U.S. DEPARTMENT OF
ENERGY

Prepared for the U.S. Department of Energy
Under Contract DE-AC05-76RL01830

PNNL- 18395

FRAMES-2.0 Software System: Frames 2.0 Pest Integration (F2PEST)

KJ Castleton
PD Meyer

May 2009

Pacific Northwest
NATIONAL LABORATORY

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

This document was printed on recycled paper.
(9/2003)

FRAMES-2.0 Software System: Frames 2.0 Pest Integration (F2PEST)

Karl J. Castleton
Philip D. Meyer

May 2009

Prepared for The U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Division of Systems Analysis & Regulatory Effectiveness
Washington, DC 20555-0001
under a Related Services Agreement with
the U.S. Department of Energy
Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Summary

The implementation of the FRAMES 2.0 F2PEST module is described, including requirements, design, and specifications of the software. This module integrates the PEST parameter estimation software within the FRAMES 2.0 environmental modeling framework. A test case is presented.

Contents

Tables

Figures

1.0 Introduction

The U.S. Nuclear Regulatory Commission (NRC) staff members are responsible for reviews of Early Site Permit (ESP) and Combined License (COL) applications for new nuclear power plants. Using site-specific information from the ESP and COL applications, NRC staff draft environmental impact statements (EIS) and perform confirmatory safety-related analyses for these proposed new plants. Many important technical-review issues surround the assessment of site-specific ground-water and surface-water conditions and behaviors that influence site hydrology and radionuclide transport. Under a variety of reasonable release scenarios, radionuclide transport through the ground-water pathway is a key component in assessing potential exposures to the public. Technical review issues related to groundwater transport of radionuclides include the formulation of hydrogeologic site conceptual models, the identification of plausible radionuclide transport pathways, the appropriate degree of complexity and conservatism in modeling transport, and the adequacy of site characterization data and monitoring plans. Resolution of these issues can be facilitated by a systematic effort to acknowledge and assess the important uncertainties in the characterization and representation of site-specific groundwater transport.

A systematic and quantitative approach for combining estimates of uncertainties in the data, conceptual models, parameters, and scenarios related to hydrogeology was presented by Meyer et al. (2007). An application to uranium transport at the U.S. Department of Energy Hanford Site 300 Area illustrated the practical benefits of the approach to provide better estimates of predictive uncertainty, quantitative results for use in assessing risk, and an improved understanding of the system behavior and limitations of the models. Although the approach and its applications were specifically designed for reviews of radionuclide transport at complex decommissioning sites, the approach is applicable to any analysis involving hydrologic modeling, including assessing nuclear facility siting, designing ground-water monitoring programs, remediating ground-water, and identifying and selecting strategies to preclude offsite migration of abnormal radionuclide releases.

To facilitate application of a systematic approach to uncertainty assessment in ESP and COL reviews, some of the methods described in Meyer et al. (2007) have been integrated with the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES) version 2 software. FRAMES[1] is a software platform that allows users the ability to select and implement environmental software models for risk assessment and management problems. This program is a flexible and holistic approach to understanding how activities affect humans and the environment. It links models that integrate across scientific disciplines, allowing for tailored solutions to specific activities, and it provides meaningful information to business and technical managers. FRAMES is the key to identifying, analyzing, and managing potential environmental, safety, and health risks. The purpose of FRAMES is to assist users in developing environmental

[1] http://mepas.pnl.gov/FramesV2/index.stm

scenarios and to provide options for selecting the most appropriate computer codes to conduct human and environmental risk management analyses.

The uncertainty approach of Meyer et al. (2007) uses information resulting from a maximum likelihood or least squares model parameter estimation to provide comparative measures of model predictive performance. This report describes the integration within FRAMES of the parameter estimation software PEST[2] (Doherty 2004). The module implementing this integration will be referred to as F2PEST. PEST is model-independent software implementing a variety of parameter estimation methods, including nonlinear regression of a generalized least squares objective function, from which the model comparison statistics are computed. During execution PEST will iteratively try different values for a model's input parameters and then compare a selected set of simulated model outputs to a given set of expected outputs, typically observed (measured) values, using the results of this comparison to compute new input parameter values.

While the complete documentation for the file formats PEST uses is available in the PEST documentation, a summary is provided here as context for the design. PEST interfaces to a model through three file types. The first file type that PEST uses is an input file template. This is a file that describes an input file (or data file) that needs to be modified by PEST, typically a model input file containing model parameters. The template file uses a mark-up approach to note where important values are and assigns aliases to those values. Associated with the input file template is the model input file that will be read. Table 1.1 is an example of an input file and associated template file.

Table 1-1. Example model input file and corresponding PEST template file

Example Model Input (Data) File	*Corresponding Input File Template*
Model input	ptf #
12 Number of months	Model input
40 Number of years	#NMonths # Number of months
65 Concentration	#NYears # Number of years
	#Conc # Concentration

The lines on the left of Table 1.1 represent the original input file including comments and descriptive information. The lines on the right represent the associated template file that contains the aliases NMonths, NYears, and Conc that should be associated with the values of 12, 40, and 65 respectively.

The second type of file PEST uses is an instruction file that describes the location of selected model output values and assigns alias names. Because reading an output file requires certain values to be found at locations marked by text labels, PEST allows for commands to be provided

[2] http://www.sspa.com/pest/

specifying how to read the data that follows. In the F2PEST tools the format is very simple because the F2PEST module writes the model output. In this case, the PEST "l1" command ("ell-1") is sufficient to read the corresponding output from a FRAMES model system. Each line of the instruction file illustrated in Table 1-2 tells PEST to advance one line in the model output file and read the output value between columns 1 and 7 for the observation named in parentheses, e.g., c3.

Table 1-2. Example model output file and corresponding PEST instruction file

Example Model Output File	*Corresponding Instruction File*
0.11948	pif #
0.29620	l1 (c1)1:7
0.50162	l1 (c2)1:7
0.67325	l1 (c3)1:7
	l1 (c4)1:7

The third file used by PEST is the control file, which is given an extension of .pst. This file defines the methods used by PEST in the parameter estimation, contains the initial and allowed values for the model input parameters, and contains the observed (or measured) outputs and weights.

2.0 Requirements

Requirements are characteristics and behaviors that a piece of software must possess to function adequately for its intended purpose. The Frames 2.0 Pest Integration (F2PEST) represents the ability in FRAMES 2.0 to perform a parameter estimation analysis across a set of environmental models for a single site. The F2PEST will be constructed in such a way that allows for multiple invocations of the PEST code within a single FRAMES 2.0 conceptual site model.

The F2PEST is a module that allows PEST to operate on a subset of FRAMES 2.0 modules that are linked together in the FRAMES conceptual site model diagram. Figure 2-1 shows a diagram and the F2PEST as part of that diagram. The environmental models (A1 and B1) shown could be Groundwater Modeling System[3] or other FRAMES 2.0 compatible environmental models. From a diagrammatic standpoint, PEST manipulates models A1 and B1 separately which means that separate PEST templates, instructions, and control files exist for both instances. The Multi-model Summarizer module shown in Figure 2-1 represents a capability to use the parameter estimation results (and other outputs from the models or uncertainty modules) to compute model comparison statistics and multi-model predictions.

The F2PEST will consume a number of modules in a way that provides inputs and outputs from the module. This means that the F2PEST needs to run as a system processor.

The main requirements of the F2PEST are:

1. Run as part of a FRAMES 2.0 simulation.
2. Consume any number of module outputs and inputs from models connected to the F2PEST module. These are the list of modules that will have their results analyzed and inputs manipulated.
3. Produce outputs that include the final objective function value, parameter estimates, the parameter estimation covariance matrix, observed values and weights, simulated values, residuals, standard error variance, and the model selection criteria (as computed by PEST)[4].
4. Accept the arguments of FRAMES 2.0 path, simulation name, and module name when invoked.
5. Run as a system level component of FRAMES 2.0
6. Have a user interface that gathers the inputs and outputs to be processed in a way that

[3] http://chl.erdc.usace.army.mil/gms

[4] The F2PEST code implemented reads additional PEST output: observed values and weights, simulated values, residuals, and the standard error variance. These are not currently produced as output by F2PEST.

allows a user to select any input or output value from any model that feeds data to the F2PEST module. This is similar to the behavior of the FRAMES Sensitivity and Uncertainty Module, which allows the user to choose a subset of inputs and outputs from all the connected models.

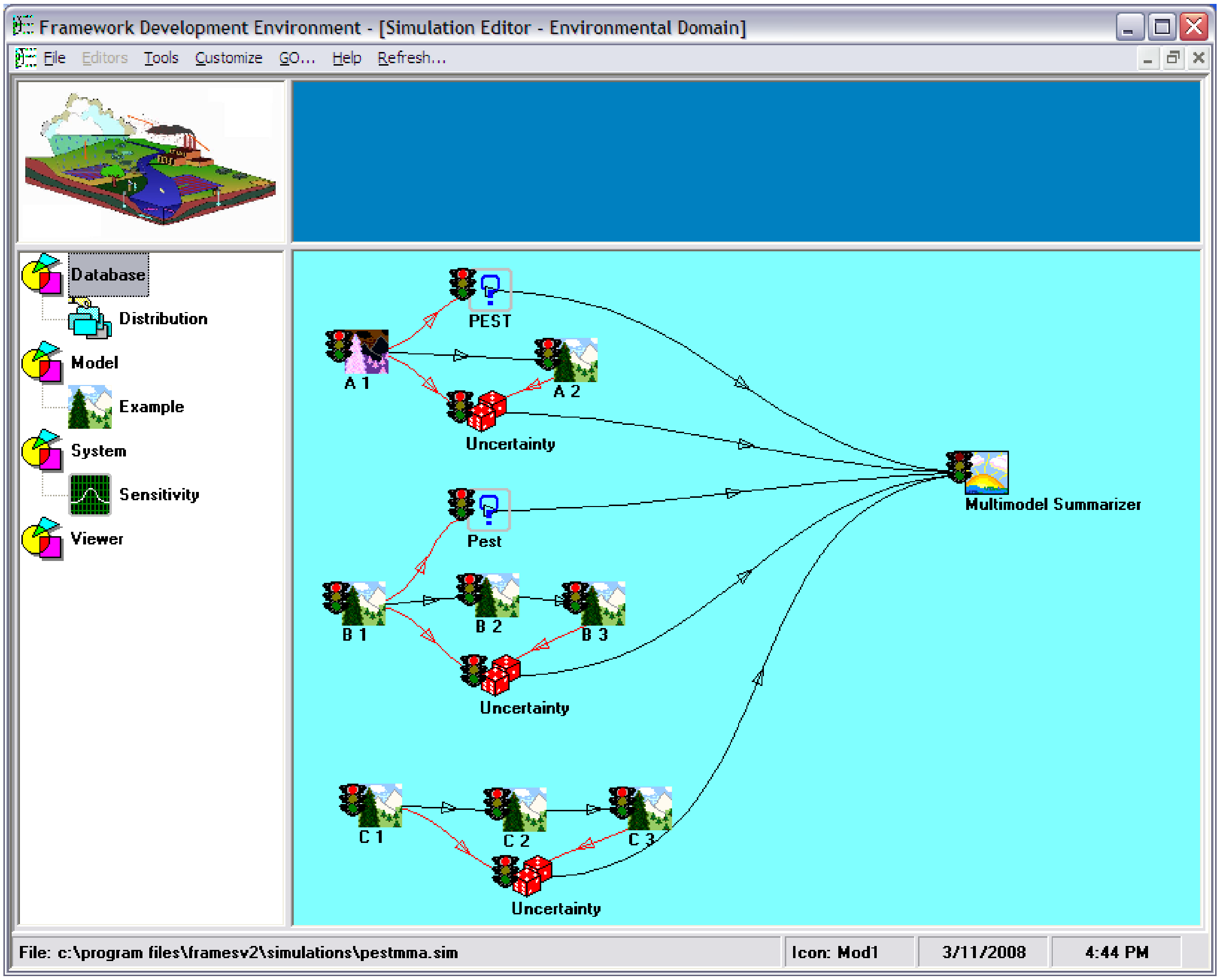


Figure 2-1. An example FRAMES 2.0 diagram that includes F2PEST modules

3.0 Design

Design elements are strategies for meeting requirements. The F2PEST is designed to meet the requirements identified in Section 2.0. Key to meeting those requirements is the interface between the FRAMES 2.0 datasets and PEST. The design for F2PEST is staged and straightforward. In the first stage the focus is to have PEST execute on a series of FRAMES modules. In Figure 2-1 PEST is shown to only connect to one module, but in general it needs to be able to connect to any number of modules. The effort put in to executing one model is just the same as for many because FRAMES 2.0 has a Run Between function that allows a system processor to execute all modules between two modules. If a single module is desired then Run Between is give the same module identifier for both the start and the end.

When the F2PEST interface is run, the user is prompted for the User Interface datasets and associated variables that need to be manipulated for the calibration. The user is also prompted for the Boundary Condition datasets (usually the result of computation) and associated variables that represent the observed values and model outputs to be used in the parameter estimation. Examples of the user interface screens for input of estimated parameters and observed values are shown in Figure 5-3 and Figure 5-4. Lastly a PEST control file is created that is initially populated for all the inputs and the observations when the user switches to the tab showing this information. An example of this screen is shown in Figure 5-5. With this approach, expert PEST users will find it very convenient to modify the control file (using a standard text editor) to have PEST perform exactly as they wish. A more novice PEST user can accept the default control parameters. For future development, a helpful application called a "Wizard" could be written that asks clear questions of the novice user and then generates a PEST control file with exactly what they need.

To implement F2PEST, three executables need to be created. The first is the F2PEST User Interface described above. The second is the PEST Pre/Post processor that runs PEST with the created PEST control file. This application reads the FRAMES 2.0 datasets and writes the PEST template file and an example text file with current values in appropriate locations. A PEST instruction file is also generated along with an example output file. So from PEST's view it is invoked with a template file, instruction file, and control file. When PEST invokes the "model" the last executable is invoked.

The last executable takes the input file, which has been updated by PEST, and updates the FRAMES 2.0 datasets. It then executes Run Between on all modules connected to the PEST icon. When Run Between completes successfully the output datasets are read and written to the output file, which is then read by PEST. Figure 3-1 is a block diagram of F2PEST integration components.

The location of PEST is assumed to be given in the PEST_PATH environment variable. All PEST files are written to the location specified in PEST_PATH.

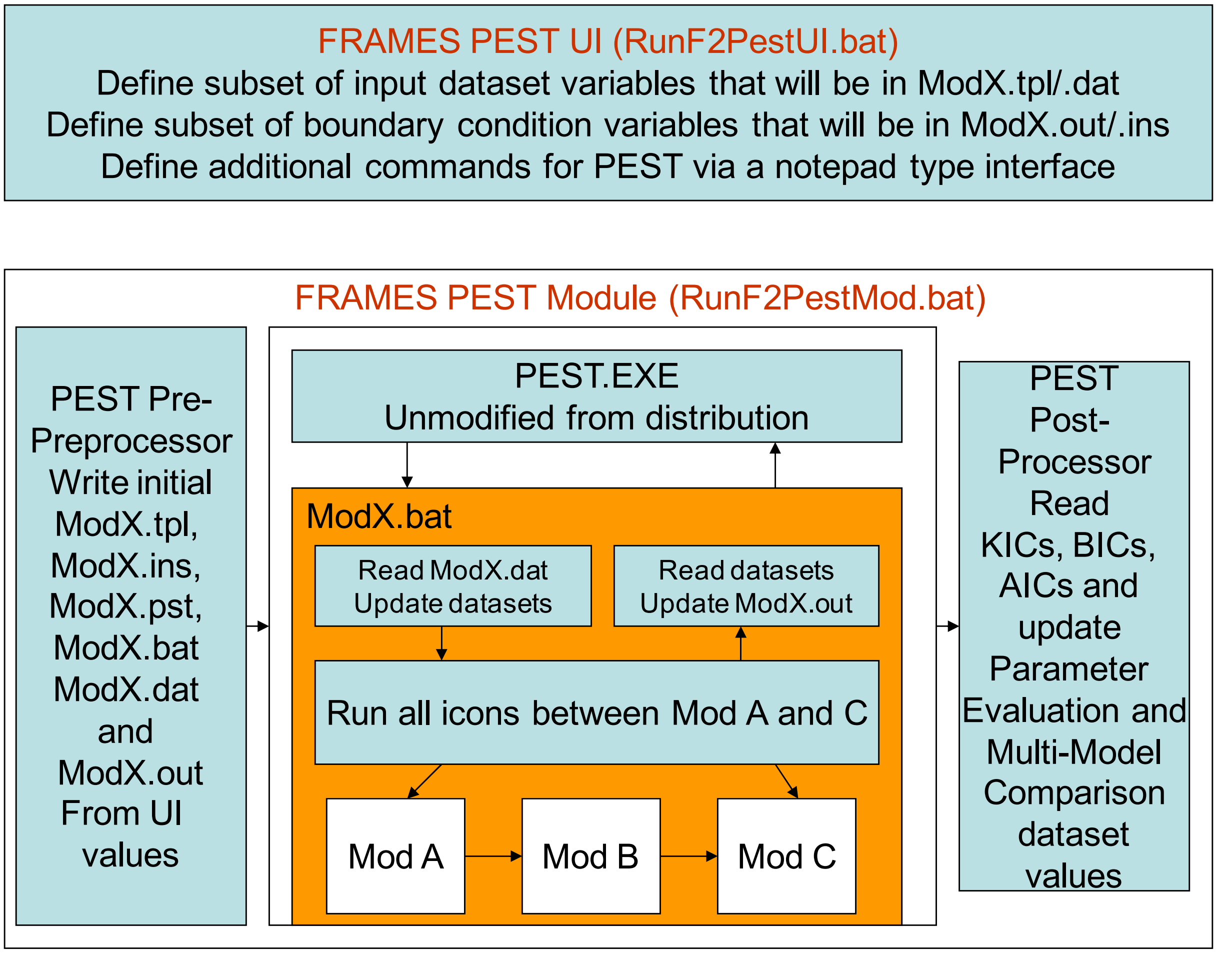

FRAMES PEST UI (RunF2PestUI.bat)
Define subset of input dataset variables that will be in ModX.tpl/.dat
Define subset of boundary condition variables that will be in ModX.out/.ins
Define additional commands for PEST via a notepad type interface
FRAMES PEST Module (RunF2PestMod.bat)
PEST Pre-Preprocessor Write initial ModX.tpl, ModX.ins, ModX.pst, ModX.bat ModX.dat and ModX.out From UI values
PEST.EXE
Unmodified from distribution
ModX.bat
Read ModX.dat
Update datasets
Read datasets
Update ModX.out
Run all icons between Mod A and C
Mod A
Mod B
Mod C
PEST Post-Processor Read KICs, BICs, AICs and update Parameter Evaluation and Multi-Model Comparison dataset values

Figure 3-1. F2PEST integration diagram

3.1 Limitations and Issues

F2PEST is built on top of the PEST executables 11.5. Any limitations in capability are expected limitations of F2PEST.

F2PEST in its initial phase of implementation does not intend to build a complete PEST user interface for PEST. Instead it intends to use defaults (i.e., recommended values in the PEST documentation) for some of the control parameters.

4.0 Specifications

Specifications are the descriptions of input and output file that are used during execution of a component. Alongside these input and output descriptions how the component is invoked and how the component was compiled are also discussed. If a virtual environment like Java is used for execution, the version of that environment is also a specification.

The F2PEST module reads any data set that is a module input or boundary condition used by subordinate modules. The data (.dat) and template (.tpl) files generated by F2PEST (discussed in the Introduction) have a simple format. Every alias specified in the user interface will be written to a line of the data file. The template file will contain the appropriate pound sign (#) framing described in the PEST documentation with the alias between the pound signs. There are always 12 character positions allocated for writing values to the data file. The data file is manipulated by PEST, the template file is read by PEST. See Table 1-1 for an example of a data file and template file pair. Note: that comments are not currently generated by F2PEST.

The formats used in the instruction (.ins) and the output (.out) file are similar to the data and template file. The output file contains one observed value on each line of the file. The instruction file also contains one line for every observed value in the output file and contains the starting 'pif' command. Table 1-2 is an example of the instruction and the output file pair. These files are read by PEST and follow format specified by PEST.

The index (idx1, idx2, .. idxN) columns seen in the F2PEST user interface (e.g., Figure 5-3 and Figure 5-4) allow a user to specify which subset of values are to be used in an analysis or manipulated. For parameters, the indices specify what values will be manipulated in a dataset. In the current implementation only a single value should be specified for a parameter. If a variable from a FRAMES 2.0 dataset that contains more than one dimension is specified as a parameter to be estimated, all indices should be specified with a single integer value to define which value in the collection of values is to be the parameter. If two values in a single variable need to be manipulated (i.e., a single variable but two different index specifications), two aliases are needed and the different indices for the two parameters need to be provided by the user. Similar logic is expected for observed values. Each alias should represent only one model output value even if the FRAMES 2.0 dataset variable has more than one dimension. The user should specify a set of indices that identify the single model output value that is to be compared to an observed value. If a single variable with different indices is needed for comparison with multiple observed values, then multiple aliases should be added, one for each specific set of indices/observed value.

F2PEST writes the ParameterEvaluation and MultiModelComparison dictionaries. The ParameterEvaluation dictionary contains the results of the PEST parameter estimation, such as the optimal parameter estimates and the parameter estimation covariance matrix. The MultiModelComparison dictionary contains the model comparison statistics computed by PEST. The contents and characteristics of the dictionaries produced by F2PEST are given in Table 4-1.

As noted above, the location of PEST is assumed to be given in the PEST_PATH environment variable. All PEST files are written to the location specified in PEST_PATH.

4.1 Command Line

The command line for the F2PEST is the standard FRAMES module command line. The module when invoked is passed the FRAMES 2.0 path, the Simulation Name, and the Module Name. These three pieces of information are required to connect to the FRAMES system with the SystemIO.dll. The FRAMES 2.0 path is assumed to be contained in the FRAMES_PATH environment variable.

4.2 Compiler Version

The user interface is compiled using Java. The project file is F2PESTUI.mak. The computational engine of the F2PEST is compiled using c++ compiler and gcc compiler version 3.4 or better. The project file for the computation component is F2PEST.mak

4.3 Java Runtime Environment Version

Java 1.3 or better is required to run the user interface.

Table 4-1. Dictionaries produced by F2PEST: ParameterEvaluation containing parameter estimation results and MultiModelComparison containing model comparison statistics

ParameterEvaluation

Name	Description	Unit	Measure	Type	Range	S	D	U	K	Prep	Indices
Alias	Alias of the parameter that is being reported			STRING	0-80	Y	1	N	N		
estimates	Estimate for the parameter			FLOAT	-1E+38-1E+38	N	1	N	N		Alias
Indices	Indices of the parameter being evaluated			STRING	0-80	Y	2	N	N		Alias
lowerLimit	95% lower limit of parameter estimate			FLOAT	-1E+38-1E+38	N	1	N	N		Alias
parameterCorrelation				FLOAT	-1E+38-1E+38	N	2	N	N		Alias
parameterCovariance				FLOAT	-1E+38-1E+38	N	2	N	N		Alias
Set	Dataset associated with the Alias			STRING	0-80	N	1	N	N		Alias
Type	Type of variables being analyzed			STRING	0-80	N	1	N	N		Alias
Units	Units of the variable being analyzed			STRING	0-80	N	1	N	N		Alias
upperLimit	95% confidence upper limit of the parameter values			FLOAT	-1E+38-1E+38	N	1	N	N		Alias
Variables	Variable names for the parameter being analyzed			STRING	0-80	N	1	N	N		Alias

MultiModelComparison

Name	Description	Unit	Measure	Type	Range	S	D	U	K	Prep	Indices
AIC	Akaike Information Criterion			FLOAT	-1E+38-1E+38	N	0	N	N		
AICC	Modified Akaike Information Criterion			FLOAT	-1E+38-1E+38	N	0	N	N		
AICCNotComputed	Modified Akaike Information Criterion available			LOGICAL	0-1	N	0	N	N		
BIC	Bayesian Information Criterion			FLOAT	-1E+38-1E+38	N	0	N	N		
KIC	Kashyap Information Criterion			FLOAT	-1E+38-1E+38	N	0	N	N		

Legend

Column Name	Meaning
S	Self-Indexed
D	Dimensional Size
U	Uncertainty can apply (Stochastic)
K	Is the variable a key to others

5.0 Testing Approach and Results

A test case was developed to test the performance of the F2PEST module integration with FRAMES. The test case uses a model of groundwater transport with observations derived from the true model output with the addition of a random error term. F2PEST was used with the observation values to estimate the parameters of the model.

The one-dimensional advection-dispersion equation for transport of a conservative solute in a uniform, steady flow field can be written as

$$D_l \frac{\partial^2 C}{\partial l^2} - v \frac{\partial C}{\partial l} = \frac{\partial C}{\partial t} \tag{1}$$

where C is the solute concentration, D_l is the dispersion coefficient along l, the direction of flow, v is the groundwater velocity, and t is time. For a semi-infinite domain with an initial concentration of zero and a step function source of $C = C_0$ at $l = 0$, the solution to Eq. 1 can be written as (Freeze and Cherry 1979)

$$\frac{C}{C_0} = \frac{1}{2}\left[erfc\left(\frac{l - vt}{2\sqrt{D_l t}} \right) + \exp\left(\frac{vl}{D_l} \right) erfc\left(\frac{l + vt}{2\sqrt{D_l t}} \right) \right] \tag{2}$$

where *erfc* is the complementary error function.

Eq. 2 was solved for concentration as a function of time at a distance of $l = 0.1$, using a source concentration of $C_0 = 1.0$ and parameters $v = 1.0\times10^{-2}$ and $D_l = 5.0\times10^{-4}$ (all quantities with arbitrary consistent units). Concentration observations for use in F2PEST were based on the solutions to Eq. 2 at times of 3, 5, 8, 12, 15, 17, and 20. Each concentration at these times was perturbed by an amount randomly sampled from a normal distribution with a mean of 0.0 and a standard deviation equal to 20 percent of the concentration. Calculated solution concentrations and randomly perturbed observed values are listed in Table 5-1. A plot of the calculated solution along with the perturbed observations is given as Figure 5-1. Note that the observed values are biased low compared to the calculated solution from which they were derived.

A FRAMES module implementing Eq. 2 as a model was written for the test case. For this test, the observations from Table 5-1 were used. C_0 was assumed to be known, while v and D_l were fit to the observations using F2PEST. The initial values used were $v = 2.0\times10^{-2}$ and $D_l = 1.0\times10^{-4}$. The F2PEST parameter estimation module was connected to the model in the FRAMES simulation editor, as shown in Figure 5-2. The F2PEST user interface contains three separate tabs, one for selecting model parameters and entering parameter information (such as initial values, upper and lower bounds, and transformations), one for selecting model output variables and entering observation values and weights, and a third tab for the PEST control file. The highlighted portion of the control file is not editable from the F2PEST UI. All control parameters are available for editing using a text editor on the PEST control file itself, however. The three tabs of the F2PEST UI are shown in Figure 5-3, Figure 5-4, and Figure 5-5.

Table 5-1. Solution concentrations from Eq. 2 and randomly perturbed observations for use in F2PEST

Time	Solution Concentration	Perturbed Observation
3	0.165727	0.071169
5	0.364976	0.367855
8	0.574724	0.427322
12	0.737246	0.706692
15	0.810768	0.777205
17	0.846108	0.870955
20	0.885475	0.739659

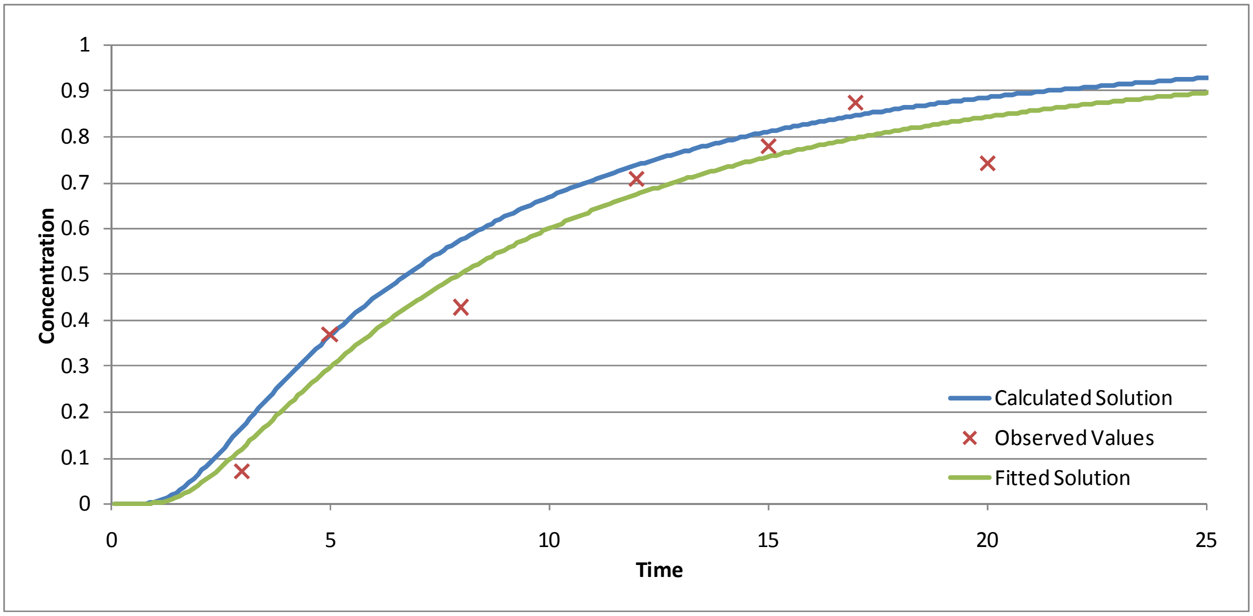
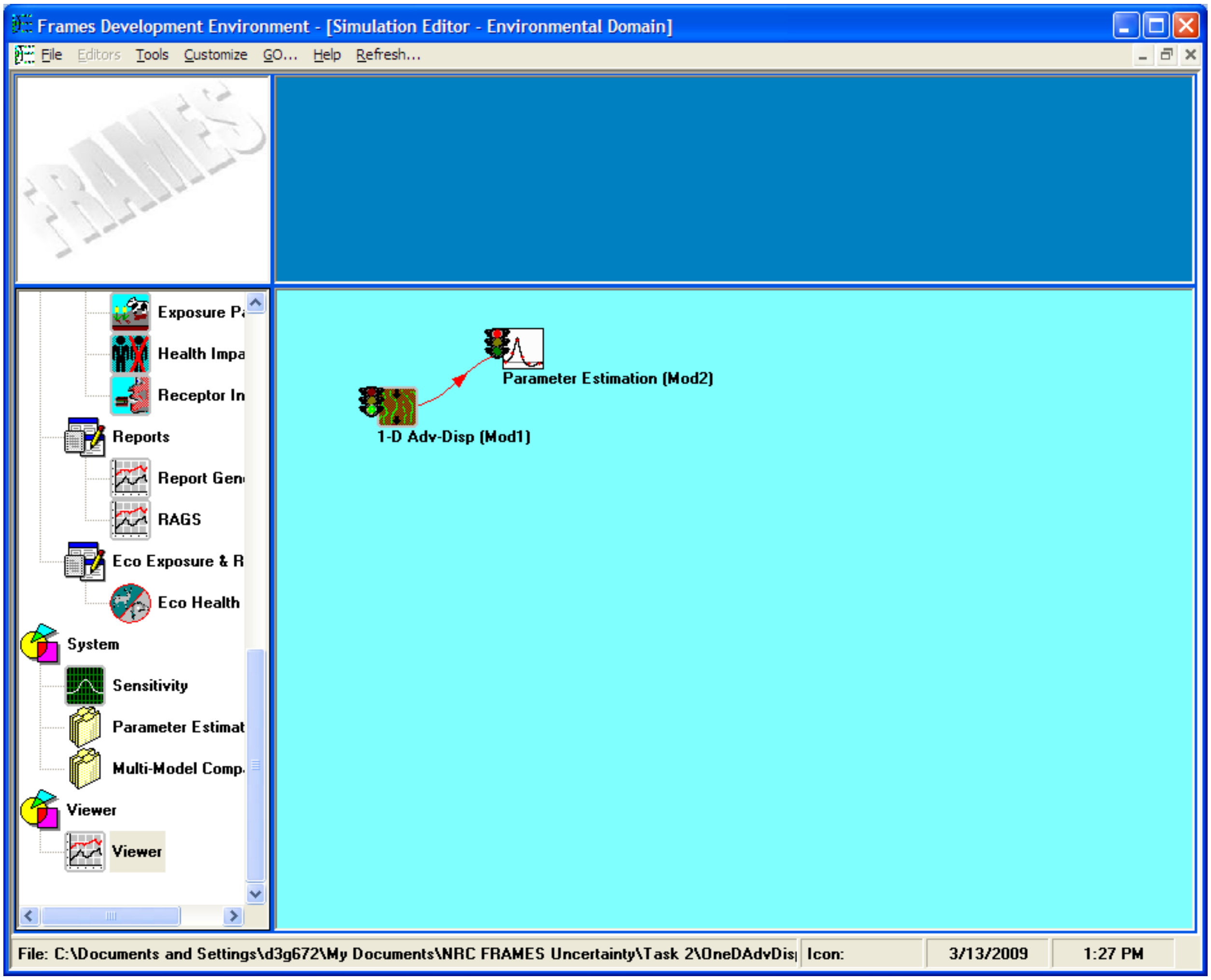


Figure 5-1. Calculated solution to Eq. 2 and randomly perturbed observation values used in F2PEST.

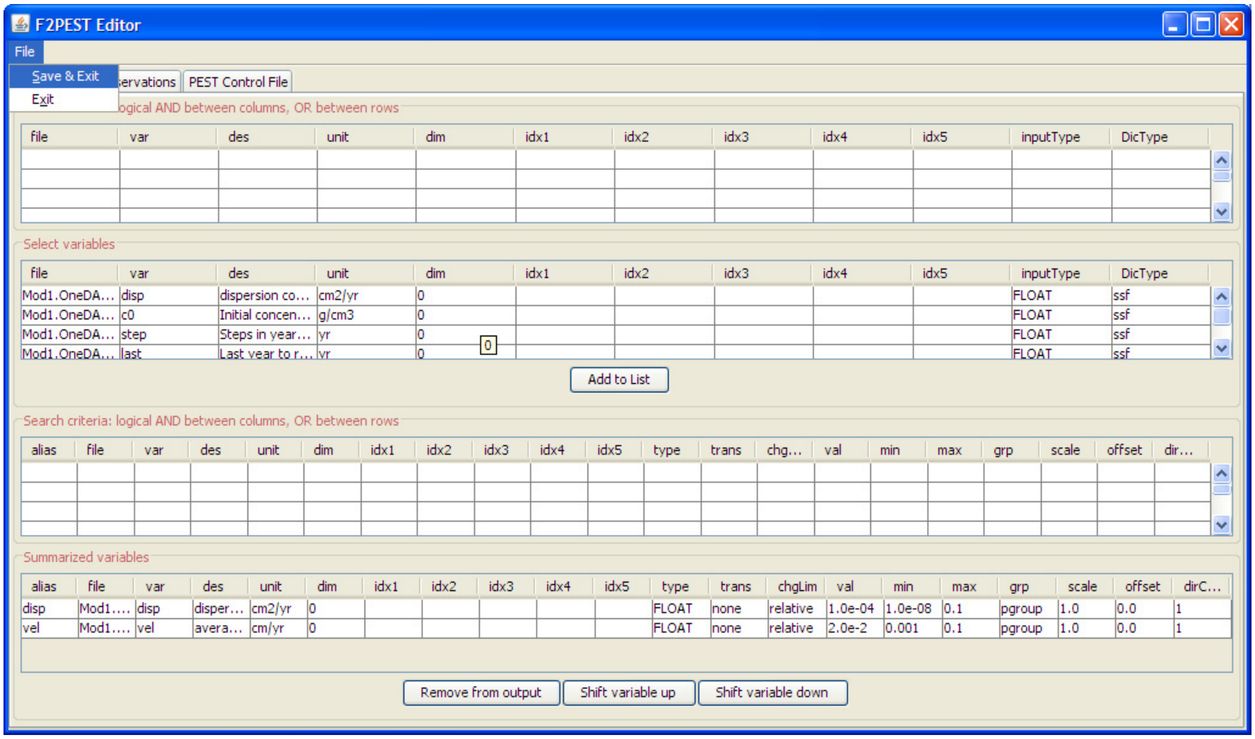
Execution of F2PEST for the test case produced the PEST .tpl, .ins, .dat, and .out files. The .dat file is the model input file containing the current iteration's values of the model parameters being estimated. The .tpl file is the template file instructing PEST on how to read and write to the .dat file. The .out file is the model output file containing the current iteration's values of the model-simulated observations. The .ins file instructs PEST on how to read the .out file. The final state of the files is shown in Table 5-2. The PEST history of execution and resulting solution is shown in the PEST .rec file (Appendix A). Final estimated parameter values were $v = 8.408 \times 10^{-3}$ and $D_l = 4.316 \times 10^{-4}$. The fitted concentration solution using these parameter values is shown in Figure 5-1.

The dictionaries produced by the F2PEST module for the test case are shown in Figure 5-6 through Figure 5-8. Values contained in the dictionaries are the same as those in the PEST .rec file demonstrating the correct execution of F2PEST for the test case.



File: C:\Documents and Settings\d3g672\My Documents\NRC FRAMES Uncertainty\Task 2\OneDAdvDis| Icon: 3/13/2009 1:27 PM

Figure 5-2. Simulation editor in FRAMES illustrating the F2PEST test case



ogical AND between columns, OR between rows

file	var	des	unit	dim	idx1	idx2	idx3	idx4	idx5	inputType	DicType

Select variables

file	var	des	unit	dim	idx1	idx2	idx3	idx4	idx5	inputType	DicType
Mod1.OneDA…	disp	dispersion co…	cm2/yr	0						FLOAT	ssf
Mod1.OneDA…	c0	Initial concen…	g/cm3	0						FLOAT	ssf
Mod1.OneDA…	step	Steps in year…	yr	0						FLOAT	ssf
Mod1.OneDA…	last	Last year to r…	yr	0						FLOAT	ssf

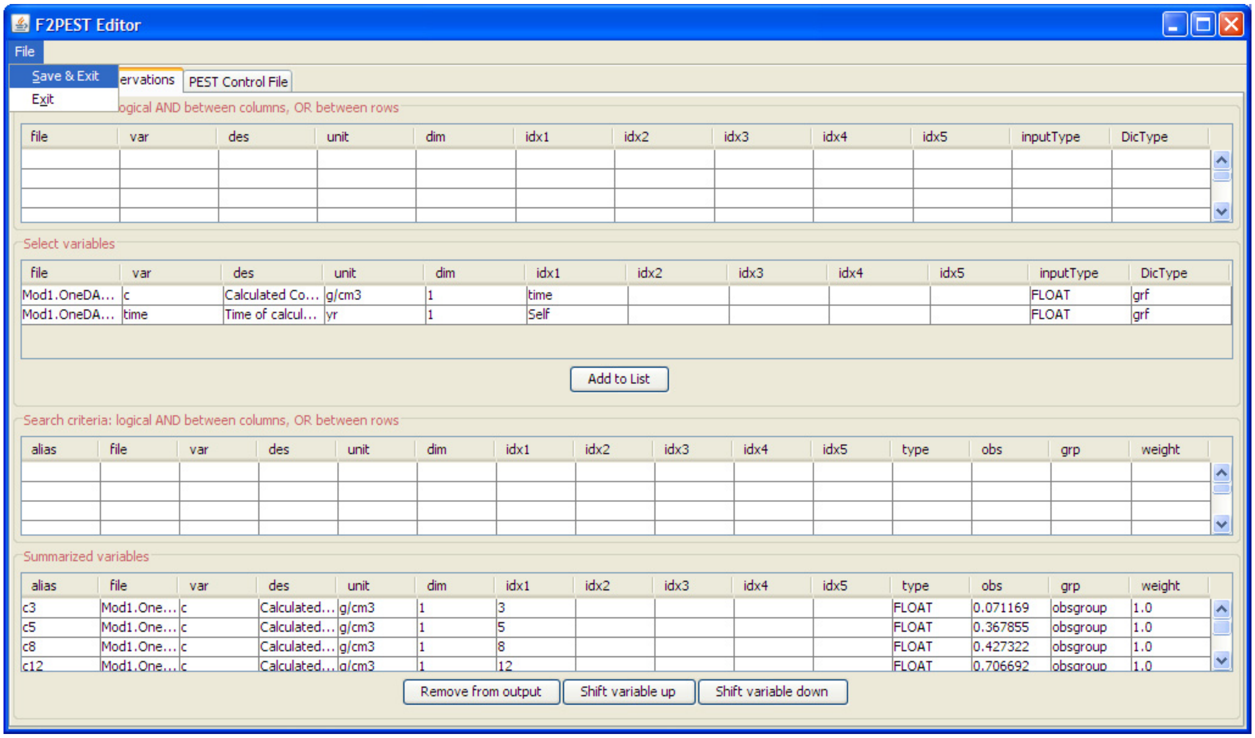
Search criteria: logical AND between columns, OR between rows

alias	file	var	des	unit	dim	idx1	idx2	idx3	idx4	idx5	type	trans	chg…	val	min	max	grp	scale	offset	dir…

Summarized variables

alias	file	var	des	unit	dim	idx1	idx2	idx3	idx4	idx5	type	trans	chgLim	val	min	max	grp	scale	offset	dirC…
disp	Mod1.…	disp	disper…	cm2/yr	0						FLOAT	none	relative	1.0e-04	1.0e-08	0.1	pgroup	1.0	0.0	1
vel	Mod1.…	vel	avera…	cm/yr	0						FLOAT	none	relative	2.0e-2	0.001	0.1	pgroup	1.0	0.0	1

Figure 5-3. Parameter selection tab from the F2PEST UI



ogical AND between columns, OR between rows

file	var	des	unit	dim	idx1	idx2	idx3	idx4	idx5	inputType	DicType

Select variables

file	var	des	unit	dim	idx1	idx2	idx3	idx4	idx5	inputType	DicType
Mod1.OneDA…	c	Calculated Co…	g/cm3	1	time					FLOAT	grf
Mod1.OneDA…	time	Time of calcul…	yr	1	Self					FLOAT	grf

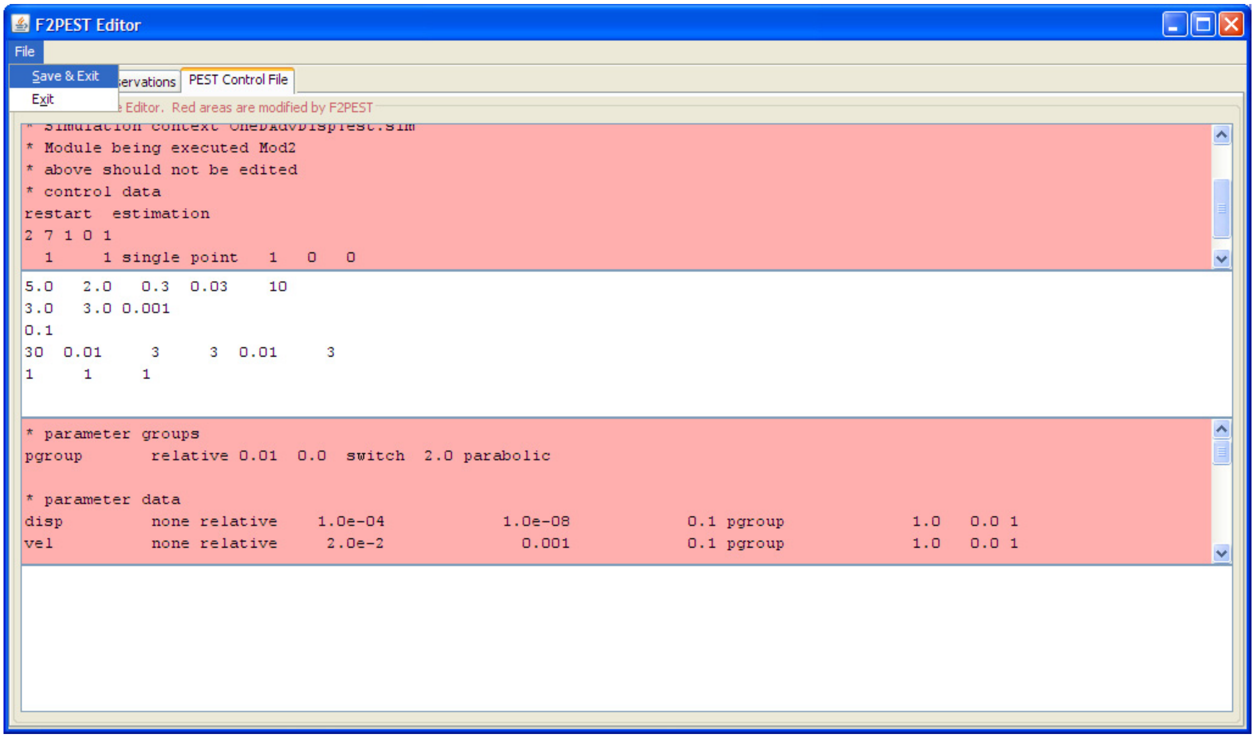
Search criteria: logical AND between columns, OR between rows

alias	file	var	des	unit	dim	idx1	idx2	idx3	idx4	idx5	type	obs	grp	weight

Summarized variables

alias	file	var	des	unit	dim	idx1	idx2	idx3	idx4	idx5	type	obs	grp	weight
c3	Mod1.One…	c	Calculated…	g/cm3	1	3					FLOAT	0.071169	obsgroup	1.0
c5	Mod1.One…	c	Calculated…	g/cm3	1	5					FLOAT	0.367855	obsgroup	1.0
c8	Mod1.One…	c	Calculated…	g/cm3	1	8					FLOAT	0.427322	obsgroup	1.0
c12	Mod1.One…	c	Calculated…	g/cm3	1	12					FLOAT	0.706692	obsgroup	1.0

Figure 5-4. Observation selection tab from the F2PEST UI



```
* Simulation context OneDAdvDispTest.sim
* Module being executed Mod2
* above should not be edited
* control data
restart  estimation
2 7 1 0 1
  1      1 single point   1   0   0
```

```
5.0   2.0   0.3  0.03    10
3.0   3.0 0.001
0.1
30  0.01     3     3  0.01     3
1     1     1
```

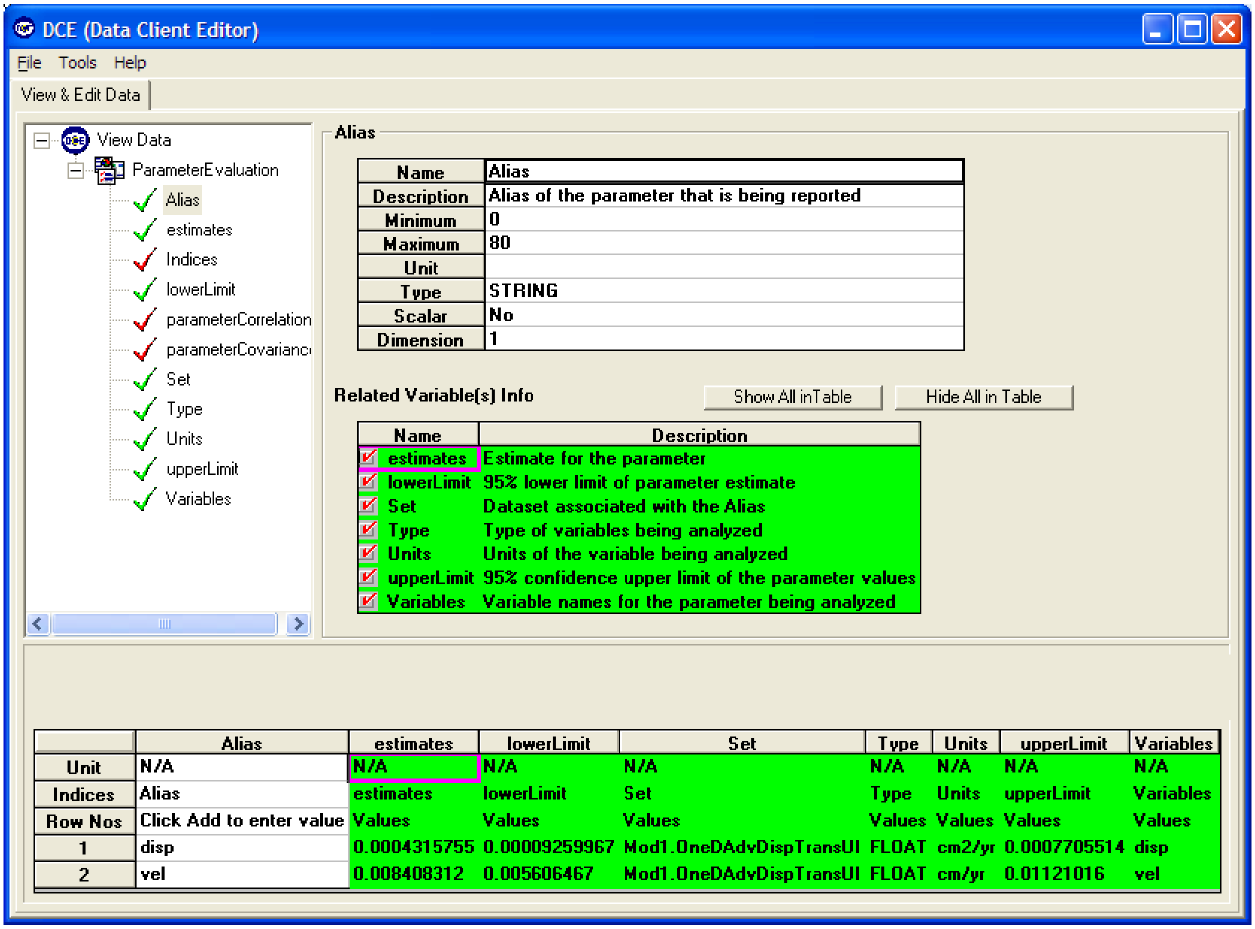
```
* parameter groups
pgroup       relative 0.01  0.0  switch  2.0 parabolic

* parameter data
disp         none relative    1.0e-04            1.0e-08           0.1 pgroup           1.0   0.0 1
vel          none relative     2.0e-2              0.001           0.1 pgroup           1.0   0.0 1
```

Figure 5-5. PEST control file tab from the F2PEST UI

Table 5-2. PEST files generated by F2PEST (final state)

.tpl	.dat	.ins	.out
ptf #	4.3157554E-04	pif #	0.11948322
# disp #	8.4083119E-03	l1 (c3)1:12	0.29619867
# vel #		l1 (c5)1:12	0.50162905
		l1 (c8)1:12	0.6732685
		l1 (c12)1:12	0.75530654
		l1 (c15)1:12	0.7960535
		l1 (c17)1:12	0.8427709
		l1 (c20)1:12	



Alias

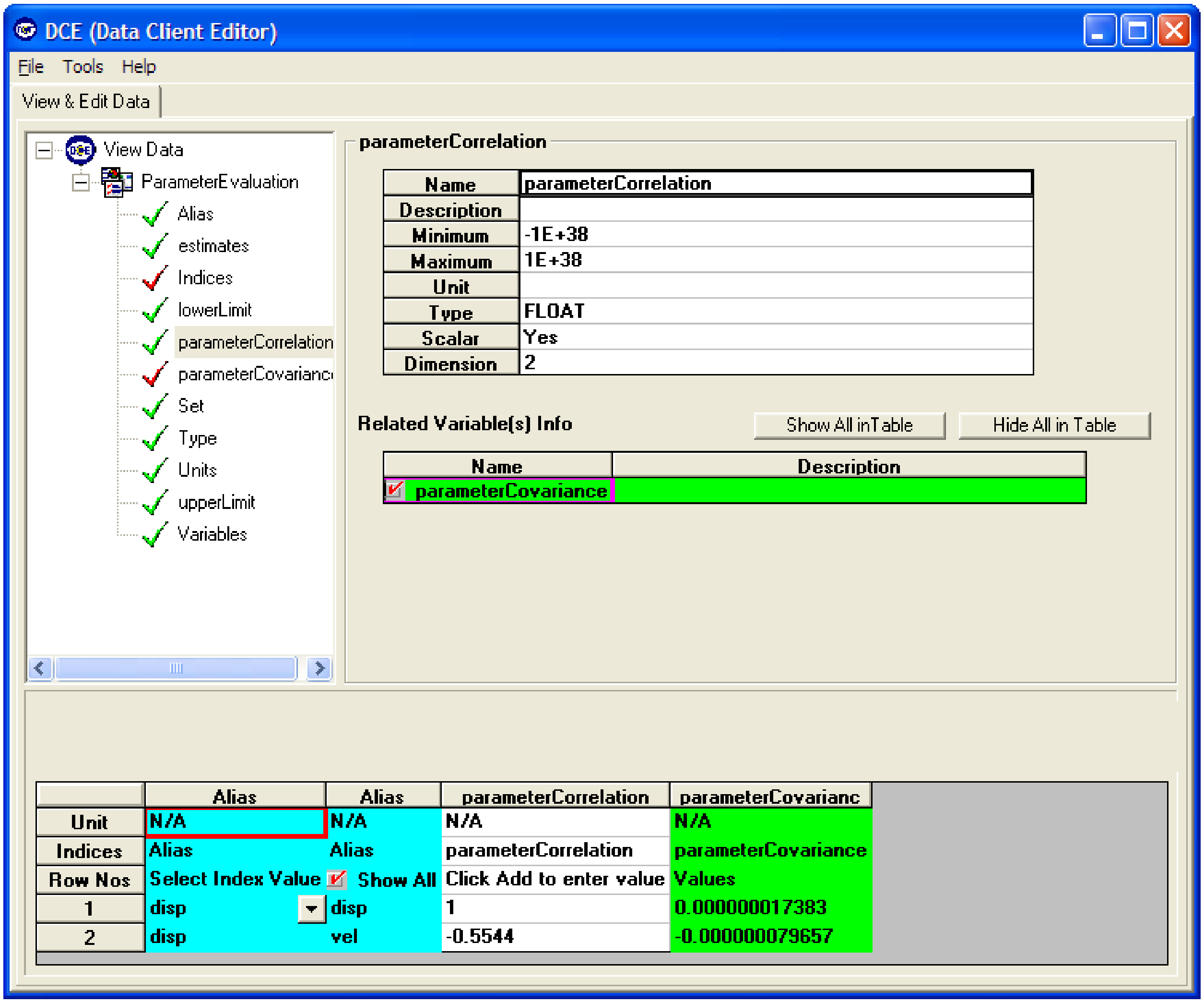
Name	Alias
Description	Alias of the parameter that is being reported
Minimum	0
Maximum	80
Unit	
Type	STRING
Scalar	No
Dimension	1

Related Variable(s) Info

Name	Description
☑ estimates	Estimate for the parameter
☑ lowerLimit	95% lower limit of parameter estimate
☑ Set	Dataset associated with the Alias
☑ Type	Type of variables being analyzed
☑ Units	Units of the variable being analyzed
☑ upperLimit	95% confidence upper limit of the parameter values
☑ Variables	Variable names for the parameter being analyzed

	Alias	estimates	lowerLimit	Set	Type	Units	upperLimit	Variables
Unit	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Indices	Alias	estimates	lowerLimit	Set	Type	Units	upperLimit	Variables
Row Nos	Click Add to enter value	Values	Values	Values	Values	Values	Values	Values
1	disp	0.0004315755	0.00009259967	Mod1.OneDAdvDispTransUI	FLOAT	cm2/yr	0.0007705514	disp
2	vel	0.008408312	0.005606467	Mod1.OneDAdvDispTransUI	FLOAT	cm/yr	0.01121016	vel

Figure 5-6. ParameterEvaluation dictionary produced by F2PEST including parameter estimates and 95% confidence limits

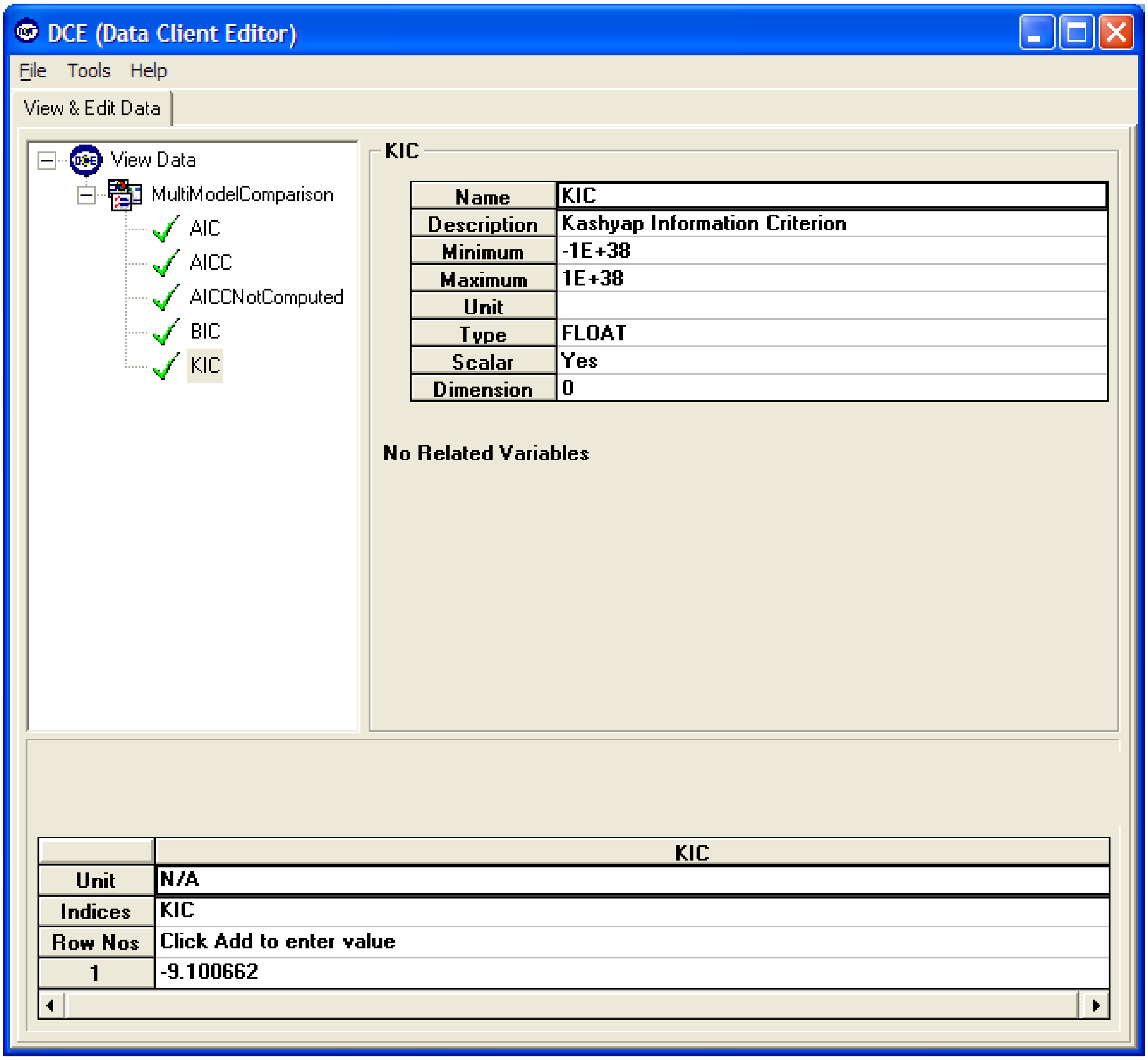


Name	parameterCorrelation
Description	
Minimum	-1E+38
Maximum	1E+38
Unit	
Type	FLOAT
Scalar	Yes
Dimension	2

Name	Description
☑ parameterCovariance	

	Alias	Alias	parameterCorrelation	parameterCovarianc
Unit	N/A	N/A	N/A	N/A
Indices	Alias	Alias	parameterCorrelation	parameterCovariance
Row Nos	Select Index Value	☑ Show All	Click Add to enter value	Values
1	disp ▾	disp	1	0.000000017383
2	disp	vel	-0.5544	-0.000000079657

Figure 5-7. ParameterEvaluation dictionary produced by F2PEST including parameter covariance and correlation matrices



KIC

Name	KIC
Description	Kashyap Information Criterion
Minimum	-1E+38
Maximum	1E+38
Unit	
Type	FLOAT
Scalar	Yes
Dimension	0

No Related Variables

	KIC
Unit	N/A
Indices	KIC
Row Nos	Click Add to enter value
1	-9.100662

Figure 5-8. MultiModelComparison dictionary produced by F2PEST

6.0 References

Doherty, J. 2004. *PEST, Model-Independent Parameter Estimation, User's Manual: 5th Edition*, Watermark Numerical Computing, Australia. http://www.sspa.com/pest/

Freeze, R.A. and J.A. Cherry. 1979. *Groundwater*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Meyer, P.D., M. Ye, M.L. Rockhold, S.P. Neuman, and K.J. Cantrell. 2007. *Combined Estimation of Hydrogeologic Conceptual Model, Parameter, and Scenario Uncertainty with Application to Uranium Transport at the Hanford Site 300 Area*, NUREG/CR-6940, U.S. Nuclear Regulatory Commission, Washington, DC.

APPENDIX A

PEST .rec File for Test Case

APPENDIX A

PEST .rec File for Test Case

```
                       PEST RUN RECORD: CASE mod2


PEST run mode:-

   Parameter estimation mode


Case dimensions:-

   Number of parameters                           :     2
   Number of adjustable parameters                :     2
   Number of parameter groups                     :     1
   Number of observations                         :     7
   Number of prior estimates                      :     0


Model command line(s):-

   Mod2.bat


Jacobian command line:-

   na


Model interface files:-

   Templates:
      Mod2.tpl
   for model input files:
      Mod2.dat

   (Parameter values written using single precision protocol.)
   (Decimal point always included.)

   Instruction files:
      Mod2.ins
   for reading model output files:
      Mod2.out


PEST-to-model message file:-
```

```
   na


Derivatives calculation:-

Param        Increment   Increment   Increment   Forward or   Multiplier  Method
group        type                    low bound   central      (central)   (central)
pgroup       relative    1.0000E-02   none        switch        2.000     parabolic


Parameter definitions:-

Name         Trans-            Change       Initial        Lower          Upper
             formation         limit        value          bound          bound
disp         none              relative   1.000000E-04   1.000000E-08   0.100000
vel          none              relative   2.000000E-02   1.000000E-03   0.100000

Name         Group          Scale         Offset        Model command number
disp         pgroup        1.00000        0.00000            1
vel          pgroup        1.00000        0.00000            1


Prior information:-

   No prior information supplied


Observations:-

Observation name        Observation       Weight       Group
 c3                     7.116900E-02      1.000       obsgroup
 c5                     0.367855          1.000       obsgroup
 c8                     0.427322          1.000       obsgroup
 c12                    0.706692          1.000       obsgroup
 c15                    0.777205          1.000       obsgroup
 c17                    0.870955          1.000       obsgroup
 c20                    0.739659          1.000       obsgroup


Control settings:-

   Initial lambda                                               :  5.0000
   Lambda adjustment factor                                     :  2.0000
   Sufficient new/old phi ratio per optimisation iteration      : 0.30000
   Limiting relative phi reduction between lambdas              : 3.00000E-02
   Maximum trial lambdas per iteration                          :  10

   Perform Broyden's update of Jacobian matrix                  : no

   Maximum  factor  parameter change (factor-limited changes)   :   na
   Maximum relative parameter change (relative-limited changes) :  3.0000
   Fraction of initial parameter values used in computing
   change limit for near-zero parameters                        : 1.00000E-03
   Allow bending of parameter upgrade vector                    : no
   Allow parameters to stick to their bounds                    : no
   Relative phi reduction below which to begin use of
   central derivatives                                           : 0.10000
   Iteration at which to first consider derivatives switch      :   1

   Relative phi reduction indicating convergence                 : 0.10000E-01
   Number of phi values required within this range               :   3
   Maximum number of consecutive failures to lower phi           :   3
   Minimal relative parameter change indicating convergence      : 0.10000E-01
   Number of consecutive iterations with minimal param change   :   3
   Maximum number of optimisation iterations                     :  30

   Attempt automatic user intervention                           : no

   Attempt reuse of parameter sensitivities                      : no


File saving options: -

   Save multiple JCO files                                       : no
   Save multiple REI files                                       : no



                            OPTIMISATION RECORD


INITIAL CONDITIONS:
   Sum of squared weighted residuals (ie phi) =  0.53293

      Current parameter values
      disp            1.000000E-04
      vel             2.000000E-02


OPTIMISATION ITERATION NO.        :    1
   Model calls so far             :    1
   Starting phi for this iteration:  0.53293

       Lambda =   5.0000     ----->
          Phi =  0.41251      (  0.774 of starting phi)

       Lambda =   2.5000     ----->
          Phi =  0.35163      (  0.660 of starting phi)

       Lambda =   1.2500     ----->
          Phi =  0.28363      (  0.532 of starting phi)

       Lambda =  0.62500     ----->
          Phi =  0.22508      (  0.422 of starting phi)

       Lambda =  0.31250     ----->
          Phi =  0.19501      (  0.366 of starting phi)

       Lambda =  0.15625     ----->
```

```
          Phi =  0.18072        (  0.339 of starting phi)

       Lambda =  7.81250E-02 ----->
          Phi =  0.17408        (  0.327 of starting phi)

       Lambda =  3.90625E-02 ----->
          Phi =  0.17095        (  0.321 of starting phi)

   No more lambdas: relative phi reduction between lambdas less than 0.0300
   Lowest phi this iteration:  0.17095

      Current parameter values                 Previous parameter values
      disp            3.575853E-04             disp            1.000000E-04
      vel             1.366885E-02             vel             2.000000E-02
   Maximum relative change:  2.576     ["disp"]


OPTIMISATION ITERATION NO.        :    2
   Model calls so far             :   11
   Starting phi for this iteration:  0.17095

       Lambda =  1.95313E-02 ----->
          Phi =  3.37609E-02  (  0.197 of starting phi)

   No more lambdas: phi is less than 0.3000 of starting phi
   Lowest phi this iteration:  3.37609E-02

      Current parameter values                 Previous parameter values
      disp            5.121448E-04             disp            3.575853E-04
      vel             7.636080E-03             vel             1.366885E-02
   Maximum relative change: 0.4414     ["vel"]


OPTIMISATION ITERATION NO.        :    3
   Model calls so far             :   14
   Starting phi for this iteration:  3.37609E-02

       Lambda =  9.76563E-03 ----->
          Phi =  3.08631E-02  (  0.914 of starting phi)

       Lambda =  4.88281E-03 ----->
          Phi =  3.08624E-02  (  0.914 of starting phi)

   No more lambdas: relative phi reduction between lambdas less than 0.0300
   Lowest phi this iteration:  3.08624E-02
   Relative phi reduction between optimisation iterations less than 0.1000
   Switch to central derivatives calculation

      Current parameter values                 Previous parameter values
      disp            4.223738E-04             disp            5.121448E-04
      vel             8.466615E-03             vel             7.636080E-03
   Maximum relative change: 0.1753     ["disp"]


OPTIMISATION ITERATION NO.        :    4
```

```
   Model calls so far             :   18
   Starting phi for this iteration:  3.08624E-02

       Lambda =  2.44141E-03 ----->
          Phi =  3.08296E-02  (  0.999 of starting phi)

       Lambda =  1.22070E-03 ----->
          Phi =  3.08296E-02  (  0.999 of starting phi)

       Lambda =  4.88281E-03 ----->
          Phi =  3.08296E-02  (  0.999 of starting phi)

   No more lambdas: phi rising
   Lowest phi this iteration:  3.08296E-02

      Current parameter values                 Previous parameter values
      disp            4.309279E-04             disp             4.223738E-04
      vel             8.414433E-03             vel              8.466615E-03
   Maximum relative change: 2.0252E-02 ["disp"]


OPTIMISATION ITERATION NO.        :    5
   Model calls so far             :   25
   Starting phi for this iteration:  3.08296E-02

       Lambda =  2.44141E-03 ----->
          Phi =  3.08294E-02  (  1.000 of starting phi)

       Lambda =  1.22070E-03 ----->
          Phi =  3.08294E-02  (  1.000 of starting phi)

       Lambda =  4.88281E-03 ----->
          Phi =  3.08294E-02  (  1.000 of starting phi)

   No more lambdas: phi rising
   Lowest phi this iteration:  3.08294E-02

      Current parameter values                 Previous parameter values
      disp            4.315755E-04             disp             4.309279E-04
      vel             8.408312E-03             vel              8.414433E-03
   Maximum relative change: 1.5028E-03 ["disp"]

   Optimisation complete: the  3 lowest phi's are within a relative distance
                          of eachother of 1.000E-02
   Total model calls:     32

   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.


                            OPTIMISATION RESULTS


Parameters ----->
```

```
Parameter        Estimated        95% percent confidence limits
                 value            lower limit       upper limit
 disp           4.315755E-04      9.259967E-05      7.705514E-04
 vel            8.408312E-03      5.606467E-03      1.121016E-02

Note: confidence limits provide only an indication of parameter uncertainty.
      They rely on a linearity assumption which  may not extend as far in
      parameter space as the confidence limits themselves - see PEST manual.

See file mod2.sen for parameter sensitivities.


Observations ----->

Observation            Measured       Calculated     Residual       Weight     Group
                       value          value
c3                    7.116900E-02   0.119483      -4.831422E-02    1.000      obsgroup
c5                    0.367855       0.296199       7.165633E-02    1.000      obsgroup
c8                    0.427322       0.501629      -7.430705E-02    1.000      obsgroup
c12                   0.706692       0.673269       3.342350E-02    1.000      obsgroup
c15                   0.777205       0.755307       2.189846E-02    1.000      obsgroup
c17                   0.870955       0.796054       7.490150E-02    1.000      obsgroup
c20                   0.739659       0.842771      -0.103112        1.000      obsgroup

See file mod2.res for more details of residuals in graph-ready format.

See file mod2.seo for composite observation sensitivities.


Objective function ----->

  Sum of squared weighted residuals (ie phi)                =  3.0829E-02


Correlation Coefficient ----->

  Correlation coefficient                                   =  0.9682


Analysis of residuals ----->

  All residuals:-
     Number of residuals with non-zero weight                       =     7
     Mean value of non-zero weighted residuals                      = -3.4076E-03
     Maximum weighted residual [observation "c17"]                  =  7.4902E-02
     Minimum weighted residual [observation "c20"]                  = -0.1031
     Standard variance of weighted residuals                        =  6.1659E-03
     Standard error of weighted residuals                           =  7.8523E-02

     Note: the above variance was obtained by dividing the objective
     function by the number of system degrees of freedom (ie. number of
     observations with non-zero weight plus number of prior information
     articles with non-zero weight minus the number of adjustable parameters.)
     If the degrees of freedom is negative the divisor becomes
     the number of observations with non-zero weight plus the number of
     prior information items with non-zero weight.


K-L information statistics ----->


  AIC   =  -31.97638
  AICC  =  -23.97638
  BIC   =  -32.13865
  KIC   =  -9.100662

Parameter covariance matrix ----->

                 disp         vel
disp           1.7383E-08  -7.9657E-08
vel           -7.9657E-08   1.1876E-06


Parameter correlation coefficient matrix ----->

                 disp         vel
disp            1.000      -0.5544
vel           -0.5544        1.000


Normalized eigenvectors of parameter covariance matrix ----->

              Vector_1     Vector_2
disp           0.9977      -6.7600E-02
vel            6.7600E-02   0.9977


Eigenvalues ----->

               1.1986E-08   1.1930E-06
```