GeoStudio/PEST – Multistep Outflow

GEOSLOPE International Ltd. | www.geoslope.com

1200, 700 - 6th Ave SW, Calgary, AB, Canada T2P 0T8
Main: +1 403 269 2002 | Fax: +1 888 463 2239

Introduction

Inverse modeling, often referred to as parameter estimation or parameter identification, is one of the most important and well-studied subjects in science and engineering. In contrast to conventional modeling, inverse modeling starts with the results and then calculates the causal factors that produced them. To do so, the problem is generally formulated as an optimization problem in which an objective function is used to find the optimal set of model parameters among all feasible sets. Although traditionally found by trial-and-error, the optimal set of model parameters is now generally found using efficient and objective automatic estimation procedures. Countless books and papers have been devoted to these procedures, and subroutines are available in a number of well-known mathematical libraries. In order to take advantage of these subroutines, significant changes must be made to the source code so that it meets the interface requirements and/or programming language of the subroutine. Model-independent programs, on the other hand, interact with a model through its own input and output files. In so doing, the estimation procedures can be used without making any changes to the model. The industry standard software packages for inverse modeling in geo-environmental engineering are PEST/PEST++ and UCODE. These packages, or toolboxes, also contain numerous tools to assist the estimation process. All of these tools are started from the command line prompt and are configured through answers to a series of command line prompt questions and/or configuration files. This document provides the command line prompts, sample files, and results of a typical parameter estimation problem using GeoStudio and PEST.

Background

The objective of inverse modeling is to find the optimal (or best) set of parameters that provides the minimum value of the objective function. To do so, the model-independent programs start by reading the observations, as well as the parameters, the process model command line, the template files, and the instruction files. Once this is completed, the problem is regularized and optimized. During optimization, algorithmically-generated parameters are transferred to the process model, which solves the analysis. The model results are then extracted from the output files, and the objective function is calculated. These steps are repeated until the end of the optimization process.

Objective Function

The objective function expresses the difference between observed and computed model responses. In most common approaches, the objective function is defined as the sum of weighted least-squares:

$$\Phi(\vec{\beta}) = \sum_{j=1}^{M}\sum_{i=1}^{N_{\vartheta,j}} w_{i,j}\left[\vartheta_{i,j} - \hat{\vartheta}_{i,j}(\vec{\beta})\right]^2 + \sum_{k=1}^{N_\alpha} w_k\left[\alpha_k - \hat{\alpha}_k(\vec{\beta})\right]^2 \qquad \text{Equation 1}$$

where $\vec{\beta} = (\beta_1,\ldots,\beta_n)$ is the parameter vector, N is the number of parameters, M is the number of observation types, $N_{\vartheta,j}$ is the number of observations of *j*th type, N_α is the number of prior information, $w_{i,j}$ is the weighting factor for the *i*th observation of *j*th type, w_k is the weighting factor for the *k*th prior information, $\vartheta_{i,j}$ is the *i*th observation of *j*th type, $\hat{\vartheta}_{i,j}$ is the *i*th estimation of *j*th type, α_k is the *k*th prior information, and $\hat{\alpha}_k$ is the *k*th prior information estimation. The first and second terms on the right-hand side of this equation are often referred to as the measurement and regularization components of the objective function. The equation can also be stated as follows:

$$\Phi(\vec{\beta}) = \sum_{j=1}^{M}\sum_{i=1}^{N_{\vartheta,j}} w_{i,j}\Delta\vartheta_{i,j}^{\ 2} + \sum_{k=1}^{N_\alpha} w_k\Delta\alpha_k^{\ 2} \qquad \text{Equation 2}$$

where $\Delta\vartheta_{i,j} = \vartheta_{i,j} - \hat{\vartheta}_{i,j}(\vec{\beta})$ and $\Delta\alpha_k = \alpha_k - \hat{\alpha}_k(\vec{\beta})$ are called residuals. The objective function is most appropriate when the residuals are uncorrelated and heteroscedastic (do not have the same variance). In order to provide independent residuals, the weighting factors should ideally be set equal to the inverse of the error variance-covariance matrix (Draper and Smith, 1998). In the case of uncorrelated residuals, the weight matrix is diagonal and each non-zero element of the weight matrix equals one over the variance of measurement error, $w_{i,j} = 1/\sigma_{i,j}^{\ 2}$.

Optimization Algorithms

The core components of the different software packages are the Gauss-Marquardt-Levenberg and Gauss-Newton local optimization algorithms, which are used to minimize the objective function. These algorithms start from a candidate solution and iteratively move to a neighboring solution using the Jacobian matrix, which consists of the partial derivatives of the residuals with respect to each parameter. These partial derivatives are generally approximated by finite differences in which each parameter is changed incrementally and the model is run to determine the residuals. The computational cost of these derivatives is generally much greater than that of the iterative process. In its search for the minimum value of the objective function, the Gauss-Levenberg-Marquardt algorithm interpolates between the Gauss-Newton algorithm and the method of gradient descent in which steps are taken proportionally to the steepest gradient of the objective function at the current point. In contrast to

Newton's method, the Gauss-Newton algorithm does not require the computation of second derivatives. The interpolation between the Gauss-Newton algorithm and the gradient descent method is controlled by the Marquardt lambda, which is adjusted at each iteration. Larger values of lambda bring the algorithm closer to the method of gradient descent, and should be used to accommodate for the non-ellipticity of the objective function early in the parameter estimation process. Smaller values of lambda, which bring the algorithm closer to the Gauss-Newton algorithm, are generally used later in the optimization process in order to avoid a phenomenon known as hemstitching. It must be emphasized that the minimum value of the objective function provided by this type of algorithm is not necessarily the global minimum. It is therefore possible that different user-supplied initial parameter values result in different optimal parameters.

In highly nonlinear problems, the objective function may be topographically complex, and contain multiple local minima. This makes it far more difficult to find the global minimum. To circumvent this difficulty, PEST provides implementations of two global optimization algorithms. The first is the shuffled complex evolution algorithm from the University of Arizona (SCEA-UA) and the second is the covariance matrix adaptation evolutionary strategy algorithm (CMA-ES). These algorithms do not require computation of the derivatives of the model outputs with respect to the adjustable parameters, and thus operate successfully in the presence of numerical noise and discontinuities. The shuffled complex evolution algorithm was developed for calibration of surface water models in which local optima are common occurrence (Duan et al., 1992, 1993, 1994). The covariance matrix adaptation evolutionary strategy is an evolutionary algorithm for difficult nonlinear optimization problems. The algorithm uses a rank-based strategy in which the best of the offspring form the next parent generation. It also generates new population members by sampling a probability distribution that is constructed during the optimization process. The algorithm also goes to considerable lengths in its handling of parameter bounds. Parameter bounds are never violated, and the tendency to violate bounds is mitigated by adding a penalty to the objective function as this occurs (Hansen and Ostermeier, 2001; Hansen et al., 2003).

Parameter Uniqueness

In most geo-environmental problems, the objective function contains multiple local minima or a global minimum that occurs over a range of parameters. In these cases, the objective function is said to be nonconvex, and the solution and problem are referred to as nonunique and ill-posed, respectively. The convexity of the objective function can be enhanced by including prior information or by simplifying the parameter vector. The process of enhancing the objective function is known as regularization, and can be done manually (prior to the estimation process) or automatically (during the estimation process). Manual regularization is generally accomplished by fixing insensitive parameters to their prior estimates and/or estimating parameter combinations rather than individual parameters. In a spatial model, the parameter vector can also be simplified by endowing large areas of the domain with the same set of material properties (or parameters). Manual regularization using prior information is strongly discouraged given that it is virtually impossible for the user to define the weighting factor, which is related to the error (or uncertainty) associated with his/her expert knowledge (Doherty, 2015).

Automatic, or mathematical, regularization can be separated into two broad approaches: subspace methods and Tikhonov. Subspace methods work through subdividing parameters into one subspace comprised of individual or combinations of estimable parameters, and another subspace comprised of non-estimable parameters. In contrast to manual regularization, the inversion process itself determines the estimable parameters by inspecting the would-be eigenvalues, or singular values, of an a priori post-inversion parameter error covariance matrix. If the ratio of a specific eigenvalue to the highest eigenvalue is particularly large, the eigenvector defines a direction of relative sensitivity in parameter space. Singular value decomposition is invoked as the eigenvalue ratio reaches a user-specified threshold. In so doing, the estimable parameters with smaller eigenvalue ratios are omitted from the inversion process thereby enhancing the convexity of the objective function. In real world problems, where system behavior and models are nonlinear, the a priori post-inversion parameter error covariance matrix is based on the Jacobian matrix. As stated earlier, computation of the Jacobian matrix comes at a great computational cost when the derivatives are calculated using finite differences. Numerically, the process becomes very slow once the parameters start to number more than about two thousand. It is then preferable to use an iterative solution algorithm such as LSQR. In an SVD-assisted inversion process, on the other hand, the global Jacobian matrix is computed only once. The parameter space is then decomposed into estimable and non-estimable parameter combinations, or super parameters, which often consist of linear combinations of the model parameters. From that point on, the inversion process is reformulated to estimate the super parameters. This significantly reduces the computational burden given that the Jacobian matrix is now formulated in terms of a few super parameters and not the hundreds, or even thousands, of model parameters.

Unlike subspace methods that reduce the number of parameters, Tikhonov regularization seeks to enhance the objective function by supplementing it with expert knowledge of all parameters. This expert knowledge is most often, but not exclusively, expressed as prior information for which the weight is inversely proportional to the prior parameter covariance matrix. Its influence on the inversion process is therefore proportional to the strength of the expert knowledge. In order to avoid over-fitting, Tikhonov regularization is implemented as a constrained minimization problem in which the regularization component of the objective function is minimized while ensuring that the measurement component of the objective function is equal to its user-supplied target. If the user-supplied target is too low, the measurement component of the objective function is lowered as much as possible. In so doing, the inversion process rejects the calibrated models with unrealistic parameter values. Both the subspace methods (singular value decomposition and LSQR) and Tikhonov have advantages and disadvantages. It is generally best practice to use both approaches together when using local optimization algorithms. Unfortunately, it is difficult, if not impossible, to use these sophisticated regularization schemes with global optimization algorithms.

Sample Case Study

Groundwater flow models hinge on an appropriate description of the unsaturated hydraulic properties (volumetric water content and hydraulic conductivity functions) of the materials. Although numerous laboratory and field methods can be used to determine these nonlinear properties, most of the methods require that the experiments reach steady-state conditions. In contrast to these time-consuming methods, inverse modelling can be used to estimate the properties using transient groundwater flow experiments with various boundary conditions. These experiments are carried out under controlled conditions with measurement of various flow variables. An algorithmically-controlled process model is then iteratively solved to find the best parameters of the functional representation of the hydraulic properties. Additional information on determining the hydraulic properties by inverse modeling can be found in Hopmans et al. (2002).

Materials and Methods

Lebeau and Konrad (2015) conducted a multistep outflow experiment with standard 20/30 silica sand manufactured by the Unimin Corporation. To achieve saturated conditions, the sample was prepared by wet pluviation with de-aired and demineralized water. The sample was then compacted using a vibratory table, which resulted in a porosity of 0.348. Once these steps were completed, the sample was connected to the experimental apparatus shown in Figure 1, and suction was increased to a value exceeding the air-entry value of the sample. This resulted in a continuous gaseous phase, which reduced the problems associated with the occurrence of non-uniform flow at the onset of the multistep experiment. The sample was subsequently subjected to five (5) consecutive computer-controlled steps of increasing suction. The applied values of suction were 0.91, 1.12, 1.31, 1.51, and 1.71 kPa. Cumulative outflow and internal measurements of suction were sampled simultaneously at 10 second intervals. As is generally recommended, approximately 100 validated space-time flow variables of each measurement type were selected exponentially for inverse modelling purposes.

Process Model – GeoStudio

The gaseous phase is herein considered continuous and groundwater flow is assumed to follow Richards' equation, which can be written as follows:

$$\rho_w g\left(\frac{V_w}{V_o}\beta_w + \beta_s\right)\frac{\partial h}{\partial t} + \rho_w g m_w \frac{\partial h}{\partial t} = \frac{\partial}{\partial y}\left(k_w \frac{\partial h}{\partial y}\right)$$

Equation 3

where ρ_w is the fluid density, g is the gravitational acceleration, V_w is the volume of fluid, V_o is the control volume, β_w is the fluid compressibility at constant temperature, β_s is the soil-structure compressibility, h is the total head, t is the time, m_w is the coefficient of water volume change, y is a coordinate, and k_w is the hydraulic conductivity. For the sake of simplicity, the fluid volume to control volume ratio is taken equal to the volumetric water content. It must also be noted that the soil is assumed incompressible under unsaturated conditions. The hydraulic properties are herein represented by the following functional relationships (van Genuchten, 1980):

$$\Theta = \frac{\theta_w - \theta_r}{\theta_s - \theta_r} = \left[1 + (\alpha\psi)^n\right]^{-m}$$

Equation 4

and

$$k_w = k_s k_r = k_s \frac{\left[1 - (\alpha\psi)^{mn}\left[1 + (\alpha\psi)^{n}\right]^{-m}\right]^2}{\left[1 + (\alpha\psi)^{n}\right]^{ml}}$$ Equation 5

where Θ is the normalized volumetric water content, θ_w is the volumetric water content, θ_s is the saturated volumetric water content, θ_r is the residual volumetric water content, ψ is the suction, $\alpha = 1/a, n, m = 1 - 1/n$ are model parameters, and l is a lumped parameter that accounts for both pore tortuosity and connectivity. By default, most functional representations are updated in GeoStudio's user interface, and cannot be altered externally. This difficulty is circumvented by using the PAR2PAR utility distributed with PEST. Another alternative would be to use Add-In functions, such as those provided on the GEOSLOPE website. It must also be noted that the GeoStudio file must be saved in uncompressed format (*.xml*) to avoid file corruption.

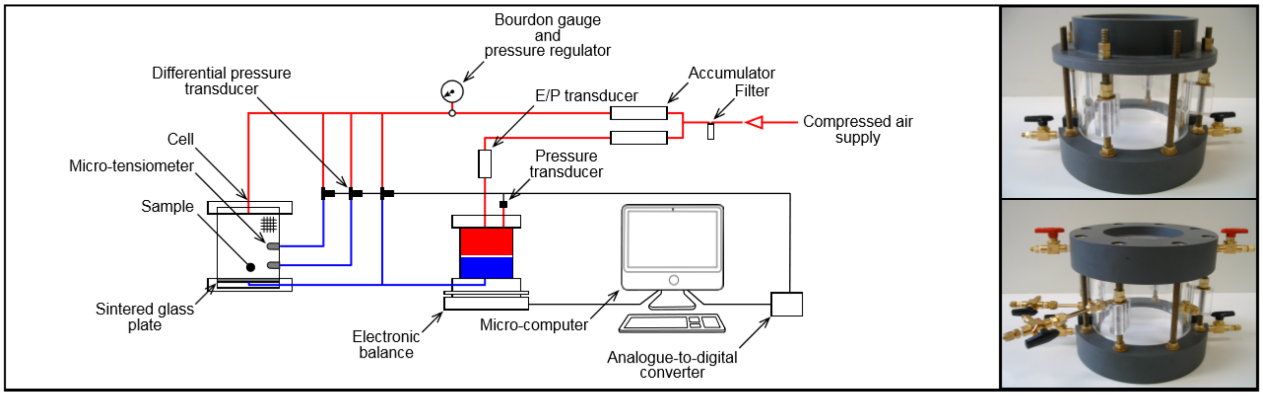


Figure 1. Diagram and photographs of the outflow experiment apparatus (Lebeau and Konrad, 2015).

Inverse Model – PEST

Control File

The PEST control file contains the parameters, the observations, the model command line, as well as the name and location of the template and instruction files. Table 1 provides a general overview of the control file used in this analysis. As is often the case, the saturated volumetric water content was set equal to the independently determined value of porosity, and the parameter vector only contains parameters θ_r, a, n, k_s, and l. For the sake of tractability, the observations were separated into two groups in which the weighting factor was set equal to one over the variance of measurement error. The command line was herein stored in a batch file, and the GeoCmd application was used to launch GeoStudio.

Template File

PEST and its utilities must be taught how to access the adjustable parameters in the model and/or utility input files. The template files, which are assigned *.tpl* extensions, are simply copies of the input files in which the spaces occupied by adjustable parameters are replaced by sequences of characters. The first line of a template file must contain the letters *ptf* followed by a space and a parameter delimiter. The parameter delimiter must be a character that does not appear anywhere within the template file except in its capacity as a parameter delimiter. Commonly used parameter delimiters are @ and *!*. Each parameter space must contain the name of the parameter, as well as blank spaces written between two delimiters. The blank spaces are important given that larger widths will allow parameters to be represented with greater precision. To ensure compatibility, the character encoding on the second line of the GeoStudio template file must be set equal to US-ASCII.

Instruction Files

PEST must also be taught how to read a model output file and identify the numbers to extract from that file. Unfortunately, the variability of certain output files inhibits the use of the template concept. A list of instructions must therefore be provided. These instructions consist of primary markers, line advances, secondary markers, whitespaces, tabs, and fixed observations. These instructions are stored in instruction files, which are assigned *.ins* extensions. The first line of the instruction files must contain the letters *pif* followed by a space and a marker delimiter. As in the template file, the marker delimiter must be a character that does not appear anywhere within the instruction files. The instruction files are herein rather simple given that the graphed model results, which coincide with the observations, are saved in separate files with comma-spaced values.

Table 1. PEST control file with parameter values in parenthesis.

```
pcf
* control data
RSTFLE (norestart) PESTMODE (estimation)
NPAR (5) NOBS (190) NPARGP (1) NPRIOR (0) NOBSGP (2)
NTPLFLE (1) NINSFLE (2) PRECIS (single) DPOINT (point) NUMCOM (1) JACFILE (0) MESSFILE (0)
RLAMBDA1 (10) RLAMFAC (-3) PHIRATSUF (0.3) PHIREDLAM (0.03) NUMLAM (10)
RELPARMAX (10) FACPARMAX (10) FACORIG (0.001)
PHIREDSWH (0.1)
NOPTMAX (50) PHIREDSTP (0.005) NPHISTP (4) NPHINORED (3) RELPARSTP (0.005) NRELPAR (3)
ICOV (1) ICOR (1) IEIG (1)
* parameter groups
PARGPNME (VANGENUCHTEN) INCTYP (relative) DERINC (0.01) DERINCLB (0.0) FORCEN (switch) DERINCMUL (2.0) DERMTHD (parabolic)
* parameter data
PARNME (VWCr)   PARTRANS (none) PARCHGLIM (relative) PARVAL1 (+3.0000E-002)    PARLBND (+1.0000E-012)    PARUBND (+3.5000E-002) PARGP (VANGENUCHTEN) SCALE (+1.0000E+000) OFFSET (+0.0000E+000) DERCOM (1)
PARNME (A)      PARTRANS (none) PARCHGLIM (relative) PARVAL1 (+1.0000E+000)    PARLBND (+8.0000E-001)    PARUBND (+1.0000E+001) PARGP (VANGENUCHTEN) SCALE (+1.0000E+000) OFFSET (+0.0000E+000) DERCOM (1)
PARNME (N)      PARTRANS (none) PARCHGLIM (relative) PARVAL1 (+5.0000E+000)    PARLBND (+5.0000E+000)    PARUBND (+1.5000E+001) PARGP (VANGENUCHTEN) SCALE (+1.0000E+000) OFFSET (+0.0000E+000) DERCOM (1)
PARNME (Ks)     PARTRANS (none) PARCHGLIM (relative) PARVAL1 (+5.0000E-004)    PARLBND (+5.0000E-005)    PARUBND (+1.0000E-002) PARGP (VANGENUCHTEN) SCALE (+1.0000E+000) OFFSET (+0.0000E+000) DERCOM (1)
PARNME (L)      PARTRANS (none) PARCHGLIM (relative) PARVAL1 (+5.0000E-001)    PARLBND (+2.5000E-001)    PARUBND (+1.2500E+000) PARGP (VANGENUCHTEN) SCALE (+1.0000E+000) OFFSET (+0.0000E+000) DERCOM (1)
* observation groups
OBGNME (CUMFLUX)
OBGNME (PRESHEAD)
* observation data
OBSNME (CUMFLUX1)     OBSVAL (-0.000994)    WEIGHT (1.8024E+005)    OBGNME (CUMFLUX)
OBSNME (...)          OBSVAL (...)          WEIGHT (...)            OBGNME (...)
OBSNME (CUMFLUX95)    OBSVAL (-0.011639)    WEIGHT (1.8024E+005)    OBGNME (CUMFLUX)
OBSNME (PRESHEAD1)    OBSVAL (-0.117649)    WEIGHT (1.2953E+002)    OBGNME (PRESHEAD)
OBSNME (...)          OBSVAL (...)          WEIGHT (...)            OBGNME (...)
OBSNME (PRESHEAD95)   OBSVAL (-0.188987)    WEIGHT (1.2953E+002)    OBGNME (PRESHEAD)
* model command line
COMLINE (MSO.bat)
* model input/output
TEMPFLE (MSO-PAR2PAR.tpl)    INFLE ("C:\...\par2par.inp")
INSFLE (MSO-CUMFLUX.ins)     OUTFLE ("C:\...\MSO\Experiment\graph_7_18731.csv")
INSFLE (MSO-PRESHEAD.ins)    OUTFLE ("C:\...\MSO\Experiment\graph_8_17730.csv")
```

Table 2. Shuffled complex evolution algorithm input file with parameter values in parenthesis.

NGS (5)	is the initial number of complexes.
MINGS (5)	is the minimum number of complexes.
NPG (9)	is the number of parameter sets in each complex.
NPS (5)	is the number of parameters per sub-complex.
INIFLG (n)	is a flag indicating whether or not the initial parameter set should be included in the population.
NSPL (9)	is the number of evolution steps before shuffling.
ISEED (555)	is the random number seed.
IPRINT (v)	is a flag indicating whether or not every point of the population should be printed after each shuffling loop.
PCENTO (0.01)	is the relative objective function reduction criteria.
KSTOP (5)	is the maximum successive relative objective function reductions.
MAXN (5000)	is the maximum number of iterations.

Run Commands

It is strongly recommended that the entire input dataset be checked using the PESTCHEK utility prior to execution. The integrity of the template and instruction files should also be checked using the TEMPCHEK and INSCHEK utilities, respectively. These utilities are run by typing the name of the utility at the command line followed by the name of the file. In this particular example, the commands are:

```
pestchek MSO.pst

tempchek MSO-PAR2PAR.tpl

tempchek MSO.tpl

inschek MSO-CUMFLUX.ins

inschek MSO-PRESHEAD.ins
```

In this highly nonlinear problem, inverse modeling was conducted with the shuffled complex evolution algorithm from the University of Arizona (SCEA-UA). Although the control variables and termination criteria are generally entered through answers to a series of command prompt questions, they are herein stored in an input file (refer to Table 2). The command line for launching the algorithm is:

```
sceua_p MSO.pst < SCEUA.inp
```

The inversion process can be stopped, paused, or restarted using the PSTOP, PPAUSE, and PUNPAUSE commands, respectively. The optimization progress is reported on screen, as well as in the PEST run record file named *MSO.rec*. A detailed report of the global optimization process can be found in the SCEA-UA run record file named *sceout.dat*. Note that the best parameter values are also reported in *MSO.par*.

Getting Started

The following steps (and the accompanying tutorial video) will help you get the inverse modeling process up and running on your computer.

1. Download the latest version of PEST.
2. Copy the following files over to an empty working folder.

   pestchek.exe
   tempchek.exe
   inschek.exe
   par2par.exe
   sceua_p.exe
3. Download the GeoStudio data files.
4. Copy all of the files over to the working folder.

   GeoStudio-PEST – Multistep Outflow.gsz
   MSO.bat
   MSO.pst
   MSO.tpl
   MSO-PAR2PAR.tpl
   MSO-CUMFLUX.ins
   MSO-PRESHEAD.ins
   PAR2PAR.inp
   SCEUA.inp
5. Create a subfolder called *MSO*.
6. Open *GeoStudio-PEST – Multistep Outflow.gsz* with GeoStudio, and save it in uncom-pressed format (…\MSO\MSO.xml).
7. Open *MSO.xml* with GeoStudio, solve the analysis, and visually compare the results to those shown in Figure 3.
8. Open *MSO.bat* with a text editor, and change the *par2par.exe* and *GeoCmd.exe* file paths.
9. Open *MSO.pst* with a text editor, and change the input and output file paths and graph file names (in the model input/output section at the bottom of the file).
10. Open *MSO-PAR2PAR.tpl* with a text editor, and change the input file path (in the template and model input files section at the bottom of the file).
11. Open a Command Prompt, and redirect to the working directory.
12. Run the PESTCHEK, TEMPCHEK, and INSCHEK utilities.
13. Launch the optimization process. Note that the process may take several hours to complete.

Results and Discussion

Figure 2 shows the evolution of the objective function with the number of model runs. In this case, the algorithm reaches an optimal value within 800 model runs. The parameters of this optimal (or best) solution are θ_r = 0.029, a = 1.101 kPa n = 10.190, k_s = 9.009×10^{-4} m/s, and l = 0.295. It must be emphasized that the value of saturated hydraulic conductivity has no physical meaning given that the outflow experiment was started under unsaturated conditions.

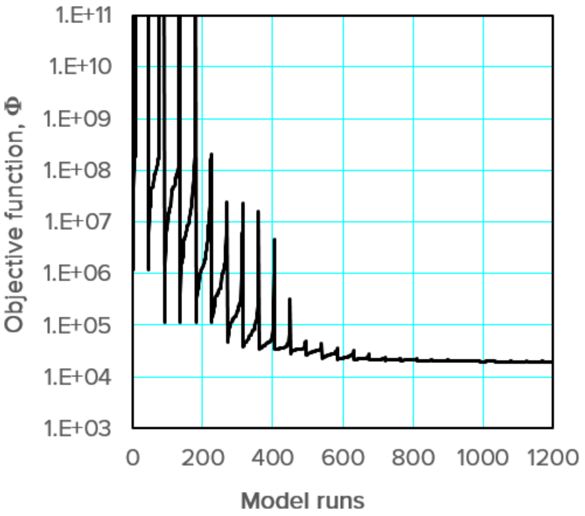
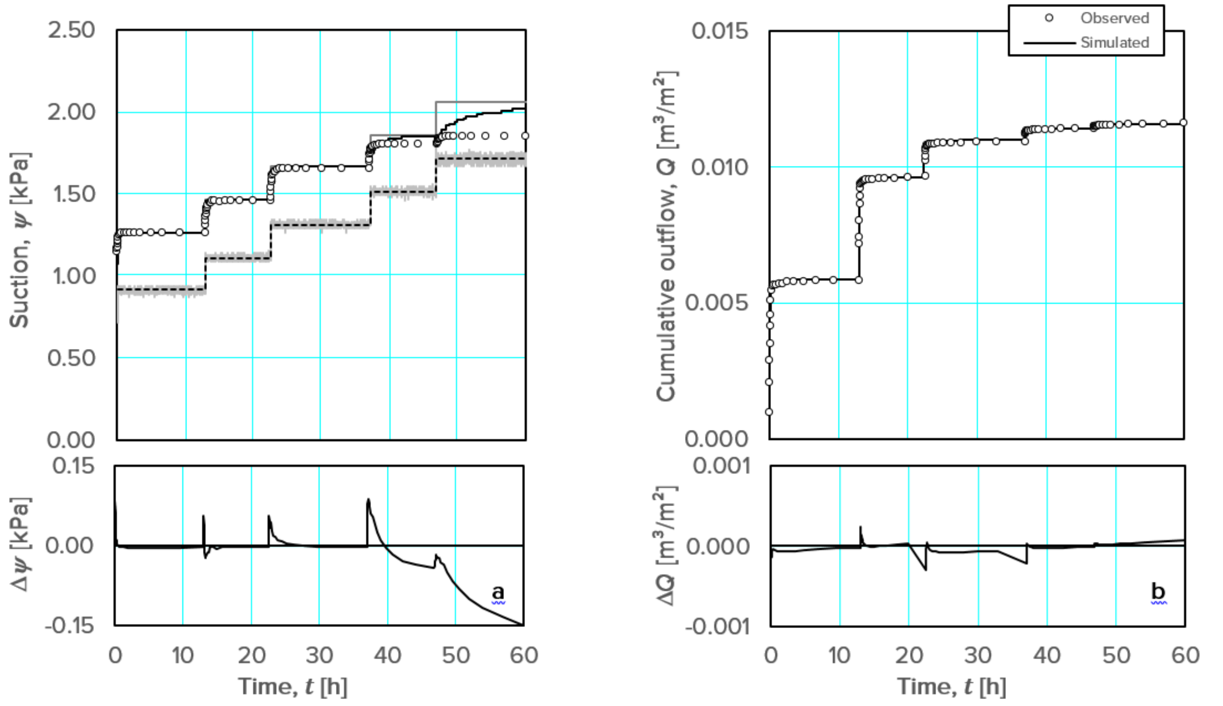


Figure 2. Convergence graph for the multistep outflow problem.

Figure 3 shows the applied and modeled lower boundary conditions, as well as the resulting cumulative outflow and suction. In general, the model is shown to be in close agreement with experimental measurements. Yet, as often occurs in coarse-grained soils, the model fails to capture the suction response during the last two steps of the outflow experiment. This is clearly reflected in the residual plots, which also show that some error occurs at the onset of the stepwise increases in suction. Despite



these discrepancies, the coefficient of determination is equal to 0.97, which suggests that the model generally reproduces the dynamics of the outflow experiment.

Figure 3. Results of the multistep outflow problem. (a) Suction. (b) Cumulative outflow.

Figure 4 compares the estimated hydraulic properties with those measured by Schroth et al. (1996). Although determined with very different experimental methods, the volumetric water content function is in very close agreement with the independent data. Note that the estimated function lies between the experimental data and the functional representation used to account for the non-uniform water content within the experimental cell under static equilibrium conditions (Schroth et al., 1996). The estimated hydraulic conductivity function is also shown to be in very close agreement with the independent experimental data, which were measured using the steady-state flux control method.

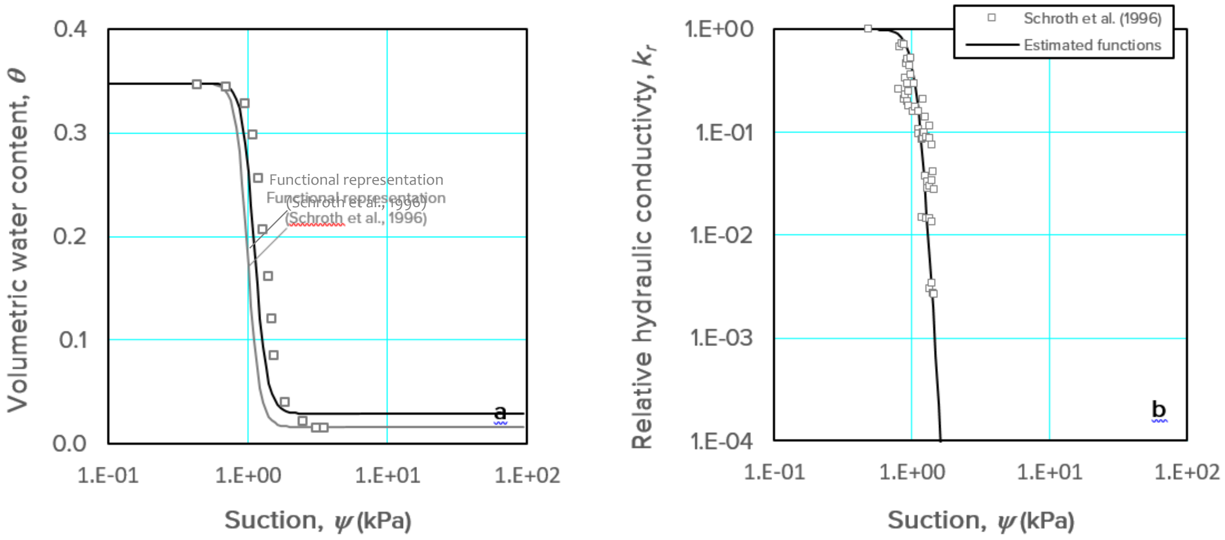


Figure 4. Hydraulic properties of the 20/30 silica sand sample. (a) Volumetric water content function. (b) Hydraulic conductivity function.

Summary and Conclusions

The objective of this example was to demonstrate the ability to solve inverse modeling problems using GeoStudio and PEST, and to provide users with typical sample files and command line prompts. The capabilities of the models were assessed through a sample inverse modeling problem in which the hydraulic properties of a silica sand sample were estimated using the results of a multistep outflow experiment. The estimated hydraulic properties were shown to be in close agreement with those independently determined using steady-state methods.

References

Doherty, J.E. 2015. Calibration and uncertainty analysis for complex environmental models. PEST: Complete theory and what it means for modelling the real world. Watermark Numerical Computing. pp. 227.

Draper, N.R., and Smith, H. 1998. Applied regression analysis. 3rd ed. John Wiley & Sons, Inc. pp. 736.

Duan, Q., Soorooshian, S. and Gupta, V.K. 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resources Research 28(4), 1015-1031.

Duan, Q., Gupta, V.K. and Sorooshian, S. 1993. A Shuffled Complex Evolution approach for effective and efficient global minimization. Journal of Optimization Theory and its Applications 76(3), 501-521.

Duan, Q., Sorooshian, S. and Gupta, V.K. 1994. Optimal use of the SCE-UA global optimization method for calibrating watershed models. Journal of Hydrology 158, 265-284.

Hansen, N. and Ostermeier, A. 2001. Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation 9(2), 159-195

Hansen, N., S.D. Muller, and Koumoutsakos, P. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation 11(1), 1-18.

Hopmans, J.W., Šimůnek, J., Romano, N., and Durner, W. 2002. Inverse methods. In Methods of soil analysis. Part 4. Physical methods. Edited by J.H. Dane and G.C. Topp. Soil Science Society of America. pp. 963-1008.

Lebeau, M., and Konrad, J.-M. 2015. A generalized global optimization application and its use in determining the hydraulic properties of unsaturated sand. In 68th Canadian Geotechnical Conference and 7th Canadian Permafrost Conference, Quebec City, Canada.

Schroth, M.H., Ahearn, S.J., Selker, J.S., Istok, J.D. 1996. Characterization of Miller-similar silica sands for laboratory hydrologic studies. Soil Science Society of America Journal 60(5), 1331-1339.

van Genuchten, M. Th. 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Science Society of America Journal 44, 892-898.