points to compute the distribution of **K_{yy}** based on the distribution of the parent parameter pilot points.

Repeat the steps for **K_{zz}**.

Click on **Bu** **Apply** to assign all changes made in the **Parameter** section.

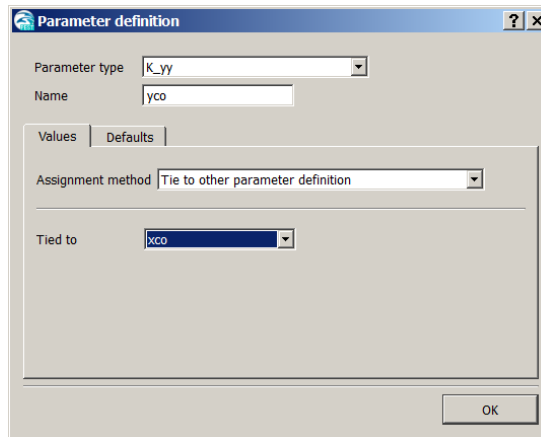


Figure 10.10 PEST parameters for parameter definition (tied parameters).

10.8 Observations Definition

In the **Observations** section of the **Problem Settings** dialog, all field measurements (**Hydraulic head**, **Mass concentration**, **Temperature**, **Ground-water age**, etc.) to be used in the PEST optimization problem are specified.

Create a new observation definition by clicking on the **New** button. In the **Observation definition** dialog, indicate **Hydraulic head** as **Observation type**. As Source method, select **Reference values in the FEM** (Figure 10.11). This option will automatically retrieve the information for all **Observation Points** (Figure 10.2) from the FEFLOW file.

In the **Defaults** tab in the **Observation definition** dialog, indicate a **Weight** of 10^3 2.621 for this observation group. The weight value will reflect to which degree an observation group affects the measurement objective function. Different **Weights** can be used between observation groups to include different uncertainties of the measurements. As a rule of thumb, observation weights can be calculated as the inverse of the measurement standard deviation.

Click on **Bu OK** to add this observation group to the FePEST problem (Figure 10.11).

Since also the **Observation Points** located on **Slice 4** were imported into FePEST, they need to be deactivated so that they do not influence the objective function. Double-click on the observation line corresponding to **GWM2** and change **Weight** to **123 0**. Click on **Bu OK** to confirm the modifications. Repeat these steps for **GWM5, GWM8, GMW10, GMW11** and **GMW12**.

The second set of observation points is added directly using an external file. This group has a higher uncertainty (standard deviation of 0.35 m).

- Click on **Bu New** to create a second observation definition. Define **Hydraulic head** as **Observation type**.
- Select **External file** as **Source** for importing observation points.
- In the file dialog, select the **ASCII Table (*.dat)** named **new_observations.dat**.
- In the **Defaults** tab, insert a **Weight** of **123 0.545** as shown in Figure 10.11.
- Click on **Bu OK** to complete definition of this second observation group.

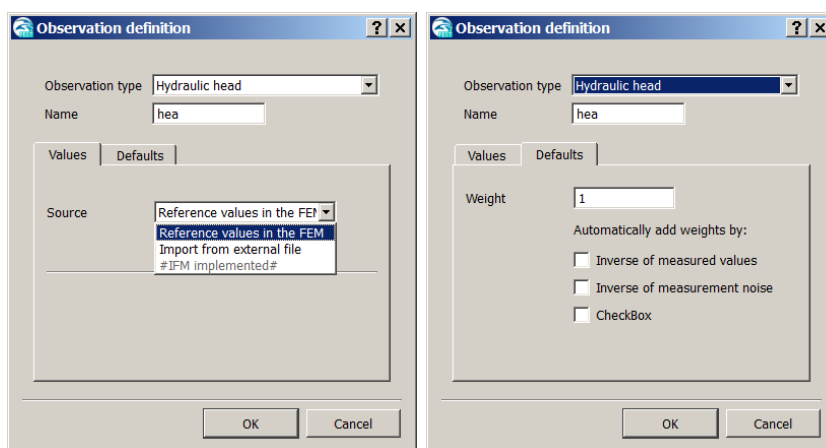


Figure 10.11 Observation definition dialog.

A FEFLOW triplet file can be easily prepared in a spreadsheet or text editor. This file type has a very simple structure and can accommodate geographical coordinates and observation values, as shown in Figure 10.12. Further details can be found in the FEFLOW Help System.



```

new_observations.trp - Notepad
File Edit Format View Help
! Additional observation points for FePEST demo exercise
3410696.218 5816945.212 40.01
3411007.883 5816345.141 40.8
3410449.665 5816552.139 38.95
3411110.426 5817337.974 41.9
END
    
```

Figure 10.12 Structure of a FEFLOW triplet file used to import observations.

Information for all the observations considered in the FePEST optimization problem is given in the lower part of the **Observation** section in the **Problem Settings** dialog (Figure 10.13), including observation type, name, coordinates, value, weights, etc. Any time series associated with FEFLOW **Observation Points** would also be listed here.

Observation items can be edited directly within FePEST. Additionally, FePEST provides the possibility to edit all or some of the observations in an external spreadsheet. Clicking **Bu Copy** will generate a copy of the entire observation list (this also applies for the parameter list).

The screenshot shows the 'Problem Settings' dialog box with the 'Observations' section selected. The 'Definitions' list contains 'Hydraulic head (hea)' and 'Hydraulic head (hea2)'. The main table lists 16 observations with columns for 'Def.', 'Observation type', 'Obs. point', and 'X'. Buttons for 'New...', 'Edit...', 'Delete', 'Copy', 'Paste', and 'Refresh' are visible on the right side.

Def.	Observation type	Obs. point	X
hea	Hydraulic head	GW1	3409115.25 5i
hea	Hydraulic head	GW2	3410265.37 5i
hea	Hydraulic head	GW3	3410123.80 5i
hea	Hydraulic head	GW4	3408241.78 5i
hea	Hydraulic head	GW5	3409557.58 5i
hea	Hydraulic head	GW6	3408938.73 5i
hea	Hydraulic head	GW7	3408361.60 5i
hea	Hydraulic head	GW8	3409953.31 5i
hea	Hydraulic head	GW9	3408565.11 5i
hea	Hydraulic head	GW10	3408112.63 5i
hea	Hydraulic head	GW11	3407877.86 5i
hea	Hydraulic head	GW12	3408487.49 5i
hea	Hydraulic head	GW13	3408078.39 5i
hea2	Hydraulic head	NEW1	3410696.22 5i
hea2	Hydraulic head	NEW2	3411007.88 5i
hea2	Hydraulic head	NEW3	3410449.67 5i
hea2	Hydraulic head	NEW4	3411110.43 5i

Figure 10.13 Final list of observations used for PEST optimization.

10.9 Parallelization Mode

FePEST allows a PEST run in parallel mode. FePEST uses BeoPEST (third-part software) as tool for the communication between slaves and master.

In the **Parallelization** section of the **Problem Settings** dialog, click on **OK** **New** to create a **Slave definition**. In the **Slave definition** dialog, the server to be used is specified under **Host name**. In this exercise, **localhost** is used for sake of simplicity. This means that multiple FEFLOW executions will simultaneously run on different cores of the same computer. As **No. of slaves**, insert a value of **3**. Make sure **Active** is selected (Figure 10.14).

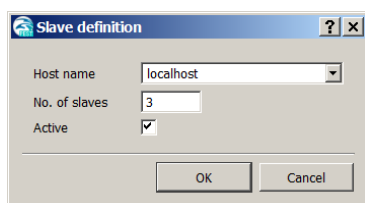


Figure 10.14 Definition of slaves for a PEST run in parallel mode.

Click **OK** to close the **Slave definition** dialog. Click **Apply** to confirm the changes in the **Parallelization** section.

Similarly, slave definitions can be created if multiple slave servers are available. The status (**Active** or **Inactive**) of a slave definition can be modified any time before the PEST problem is initiated.

For reference, all the FePEST runs in this exercise were computed using an Intel(R) Core™ i7 – 3840 QM CPU with 2.80 GHz and 16.0 GB RAM. Depending on the available hardware, FePEST run times may vary.

10.10 Estimation Method and Regularization Setup

FePEST graphical interface provides all the combination of regularization supported by PEST such as:

- Regularization methods
- Tikhonov Regularization
- Singular Value Decomposition (SVD)
- SVD-Assist
- Least Squares Regularization (LSQR)

By default, FePEST uses **Singular Value Decomposition**. This is a very powerful option for FEFLOW models, as discussed in the theoretical part of this chapter. For this exercise, scenarios are implemented with and without Tikhonov regularization in order to explore the importance of the prior knowl-



edge in the calibration process. **SVD-Assist** is used in this tutorial to cut off the number of dimensions of the problem, i.e. a rapid calibration.

10.10.1 Regularization Setup

In the **Problem Settings** dialog, go to the section **Regularization** and activate **Tikhonov**. In the same page include **SVD-Assist** (Figure 10.15).

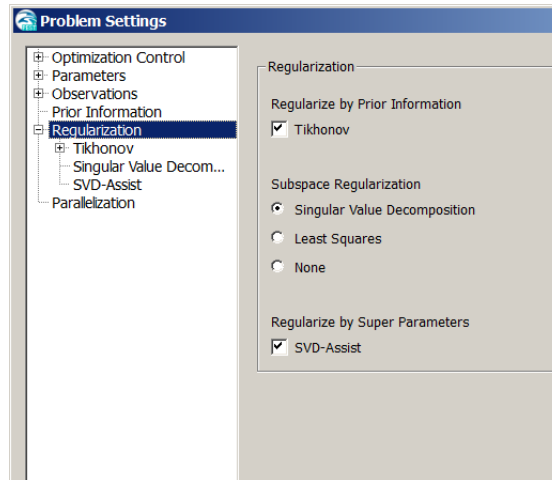


Figure 10.15 Regularization settings in FePEST.

In the section of Tikhonov, activate the three different options of prior information as shown in Figure 10.16.

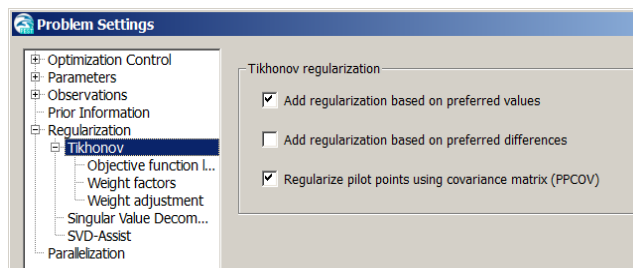


Figure 10.16 Tikhonov regularization settings: Prior information.

In the **Objective function** settings of **Tikhonov**, specify values of **0.25**, **0.5** and **0.1** for the PEST parameters **PHIMLIM**, **PHIMACCEPT**, and **FRACPHIM**, respectively.

Click on **Apply** to confirm the modifications.


10.10.2 Settings of SVD-Assist

In the **Problem Settings** dialog, go to the section **Regularization > SVD-Assist**, the option **Automatically set using SUPCALC** is selected by default as the option to determine the number of super parameters. Click on **Apply** and **OK** to confirm the changes.

Save the changes to **fepest.fps**.

10.11 Running PEST

In the active view FePEST shows pilot points, observation points and parameter zones (Figure 10.17), as indicated in the **Problem Settings** dialog.

Navigation through different slices and layers of the FEFLOW domain is possible via the  **Map Contents** panel (if not visible, this is found in menu **View > Map Contents**).

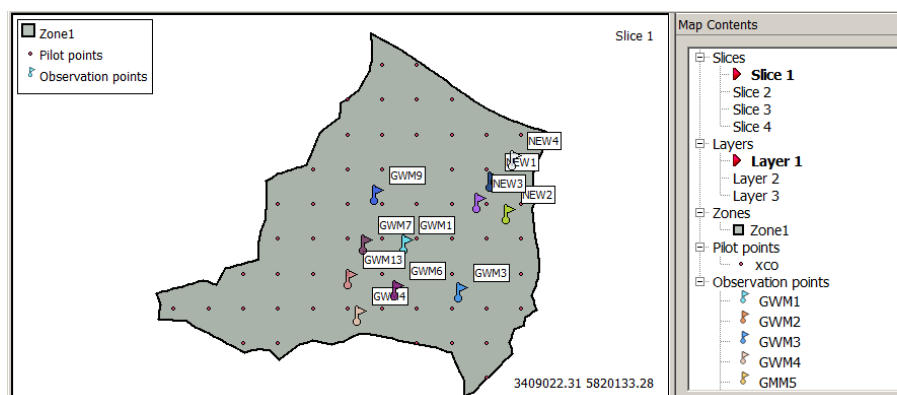


Figure 10.17 FEFLOW model with assigned pilot and observation points for FePEST project.

Before initiating a PEST run, all configuration files need to be created. Click on **Checks** to generate the three main PEST files (Control file, Template file and Instruction file) and batch files for FEFLOW and PEST. At the same time, the PEST check tools **PESTCHEK**, **TEMPCHEK**, and **INSCHEK** are executed. A dialog in FePEST will show any errors and/or warnings found.

If errors are found, modifications can be made via the **Problem Settings** dialog. As a general recommendation, the **Checks** tools should always be run after any modification to the FePEST project.

PEST is initiated with a click on **Run**. FePEST still provides the possibility to re-create PEST files if modifications have been made in the **Problem Settings** dialog. If a FePEST project involves the use of pilot points, this option



should be marked as the pilot-point files **PLPROC** are not automatically created by the **Checks** tool.

In the **Run** dialog, activate **Create the files required by PEST** and click **OK** to initiate the PEST optimization.

The **Output** panel appears after the PEST initialization. In the case of running PEST in parallel mode, additional **Output** panels will appear on the master display for each slave. The information presented in the **Output** panels is identical to that appearing on the console in a classic command-line PEST execution.

In a similar manner the rest of FePEST projects can be executed.

10.12 Postprocessing PEST Results

10.12.1 Optimization Results in FePEST

Immediately after the first FEFLOW model execution, FePEST provides graphical feedback of the optimization problem and the status of the objective function (Figure 10.18). In the **Simulated vs. Observed** panel, the current misfit between FEFLOW results and observations is shown.

Status	
Time elapsed	00:11:50
Objective function Phi initial	255.81
OPTIMISATION ITERATION NO.	11
Model calls so far	310
Starting Phi for this iteration	13.481
Lambda	
Objective function Phi	
Fraction of starting Phi	
Lowest Phi for this iteration	

Figure 10.18 Status panel in FePEST with current PEST iteration information.

After the PEST optimization process has finished, results are available automatically or they can be retrieved via the **Estimation > Show results** menu.

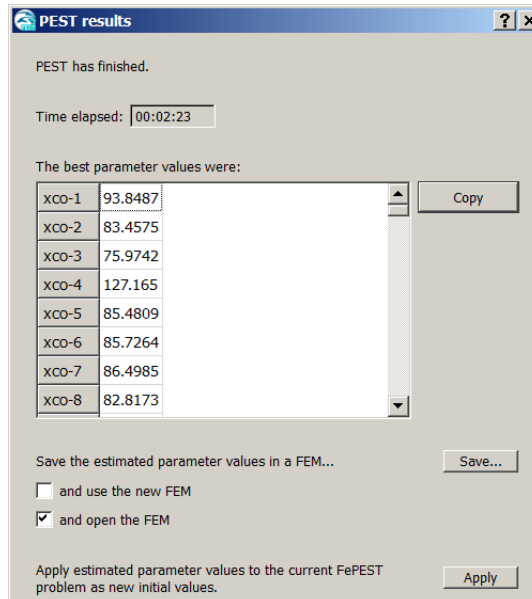



Figure 10.19 Estimation results and different export options.

In the **Estimation results** dialog, the final estimated value of K_{xx} for each pilot point is shown. From here, several options are available:

- Click **Bu Save** to save the results in a FEFLOW fem file with final parameter estimates (automatic assignment of material properties in FEFLOW).
- Click **Bu Save** and select **Use the new FEM** to connect a new FEFLOW fem file (with new material properties) to the current FePEST project.
- Click **Bu Apply** to create a new FePEST project based on the final parameter estimates.

Use the first option to review PEST results directly in FEFLOW. Save the new file as **fepest3.fem**. We will leave the post-processing in FEFLOW for the next steps.

In FePEST you can visualize the  **Simulated vs. Observed** panel (Figure 10.20) to verify the graphically the mismatch between model and historical data. Exact numbers can be visualized faster using the **Tooltip** option (activated via right-click into the chart), and placing the mouse cursor over a data point in the chart.

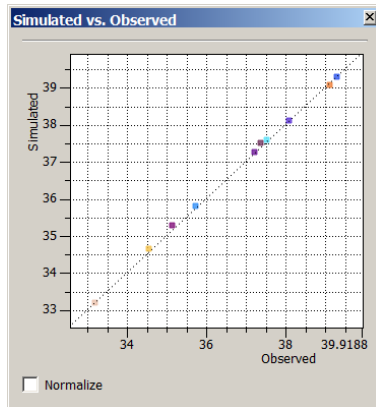


Figure 10.20 Simulated versus observed values in FePEST.

The final misfit can be reviewed in the **Residuals** panel (Figure 10.21), which also contains information on the weighted residuals.

Name	Group	Measured	Modelled	Residual
hea-gwm1	heagp	37.52000	37.51635	3.6533573E-03
hea-gwm2	heagp	39.10000	39.05638	4.3616142E-02
hea-gwm3	heagp	35.72000	35.71407	5.9327724E-03
hea-gwm4	heagp	33.17000	33.16604	3.9580596E-03
hea-gwm5	heagp	34.53000	34.53599	-5.9917698E-03
hea-gwm6	heagp	35.13000	35.12521	4.7869083E-03
hea-gwm7	heagp	37.36000	37.35896	1.0383431E-03
hea-gwm8	heagp	37.19000	37.21094	-2.0940914E-02
hea-gwm9	heagp	39.27000	39.26920	8.0071852E-04
hea-gwm10	heagp	38.06000	38.06263	-2.6297578E-03
hea-gwm11	heagp	36.73000	36.72633	3.6699135E-03
hea-gwm12	heagp	35.18000	35.19137	-1.1366763E-02
hea-gwm13	heagp	35.99000	35.98816	1.8383352E-03
hea2-new1	hea2gp	40.01000	40.45865	-0.4486458
hea2-new2	hea2gp	40.80000	40.29291	0.5070910
hea2-new3	hea2gp	38.95000	39.56311	-0.6131120
hea2-new4	hea2gp	41.90000	42.01934	-0.1193414

Figure 10.21 Residuals panels shows the final misfit between FEFLOW simulated values and measured values.

10.12.2 Results Postprocessing in FEFLOW

Open the file **fepest3.fem** in FEFLOW. Make sure that **Layer 1** is selected in the **Spatial Units** panel. Double-click on **K_{xx}** under **Material Properties > Fluid flow > Conductivity** in the **Data** panel to plot this parameter in the **Slice** view.

In the **Maps** panel, right-click on **ASCII Table Files** and choose **Add Map(s) to folder ASCII Table Files**. Select **new_observation_wells.dat** to include a file with new observation points (second group in FePEST).

Right-click on this new entry and choose the option **Convert to > Observation Points**. Click **OK** to accept the defaults settings.

Double-click on **Observation Points** in the **Spatial Units** panel to plot them in the active view.

The distribution of K_{xx} estimated by PEST is shown in Figure 10.22. Observation points **NEW1-NEW4** strongly influence the conductivity values.

The distribution of conductivity K_{xx} using **Tikhonov, Singular Value Decomposition** and **SVD-Assist** is shown in Figure 10.22. Note that the estimated distribution is significantly smoother due to the regularization techniques. In this exercise, the observation points **NEW1-NEW4** contain a significant measurement noise. From a first inspection to the conductivity field, it seems to be that these do not influence significantly the regularized calibration. The reason of this behaviour falls into the distribution of the observation weights in the FePEST problem. The observation weights have been assigned in such a manner that all the observation groups contribute equally to the measurement objective function, i.e. observations **NEW1-NEW4** must have a lower weight than others. Moreover, the objective function limits for the Tikhonov regularization has been significantly increased compared to the default values in PEST / FePEST. This information instructs PEST to avoid the over-fitting.

Areas of high conductivity (shown in red in Figure 10.22) are even far from reaching the bounds of the parameter definition in FePEST.

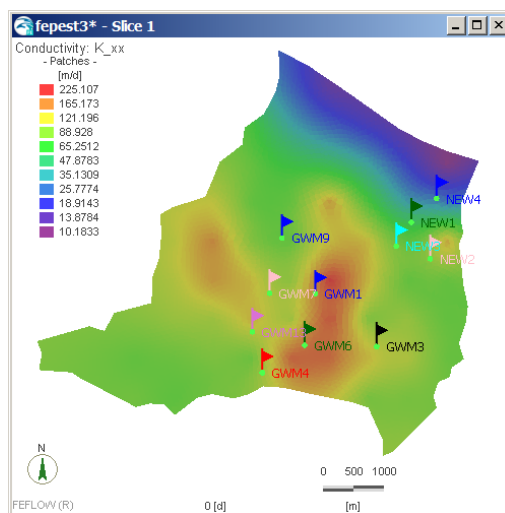


Figure 10.22 Distribution of conductivity K_{xx} estimated including regularization methods in FePEST.



The initial distribution of conductivity K_{xx} has been previously stored as a **User Data > initial_conductivity** in the **Data** panel. Double-click on **initial_conductivity** to plot its distribution in the **Slice** view. By default, **User Data** are plotted using a linear colour mapping. Double-click on **initial_conductivity** in the **View Components** panel to open the **Properties** panel. Here, change to the **Logarithmic** option and click **Apply** to confirm. The distribution of the original (initial) conductivity K_{xx} is shown in Figure 10.23.

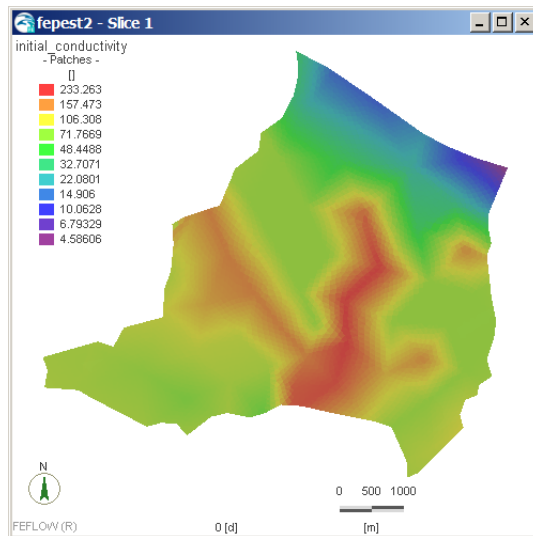


Figure 10.23 Distribution of initial conductivity K_{xx} interpolated by an Akima regionalization technique.

The initial distribution of conductivity K_{xx} looks somewhat “bumpy” due to the Akima regionalization applied.

To compute the relative parameter deviation, the initial distribution **initial_conductivity** is used as reference. In the **Data** panel, right-click on **User Data** to open the context menu and select **Add Elemental Expression**. Rename this elemental expression to **Parameter-Deviation**. Right-click on **Parameter-Deviation** to open its context menu and select **Edit Parameter Expression**. In the **Expression Editor**, write the same expression as shown in Figure 10.24.

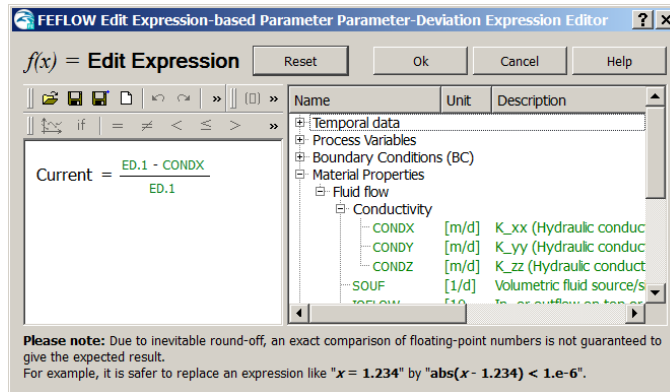


Figure 10.24 Use of the Expression Editor in FEFLOW to compute the parameter deviation in respect to the initials.

Click **OK** to confirm the changes, leave the dialog and plot the element expression in the active view by double-clicking on **Parameter-Deviation** (Figure 10.24). We can identify the areas, where the maximum changes occurred during the calibration by plotting the distribution with the option **Fringes** in the **View Components** panel. Open the **Properties** panel of **Fringes** and select **Arbitrary** option. Click **Edit** to provide the list as shown in Figure 10.25. Finally Click **Apply**.

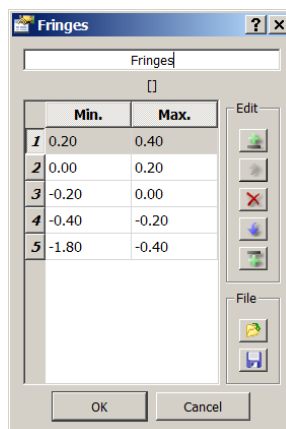


Figure 10.25 Fringes of parameter deviation.

The distribution of the parameter deviation shows that the regularized calibration has indeed modified the conductivity distribution in the model. The areas near the new four observation wells have suffered the maximum changes.

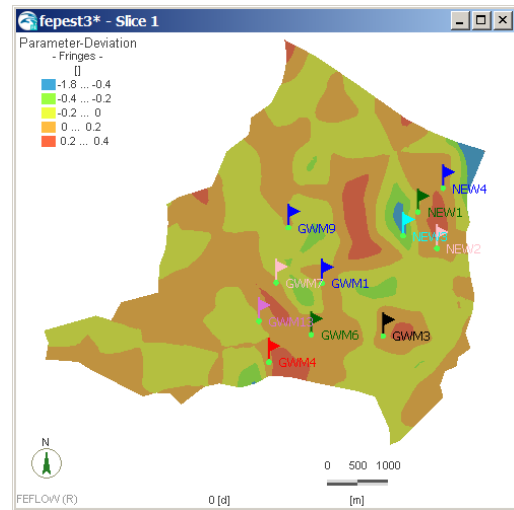


Figure 10.26 Distribution of the parameter deviation between calibrated set and initial set.

Save the changes in FEFLOW 6.2 to **fepest4.fem** and close FEFLOW.

10.12.3 Sensitivity Estimation

In the previous steps we have learnt that some observation points influenced the regularized calibration more than others. Now in FePEST we will intend to calculate the sensitivity distribution

Go back to FePEST and go to the menu **Estimation - Utilities - Sensitivity**. The sensitivity distribution will be computed using the **JROW2VEC** PEST utility. FePEST extracts information from the **Jacobian Matrix** to compute the sensitivity distribution. In the **Sensitivities** dialog and check **Select / deselect all** to consider all the observations. Click **Recalculate** to initiate the calculation. Check **and open the FEM** in order to save the distributions in FEFLOW. You can save this file as **fepest5.fem**. Click **Close** to leave the **Sensitivities** dialog.

Open FEFLOW and load **fepest5.fem** to visualize the sensitivities in the model domain.

Alternatively, the sensitivities table can be copy to any text editor or spreadsheet editor for further post-processing. If that is the case, file can be anyway imported as usual in FEFLOW with the following steps:

- Click **Add Nodal Distribution** under **User Data** in the **Data** panel and rename the new entry to **Sensitivity**.

- In the **Maps** panel, right-click on **ASCII Table Files** and select **Add Map(s) to Folder ASCII Table Files**. Choose the file **pilot_point_sensitivities.dat**. Right-click on this new file and select **Link to Parameter(s)**. In the **Parameter Association** dialog, create a link between **HEA-9** and **User Data > Sensitivity**. Keep the default settings (**Regionalization method Inverse distance**, **Neighbors 4** and **Exponent 2**). Click on **OK** to confirm the changes.
- Activate the link with a double-click on **HEA-9 -> sensitivity** in the **Data** panel. Select **Slice 1** in the **Spatial Units** panel and select all nodes with a click on **Select All**. Reduce the **Snap distance** to **0 m** in the **Snap-Distance** toolbar and click on **Assign**.

The sensitivity distribution for observation **HEA-9**, corresponding to observation point named **GWM9**, is plotted in Figure 10.27. The distribution of sensitivity from the **Jacobian Matrix** reveals that **K_{xx}** values from pilot points placed on the southern border (around observation points **GWM13**, **GWM6** and **GWM4**) highly influence the values for the observation point **HEA-9**.

The negative sensitivity values reflect the tendency of a decreasing **Hydraulic head** in **HEA-9** with increasing **K_{xx}** at these pilot points.

Save the changes to the FEFLOW file **fepest5.fem**.

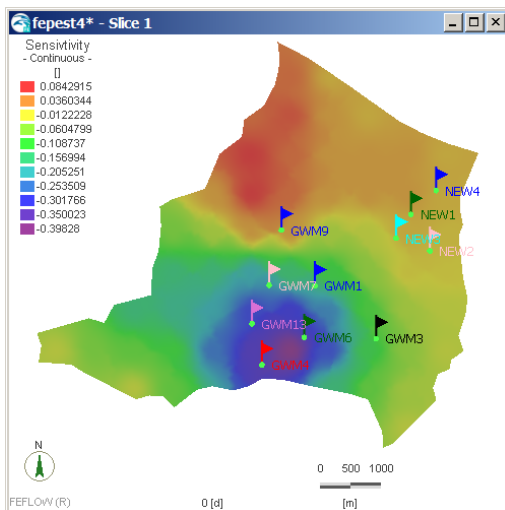


Figure 10.27 Distribution of sensitivity for observation point GWM9.

10.12.4 Additional Exercises

Run the same FePEST problem under these two cases. Use as a starting point the file **fepest.fps**.



- **Case 1:** Decrease significantly the value of the objective function limits for the Tikhonov regularization.
- **Case 2:** Deactivate the Tikhonov option in the regularization.

In the first case, you will notice the problem of over-fitting. There will be a perfect match between the observations and the FEFLOW results, but the conductivity calibrated is far from being realistic (i.e. according to the imposed prior knowledge).

In the second case, there is no prior information supported in the calibration. PEST will find a solution, which perfectly fits the observations, however this is again not realistic. Here you end up with the problem of non-uniqueness in the system, i.e. several solutions provide the same answer.





INDEX



E	
Export FePEST Results	94
F	
FePEST File Configuration	79
I	
IFM Implemented	83
Interpolation from Pilot Points	83
J	
Jacobian Matrix	99
JROW2VEC	99
K	
Kriging Configuration	86
Kriging Interpolation	84
N	
NOPTMAX	81
O	
Objective Function	93
Observation Definition	87
Observation Weights	87
P	
Parallelization	89
PEST Parameter Types	86
PEST Termination Criteria	81
Pilot Point Generation	84
R	
Regularization Setup	90
Residuals	95
Running PEST	92
S	
Sensitivity	99, 100
T	
Tikhonov Regularization	91
V	
Variograms	85







**Numerics**

3D Discretization 56

A

Advancing Front 54

Age 175

Anisotropy 75

Auxiliary Data 111

B

Borehole Heat Exchanger 108

Boundary Conditions 105

Boundary Constraint Conditions . . 109

Boundary-Condition Deactivation . 114

Budget Analysis 152

Budget Groups 154

C

Capture Zone Delineation 181

Clipping 94

Command-Line Mode 13

Constraint Conditions 109

Content Analysis 155

Convergence 144

Coordinate Systems 33

D

Database 30

Dimension 68

Discrete Features 112

Discretization 53

Documentation 14

E

Element Deactivation 56

Error Tolerance 70

ESRI Geodatabase 30

Exit Probability 176

Export 37, 157, 176

Expression Editor 118

F

FEFLOW Book 14

FE-LM2 14

FePEST 13

FEPLOT 13



Free Surface 71

G

Geodatabases 30

Georeferencing 34

GridBuilder 54

H

Head Limits 74

I

Inactive Elements 56

Inspection 95

Interactive 1D Interpolation125

Interface Manager IFM185

Internal Flows155

Interpolation117

L

Lifetime Expectancy175

Lookup Tables119

M

Map Data 36

Maps 29

Material Properties110

Mean Age175

Mean Lifetime Expectancy175

Measure Tool 56

Mesh Editing 55

Mesh Generation 53

Mesh Import 55

Mesh-Property Check 55

Modulation Function110

Multilayer Well108

N

Notation 15

O

Observation Points151

Oracle 30

P

Parameter Assignment103

Pathlines155



Péclet Number	111
Phreatic	73
Planes	94
PostGIS	30
Process Variables	104
Programming Interface	185
Projection	68

Q

Quick Import	118
------------------------	-----

R

Random-Walk Particle-Tracking	157
Raster Maps	29
Regionalization	117

S

Scatter Plot	151
Scene Library	95
Selections	83
Stereoscopic Visualization	96
Subdomain-Boundary Budget	153
Supermesh	43
Supermesh Export	45

T

Time Series	113
Time Stepping	69
Time-Series Gap	114
Time-Varying Material Properties	121
Transport Mapping	54
Travel Time	179
Triangle	54

U

Units	125
Use Parameter Expression	125
User Data	111

V

Vector Maps	29
View Windows	25, 93
Viewer Mode	13
Visualization	93



W

WGEO 13

**Numerics**

3D Discretization 56

A

Advancing Front 54

Age 175

Anisotropy 75

Auxiliary Data 111

B

Borehole Heat Exchanger 108

Boundary Conditions 105

Boundary Constraint Conditions . . 109

Boundary-Condition Deactivation . 114

Budget Analysis 152

Budget Groups 154

C

Capture Zone Delineation 181

Clipping 94

Command-Line Mode 13

Constraint Conditions 109

Content Analysis 155

Convergence 144

Coordinate Systems 33

D

Database 30

Dimension 68

Discrete Features 112

Discretization 53

Documentation 14

E

Element Deactivation 56

Error Tolerance 70

ESRI Geodatabase 30

Exit Probability 176

Export 37, 157, 176

Expression Editor 118

F

FEFLOW Book 14

FE-LM2 14

FePEST 13

FEPLOT 13



Free Surface 71

G

Geodatabases 30

Georeferencing 34

GridBuilder 54

H

Head Limits 74

I

Inactive Elements 56

Inspection 95

Interactive 1D Interpolation125

Interface Manager IFM185

Internal Flows155

Interpolation117

L

Lifetime Expectancy175

Lookup Tables119

M

Map Data 36

Maps 29

Material Properties110

Mean Age175

Mean Lifetime Expectancy175

Measure Tool 56

Mesh Editing 55

Mesh Generation 53

Mesh Import 55

Mesh-Property Check 55

Modulation Function110

Multilayer Well108

N

Notation 15

O

Observation Points151

Oracle 30

P

Parameter Assignment103

Pathlines155



Péclet Number	111
Phreatic	73
Planes	94
PostGIS	30
Process Variables	104
Programming Interface	185
Projection	68

Q

Quick Import	118
------------------------	-----

R

Random-Walk Particle-Tracking	157
Raster Maps	29
Regionalization	117

S

Scatter Plot	151
Scene Library	95
Selections	83
Stereoscopic Visualization	96
Subdomain-Boundary Budget	153
Supermesh	43
Supermesh Export	45

T

Time Series	113
Time Stepping	69
Time-Series Gap	114
Time-Varying Material Properties	121
Transport Mapping	54
Travel Time	179
Triangle	54

U

Units	125
Use Parameter Expression	125
User Data	111

V

Vector Maps	29
View Windows	25, 93
Viewer Mode	13
Visualization	93



W

WGEO 13

**Numerics**

3D Discretization 56

A

Advancing Front 54

Age 175

Anisotropy 75

Auxiliary Data 111

B

Borehole Heat Exchanger 108

Boundary Conditions 105

Boundary Constraint Conditions . . 109

Boundary-Condition Deactivation . 114

Budget Analysis 152

Budget Groups 154

C

Capture Zone Delineation 181

Clipping 94

Command-Line Mode 13

Constraint Conditions 109

Content Analysis 155

Convergence 144

Coordinate Systems 33

D

Database 30

Dimension 68

Discrete Features 112

Discretization 53

Documentation 14

E

Element Deactivation 56

Error Tolerance 70

ESRI Geodatabase 30

Exit Probability 176

Export 37, 157, 176

Expression Editor 118

F

FEFLOW Book 14

FE-LM2 14

FePEST 13

FEPLOT 13



Free Surface 71

G

Geodatabases 30

Georeferencing 34

GridBuilder 54

H

Head Limits 74

I

Inactive Elements 56

Inspection 95

Interactive 1D Interpolation125

Interface Manager IFM185

Internal Flows155

Interpolation117

L

Lifetime Expectancy175

Lookup Tables119

M

Map Data 36

Maps 29

Material Properties110

Mean Age175

Mean Lifetime Expectancy175

Measure Tool 56

Mesh Editing 55

Mesh Generation 53

Mesh Import 55

Mesh-Property Check 55

Modulation Function110

Multilayer Well108

N

Notation 15

O

Observation Points151

Oracle 30

P

Parameter Assignment103

Pathlines155



Péclet Number	111
Phreatic	73
Planes	94
PostGIS	30
Process Variables	104
Programming Interface	185
Projection	68

Q

Quick Import	118
------------------------	-----

R

Random-Walk Particle-Tracking	157
Raster Maps	29
Regionalization	117

S

Scatter Plot	151
Scene Library	95
Selections	83
Stereoscopic Visualization	96
Subdomain-Boundary Budget	153
Supermesh	43
Supermesh Export	45

T

Time Series	113
Time Stepping	69
Time-Series Gap	114
Time-Varying Material Properties	121
Transport Mapping	54
Travel Time	179
Triangle	54

U

Units	125
Use Parameter Expression	125
User Data	111

V

Vector Maps	29
View Windows	25, 93
Viewer Mode	13
Visualization	93



W

WGEO 13

**Numerics**

3D Discretization 40

A

Advancing Front 38

Age 127

Animation 123

Anisotropy 53

Auxiliary Data 81

B

Borehole Heat Exchanger 79

Boundary Conditions 76

Boundary Constraint Conditions 79

Boundary-Condition Deactivation 83

Budget Analysis 112

Budget Groups 113

C

Capture Zone Delineation 131

Clipping 68

Combined Regularization 183

Command-Line Mode 9

Constraint Conditions 79

Content Analysis 114

Convergence 106

Coordinate Systems 24

D

Database 22

Dimension 48

Discrete Features 81

Discretization 37

Documentation 10

E

Element Deactivation 39

Error Tolerance 49

ESRI Geodatabase 22

Exit Probability 128

Export 27, 115, 128

Export FePEST Results 185

Expression Editor 86

F

FEFLOW Book 10



FE-LM2 10
FePEST 9
FePEST File Configuration 173
FEPLOT 9
Finite-Element Mesh 37
Free Surface 50

G

Geodatabases 22
Georeferencing 25
GridBuilder 38
Groundwater Age 127

H

Head Limits 52

I

IFM Implemented 175
Inactive Elements 39
Inspection 69
Interactive 1D Interpolation 91
Interface Manager IFM 193
Internal Flows 114
Interpolation 85
Interpolation from Pilot Points 175

J

Jacobian Matrix 190
JROW2VEC 190

K

Kriging Configuration 177
Kriging Interpolation 176

L

Lifetime Expectancy 128
Lookup Tables 87

M

Map Data 26
Maps 21
Material Properties 80
Mean Age 127
Mean Lifetime Expectancy 128
Measure Tool 39
Mesh Editing 39



Mesh Generation 37
Mesh Import 38
Mesh-Property Check 39
Modulation Function 80
Multilayer Well 78

N

NOPTMAX 174
Notation 11

O

Objective Function 184
Observation Definition 179
Observation Points 111
Observation Weights 179
Oracle 22

P

Parallelization 180
Parameter Assignment 75
Pathlines 114
Péclet Number 81
PEST Parameter Types 178
PEST Termination Criteria 174
Phreatic 51
Pilot Point Generation 176
Planes 68
PostGIS 22
Problem Settings 47
Process Variables 76
Programming Interface 193
Projection 48

Q

Quick Import 85

R

Random-Walk Particle-Tracking 115
Raster Maps 21
Regionalization 85
Regularization Setup 181
Residuals 189
Results Evaluation 111, 127
Running PEST 183

S

Scatter Plot 111



- Scene Library 69
- Selections 59
- Sensitivity 190, 191
- Simulation 105, 135
- Singular Value Decomposition 183
- Stereoscopic Visualization 69
- Subdomain-Boundary Budget 113
- Supermesh 31
- Supermesh Export 33

T

- Tikhonov Regularization 182
- Time Series 82
- Time Stepping 49
- Time-Series Gap 83
- Time-Varying Material Properties 88
- Transport Mapping 38
- Travel Time 130
- Triangle 38

U

- Units 91
- Use Parameter Expression 91
- User Data 81
- User Interface 17

V

- Variograms 177
- Vector Maps 21
- Video 123
- View Windows 18, 67
- Viewer Mode 9
- Visualization 67

W

- WGEO 9

