INEEL/CON-03-00154
PREPRINT

New Data and File Requirements for TET^{-1}

G. Michael Shook

January 27, 2003

Twenty-Eighth Workshop on Geothermal Reservoir Engineering

PROCEEDINGS, Twenty-Eighth Workshop on Geothermal Reservoir Engineering
Stanford University, Stanford, California, January 27-29, 2003
SGP-TR-173

NEW DATA AND FILE REQUIREMENTS FOR TET^{-1}

G. Michael Shook

Idaho National Engineering and Environmental Laboratory
PO Box 1625
Idaho Falls, ID 83415-2107
ook@inel.gov

ABSTRACT

A new version of Tet^{-1} is being finalized for distribution and beta-testing. The new version allows the user to create and modify the TETRAD input deck either graphically or manually. By defining Regions within the TETRAD domain and defining parameters within the regions, parameter estimation is done external to any proprietary software. The new version of Tet^{-1} can be run on any existing version of TETRAD, and makes identifying the TETRAD parameters to be estimated easier and more general than was previously possible. A beta-test version will be available in April 2003.

INTRODUCTION

A project started last year at the Idaho National Engineering and Environmental Laboratory has continued with the development of a test version of Tet^{-1}. Based on the public domain Parameter ESTimation model PEST (Doherty, 2000), Tet^{-1} consists of a suite of I/O and instruction files that runs a forward model (TETRAD), creates observation and prediction output files used in determining parameter estimate updates, modifies the input parameters, etc. until pre-set parameter estimation convergence criteria are met. The previous version of Tet^{-1} had most of the instructions and file names "hard wired," and thus was neither flexible nor user-friendly. By taking advantage of certain input/output features of TETRAD, we have increased the ease of using this code. The new features are described below; a new version of Tet^{-1} will be available to interested users in April 2003.

PEST AND THE INVERSE PROBLEM

The inverse problem, i.e., determining reservoir properties from field observations, is solved by minimizing the sum of squared differences between field observations, **c**, and model predictions, **b**.

$$\Phi = (\mathbf{c} - \mathbf{b}(\mathbf{x}))^t (\mathbf{c}-\mathbf{b}(\mathbf{x})) \quad (1)$$

The model predictions, **b**, are functions of the parameters, **x**, that we wish to estimate. Because model predictions (e.g., production enthalpy) are generally non-linear functions of the model parameters, we first linearize model response using Taylor's theorem:

$$\mathbf{b} = \mathbf{b}_o + \frac{\partial \mathbf{b}}{\partial \mathbf{x}} \delta\mathbf{x} = \mathbf{b}_o + \mathbf{J} \cdot \delta\mathbf{x}$$

The matrix of partial derivatives is known as the sensitivity matrix or Jacobian, and is denoted as **J**. Calculating **J** requires a large number of forward model runs, as the individual elements are the sensitivity of each model prediction to each parameter being estimated.

Different field observations have different degrees of accuracy, suggesting some observations carry more weight than others. Different types of observations also have different magnitudes, and often exert different influence of the objective function. We can assign different weights to the observations by defining **Q** as a diagonal matrix comprising (user-defined) field observation weights. The inverse problem can now be defined as deriving a set of model parameters (and therefore model predictions) which minimizes the objective function Φ

$$\text{Min: } \Phi = (\mathbf{c}-\mathbf{J}\mathbf{x})^t \mathbf{Q} (\mathbf{c}-\mathbf{J}\mathbf{x}) \quad (2)$$

We obtain an updated parameter set $\delta\mathbf{x}$ by solving

$$\delta\mathbf{x} = (\mathbf{J}^t \mathbf{Q}\mathbf{J})^{-1} \mathbf{J}^t \mathbf{Q}(\mathbf{c}-\mathbf{b})$$

The above discussion is intentionally simplified, ignoring specifics of PEST's implementation of the method, variable transformations available, methods of accelerating parameter updates (Marquardt, 1963; Levenberg (1944)), and other PEST features. One important by-product of the parameter estimation process is an extensive sensitivity and error analysis

which can identify the key parameters governing system responses and identify parameter correlations. A detailed discussion of these features is given by Doherty (2000). A discussion on their use and impact on parameter estimation within the framework of Tet^{-1} is given by Shook and Doherty (2003).

TET^{-1}

Tet^{-1} is essentially an interface that executes forward TETRAD models and pre- and post-processes TETRAD I/O files for use in PEST's parameter estimation calculation discussed above. PEST is a particularly useful inverse model for this project for a variety of reasons. First, PEST is in the public domain, and therefore available to a wide range of users at a nominal price. PEST is also model independent, working only on input and output files. Because most TETRAD users do not have access to source code, this feature preserves the code developer's interests. Also, PEST operates in a script mode; any number of commands (models) can be executed in the script. This is the single most important feature of PEST for later stages of this project, since we plan to combine both reservoir modeling and geophysical modeling in future parameter estimation software.

There are several new input files required by this new version of Tet^{-1}. These files are intended to make input generation and parameter estimation simpler and more flexible than in the previous version. New input file requirements are discussed below. TETRAD normally appends a file extension on the "root name" of the input file (e.g., from TEST1.DAT TETRAD generates TEST1.OUT). For the purpose of discussion below, "*" is used in lieu of the root file name. An example problem discussed in a following section shows the structure of each of the new files and input requirements.

Input Files

The input file structure for a TETRAD run has not changed; however, Tet^{-1} now requires the previously-optional *.IS file as well. If PetroSim or Intersim is used in the file construction, this file is already present. If a GUI is not used in input file preparation, the keyword "ISWRITE" should be invoked after reading in model properties, and a *.IS file written. This file contains model dimensions and grid size, and (optionally) a number of model parameters, initial and boundary conditions, etc. It may contain appreciably more, *but must at minimum contain all parameters to be estimated in Tet^{-1}*, as the instruction programs discussed below modify this input file.

An example of what might be written to the *.IS using 'ISWRITE' is given below. If we were trying to estimate only reservoir porosity and permeability (assumed isotropic here), either of the following two lines should be included in the input file *.DAT:

```
'ISWRITE'  0  'TEST1.IS'
2    'POR'  'kX'
```

or

```
'ISWRITE'  0  'TEST1.IS'
6    'POR'  'P'  'T'  'SG'  'AQU'  'kX'
```

The first example writes only the minimum data required in the *.IS file; the second clearly does more. Either TEST1.IS file works with Tet^{-1}. Having generated this *.IS file, 'ISWRITE' would then be removed from the input deck, and the following line would be included:

```
'ISREAD'  0  'TEST1.IS'
```

This tells TETRAD that there exists reservoir properties and grid information residing TEST1.IS.

Three new input files are also required to run Tet^{-1}. The first file defines where "regions" exist within the grid domain (a "Regions Definition File, or *.RDF). Regions are defined as a group of grid blocks from "row" I1 to I2, "column" J1 to J2 and layer K1 to K2. Note that every grid block can be a different region, if desired. The file format is as shown in Table 1 below (explanations in italics):

Table 1. File structure of a *.RDF file

NX NY NZ	*grid dimensions*
NREGIONS	*# regions to be defined*
NAME1 Nlines	*Region name, # lines required to define*
I1 I2 J1 J2 K1 K2	*Region boundaries, repeat*
…	*this line Nline times*
NAME2 Nlines	*etc.*

Multiple lines defining a Region may be necessary to capture field features (e.g., a dipping fault that migrates basin-ward with depth). A program checks to ensure all grid blocks are assigned to only 1 region. This program essentially maps properties defined on a regional basis to individual grid blocks. Regional properties are defined below.

A second new input file required is a "Regional Properties File (*.RPF). This file has the following structure (comments in italics).

Table 2. Contents of a *.RPF file

NREGIONS NVAR	*# Regions, # Variables*
REGION 'Param1' ... 'ParamN'	*This is a header*
NAME1 Value1 … ValueN	*Parameter values*
NAME2 …	*Continued NREGIONS times*

Where the word Param1 for example is replaced by the name of the parameter to be estimated (e.g.,

'POR'). Parameter names used in this file must correspond to the *.IS naming convention.

This file defines all of the properties that *may* be estimated by Tet^{-1} for a particular problem. Not all parameters listed have to be estimated in every region, but every one listed should be estimated by Tet^{-1} in at least one region. The template file discussed below specifies which parameters in which regions are variable (i.e., being estimated) and which are fixed. Values are entered for all parameters. For parameters to be estimated, these are initial guesses; for fixed parameters these are their known value. This file is updated by PEST. Updated parameter values are then written to the *.IS file using the mapping information from the *.RDF file. The *.IS file (with updated parameters) is read by TETRAD every forward model run.

The final new input file required is the template file (*.TPL) that PEST uses to identify and update the parameters to be estimated. This file is a copy of the *.RPF file with 2 changes. A new line 1 is inserted indicating how parameters are to be delineated in the file. The delineation character can be anything not otherwise encountered in the TPL file. Secondly, the values of parameters to be estimated are replaced with the parameter's variable name (as known by PEST). A simple example is given in Table 3 below.

Table 3. A Template file (*.TPL)

```
ptf #
NREGIONS   NVAR
REGION   'POR'        'kX'
NAME1    0.05        #kx1     #
NAME2    #por2    #  100.
```

In this example, there are two regions. Porosity in region 1 is 'known," as is permeability in region 2. The two parameters being estimated are delineated with the " # " character, and are written with sufficient accuracy so the Jacobian elements are calculated appropriately.

Processing TETRAD Output

Having created the input files described above, a forward simulation can then be run. It is assumed that field observations and model output (predictions) vary in time, and are associated with wells or other point measurements. They are, therefore, data TETRAD can write to a history plot file (a *.PLT file) using the 'PLOT' keyword. Two utility programs re-format the *.PLT file, time-interpolate the model data to a similar set of field observations, and write an output file of model predictions.

A Forward Model Run

It should be clear from the above discussions that a single forward model run actually comprises a series of commands, including updating the *.IS file, executing TETRAD, and post-processing the output. Each of these steps (programs) requires input, which is either entered from the keyboard or from file redirect. Because PEST makes numerous model runs in the inversion process, input from the screen is clearly impossible. Therefore, a batch file is written that calls each program in turn, providing input to each by way of file redirection. The *.in files are most easily created by running the programs in one window and creating the appropriate input file in another.

An example batch file for a DOS environment is shown in Table 4. A similar structure would of course be required in other operating systems.

Table 4. A Batch file that runs the forward models.

```
@echo off
rem update *.is using *.rdf and *.rpf
rem reg.in tells regmap the names of the relevant files
regmap  <reg.in             >nul
rem delete tetrad outfiles as required
rem clean.in contains the file names to be deleted
clean    <clean.in          >nul
rem run tetrad
rem tet.in contains input and output file names
tetm1    <tet.in            >nul
rem post process tetrad output
rem the *.in contain I/O file names required
tet2smp  <tet2smp.in        >nul
smp2smp <smp2smp.in     >nul
```

A final file is required to run Tet^{-1}. This is known as the Pest Control File; it is created interactively by executing PESTPRPT. Responses to a series of prompts results in creating the *.PST file. This file contains information on how many parameters are to be estimated, types of variable transformations, initial guesses and bounds, etc. Defaults are used for variable transforms, initial values, etc., which may need changing (they do in the example below). The details of this file are discussed at length by Doherty (2000), and Shook and Doherty (2003).

TET^{-1} EXECUTION EXAMPLE

File Creation

The following example show the new files required by the new version of Tet^{-1}. This example is discussed in more detail elsewhere (Shook and Renner, 2002). It is a simplified example of injection

of cool liquid into an initially saturated (P_I = 17 bar; T_I = 170ºC) geothermal reservoir. Fluid production and tracer test results from each of four production wells are used to estimate porosity and permeability in the 2-D domain.). The correct domain and petrophysical properties are shown schematically in Figure 1.

This example is purely numerical in nature. That is, the reservoir permeability and porosity were first assumed known, and a set of "field observations" (production rates and pressures, tracer concentration histories) were obtained from a TETRAD simulation. An abridged version of this file (REAL.SMP) showing the required format for field observations is given in Appendix 1 (the complete file is given in the electronic version of the paper).

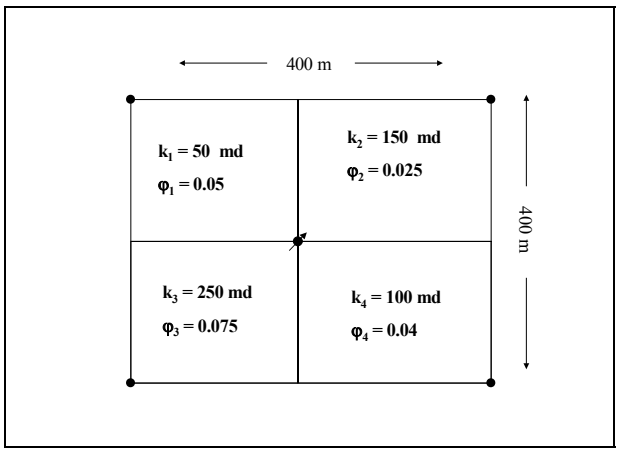

Figure 1. Schematic of reservoir description for example problem. Production wells are at each corner; injection well is in the middle.

Having generated "true" field observations, the input file was changed from the correct permeability and porosity to (incorrect) uniform conditions of k=750 md and φ = 0.1. It was assumed that all other properties of the example domain (e.g., size, initial conditions) are known, and that PetroSim was used to create 'TEST1.IS.' In this example, grid dimensions and size, permeability, porosity, pressure, temperature, and initial saturations were all entered via PetroSim. The input file TEST1.DAT is given in Appendix 2.

TEST1.RDF is then created. We have fairly simple regional definitions: 4 regions, each associated with a quadrant of the domain. TEST1.RDF is given in Table 5 below. We are estimating permeability and porosity for each of the 4 regions, so TEST1.RPF is as shown in Table 6. The template file TEST1.TPL is also given in Table 6.

Before Tet^{-1} can be run, a forward simulation must be made. This run is necessary to determine the initial model predictions and calculate the objective function. The well history plot file (TEST1.PLT) is post-processed to create an initial prediction file with format consistent with the Field Observations file. While doing so, the input files for TET2SMP and SMP2SMP can be written, as shown in Table 7.

Table 5. The RDF file for Example TEST1

```
21   21        1
4
QUAD1          1
1    10        1          10        1         1
QUAD2          1
11   21        1          10        1         1
QUAD3          1
1    10        11         21        1         1
QUAD4          1
11   21        11         21        1         1
```

Table 6. The RPF (a) and TPL file (b) for TEST1

```
(a)
4    2
REGION     'POR'    'kX'
QUAD1               0.1       750.
QUAD2               0.1       750.
QUAD3               0.1       750.
QUAD4               0.1       750.

(b)
ptf #
4    2
REGION     'POR'              'kX'
QUAD1               #por1    # #kx1     #
QUAD2               #por2    # #kx2     #
QUAD3               #por3    # #kx3     #
QUAD4               #por4    # #kx4     #
```

Table 7. TET2SMP.in (a) and SMP2SMP.in (b) for TEST1 (comments in italics)

```
(a)
test1.plt          the input file
01/01/2000         starting date
n
test1.smp          the reformated file

(b)
real.smp           field observations file
test.smp           tetrad reformatted file
0
model.smp          model predictions file
```

Finally, the batch file can be prepared as shown in Table 4, and PESTPRPT is run. The resulting Pest Control File, TEST1.PST is given in Appendix 3.

Tet^{-1} is invoked by executing the following command.

PEST TEST1

Example Results

Results from this example problem are discussed by Shook and Renner (2002). Tet^{-1} made a total of 275 model runs before satisfying termination criteria and stopping. PEST correctly reproduced the unknown parameters (maximum relative error is 0.1%). While that is of great comfort (especially in the testing phase!), the objective of this paper was to demonstrate the simplicity of creating the Tet^{-1} interface, not in generating parameter estimates. An extensive example set, including all files discussed here is in preparation (Shook and Doherty, 2003). This document will discuss the nuances of Tet^{-1} in greater detail, including model setup, file generation, and interpretation of the statistics generated by PEST. This document will be available along with a beta-test version of Tet^{-1} to interested users in April 2003.

SUMMARY

A new version of Tet^{-1} has been written that both simplifies and generalizes the process of identifying TETRAD input parameters that are to be estimated. The new method requires the user to create three new input files that describe reservoir properties (parameters that might be estimated) on a "regional" basis. New programs map these values onto the existing TETRAD domain. This process, and those that post process the output, is transparent to the user.

Final testing of this new version of Tet^{-1} is ongoing, but will be finished shortly. A beta-test version of the code, along with more extensive discussion of the program structure and output will be available in April 2003.

ACKNOWLEDGEMENTS

The author gratefully thanks John Doherty of Watermark Numerical Computing for his support in linking PEST to TETRAD, and for illuminating the nuances of PEST. Funding for this work was provided by the U.S. Department of Energy, Office of Geothermal and Wind Technologies, under contract DE-AC07-99ID13727.

REFERENCES

Doherty, J.L, 2000, "PEST: Model Independent Parameter Estimation, Preface to the 4th Edition," Watermark Numerical Computing.

Levenberg, K., 1944, "A Method for the Solution of Certain Nonlinear Problems in Least Squares," *Q. Appl. Math*, V. 2, pp 164-168.

Marquardt, D.W., 1963, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *J. Soc. Ind. Appl. Math*., V. 11(2), pp 431-441.

Shook, G.M., and J.L. Renner, 2002, "An Inverse Model for TETRAD: Preliminary Results," Trans. Geothermal Resources Council,Vol. 26, pp. 119-122.

Shook, G.M. and J.E. Doherty, 2003, "Tet^{-1}: An Interface between PEST and TETRAD for Parameter Estimation in Thermal, Multiphase Flow Problems," INEEL report, in preparation.

APPENDIX 1. EXAMPLE OF "FIELD OBSERVATIONS" FILE

```
P1      QTOTMASS     01/02/2000     00:00:00      -0.1888886
P1      QTOTMASS     01/02/2000     06:00:00      -0.1817076
P1      QTTOTMAS     01/02/2000     12:00:00      -0.1761414
** extensive additional rate data deleted in this example
P1      QFCGP2       01/02/2000     00:00:00       0.000000
P1      QFCGP2       01/02/2000     06:00:00       0.000000
P1      QFCGP2       01/02/2000     12:00:00       0.000000
** extensive additional mass fraction of tracer data deleted
P1      QTOTENTH     01/02/2000     00:00:00       1329.717
P1      QTOTENTH     01/02/2000     06:00:00       1323.416
P1      QTOTENTH     01/02/2000     12:00:00       1319.561
** extensive additional enthalpy data deleted
P1      PBHW         01/02/2000     00:00:00       900.0000
P1      PBHW         01/02/2000     06:00:00       900.0000
P1      PBHW         01/02/2000     12:00:00       900.0000
** extensive additional bottom hole pressure data deleted
** similar data for other wells (P2, P3, P4) deleted in this abbreviated file.
```

APPENDIX 2. THE EXAMPLE INPUT FILE

```
'COMMENT'  'Run # Test1'
'TYPE'       4    2    2    0.0     0.0
'UNITS'       1      1      0       1
'DIMEN'     21   21     1   5000
'ISREAD'  'TEST1.IS'
'PLOT'  0  0
4  'QTOTMASS'  'QFCAP2'   'QTOTENTH'  'PBHW'
'PERMMULT'  1  441  1  1.  -1.  -1.
'RELANAL'  0  1  0  1.
240.  240.  1.E-6  0.03  1.E-6  0.03
.3   1.  1.5
0.  0.  1.1    0.  2.
0.  1.  1.5
0.  0.  1.5  0.  1.
'COMMENT'  '************************************************************'
'RECUR'
'TIMEYR'   0
'TIME'  0.0   0.0001
'PRODUCER'  'P1'  1  3
1
0.133  0.07,,,,,,,,,,
,,,,,,,,,-20.
2.5,,,,,,,,,,,,,,
[ other well definitions deleted ]
'R'  'I1'  0.  0.  1.
'P'  'P1'  0.1  900.  0.  0.   -1.
'P'  'P2'  0.1  900.  0.  0.  -1.
'P'  'P3'  0.1  900.  0.  0.  -1.
'P'  'P4'  0.1  900.  0.  0.  -1.
'TIME'   1.  -1.
'R'  'I1'  0.  1.   0.
'TIME'  1.5   -1.
[ other 'TIME' lines deleted ]
'TIME'  100.  0.
```

APPENDIX 3. THE PEST CONTROL FILE

```
pcf
* control data
restart  estimation
     8   340     2     0     4
    1     1 single  point  1   0   0
 10.0  2.0  0.3  0.03  10
 10.0  10.0  0.001
 0.1
 30  0.005  4  4  0.005  4
 1  1  1
* parameter groups
por      relative    0.01  0.0  switch  2.0 parabolic
perm     relative    0.01  0.0  switch  2.0 parabolic
* parameter data
por1         log  factor   .1  1.0e-10  1.0e10 por     1.0  0.0  1
por2         log  factor   .1  1.0e-10  1.0e10 por     1.0  0.0  1
por3         log  factor   .1  1.0e-10  1.0e10 por     1.0  0.0  1
por4         log  factor   .1  1.0e-10  1.0e10 por     1.0  0.0  1
perm1        log  factor   750.  1.0e-10  1.0e10 perm     1.0  0.0  1
perm2        log  factor   750.  1.0e-10  1.0e10 perm     1.0  0.0  1
perm3        log  factor   750.  1.0e-10  1.0e10 perm     1.0  0.0  1
perm4        log  factor   750  1.0e-10  1.0e10 perm     1.0  0.0  1
* observation groups
 1
 2
 3
 4
observation data
[ The Observation data was removed from this Appendix due to excessive length.
            It is the same data as is written in the Field Observation File – REAL.SMP
            found in Appendix 1 in this example ]
* model command line
 test1.bat
* model input/output
 test1.tpl  test1.dat
 test1.ins  model.smp
* prior information
```