Using PEST to Calibrate a HydroGeoSphere Model

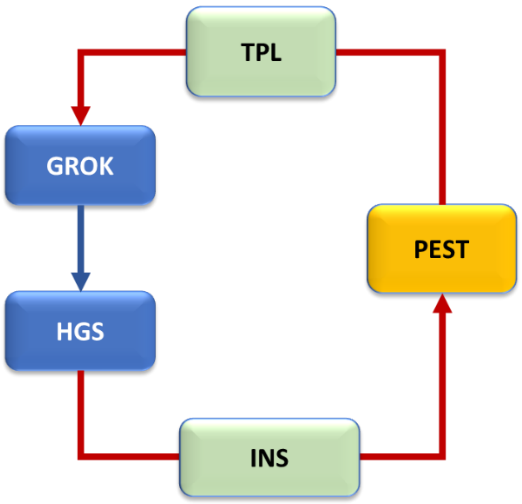


Table of Contents

1 Introduction

Thank you for your interest in this tutorial, which is designed to introduce the basic workflow for integrating your HydroGeoSphere (HGS) models with '**PEST: Model-Independent Parameter Estimation and Uncertainty Analysis**'.

Please note that this tutorial assumes the reader has a general understanding of how to build and review HydroGeoSphere models, and a basic understanding of PEST itself. In the simplest terms, PEST is a software package which automates calibration, and calibration-constrained uncertainty analysis of any numerical model. If you are unfamiliar with PEST, we highly recommend reviewing the supplementary materials listed below to help you understand the theory behind this software, and the limitations of an automated calibration process. To paraphrase John Doherty (creator of PEST) – "just because you've calibrated a model doesn't mean you're going to get the right answer".

This tutorial accompanies a set of HGS and PEST input files which you can modify to suit your own needs. However, the example problem introduced here is a relatively simple one which only introduces a workflow for implementing the most basic capabilities offered by PEST (specifically the basic parameter estimation process). This tutorial does not discuss the use of PEST for parameter uncertainty analysis, nor does it introduce more advanced parameter estimation methods such as Tikhonov regularization, singular value decomposition, or the use of pilot points. Finally, it should also be noted that many of the input instructions for the PEST control/input files are not discussed herein, and the analysis of PEST output is left to the reader. The aim of this tutorial is to review key PEST input files/instructions, and to provide a practical example on how to apply PEST to an HGS model.

Note: this tutorial was based on the May, 2021 (REVISION 2255) edition of HydroGeoSphere.

2 Supporting Materials

More information regarding advanced PEST methods, detailed information on PEST input files, and the interpretation of PEST output files are provided in the supplementary materials below. A review of these materials combined with the present tutorial should provide you with a solid foundation to apply PEST to any HGS model, and guide further study.

The philosophy of model calibration: Darcy Lecture by John Doherty, creator of PEST

- https://www.youtube.com/watch?v=Gb_bx6Ui3vA

PEST Manuals and Documentation

- https://pesthomepage.org/documentation

Getting Started with PEST: an introduction to the workings of PEST by Zhulu Lin (University of Georgia)

- https://www.ndsu.edu/pubweb/~zhulin/pdf/teaching/starting%20pest.pdf

Lessons by John Doherty about PEST and more: YouTube playlist for simple to advanced PEST lessons

- https://youtube.com/playlist?list=PL_uHV4vwQCU9AqUlkKh6gPbHCSwCKP0K0

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

- https://pubs.usgs.gov/sir/2010/5169/pdf/GWPEST_sir2010-5169.pdf

3 The HydroGeoSphere Model

You can download the sample model files accompanying this tutorial from the hyperlink below:

- **Download the Abdul_PEST Model Files**
  https://community.aquanty.com/assets/uploads/files/1621425924753-abdul_pest.zip

3.1 General Review: Model Size, Duration, Active Domains/Processes

This is an iteration on the popular Abdul verification problem. The model domain is approximately 80 m × 16 m areally and up to 4 m deep. A man-made stream channel lies approximately 1.2 m below the surrounding grassy land. The channel is initially dry prior to the application of the artificial rainfall via irrigation sprinklers. The initial water table lies around 22 cm below the streambed with the artificial recharge applied at a rate of 2 cm/hour for 50 minutes (3000 seconds). Infiltration in upland regions, discharge in lower regions and runoff, all govern the behavior of the system. The model is a simple surface/subsurface transient flow model with no additional processes (e.g. transport, evapotranspiration, freeze/thaw, etc.). The entire model duration is 100 minutes (6000 seconds).

The model has been simplified somewhat (e.g. removing model output times, reducing the number of monitoring wells, removed hydrograph nodes, etc.) compared to the typical Abdul verification problem in order to minimize the model run-time. When calibrating an HGS model with PEST the model will be run many *many* times (usually hundreds or even thousands of times). As such, stripping superfluous tasks out of your HGS model can potentially save you hours of PEST run-time.

3.2 Observation Data

To calibrate any model, PEST relies on observation data which it compares to the model output. The sum of squares of the weighted error between these observations and model-generated outputs defines the PEST objective function (ϕ). The objective function is the metric against which PEST determines whether changes to parameter values are ‘good’ or ‘bad’ (i.e. whether the calibration is improving).

One strength of PEST is that it can calibrate the model against a wide variety of different observation data types (e.g. heads, concentrations, fluxes, saturations, etc.). For this tutorial we will rely on overland flow fluxes, groundwater heads, and soil saturation levels.

Note: the observation data in this tutorial bears no relation to reality. Observation data types, values and locations used here are merely applied for the purposes of illustration.

Within the Abdul_PEST.grok file we note the presence of a critical depth boundary condition (‘*CritDepth_outlet’*) at the model ‘outlet’ (i.e. the nodes associated with the downstream reach of the channel). The total water flux within this channel represents the first type of observation data which PEST will rely on for the parameter estimation process. Since the model design includes ‘*CritDepth_outlet*’ as a boundary condition the resulting fluxes for all timesteps will be written automatically to the ‘*abdul_PESTo.water_balance.dat*’ output file.

We also note the presence of two observation points (‘*point1*’ and ‘*point2*’) which correspond to the locations where head and soil moisture/saturation observations have been taken. These represent the 2nd and 3rd types of data which will be incorporated into the parameter estimation process.

Observations of channel flux (‘CritDepth_outlet’) and groundwater heads (‘point1’) have been taken frequently throughout the duration of the model (at t = 673, 1024, 1248, 1427, 1684, 1887, 2154, 2431, 2668, 2881, 3040, 3146, 3197, 3329, 3444, 3560, 3747, 3897, 4046, 4294, 4547, and 4820). To ensure that model results are written at the times corresponding to model observations, the ‘target times’ command is applied, and a list of all observation times are included. The head and flux observation values and times will both be referenced directly in the PEST input files, based on the HGS model output from the ‘*Abdul_PESTo.water_balance.dat*’ (flux) and ‘*abdul_PESTo.observation_well_flow.point1.dat*’ (head) files. Finally, the saturation results from ‘point2’ are averaged over the first and 2nd half of the model duration using the ‘compute post simulation average’ command. As such, PEST will compare saturation observations from the output file ‘*abdul_fineo.avg_val_summary.dat*’.

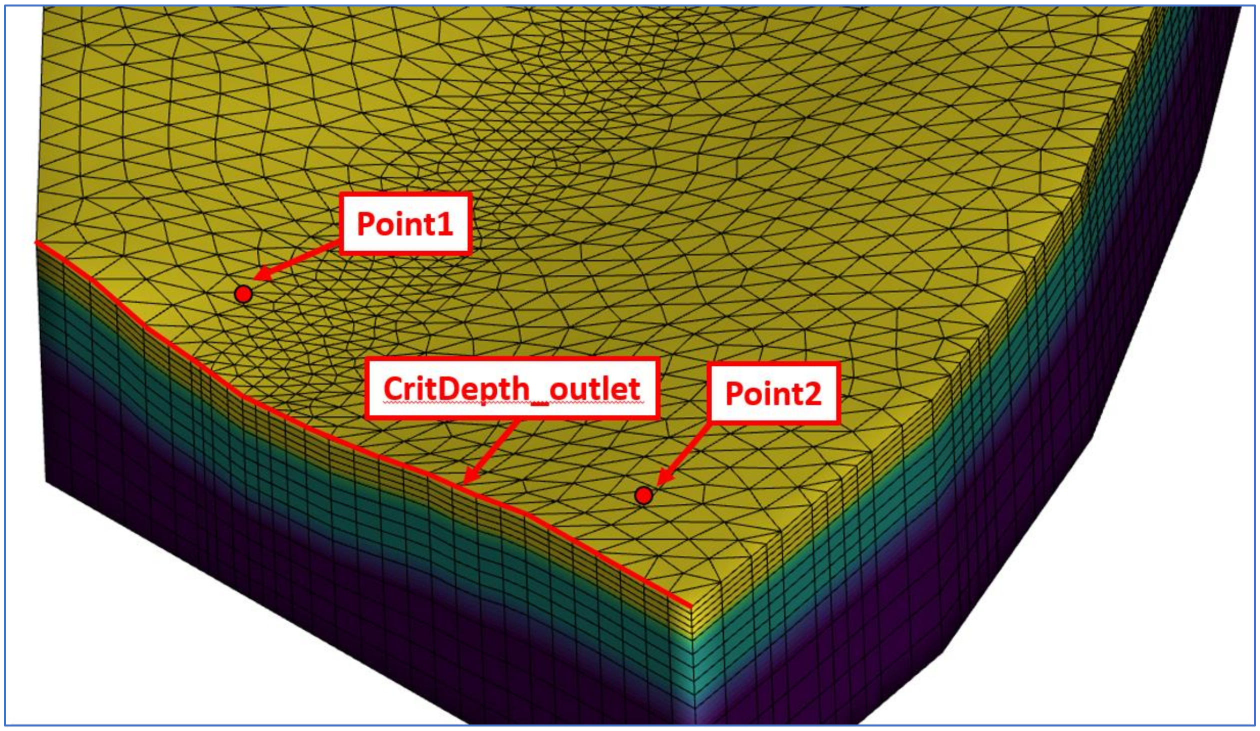


Figure 1*: *Observation Data – CritDepth_outlet (fluxes), Point1 (heads), Point2 (soil moisture)

3.3 Numerical Simulation Parameters

Please note that the numerical simulation parameters for this example are quite relaxed, favouring faster model run times over solution accuracy. In a real-world situation you will need to balance these competing goals. You can get the best of both worlds by *manually* parameterizing your model under strict numerical convergence parameters, then relaxing the HGS convergence criteria for an initial parameter estimation (PEST run). With the resulting parameter values you may consider running PEST again with stricter convergence criteria, as the updated parameter values will give PEST a better starting point to work with.

4 A Review of PEST Input Files, Processes and Output Files

The PEST parameter estimation process is controlled by four essential input files and run from the command line, much the same way that HydroGeoSphere operates. Please note that further input files may be required for advanced PEST capabilities, but for the parameter estimation application demonstrated here we require only four PEST input files, each of which is discussed in further detail later in this section:

1. **The PEST control (*.pst) file**
    - Includes all essential data controlling the PEST run
2. **Model input template (*.tpl) file(s)**
    - Allow PEST to manipulate HGS parameters
3. **Model output reading instruction (*.ins) file(s)**
    - Allow PEST to read model output and compare to observation data
4. **Model batch (*.bat) file**
    - Allows PEST to repeatedly run successive executables (e.g. grok.exe and phgs.exe)

In simple terms, the PEST parameter estimation process works like this (illustrated in ***Figure 2*** below):

1) The PEST process is initiated from the command prompt.
2) PEST reviews the PEST control (*.pst) file to read basic information about the PEST run, for example:
    a. What kind of run is being performed,
    b. How many parameters will be estimated, upper/lower limits and initial parameter values, etc.
    c. How many observation values are available, observation 'weights', etc.
    d. The name of model input template (*.tpl) file(s)
    e. The name of model output instruction (*.ins) file(s)
    f. The name of the model batch (*.bat) file
3) PEST applies initial parameter values from the *.pst file to the input template (*.tpl) file(s)
4) PEST uses the batch (*.bat) file to update HGS input files (grok.exe) and run HGS (phgs.exe)
5) PEST uses the output instruction (*.ins) file(s) to read HGS output, compares to observation data from the *.pst file and calculates the resulting objective function (ϕ) for this model run.
6) PEST updates the input template (*.tpl) file(s) with new parameter values.
7) Steps 4, 5 and 6 are repeated hundreds, or even thousands of times until there is no appreciable decrease in the objective function () value. At this point PEST will terminate and you can review output files to determine parameter estimates which resulted in the lowest possible objective function.

Note: as you may suspect, steps 6 and 7 are simplified considerably here. PEST uses an optimization algorithm to determine the best possible parameter combinations in an iterative manner using Gauss-Marquardt-Levenberg method of nonlinear parameter estimation. In other words, PEST doesn't just randomly vary parameter values and 'hope for the best'.

See the PEST User's Manual for a detailed review of the parameter estimation process.

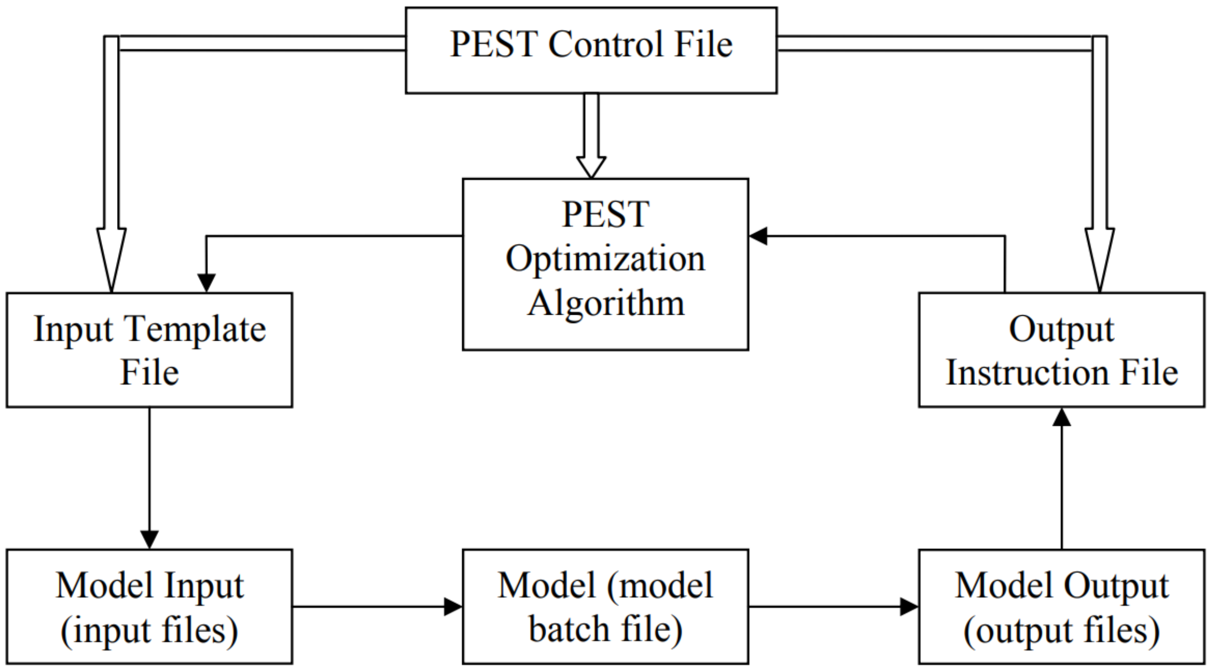


Figure 2*: *Schematic diagram of PEST optimization process (from 'Getting Started with PEST', Zhulu Lin – see supporting materials section)

4.1 The Pest Control (*.pst) File

The PEST control (*.pst) file contains all instructions that PEST will refer to while performing the parameter estimation process. All PEST control files must begin with 'pcf' in the first line and will typically contain seven sections which are denoted by a line starting with a '*' followed by the section name. For more detailed information regarding the PEST control file, please refer to Appendix A of the PEST manual (see supporting materials).

For the parameter estimation process demonstrated in this tutorial, the PEST control file contains the following seven sections (note: other PEST applications may contain various other sections requiring different information, see the PEST manual for more information):

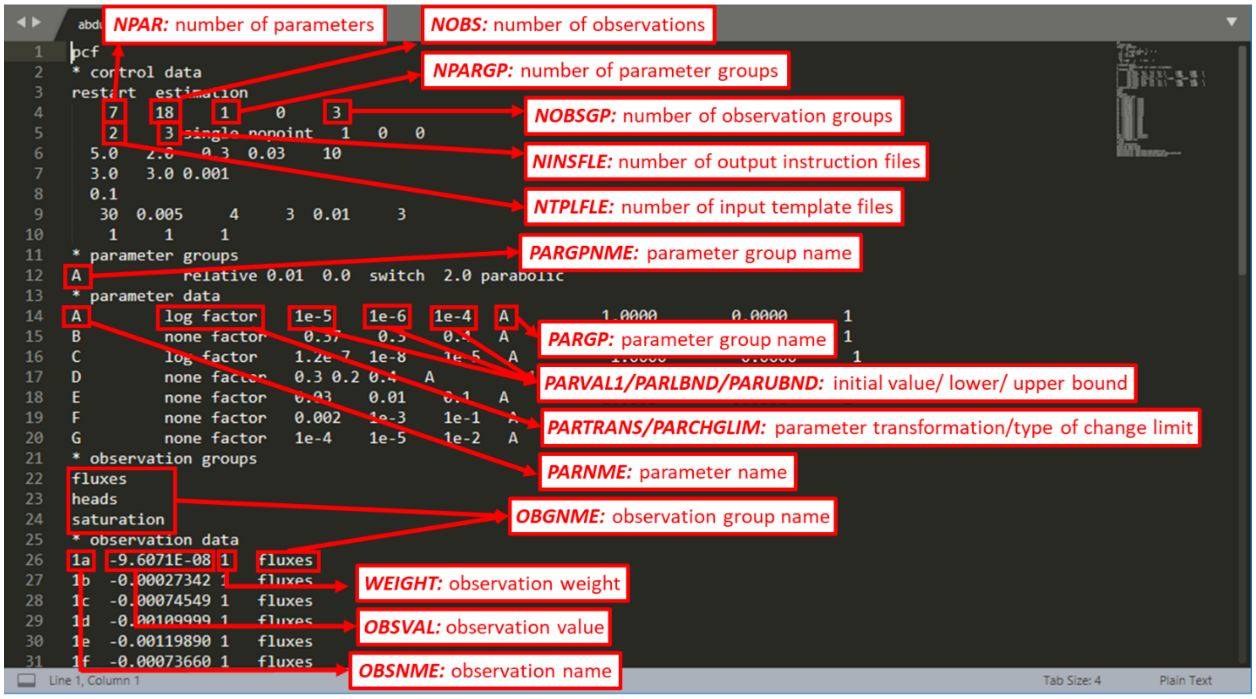
1. *** control data** – describes the type of PEST analysis (e.g. parameter estimation, regularization, predictive analysis, etc.) and includes variables for the # of parameters, # of observations, # of input template files, # of output instruction files, stopping criterion for the analysis, etc.
2. *** parameter groups** – list of parameter groups for the analysis and variables controlling the estimation of parameter groups (e.g. parameter incrementing type and size). Each parameter group requires at least one parameter, and all parameters must be assigned to one group.
3. *** parameter data** – list of parameters which PEST will use for the estimation process, and variables controlling the analysis (e.g. upper/lower bound on parameter value, initial value, parameter transformation, etc.). The parameter names listed here are used within the input

template files to identify “parameter spaces” (i.e. spots within the input file corresponding to the parameter value).

4. *** observation groups** – list of observation groups for the analysis. All individual observations must be assigned to a single observation group. This section only requires a list of observation group names, variables controlling individual observations are applied in the following section.
5. *** observation data** – a list of every individual observation value which PEST incorporates into the model objective function. This section lists the observation name, value, weight and observation group. Different weights can be applied to different types of observation data to ensure they exert a relatively similar impact on the objective function. For example, comparing head values against concentration values where the absolute value of the observations may be several orders of magnitude in difference.
6. *** model command line** – a single line referencing the batch file which allows PEST to execute grok.exe and phgs.exe in succession.
7. *** model input/output** – provides a list of model input template (*.tpl) files and model output instruction (*.ins) files associated with this parameter estimation. Each *.tpl and *.ins file should be associated with a specified model output file.

Figures 3 and ***4*** below are based on the PEST control file for this tutorial (Abdul_PEST.pst) and illustrate some of the important variables that you might need to change when applying PEST to your own HydroGeoSphere problems.

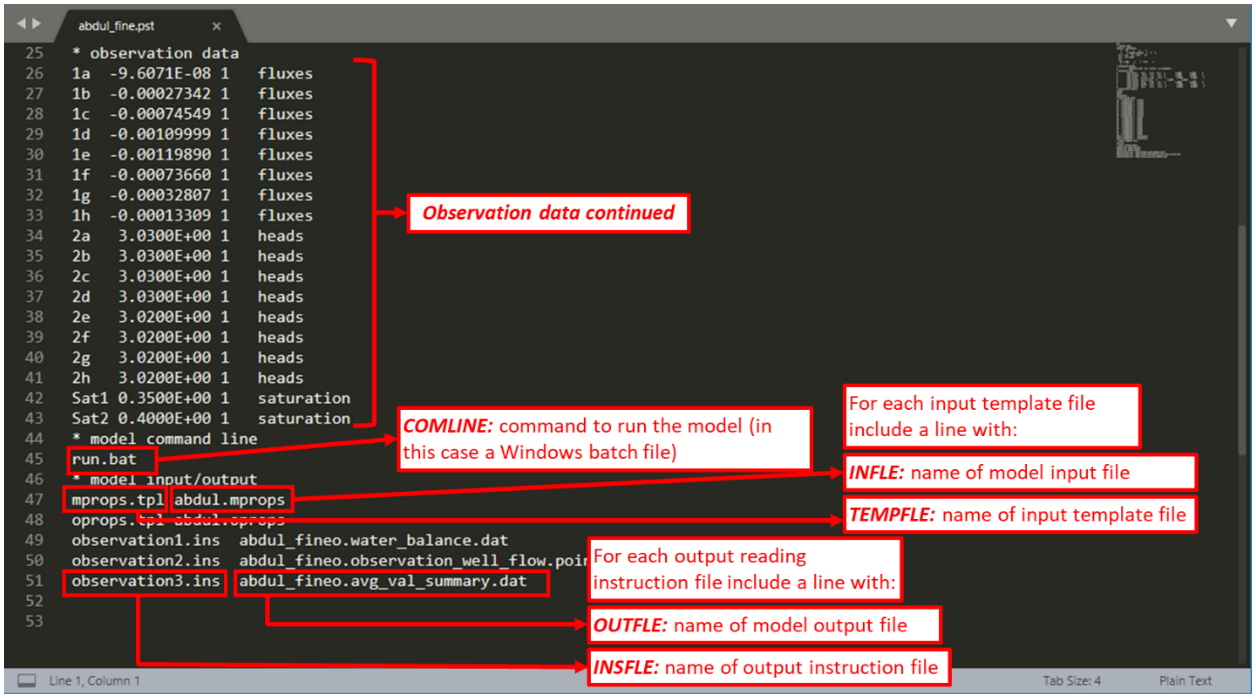
Note: bold/italicized text represent variable IDs which are referenced in the PEST manual (Appendix a).



```
pcf
* control data
restart  estimation
    7    18    1    0    3
    2     3 single nopoint   1   0   0
  5.0   2.0   0.3  0.03    10
  3.0   3.0 0.001
  0.1
   30  0.005     4     3  0.01     3
    1     1     1
* parameter groups
A            relative 0.01  0.0  switch  2.0 parabolic
* parameter data
A         log factor   1e-5    1e-6    1e-4   A          1.0000        0.0000       1
B         none factor   0.37    0.3     0.4   A                                      1
C         log factor   1.2e-7  1e-8    1e-5    A         1.0000        0.0000       1
D         none factor  0.3 0.2 0.4    A
E         none factor  0.03    0.01    0.1   A
F         none factor  0.002   1e-3    1e-1   A
G         none factor  1e-4    1e-5    1e-2   A
* observation groups
fluxes
heads
saturation
* observation data
1a  -9.6071E-08 1   fluxes
1b  -0.00027342 1   fluxes
1c  -0.00074549 1   fluxes
1d  -0.00109999 1   fluxes
1e  -0.00119890 1   fluxes
1f  -0.00073660 1   fluxes
```

abdu
NPAR: number of parameters
NOBS: number of observations
NPARGP: number of parameter groups
NOBSGP: number of observation groups
NINSFLE: number of output instruction files
NTPLFLE: number of input template files
PARGPNME: parameter group name
PARGP: parameter group name
PARVAL1/PARLBND/PARUBND: initial value/ lower/ upper bound
PARTRANS/PARCHGLIM: parameter transformation/type of change limit
PARNME: parameter name
OBGNME: observation group name
WEIGHT: observation weight
OBSVAL: observation value
OBSNME: observation name
Line 1, Column 1
Tab Size: 4
Plain Text

***Figure 3**: Description of Variables from the Abdul_PEST.pst Control File Part 1 (for full list of variables see PEST Manual Appendix A)*



```
* observation data
1a  -9.6071E-08 1   fluxes
1b  -0.00027342 1   fluxes
1c  -0.00074549 1   fluxes
1d  -0.00109999 1   fluxes
1e  -0.00119890 1   fluxes
1f  -0.00073660 1   fluxes
1g  -0.00032807 1   fluxes
1h  -0.00013309 1   fluxes
2a   3.0300E+00 1   heads
2b   3.0300E+00 1   heads
2c   3.0300E+00 1   heads
2d   3.0300E+00 1   heads
2e   3.0200E+00 1   heads
2f   3.0200E+00 1   heads
2g   3.0200E+00 1   heads
2h   3.0200E+00 1   heads
Sat1 0.3500E+00 1   saturation
Sat2 0.4000E+00 1   saturation
* model command line
run.bat
* model input/output
mprops.tpl abdul.mprops
oprops.tpl abdul.oprops
observation1.ins  abdul_fineo.water_balance.dat
observation2.ins  abdul_fineo.observation_well_flow.poi
observation3.ins  abdul_fineo.avg_val_summary.dat
```

Observation data continued

COMLINE: command to run the model (in this case a Windows batch file)

For each input template file include a line with:

INFLE: name of model input file

TEMPFLE: name of input template file

For each output reading instruction file include a line with:

OUTFLE: name of model output file

INSFLE: name of output instruction file

***Figure 4**: Description of Variables from the Abdul_PEST.pst Control File Part 2 (for full list of variables see PEST Manual Appendix A)*

4.2 Model Input Template (*.tpl) File(s)

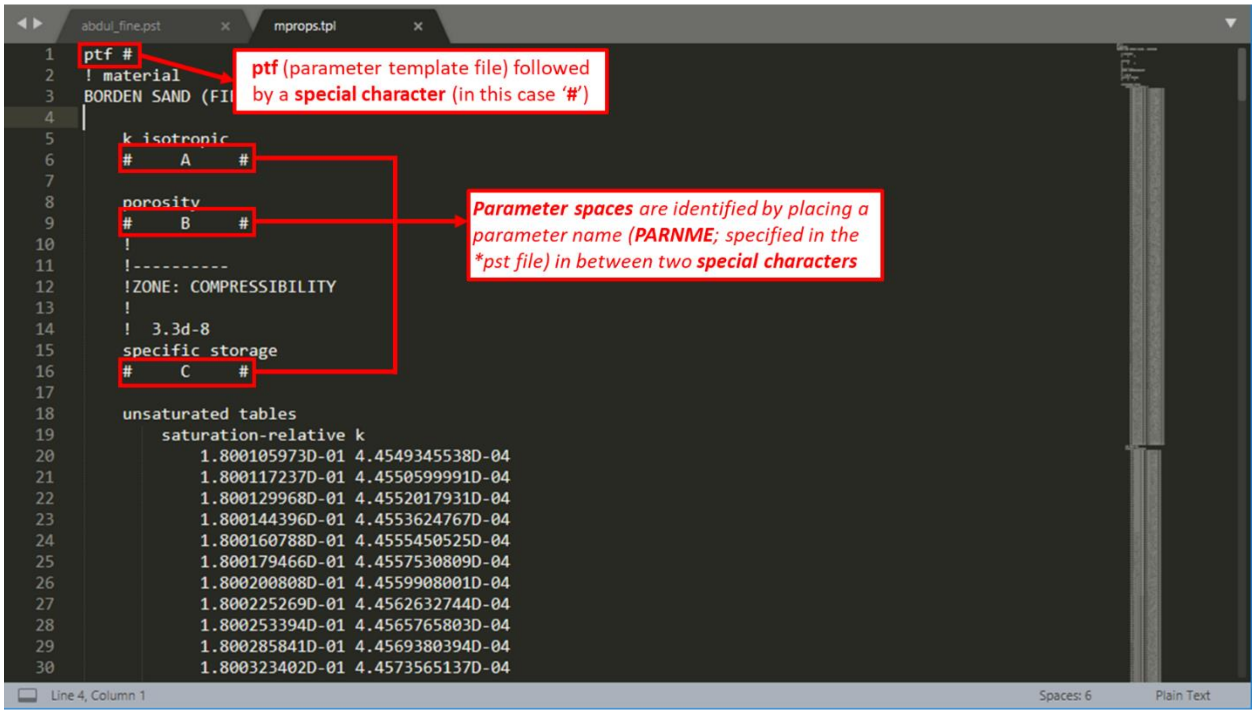
As PEST is initiated it begins reading through the PEST control file and encounters a series of parameters which it will use for the parameter estimation. PEST requires some method of updating the model and running it using a variety of parameter values. This is where the input template (*.tpl) file comes in. In the PEST control file you will identify the '***TEMPFLE***' or template input file name and the associated '***INFLE***' or model input file associated with that template (i.e. as referenced in the model *.grok file). PEST will use the *.tpl file (TEMPFLE) as a template to overwrite the named model input file (INFLE) before running the model with updated parameter values.

To create an input template file simply make a copy of the existing HydroGeoSphere property files (i.e. INFLE), rename them to the template input file name (i.e. TEMPFLE) and **change the file extension to *.tpl**. For example, in this Abdul_PEST tutorial the '*abdul.mprops*' and '*abdul.oprops*' files have been copied and renamed to '*mprops.tpl*' and '*oprops.tpl*'. Once the template files are created you must make configure it in such a way that PEST is able to identify that it is a template file, and be able to update parameter values.

Each *.tpl file must begin with an initial line containing 'ptf' (*parameter template* file) followed by a '*special character*' which PEST will use to identify the parameter spaces within the template file. Throughout the template file you then replace the location of actual parameter values with the parameter name (from the control file) between two of the special characters. ***Figure 5*** illustrates the general makeup of an input template file, based on the '*abdul.mprops*' file.

> ***Note:*** make sure the 'width' of the parameter space (i.e. the number of characters between the special characters) is large

enough to accommodate the expected parameter values. A wider parameter space provides a higher degree of precision (i.e significant digits).



```
ptf #
! material
BORDEN SAND (FI

      k isotropic
      #     A     #

      porosity
      #     B     #
      !
      !----------
      !ZONE: COMPRESSIBILITY
      !
      !  3.3d-8
      specific storage
      #     C     #

      unsaturated tables
            saturation-relative k
                  1.800105973D-01 4.4549345538D-04
                  1.800117237D-01 4.4550599991D-04
                  1.800129968D-01 4.4552017931D-04
                  1.800144396D-01 4.4553624767D-04
                  1.800160788D-01 4.4555450525D-04
                  1.800179466D-01 4.4557530809D-04
                  1.800200808D-01 4.4559908001D-04
                  1.800225269D-01 4.4562632744D-04
                  1.800253394D-01 4.4565765803D-04
                  1.800285841D-01 4.4569380394D-04
                  1.800323402D-01 4.4573565137D-04
```

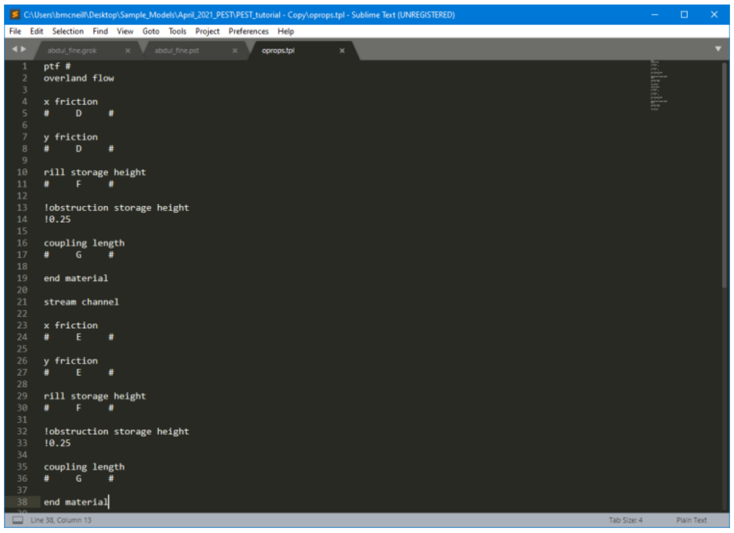
ptf (parameter template file) followed by a **special character** (in this case ‘**#**’)

Parameter spaces *are identified by placing a parameter name (**PARNME**; specified in the *pst file) in between two **special characters***

***Figure 5**: Layout of the Model Input Template (*.tpl) File(s)*

Note: make sure to select a special character that does not appear elsewhere in your model input files, otherwise PEST may have trouble differentiating parameter spaces with other portions of the input file.

Please note that an input template file may contain multiple parameter spaces for a single PEST parameter. For example, ***Figure 6*** shows the ‘*oprops.tpl*’ file and illustrates that both x and y friction parameters are defined by the same PEST parameters, and that both property zones (‘overland flow’ and ‘stream channel’) will share the same rill storage height and coupling length parameter values.



```
ptf #
overland flow

x friction
#     D     #

y friction
#     D     #

rill storage height
#     F     #

!obstruction storage height
!0.25

coupling length
#     G     #

end material

stream channel

x friction
#     E     #

y friction
#     E     #

rill storage height
#     F     #

!obstruction storage height
!0.25

coupling length
#     G     #

end material
```

***Figure 6*: Abdul_PEST 'oprops.tpl' File with Shared PEST Parameters (i.e. #D#, #E#, #F# and #G#)**

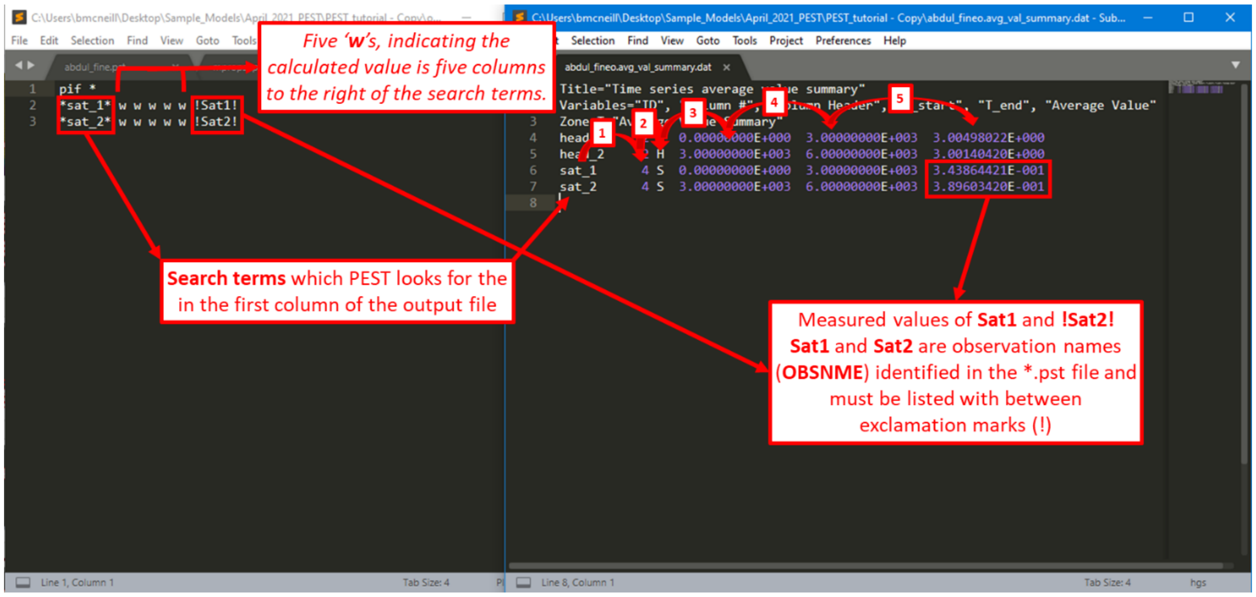
4.3 Model Output Reading Instruction (*.ins) File(s)

As PEST has run the updated model with new parameter values it requires some instruction to locate the resulting output data which should be compared with the available observation values listed in the PEST control file. In the PEST control file you should identify the '***INSFLE***' or output reading instruction file names and the associated '***OUTFLE***' or model output files associated with each data observation listed in the control file. Please note that ALL listed observations should be identified in only ONE of the listed output files.

Each *.ins file must begin with an initial line containing 'pif' (*parameter instruction* file) followed by a '*special character*' which helps PEST to read the output files.

After the first line you will use the special characters to provide PEST with a way to identify a value in the first column of the output file (i.e. the '*search term*'; frequently a time value). The *.ins file then includes one or more '**w**' characters, which indicate to PEST how many columns to the right the specified observation value can be found.

Figure 7 illustrates the general structure of an output reading instruction file, based on the '*observation3.ins*' file for the Abdul_PEST tutorial. Note that the '*observation3.ins*' file is associated with the '*abdul_fineo.avg_val_summary.dat*' output file. First a search term is provided, followed by five '**w**' characters to indicate that the desired output value is found five columns to the right of the search terms ('**sat_1**' and '**sat_2**'), followed by the observation name (**OBSNME**) listed in the control file:


C:\Users\bmcneill\Desktop\Sample_Models\April_2021_PEST\PEST_tutorial - Copy\o...
File Edit Selection Find View Goto Tools
abdul_fine.pst
Five 'w's, indicating the calculated value is five columns to the right of the search terms.
C:\Users\bmcneill\Desktop\Sample_Models\April_2021_PEST\PEST_tutorial - Copy\abdul_fineo.avg_val_summary.dat - Sub...
Selection Find View Goto Tools Project Preferences Help
abdul_fineo.avg_val_summary.dat
1
2
3
4
5
Search terms which PEST looks for the in the first column of the output file
Measured values of Sat1 and !Sat2! Sat1 and Sat2 are observation names (OBSNME) identified in the *.pst file and must be listed with between exclamation marks (!)
Line 1, Column 1
Tab Size: 4
Line 8, Column 1
Tab Size: 4
hgs

```
pif *
*sat_1* w w w w w !Sat1!
*sat_2* w w w w w !Sat2!
```

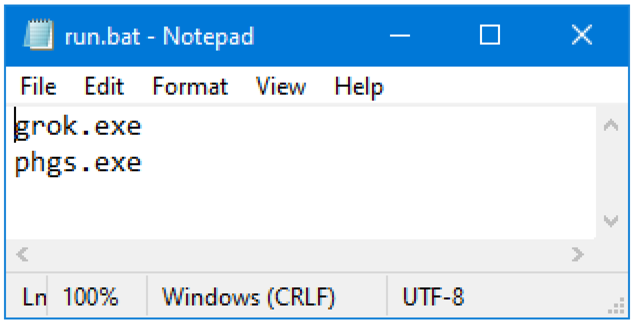
```
Title="Time series average value summary"
Variables="ID", "Column #", "Column Header", "T_start", "T_end", "Average Value"
Zone T="Average value summary"
head                0.00000000E+000  3.00000000E+003  3.00498022E+000
head_2            2 H  3.00000000E+003  6.00000000E+003  3.00140420E+000
sat_1             4 S  0.00000000E+000  3.00000000E+003  3.43864421E-001
sat_2             4 S  3.00000000E+003  6.00000000E+003  3.89603420E-001
```

***Figure 7**: Layout of the Model Input Template (*.tpl) File(s)*

Note: all observation values listed in the PEST control file must be included in one (and only one) of the output reading instruction files. If an observation value is listed in the control file and PEST does not have any instructions for finding the associated calculated value then you will encounter errors.

4.4 Model Batch (*.bat) File

Batch files are very simple but powerful script files that can be executed in the Windows environment. They simply contain a series of commands which can be used within the command prompt. When operating PEST with HGS simply include a batch file (e.g. '*run.bat*') in the folder which contains the commands 'grok.exe' and 'phgs.exe'. Once parameter values are updated PEST will initiate the batch file in order to recompile model inputs (grok.exe) and run the HGS model (phgs.exe). ***Figure 8*** shows the contents of the 'run.bat' file included in the Abdul_PEST tutorial:



```
grok.exe
phgs.exe
```

***Figure 8**: Batch File which Allows PEST to Run grok.exe and phgs.exe*

Note: your model folder should also include a batch prefix (*.pfx) file so that grok.exe and phgs.exe can automatically identify your model prefix/name.

5 Workflow for Using PEST with HGS

Now that the general workflow associated with the PEST optimization algorithm and the required input files have been introduced, let's review the actual workflow that run PEST on an HGS model.

1) To begin with, PEST always requires a model which is capable of converging under the range of parameter values used by the optimization algorithm. At the very least you must run your model once and ensure that the model converges and provides reasonable results.
   - Ensure the model outputs desired results at times associated with available observation data
2) Once a working model is available, simplifying the model (see the Tips/Tricks section below). PEST will run your model many dozens or hundreds of times, so reducing unnecessary computation will reduce the overall PEST runtime considerably.
   - Deactivate unnecessary output instructions/processes, pre-calculate property tables, etc.
3) Make a copy of the model in a PEST working folder
4) Write the PEST control file. Consider using the *Abdul_PEST.pst* file as a template and making necessary changes.
5) Make the necessary input template (**.tpl*) files by copying the associated HGS property files (e.g. **.mprops*), renaming it to the template file name (TEMPFLE; listed in the PEST control file) and changing the extension to '*.tpl*'. Use the specified '*special character*' to identify parameter spaces.
6) Write the necessary output reading instruction (**.ins*) files based on your available observations.
7) Write the model batch (**.bat*) file.
8) Copy the pest executable (*pest.exe*) into the PEST project folder.
9) Open the command prompt (type 'cmd' into Windows search bar and hit enter)
10) Type 'pest.exe' and the name/extension of the PEST control file into the command prompt and hit enter.
    - e.g. "**pest.exe Abdul_PEST.exe**"
11) PEST will now begin running the parameter optimization algorithm.

You can test this out using the Abdul_PEST tutorial by copying all files from the "PEST Input files" folder into the "PEST_Tutorial" folder and following steps 9) and 10) above.

6 Review PEST Output

Running PEST on the Abdul_PEST tutorial should take less than an hour (dependent on computer specs). It will run the model 77 times, and once it's finished you should see a message in the command prompt indicating that various run statistics (e.g. run details, parameter sensitivities, observation sensitivities and residuals) have been recorded in various ouput files. While the final PEST-estimated values will differ slightly for every unique PEST analysis, the following table indicates the general range of values expected before and after the PEST optimization analysis.

As you can see from ***Table 1***, the final parameter values estimated by PEST have resulted in a slightly better calibration (i.e. objective function reduced from 0.0801 to 0.0791).

Parameter	*PEST Parameter ID*	*Input File*	*Initial Value*	*PEST Best Estimate*
k isotropic	A	Abdul_PEST.mprops	6.1393735E-06	1.004131E-05
porosity	B	Abdul_PEST.mprops	0.30217968	0.370471
specific storage	C	Abdul_PEST.mprops	3.2915398E-07	2.287316E-08
Overland flow x/y friction	D	AbduL_PEST.oprops	0.400000	0.316727
Stream channel x/y friction	E	AbduL_PEST.oprops	2.0965694E-02	3.170640E-02
rill storage height	F	AbduL_PEST.oprops	1.00E-03	2.014355E-03
coupling length	G	AbduL_PEST.oprops	1.7866381E-04	1.024497E-04
Objective function (φ)	-	-	0.0801134	0.0790670

Table 1: Results of PEST Analysis (Abdul_PEST)

A close review of PEST output is outside the scope of this tutorial. However, a quick review of the main output files is provided to guide further study of the PEST documentation.

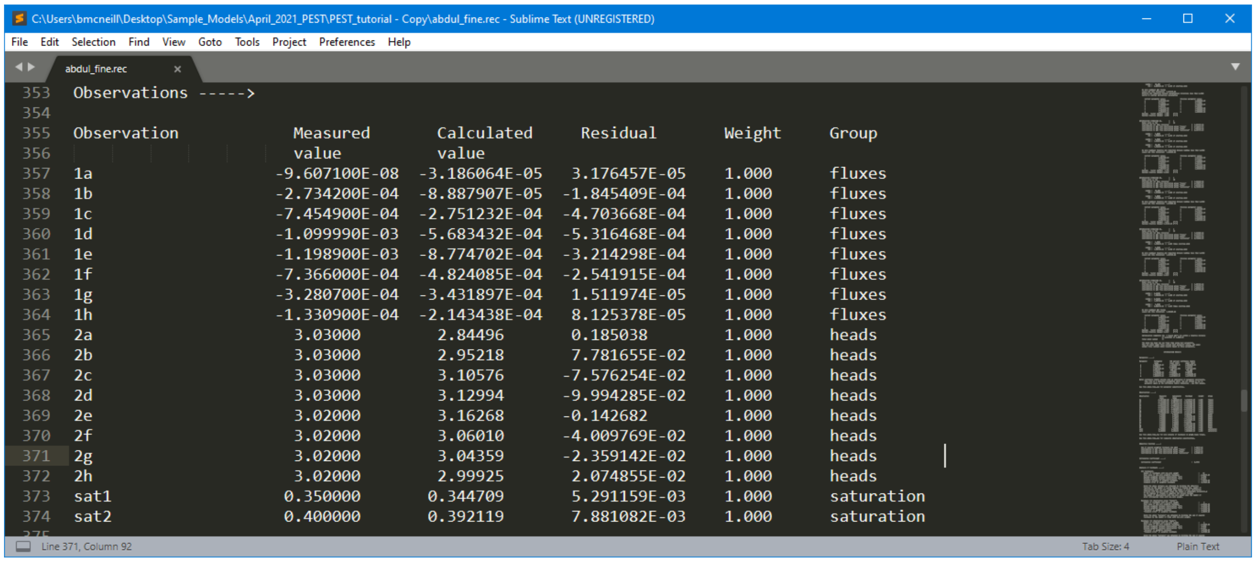
PEST results are written to four main output files:

- Abdul_PEST**.rec**: the 'record' file contains full details of the PEST optimization process, including parameter definitions, control settings, initial conditions, parameter values and objective function values during all optimization iterations.
- Abdul_PEST**.sen**: contains a record of parameter sensitivities. The sensitivity of each parameter with respect to each individual observation is recorded for each iteration of the optimization algorithm.
- Abdul_PEST**.seo**: contains a record of observation sensitivities. The sensitivity of each observation value with respect to each estimated parameter is recorded for each iteration of the optimization algorithm.
- Abdul_PEST**.res**: contains a record of observation residuals during each iteration of the optimization algorithm. Observation residuals are used at the end of each iteration to calculate the objective function (i.e. the main metric of success for the optimization algorithm).

The final sections of the *.rec file record the final/best estimated parameter values based on the PEST optimization algorithm. As indicated by the *.rec file,

> "The model has been run one final time using best parameters.
> Thus all model input files contain best parameter values, and model
> output files contain model results based on these parameters."

It should be noted that the residual values associated with the head observations (i.e. 2a-2h) are orders of magnitude larger than the residual values associated with flux or saturation observations. This is due to the difference in the absolute magnitude of these observed values, and one potential improvement to the PEST run that could be explored is applying different weights to each observation type to ensure that each exert equal control over the objective function. ***Figure 9*** shows the final residual values associated with the available observation data.


C:\Users\bmcneill\Desktop\Sample_Models\April_2021_PEST\PEST_tutorial - Copy\abdul_fine.rec - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
abdul_fine.rec
Line 371, Column 92
Tab Size: 4
Plain Text

```
Observations ----->

Observation            Measured       Calculated     Residual       Weight     Group
                       value          value
1a                   -9.607100E-08  -3.186064E-05   3.176457E-05    1.000      fluxes
1b                   -2.734200E-04  -8.887907E-05  -1.845409E-04    1.000      fluxes
1c                   -7.454900E-04  -2.751232E-04  -4.703668E-04    1.000      fluxes
1d                   -1.099990E-03  -5.683432E-04  -5.316468E-04    1.000      fluxes
1e                   -1.198900E-03  -8.774702E-04  -3.214298E-04    1.000      fluxes
1f                   -7.366000E-04  -4.824085E-04  -2.541915E-04    1.000      fluxes
1g                   -3.280700E-04  -3.431897E-04   1.511974E-05    1.000      fluxes
1h                   -1.330900E-04  -2.143438E-04   8.125378E-05    1.000      fluxes
2a                     3.03000        2.84496        0.185038       1.000      heads
2b                     3.03000        2.95218        7.781655E-02   1.000      heads
2c                     3.03000        3.10576       -7.576254E-02   1.000      heads
2d                     3.03000        3.12994       -9.994285E-02   1.000      heads
2e                     3.02000        3.16268       -0.142682       1.000      heads
2f                     3.02000        3.06010       -4.009769E-02   1.000      heads
2g                     3.02000        3.04359       -2.359142E-02   1.000      heads
2h                     3.02000        2.99925        2.074855E-02   1.000      heads
sat1                   0.350000       0.344709       5.291159E-03   1.000      saturation
sat2                   0.400000       0.392119       7.881082E-03   1.000      saturation
```

***Figure 9**: Final Residual Values Following the Optimization Algorithm*

7 Tips/Tricks/Best Practices

1) To start, you'll want to build a good initial HGS simulation that is both accurate and compact

   a. By accurate I mean you will want a decent initial calibration

   b. By compact I mean you will want to reduce unnecessary computation steps and also optimize your run settings for faster run times. PEST will be running HGS many *many* times, so cutting down the run-time as much as possible is going to save you a lot of time.

      i. You don't want PEST to calibrate a model with an unnecessary, extended spin-up period at the beginning.

      ii. You don't want PEST to calibrate a model that includes processes (e.g. transport) that are not going to impact PEST.

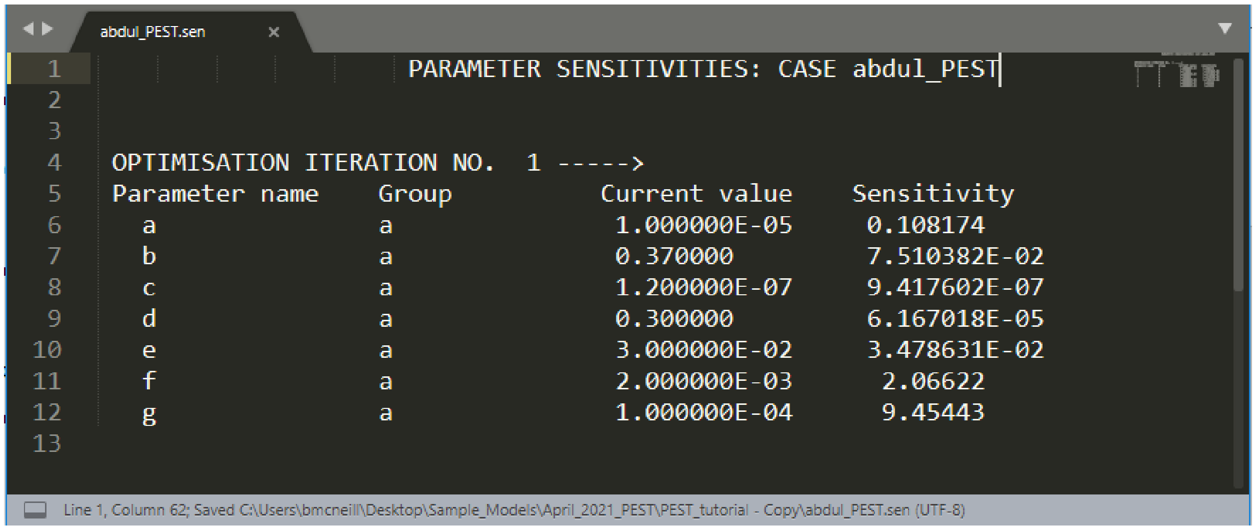
      iii. You don't want your model to include superfluous tasks like writing out hydrograph nodes, observation wells, etc. Just stick to what PEST actually needs.

2) PEST requires access to all executables involved in the modeling process (i.e. all executables listed in 'run.bat' file). Therefore, you should copy the HGS executables (grok.exe and phgs.exe) into the PEST run folder or set environment variables for HGS and PEST related executables.

3) Always remember: PEST has absolutely no idea what parameter values, or combination of parameter values, would be considered reasonable by a hydrologist or hydrogeologist. PEST should only be used alongside a healthy dose of expert knowledge! You can incorporate your knowledge of the system into the PEST process by applying appropriate upper and lower bounds on parameter values, tying various parameter values to each other (i.e. if parameter X increases then parameter Y increases too), applying higher weights to observations which you are confident about, etc.

4) Before using PEST in full parameter estimation mode, consider running an initial parameter sensitivity analysis to determine which parameters exert the greatest control over your model (i.e. the most sensitive). By focusing the parameter estimation mode only on the most sensitive parameters you can decrease runtimes and get the greatest return on your time.

    a. The simplest way to do this is simply to run PEST in parameter estimation mode and wait for the first optimization iteration to complete. A record of parameter sensitivities is written to the *.sen file, so once this file is updated with the sensitivities from the first optimization iteration you can cancel the PEST run and review the sensitivity info. An example of the *.sen file is shown below. A higher sensitivity indicates that the model output is more sensitive to changes in the parameter than those with a lower sensitivity. In the example below we see that parameter ‘g’ is the most sensitive, and parameter ‘c’ is the least sensitive. Therefore, you may consider the value of including parameter ‘c’ in the parameter estimation process.



```
                    PARAMETER SENSITIVITIES: CASE abdul_PEST


OPTIMISATION ITERATION NO.  1 ----->
Parameter name     Group          Current value    Sensitivity
  a                a               1.000000E-05     0.108174
  b                a               0.370000         7.510382E-02
  c                a               1.200000E-07     9.417602E-07
  d                a               0.300000         6.167018E-05
  e                a               3.000000E-02     3.478631E-02
  f                a               2.000000E-03      2.06622
  g                a               1.000000E-04      9.45443

```

Line 1, Column 62; Saved C:\Users\bmcneill\Desktop\Sample_Models\April_2021_PEST\PEST_tutorial - Copy\abdul_PEST.sen (UTF-8)

Figure 10: Example of *.sen Output File (Parameter G is most sensitive)

5) Your HGS model must output results at the exact same times as the available observation data. However, if you have many observations (e.g. daily) over a long time period, you may experience very long model run times. Therefore, consider providing PEST with only a representative subset of the available observation data.

    a. By using the ‘target times’ command you can force HGS to calculate a timestep at the desired time, ensuring that results will be written at the exact right time.

    b. You can also consider calibrating against averaged values using the ‘compute post simulation average’ command.

    c. Both of these options are illustrated in the Abdul_PEST example files.