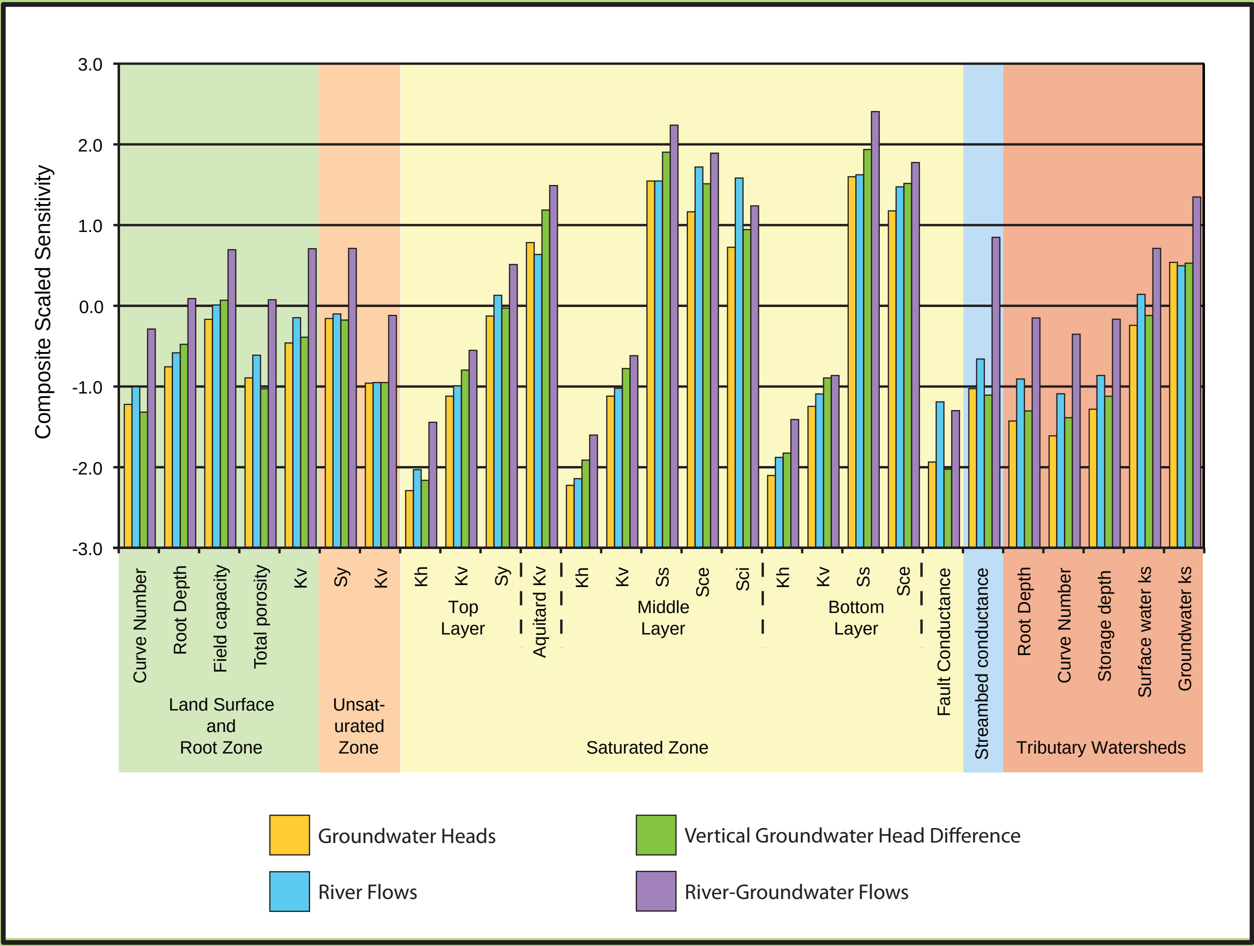
DWR Technical Memorandum

Programs to Support Calibration of Integrated Water Flow Model (IWFM) Using PEST

Documentation of PEST-IWFM Utilities

Charles F. Brush and Emin C. Dogrul



Bay-Delta Office, California Department of Water Resources, 1416 Ninth Street, Sacramento, CA 95814

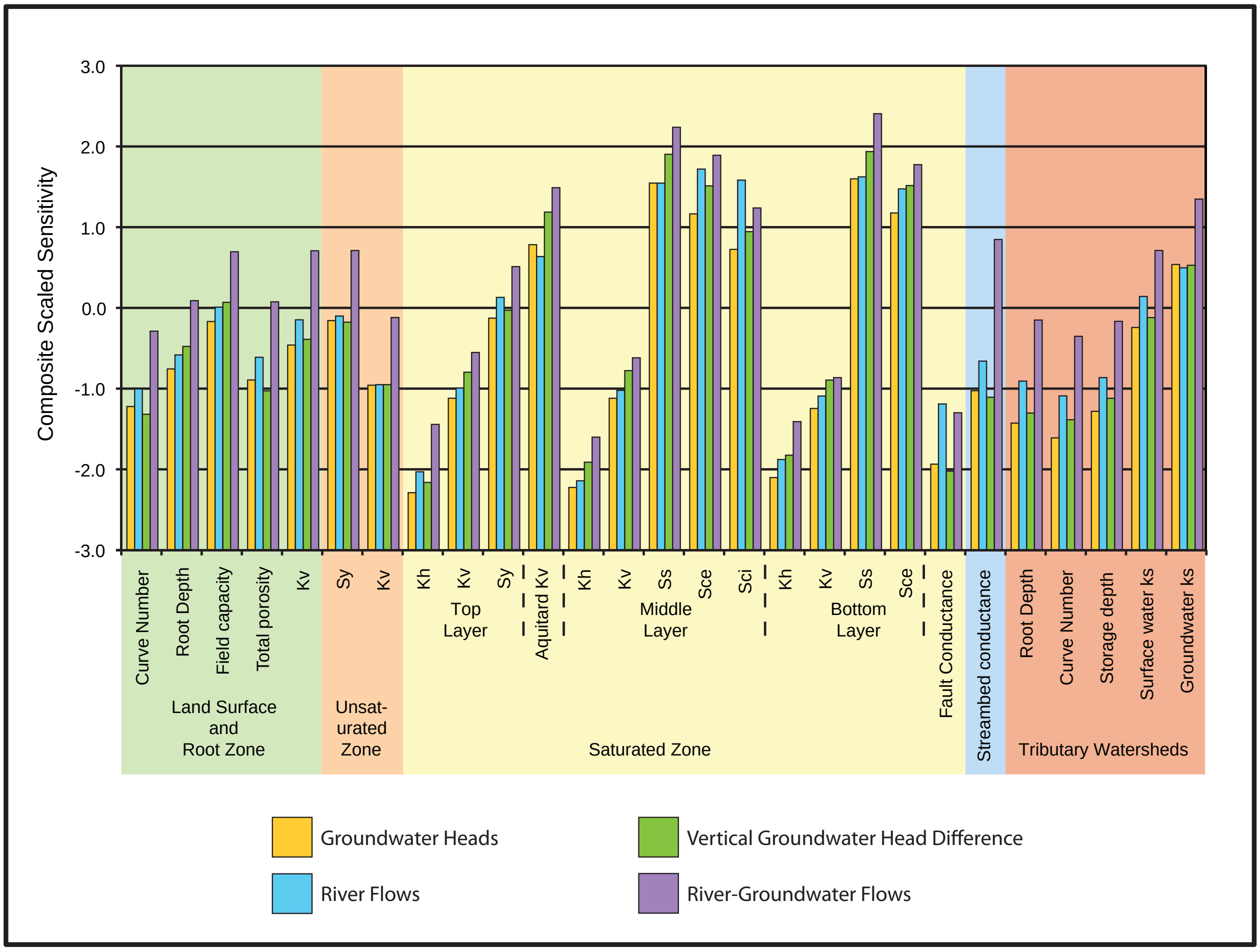
PEST-IWFM Tools for IWFM Version 3.02

DWR Technical Memorandum

Programs to Support Calibration of Integrated Water Flow Model (IWFM) Using PEST

Documentation of PEST-IWFM Utilities

Charles F. Brush and Emin C. Dogrul



Bay-Delta Office, California Department of Water Resources, 1416 Ninth Street, Sacramento, CA 95814

PEST-IWFM Tools for IWFM Version 3.02

DWR Technical Memorandum: Programs to Support Calibration of Integrated Water Flow Model (IWFM) Using PEST: Documentation of PEST-IWFM Utilities
Authors: Charles F. Brush and Emin C. Dogrul
Modeling software and documentation originated and maintained by the Bay-Delta Office, California Department of Water Resources, 1416 Ninth Street, Sacramento, CA 95814
http://baydeltaoffice.water.ca.gov/modeling/hydrology/iwfm/index_iwfm.cfm
This report describes PEST-IWFM tools for IWFM version 3.02.

Table of Contents

List of figures

Acronyms and Abbreviations

BUD	File name extension for IWFM Budget output files
CFS	Cubic Feet per Second
DWR	California Department of Water Resources
IWFM	Integrated Water Flow Model
ME	Mean Error
RMSE	Root Mean Squared Error
SMP	File name extension for PEST-compatible observation files

Introduction

The Integrated Water Flow Model (IWFM) application simulates water flow through the linked land surface, surface water and groundwater flow system. IWFM uses the land-use based approach to calculate water demands, and uses the Galerkin finite element method to calculate groundwater flow. IWFM also integrates the groundwater and surface water flow terms into a single matrix, allowing robust solutions and fast execution times. The IWFM application and supporting tools are developed by the Modeling Support Branch of the California Department of Water Resources (DWR) Bay-Delta Office. IWFM source code, execdutables and tools are released to the public under the GNU General Public License, and are freely available from the DWR web site.

Model Calibration

Model calibration is the process of modifying a model to improve its accuracy. This is generally the most challenging aspect of model development. Model calibration can involve modifying several types of components, including fixed features, fluxes, control data, and parameter values. Fixed features include for example river alignments and pump locations; fluxes include prescribed inflows, outflows, and evapotranspiration rates; and control data include for example the length of simulation time steps and the solution methods employed. Parameters describe physical properties of simulated components; examples of groundwater parameters include the horizontal hydraulic conductivity, vertical hydraulic conductivity, specific storage, specific yield, inelastic storage coefficient and elastic storage coefficient at each finite element node.

Model calibration is generally focused on selecting the set of model parameters that allow the model to best replicate historical observations. A typical IWFM model can have hundreds to thousands of parameters, and parameter optimization is a powerful model calibration method. Parameter optimization is an automated calibration method in which better and better sets of parameters are identified to improve the model's accuracy.

Inverse modeling is a very powerful parameter optimization method used in model calibration. Inverse modeling uses a statistical method, such as weighted least-squares, to quantify the difference between model results and historical observations for different sets of parameter values. This information is then used to choose a set of parameter values that improves model accuracy. Inverse modeling is commonly implemented through a set of computer programs that run the model many times with slightly different parameter values, calculate the objective function value after each run, and use this information to select a new set of parameter values that increases model accuracy.

Model calibration with PEST

PEST (Doherty, 2010) is a suite of computer programs that provides model-independent parameter optimization. The PEST software is available from the PEST web site (www.pesthome.org). PEST employs a unique approach to model calibration, inserting modified parameter values into model input files, running the model, reading values from the output files to calculate the objective function value, and using the information from many model runs to select new parameter values. The PEST software consists of a main **pest** program, and a large number of utility programs.

Figure 1. PEST flow chart for initiating the calibration of an IWFM model. Model calibration is accomplished in the final step, “Execute PEST”.

The PEST-IWFM Utilities

This document describes utility programs developed to facilitate the calibration of IWFM models with PEST. The tools described in this report work with IWFM version 3.02. The utilities were developed to be general enough to be applicable to any IWFM model. The PEST IWFM utilities were developed to achieve the following objectives:

- To support the use of the pilot point parameterization scheme for defining parameters in the IWFM model;
- To compile the parameter values calculated on the basis of pilot points, or any other scheme that outputs node-specific parameter values, into a form readable by IWFM;
- To support the assignment of Tikhonov style regularization between parameters defined on the basis of pilot points;
- To streamline the post-processing of simulated equivalents to the observations, simplifying file formats and the process required to add or remove observations from the calibration data set; and
- To log-transform surface water flow simulated equivalents.

To achieve these objectives, nine programs were developed – three based upon existing PEST utilities, one (REAL2IWFM) developed specifically for providing modified finite element parameter values to the IWFM application, three post-processors to facilitate data transfer from IWFM to PEST, and two programs to log-transform IWFM output for PEST. This document describes each of these programs, including the inputs required, program execution, and the outputs produced. A flow chart illustrates the sequence of processes that must be completed when using the programs to parameterize and calibrate an IWFM model application.

The PEST-IWFM utilities are written in FORTRAN and use FORTRAN-90 modules. The source code for each program is contained in a single file for portability, with the file extension "f90". The executables distributed with this documentation were compiled using the Intel Visual Fortran Compiler release 14 for Windows and Intel Compiler version 13 for Linux.

Acknowledgements

The original PEST-IWFM tools and documentation were developed by Steve Shultz of CH2MHill and Matt Tonkin of S.S. Papadopulos and Associates to work with IWFM version 2 (CH_2M Hilland S.S. Papadopulos, 2008). Several of these original tools were modified from PEST tools developed by John Doherty of Watermark Numerical Computing (Doherty, 2013a and 2013b).

PPK2FACI

Pilot points are used to calibrate parameters on a grid that is coarser than the model grid, significantly reducing the number of calibrated parameters. The parameter values must then be transferred from the pilot points to the model grid. This transfer can be accomplished using several geostatistical methods, including bilinear interpolation and kriging. **PPK2FACI** calculates and stores kriging factors for use in spatial interpolation from a set of pilot points to the nodes of an IWFM finite element grid, and calculates regularization information pertaining to the pilot point parameterization scheme. **PPK2FACI** processes pilot-point based parameter data into kriging factors and regularization information which are used by other programs in the regularization inverse model, including **PPKREGI** and **FAC2REALI** (Figure 1).

PPK2FACI uses kriging factors based on user-supplied (possibly nested) variograms, each with an arbitrary magnitude and direction of anisotropy. Different variograms can be used for spatial interpolation in different parts of the model domain, identified according to a Zone Specification File (described in Appendix B).

PPK2FACI is based on the PEST utility **PPK2FAC** (Doherty, 2013b), which is compatible with the MODFLOW and MT3D finite-difference models. A complete description of **PPK2FAC** is available in the PEST Groundwater Data Utilities documentation, Part B: Program Descriptions (Doherty, 2013b).

PPK2FACI Inputs

The **PPK2FACI** program requires four input files. These files are listed beow in the order the program requests them.

- Pilot Point File - contains the location and zone designation of pilot points: typically one file for each model layer. More detailed information is available in Doherty (2013b).
- Node Specification File – contains the X,Y locations for the IWFM model nodes. This file is a standard IWFM Preprocessor input file.
- Structure File – contains the definition(s) of the variogram(s) used for interpolation of the pilot point values to the IWFM model nodes. More detailed information is available in Doherty (2013b).
- Zone Specification File – contains a list of the IWFM nodes, together with an integer designation for each node. This file facilitates the use of pilot points together with zones in the definition of parameter distributions. An example of this file format is provided in Appendix B.

PPK2FACI Execution

PPK2FACI is executed at the command prompt by typing:

```
ppk2faci
```

The program prompts the user for a series of inputs, including the names of the input files (described above), the names of the output files (described below), and prompts for the variogram and the definition of parameter zones in the model. The responses to these prompts are usually provided with a list or piping file. Figure 2 shows a simple example where **PPK2FACI** calculates kriging factors from a grid with 398 pilot points and one zone to an IWFM finite element grid with 1392 nodes.

```
>ppk2faci

 Program PPK2FACI calculates point-to-cell factors by which kriging is
   undertaken from a set of pilot points to an IWFM finite-element mesh.

 Enter name of pilot points file: pp-file.dat
  - data for 398 pilot points read from pilot points file pp-file.dat

 Enter minimum allowable points separation: 1000

 Enter name of IWFM nodel X-Y coordinate file: IWFMnodes.dat
 Enter name of structure file: structure.dat
 Enter name of ASCII label file defining zonation: IWFMzones.dat

 The following zones have been detected in the ASCII zone definition file or determined from
defaults:-

    For zone characterised by integer value of 1:-
    Enter structure name (blank if no interpolation for this zone): struct1
    Perform simple or ordinary kriging [s/o]: o
    Enter search radius: 10000000
    Enter minimum number of pilot points to use for interpolation: 5
    Enter maximum number of pilot points to use for interpolation: 100

 Enter name for interpolation factor output file: factors.dat
 Is this a formatted or unformatted file? [f/u]: f
 Enter name for regularisation information output file: regul.dat

 Carrying out interpolation for mesh zone 1....

   Number of pilot points for this zone     =   398
   Mean data value for these pilot points   =   1313.6
   Data standard deviation for these points =   427.54
   Working....
   No. of grid points for which factors were calculated =  1393

  - kriging factors written to file factors.dat
  - regularisation information written to file regul.dat

>
```

Figure 2. Sample inputs (white) and outputs (black) for program ppk2faci.

PPK2FACI Outputs

The following output files are generated by **PPK2FACI**.

- Kriging Factor File – lists the pilot points, followed by the kriging factors (weights) that pertain to each of the IWFM model nodes, in a format that can be read by the program **FAC2REALI** (described below). This file is analogous to the file produced when using the standard PEST utilities described by Doherty (2013b), except it lists the number of nodes rather than the number of rows and columns.
- Regularization Information File – lists the regularization information pertaining to a pilot point parameterization scheme, in a format that can be read by the program **PPKREGI** (described below). This file is analogous to the file produced when using the MODFLOW/MT3D PEST utilities described by Doherty (2013b).

The Regularization Information File is used to introduce geostatistically-based regularization constraints to a parameter estimation problem. Generation of IWFM input data based on **PPK2FACI**-generated kriging factors is carried out by the program **FAC2REALI** (described below), and the regularization information recorded by **PPK2FACI** is processed by the program **PPKREGI** (described below) and appended to an existing PEST control file.

PPKREGI

PPKREGI incorporates regularization information for parameters described with pilot points into an existing PEST Control File. The regularization information is added to the PEST Control File in the form of prior information equations. The weights for these equations are typically determined using a variogram specified for the pilot points associated with each parameter type. The geostatistical information underlying the regularization is typically produced by the program **PPK2FACI**. **PPKREGI** processes the regularization information produced by **PPK2FACI**, and creates a new regularization PEST Control File.

PPKREGI is based on the PEST utility **PPKREG** (Doherty, 2013b), which serves the same function for the MODFLOW and MT3D finite-difference models. A complete description of **PPKREG** is available in the PEST Groundwater Data Utilities documentation, Part B: Program Descriptions (Doherty, 2013b).

PPKREGI Inputs

The **PPKREGI** program uses two input files. These files are listed below in the order the program requests them.

- PEST Control File – contains information that defines a model parameter estimation problem for PEST. The PEST control file may already contain regularization for other parameters, in which case the additional regularization will be appended to the end of the listed regularization. The PEST Control File contents are described in the PEST manual (Doherty, 2010).
- Regularization Information File - lists the regularization information pertaining to a pilot point parameterization scheme, usually produced by the program **PPK2FACI**. This file is described by Doherty (2013b).

PPKREGI Execution

PPKREGI is executed at the command prompt by typing:

```
ppkregi
```

The program then prompts the modeler for a series of inputs, including the names of the input files, the name of the output file, and several prompts regarding the regularization type and values required for defining the regularization problem. The requested information can be provided with a list or piping file. Figure 3 shows a simple example where **PPKREGI** adds regularization for a single parameter type into an existing PEST control file.

PPKREGI Outputs

The following output files are generated by **PPKREGI**.

- Replacement PEST Control File – the replacement PEST control file includes the contents of the original PEST control file, plus:
  - o Incremented counters at the top of the PEST control file to account for additional information,
  - o Regularization equations for the pilot point parameters, appended in the Prior Information section, and
  - o A regularization section at the end of the PEST control file that lists variables controlling the regularized inversion.

```
>ppkregi

 Program PPKREGI adds a prior information and regularisation section to a PEST
   control file whose parameters are based on pilot points. Regularisation
   weights are assigned based on information contained within a
   PPK2FACI-generated regularisation information file.

 Enter name of PEST control file: pest_control_file.pst
 Enter name of regularisation information file: regul.dat

 How many pilot-point-based parameter families in PEST control file
 pertain to this regularisation information file: 1

 For family number 1:-
   Enter parameter prefix (<Enter> if none):
   Apply uniform or geostatistical regularisation? [u/g]: g
   Enter weight multiplier: 1
   Enter new regularisation group name: regul_param
   Enter root name for new prior information: param

 Enter name for new PEST control file: param_regul.pst
 Enter target measurement objective function PHIMLIM: 10
 Enter initial regularisation weight factor WFINIT: 1
 Enter min. reg. weight factor WFMIN ( <Enter> if 1.0000E-10): 1e-10
 Enter max. reg. weight factor WFMAX ( <Enter> if 1.0000E+10): 1e+10

  - file regul.dat written ok.

>
```

Figure 3. Sample inputs (white) and outputs (black) for program ppkregi.

FAC2REALI

FAC2REALI uses kriging factors (weights) calculated with **PPK2FACI** to interpolate parameter values at pilot points to generate parameter values at IWFM model nodes. **FAC2REALI** is called by the batch file or shell script listed in the PEST control file, and is executed by PEST each time the simulation model is executed (figure 1). This section describes the inputs and outputs for **FAC2REALI**.

FAC2REALI is based on the PEST utility **FAC2REAL**, which is compatible with the MODFLOW and MT3D finite-difference models. A complete description of the **FAC2REAL** Model Framework is available in the PEST Groundwater Data Utilities documentation, Part B: Program Descriptions (Doherty, 2013b).

FAC2REALI Inputs

The **FAC2REALI** requires two input files. These files are listed below in the order the program requests them.

- Regularization Information File – produced by the program **PPK2FACI**, this file lists the regularization information for to a pilot point parameterization scheme. This file is analogous to the file produced when using the MODFLOW/MT3D PEST utilities described by Doherty (2013b).
- Pilot Point File - contains the location and zone designation of pilot points, typically one file for each model layer. More detailed description is given in Doherty (2013b).

FAC2REALI Execution

FAC2REALI is executed at the command prompt by typing:

```
fac2reali
```

The program prompts the modeler for a series of inputs, including the names of the input files, the name of the output file, and several other items including the interpolation limits and defaults for nodes that are not adjusted by the interpolation process. These prompts areusually provided in a list or piping file. A simple example is given below for a case where **FAC2REALI** calculates parameter values based on kriging factors produced by **PPK2FACI**.

FAC2REALI Outputs

FAC2REALI generates one output file.

- Node Property File – this file lists the parameter value for each node, created by interpolating from the values at the pilot points using information created using **PPK2FACI**. The contents of one or more Node Property Files are concatenated into a single IWFM Parameter Overwrite File by the program **REAL2IWFM** (described below).

```
>fac2reali

 Program FAC2REALI carries out spatial interpolation to IWFM v.3 node
   locations based on interpolation factors calculated by PPK2FACI and pilot
   point values contained in a pilot points file.

 Enter name of interpolation factor file: factors_pkh_l1.dat
 Is this a formatted or unformatted file? [f/u]: f

 Enter name of pilot points file [PP-PKH-L1.dat]:
  - data for 398 pilot points read from pilot points file PP-PKH-L1.dat
 Enter lower interpolation limit: 1e-10
 Enter upper interpolation limit: 1e+10
 Enter name for new node property file: kh-l1-nodes.out
 Enter value for elements to which no interpolation takes place: -999
 - file kh-l1-nodes.out written ok.

>
```

Figure 4. Sample inputs (white) and outputs (black) for program fac2reali.

REAL2IWFM

REAL2IWFM processes the Node Property Files produced by **FAC2REALI** using kriging factors produced by **PPK2FACI**. **REAL2IWFM** concatenates several Node Property Files into a single file, the IWFM Parameter Overwrite File (unit 30 in the IWFM Simulation Main Input File). The Parameter Overwrite File was developed to simplify incorporating node-specific parameter values into the IWFM simulation. This provides capabilities analogous to the pilot point implementation of spatial parameters when PEST is implemented with MODFLOW. **REAL2IWFM** is executed in the batch file or shell script listed in the PEST Control File; it is executed each time the simulation model is executed by PEST (Figure 1).

During an inverse model run, **REAL2IWFM** can read a template IWFM Parameter Overwrite File, and modify entries in that file by inserting updated values interpolated from the values at the pilot points. A value of -1 can be used in the original Parameter Overwrite File to indicate specific node/parameter combinations that should not be updated. When **REAL2IWFM** encounters a value of -1 it retains the value in the origianl Parameter Overwrite File.

REAL2IWFM Inputs

The following input files are required to execute **REAL2IWFM**. The files are listed in the same sequence as the program prompts the modeler.

- Template Parameter Overwrite File – only if **REAL2IWFM** is modifying an existing Parameter Overwrite File. An example is provided in Appendix B.
- Node Property File – one for each parameter type for each model layer that is to be modified through an Parameter Overwrite File. These files (one for each parameter type) are produced by the program **PPK2FACI** (described above).

REAL2IWFM Execution

REAL2IWFM is executed at the command prompt by typing:

```
real2iwfm
```

The program prompts the modeler for a series of inputs, including the names of the input files, the name of the output file, and several prompts to identify the model layers the parameters will be assigned. These prompts are usually provided in a list or piping file. A simple example is given below where **REAL2IWFM** creates a new Parameter Overwrite File from scratch, and inserts parameter values at nodes for multiple parameter types, with varying by-layer responses.

REAL2IWFM Outputs

REAL2IWFM always creates one new output file:

- Parameter Overwrite File – lists the nodes of the IWFM model and provides parameter values for those nodes and parameter types. These values replace parameter values listed in the original IWFM Parameter Input File. For all nodes where the input Template Parameter Overwrite File has a value of

```
>real2iwfm

 Program REAL2IWFM reads one or more output files from FAC2REALI and writes
   an IWFM Overwrite File that can be read by IWFM.


 Will REAL2IWFM create a new overwrite file from scratch? n

 Enter name of existing overwrite file:  IWFMoverwrite.tpl
 Enter name of new overwrite file:  IWFMoverwrite.dat
 How many layers in the IWFM model application?:  3
 How many nodes-per-layer in the IWFM model application? 1393
 Enter the time step units for KH, KV and KL in the IWFM model application (HEC-DSS format):
1mon
     Read 4179 items of replacement parameter data in file CVOVERWRITEIWFM.TPL


 Include data for parameter type PKH? y

Parameter value file for parameter type PKH for layer 1
  (Enter none if no overwrite for PKH for layer 1   ): pkh-l1-nodes.out
  Read values for 1393 nodes from file PKH-L1-NODES.OUT
Parameter value file for parameter type PKH for layer 2
  (Enter none if no overwrite for PKH for layer 2   ): pkh-l2-nodes.out
  Read values for 1393 nodes from file PKH-L2-NODES.OUT
Parameter value file for parameter type PKH for layer 3
  (Enter none if no overwrite for PKH for layer 3   ): pkh-l3-nodes.out
  Read values for 1393 nodes from file PKH-L3-NODES.OUT

 Include data for parameter type PS? y

Parameter value file for parameter type PS for layer 1
  (Enter none if no overwrite for PS for layer 1   ): ps-l1-nodes.out
  Read values for 1393 nodes from file PS-L1-NODES.OUT
Parameter value file for parameter type PS for layer 2
  (Enter none if no overwrite for PS for layer 2   ): ps-l2-nodes.out
  Read values for 1393 nodes from file PS-L2-NODES.OUT
Parameter value file for parameter type PS for layer 3
  (Enter none if no overwrite for PS for layer 3   ): ps-l3-nodes.out
  Read values for 1393 nodes from file PS-L3-NODES.OUT

 Include data for parameter type PN? y

Parameter value file for parameter type PN for layer 1
  (Enter none if no overwrite for PN for layer 1   ): pn-l1-nodes.out
  Read values for 1393 nodes from file PN-L1-NODES.OUT
```

Figure 5. Sample inputs (white) and outputs (black) for program real2iwfm.

–1, Parameter Overwrite File also contains a value of -1. When the IWFM Simulation program encounters a value of -1 in the Parameter Overwrite File, it uses the parameter value in the standard IWFM Parameter Input File.

```
Parameter value file for parameter type PN for layer 2
  (Enter none if no overwrite for PN for layer 2   ): pn-l2-nodes.out
  Read values for 1393 nodes from file PN-L2-NODES.OUT
Parameter value file for parameter type PN for layer 3
  (Enter none if no overwrite for PN for layer 3   ): pn-l3-nodes.out
  Read values for 1393 nodes from file PN-L3-NODES.OUT

 Include data for parameter type PV? y

Parameter value file for parameter type PV for layer 1
  (Enter none if no overwrite for PV for layer 1   ): none
Parameter value file for parameter type PV for layer 2
  (Enter none if no overwrite for PV for layer 2   ):  pv-l2-nodes.out
  Read values for 1393 nodes from file PV-L2-NODES.OUT
Parameter value file for parameter type PV for layer 3
  (Enter none if no overwrite for PV for layer 3   ): none

 Include data for parameter type PL? y

Parameter value file for parameter type PL for layer 1
  (Enter none if no overwrite for PL for layer 1   ):  pl-l1-nodes.out
  Read values for 1393 nodes from file PL-L1-NODES.OUT
Parameter value file for parameter type PL for layer 2
  (Enter none if no overwrite for PL for layer 2   ):  pl-l2-nodes.out
  Read values for 1393 nodes from file PL-L2-NODES.OUT
Parameter value file for parameter type PL for layer 3
  (Enter none if no overwrite for PL for layer 3   ):  pl-l3-nodes.out
  Read values for 1393 nodes from file PL-L3-NODES.OUT

 Include data for parameter type SCE? n


 Include data for parameter type SCI? n


 Overwrite file CVoverwriteIWFM.dat has been written. If this was created on
   the basis of an existing overwrite file, the scaling factors from that
   file have been preserved. If it was created from scratch, all scaling
   factors have been set equal to 1. In each case this assumes that the user
   has correctly accounted for the scaling.


>
```

Figure 5 (continued). Sample inputs (white) and outputs (black) for program real2iwfm.

IWFM2OBS

IWFM2OBS is a pre- and post-processing program developed primarily to interpolate the simulated equivalents of observations from the IWFM standard model output files into a PEST-compatible form. **IWFM2OBS** reads files with observed (groundwater, stream, subsidence and tile-drain hydrograph) values, then creates a PEST-compatible output file with time-interpolated simulated values that match the times and locations of the observations. **IWFM2OBS** was developed specifically for use with IWFM and is executed in the batch file or shell script listed in the PEST Control File; it is executed after the IWFM Simulation application.

IWFM2OBS can also be executed as a pre-processor. When used as a pre-processor, **IWFM2OBS** can create a PEST Instruction File and a list of observations for each observation type used in the PEST Control File. This list contains only the observations that fall within the model simulation period, and the observations are listed in the PEST Instruction File in thesame order as the occur in the PEST Control File.

IWFM2OBS Inputs

Program **IWFM2OBS** will not run unless a settings file (`settings.fig`) is present within the directory where it is run. (The format and contents of the settings file are detailed in Section 2.19 of Doherty, 2013a).

IWFM2OBS requires the following files:

- Standard IWFM Simulation Input File – contains a listing of file names including the names of the Print Control File and Hydrograph Output Files.
- IWFM Print Control File – contains unique names for all hydrographs output files. The Print Control File name is obtained from the Simulation Input File. The Print Control File is slightly modified from the standard IWFM format. Hydrograph identifiers (IDs) are added as an additional column on the right-hand side of each hydrograph print control specification section. The IDs for the groundwater and subsidence hydrographs are in column 4 (following IOUTHL for groundwater or IOUTSL for subsidence, and X and Y). The IDs for streams and tile drains are in column 2 (following IOUTR for streams and OUTTD for tile drains). These IDs are ignored by IWFM.
- IWFM Hydrograph Output Files – the names are obtained from the standard IWFM Simulation Input File. The IWFM Simulation Input File must specify at least one type of hydrograph output. IWFM unit numbers for the hydrograph output files are 41 (subsidence), 45 (tile drain), 46 (surface water), and 47 (groundwater).
- PEST Observation Files – one PEST Observation File is read for each type of hydrograph output. PEST Observation Files must be in the “bore sample file” format, as described in Section 2.7 of the PEST Groundwater Data Utilities

documentation, Part A: Overview (Doherty, 2013a). This format contains four values for each observation: ID, Date, Time, and Value.

IWFM2OBS Execution

IWFM2OBS is executed at the command prompt by typing:

```
iwfm2obs
```

The program then prompts the modeler for a series of inputs, including the name of the IWFM Simulation Main Input File. **IWFM2OBS** then loops through the same set of prompts for each type of hydrograph output that is specified in the Simulation Main Input File. These prompts include the name of the corresponding PEST Observation File, the maximum time for interpolation between model outputs, an optional PEST Instruction File, and the name of the PEST Observation output file the results will be written to. The answers to these prompts are usually provided in a list or piping file. A simple example is given below where **IWFM2OBS** performs time-interpolation on stream and groundwater observations, but not on subsidence tile drains.

IWFM2OBS Outputs

IWFM2OBS generates the following output files:

- PEST Observation File – simulated values interpolated to observation times, in the bore sample format (*.smp).
- PEST Instruction File – a pre-processor file that can optionally be created.
- PEST control "helper" file – If the option to write a PEST Instruction File is selected, an additional file is created with the identifier "**pcf_**" appended to the beginning of the Instruction File name. This file lists selected observations from the input file that have corresponding output when **IWFM2OBS** performs time-interpolation. These data are meant to be pasted into the PEST control file as observations. Use of the "helper file" also ensures correlation of numbering schemes between the instruction file and pest control file.

Note on temporary, intermediate files: **IWFM2OBS** creates several intermediate files (e.g. `gw_temp.smp` and `sw_temp.smp)`. These files are used by **IWFM2OBS** during processing but are not passed on to any other applications. These files can be deleted in the cleanup portion of the model-run batch file or shell script to insure that they are not accidently reused.

```
>iwfm2obs

 Program IWFM2OBS interpolates model output to match the times and locations
   of calibration observations and put them into a PEST-compatible output
   file.

 Enter name of IWFM main input file: Sim.in

 4 hydrograph file(s) are output by IWFM

 Enter name for subsidence observation smp file (ENTER for none): subsidence.smp
 Enter (subsidence) extrapolation threshold in days (fractional if necessary): 30
 OPTIONAL Enter name for subsidence PEST instruction file (ENTER for none):
 Enter name for subsidence output: sbout.smp
 Enter name for tiledrain observation smp file (ENTER for none):
 Enter name for stream observation smp file (ENTER for none): streams.smp
 Enter (stream) extrapolation threshold in days (fractional if necessary): 30
 OPTIONAL Enter name for stream PEST instruction file (ENTER for none):
 Enter name for stream output: stout.smp
 Enter name for groundwater observation smp file (ENTER for none): heads.smp
 Enter (groundwater) extrapolation threshold in days (fractional if necessary): 30
 Will head differences be calculated? Y
   Enter name of file listing well pairs: headdiffs.in
 OPTIONAL Enter name for groundwater PEST instruction file (ENTER for none):
 Enter name for groundwater output: gwout.smp

 - names of  1385 groundwater hydrograph ids read from file Print.dat
 - names of    34 stream hydrograph ids read from file Print.dat
 - names of    24 subsidence hydrograph ids read from file Print.dat

 - data for    444 times read from file SubsHyd.out
 - 1695 data lines written to new bore sample file sbout.smp


 - data for    444 times read from file SWhyd.out
 - 5636 data lines written to new bore sample file stout.smp


 - data for    444 times read from file GWhyd.out
 - 62964 data lines written to new bore sample file gwout.smp

NORMAL TERMINATION - IWFM2OBS

>
```

Figure 6. Sample inputs (white) and outputs (black) for program iwfm2obs.

STACDEP2OBS

STACDEP2OBS is a pre- and post-processing program developed to facilitate the use of stream-groundwater flow observations in IWFM model calibration. Stream net gain or loss data for a physical stream reach are usually determined through the following procedure:

1. Obtain total gaged flows at two (or more) stream gages.
2. Complete a base-flow separation analysis and differentiate the contribution to the total gaged flow from (a) gains from groundwater – i.e., base flow; and (b) other sources, such as overland flow and direct precipitation.

STACDEP2OBS interpolates the simulated equivalents to stream-groundwater flow observations from the IWFM Stream Reach Budget File into a form that is compatible with PEST. **STACDEP2OBS** performs time-interpolation of the contents of this file to match stream-groundwater flow observation times, and writes the simulated equivalents to the observations to a PEST Output File in the "bore sample file" format (extension ".smp"). **STACDEP2OBS** can also write a PEST Instruction File (extension ".ins") that enables PEST to read the outputs from **STACDEP2OBS**.

STACDEP2OBS is designed to be called as part of a batch file or shell script run by PEST, following execution of the IWFM Simulation and Budget applications. **STACDEP2OBS** can also be executed as a pre-processor. When used as a pre-processor, **STACDEP2OBS** can create a PEST Instruction File and a listing of enumerated observations for each observation type in the PEST Control File. This observation listing contains only those observations that fall within the model simulation period, listed in the same order as the PEST Instruction File.

STACDEP2OBS Inputs

Program **STACDEP2OBS** will not run unless a settings file (`settings.fig`) is present within the directory where it is run. (The format and contents of the settings file are detailed in Section 2.19 of the PEST Groundwater Data Utilities documentation, Part A: Overview (Doherty, 2013a).

STACDEP2OBS requires the following input files:

- IWFM Budget Input File (extension ".in") – this file contains the units and time steps used in the simulation, and conversion factors.
- IWFM Stream Reach Budget Output File (extension ".bud") – this file contains the flow balance for each simulated stream reach, including inflows, outflows, and gains from groundwater.

- Stream Reach Groups File – this file lists groups of simulated stream reaches that together correspond to a physical stream reach for which measured stream-groundwater flow observations are available for calibration. Simulated stream-groundwater flows for all members of a stream reach group are summed to provide the net stream-groundwater flow for the entire group (entire physical stream reach). Each stream reach group can contain up to four simulated stream reaches. Anexample is provided in Appendix B.
- PEST Observation File – Observed stream-groundwater flows for physical stream reaches. This file is in the "bore sample file" format, as described in section 2.7 of the PEST Groundwater Data Utilities documentation, Part A: Overview (Doherty, 2013a). This format consists of four values for each observation: ID, Date, Time, and Value.

An example stream reach groups input file is provided with the source code. This file lists 33 stream reach groups. Some stream reach groups contain only a single simulated reach, and others contain more than one.

STACDEP2OBS Execution

STACDEP2OBS is executed at the command prompt by typing:

```
stacdep2obs
```

The program prompts the user for a series of inputs, including the names of the main input files, the time unit used in the model, and the model's starting date. **STACDEP2OBS** then prompts for the name of the PEST Observation File, the maximum allowable time of interpolation between model outputs, the name of an optional PEST Instruction File, the name of the PEST Output File that will be created, the name of the Stream Reach Groups File, and the name of the IWFM Budget Input File. The responses to these prompts are usually provided in a list or piping file. Figure 7 shows a simple example where **STACDEP2OBS** performs time-interpolation on simulated stream-groundwater flows, but does not produce a PEST Instruction File.

STACDEP2OBS Outputs

STACDEP2OBS produces the following files:

- PEST Observation File – the simulated stream-groundwater flows for each stream reach group, time-interpolated to match observation times. The output is in bore sample format (*.smp).
- PEST Instruction File – an optional pre-processor file.

Note on temporary, intermediate files: **STACDEP2OBS** creates an intermediate file, `gl_temp.smp`. This files is used by **STACDEP2OBS** during processing but is not passed on to any other applications. This file can be deleted in the cleanup portion of the model-run batch file or shell script to insure that it is not accidently reused.

```
>stacdep2obs
 Enter name of IWFM BUDGET utility output file:   streamrch.bud
 Enter model time unit, (d)ay, (w)eek, (m)onth, or (y)ear: m
 Enter name for stream accretion/depletion observation smp file:   StAcDepObs.smp
 Enter extrapolation threshold in days (fractional if necessary): 30
 OPTIONAL Enter name for gainloss PEST instruction file (ENTER for none):
 Enter name for gainloss output: StAcDepObs.out
 Enter name of file listing stream reach groups: StAcDepGroups.in
 Enter name of input file to the BUDGET program: Budget.in
 Report long-term simulated average only? (Y/N): n
 Reading file: StAcDepGroups.in
 Group                    No.          Members
 Kern_1                   1            1
 Tule_1                   1            6
 Kawe_1                   1            5
 King_1                   1            2
 Sout_1                   1            3
 Nort_1                   1            4
 SanJ_1                   1            7
 SanJ_2                   1            8
 SanJ_3                   3           10          12          14
 SanJ_4                   1           16
 SanJ_5                   1           18
 SanJ_6                   2           20          22
 SanJ_7                   1           24
 Processed file: StAcDepGroups.in
 Reading file: Budget.in
 Processed file: Budget.in
 Processed file: STREAMRCH.BUD
 Read data for   25 reaches
 - 13 data lines written to new accretions-depletions file StAcDepObs.out


>
```

Figure 7. Sample inputs (white) and outputs (black) for program stacdep2obs.

GET-AVERAGES

GET-AVERAGES is a post-processing program that reads a PEST Observation File, calculates the average value for groups of observations, and writes the results to a PEST Observation File. This program is generally used where only estimated long-term average observed values are available. **GET-AVERAGES** is designed to be run within the batch file or shell script run by PEST, following execution of the IWFM Simulation and Budget programs and of other post-processors that create PEST Observation Files containing simulated equivalents to observations.

GET-AVERAGES Inputs

GET-AVERAGES requires the following input files:

- PEST Observation File – A file containing time-series of simulated equivalents in the “bore sample format” (*.smp).

GET-AVERAGES Execution

GET-AVERAGES is executed at the command prompt by typing:

```
get-averages
```

The program prompts the user for the name of one input file. The output file names are created by modifying the input file name. A simple example is shown in figure 8.

```
>get-averages
 Enter name of smp file with time-series data: StAcDepObs.smp

NORMAL TERMINATION - GET-AVERAGES

>
```

Figure 8. Sample inputs (white) and outputs (black) for program get-averages. For this case, the program created files StAcDep-avg.smp and StAcDep-one.smp.

GET-AVERAGES Outputs

GET-AVERAGES generates two output files:

- PEST Observation File – A copy of the original input file with each value replaced by the average value for the observation location, in the "bore sample file" format (*.smp).
- PEST Observation File – A file with one entry for each observation location containing the average value, in the "bore sample file" format (*.smp).

LTBUD

Flow values often range over several orders of magnitude, from zero to very high values. Log-transforming these values reduces the range of the simulated equivalent values used in PEST and gives greater impact to the weighting scheme. The log of zero is undefined, so a small value is added to zero values so log-transformed value will be a large negative value (for example, adding 0.001 results in a log-transformed value of -3).

LTBUD is a post-processing program that reads an IWFM Budget Output File, performs log transformations on the values in the budget tables, and writes the new values in the form of an IWFM Budget Output File.

LTBUD Inputs

LTBUD requires the following inputss. They are listed in the same sequence as the program prompts the user.

- IWFM Budget Output File – The name of a file containing budget tables for a particular flow process, produced by the IWFM Budget program. For example, the Stream Reach Budget file, which contains one budget table with a water balance for each stream reach in the IWFM model.
- Output file name – This is conventionally the base name of the IWFM Budget Output File (above) with the extension ".out".
- Specify a value to add to zero – Supply a number between zero and one that will be added to zero values in the table before the log transformation.

LTBUD Execution

LTBUD is executed at the command prompt by typing:

```
ltbud
```

The program prompts the user for two items: the name of the IWFM Budget Program output file to be processed, and the value (between 0 and 1) to be added to table values equal to zero. The responses to these prompts are usually provided in a

```
>ltbud

   Program LTBUD reads an output file from the IWFM Budget
   program, log-transforms all of the numbers in the budget
   tables, and writes out the results to a new file.

 Enter Budget file name: SMWSHED.BUD

 Enter output file name: SMWSHED.OUT

 Enter value to add to zero: 0.001

 => Wrote results to file SMWSHED.OUT
NORMAL TERMINATION - LTBUD

>
```

Figure 9. Sample inputs (white) and outputs (black) for program ltbud.

list or piping file. A simple example is provided in figure 9.

LTBUD Outputs

LTBUD generates one file:

- LBBUD output file – The text and dates in this file are identical to those of the input IWFM Budget Output File. All of the values in the tables are log-transformed.

LTSMP

Flow values often range over several orders of magnitude, from zero to very high values. **LTSMP** log-transforming these values and changes zero-flow values to a high negative value. The log of zero is undefined, so a small value is added to zero values so log-transformed value will be a large negative value (for example, adding 0.001 results in a log-transformed value of -3). This reduces the range of the simulated equivalent values used in PEST, increases the priority of matching zero-flow values, and gives greater impact to the weighting scheme.

LTSMP reads a PEST Observation File in "bore sample file" format, log-transforms the values, and writes the results to a new PEST Observation File.

LTSMP Inputs

LTSMP requires the following inputss. They are listed in the same sequence as the program prompts the user.

- PEST Observation File – A file containing time-series of surface-water flow observations or simulated equivalents, in the "bore sample format" (*.smp).
- Output file name – This is conventionally the base name of the input (above) with "_ltran" added, followed by the extension ".smp".
- Number of charachters to skip – This integer value is the number of characters on each line that is read from the input file and written to the output file. This includes the ID, Date and Time for each observation.
- Specify a value to add to zero – Supply a number between zero and one that will be added to zero values in the file before the log transformation.

LTSMP Execution

LTSMP is executed at the command prompt by typing:

```
ltsmp
```

The program prompts the user for four inputs: the names of the PEST Observation File, the name of the output file, the number of characters to skip on each line (the width of the first three fields of the bore-log file), and the value to be added to zero. The responses to these prompts are usually provided in a list or piping file. A simple example is provided in figure 10.

LTSMP Outputs

LTSMP generates one file:

- PEST Observation File – A copy of the input file, with the observation values (or simulated equivalent values) replaced by the log-transformed values.

```
>ltsmp

   Program LTSMP read a bore-log-formatted file containing
   observations or simulated equivalents, log-transforms
   all of the values in the file, and writes out the results
   to a new file.

 Enter input file name: stout.smp

 Enter output file name: stout_ltrans.smp

 Enter number of characters to skip: 38

 Enter value to add to zero: 0.001

 => Wrote results to file stout_ltrans.smp

NORMAL TERMINATION - LTSMP

>
```

Figure 10. Sample inputs (white) and outputs (black) for program ltsmp.

References

CH_2M Hill and S.S. Papadopulos. 2008. Programs to Support Calibration of IWFM using PEST. Sacramento, CA: CH_2M Hill.

Doherty, J. 2010. PEST – Model-Independent Parameter Estimation, User Manual, 5th Edition. Corinda, Australia: Watermark Numerical Computing.

Doherty, J. 2013a. PEST Groundwater Data Utilities, Part A: Overview. Corinda, Australia: Watermark Numerical Computing.

Doherty, J. 2013b. PEST Groundwater Data Utilities, Part B: Program Descriptions. Corinda, Australia: Watermark Numerical Computing.

Appendix A – Source Code and Binaries

This appenxix contains source code for the PEST-IWFM utilities, and compiled binary executables for Windows and Linux. The Windows executables were compiled with Intel Fortran Composer XE version 14 on an Intel Xeon processor running the Windows 7 operating system. The Linux binaries were compiled with Intel Fortran 64 Compiler XE version 13 on an Intel Xeon processor running the Ubuntu operating system.

Appendix B – Example Input and Output Files

Zone Specification File – This file contains a list of the IWFM nodes and an integer designation for each node. With this file, model nodes can be aggregated into zones for zonal parameter distributions. The Zone Specification File is used by **PPK2FACI**. The example file **`IWFMzones.dat`** contains a list of the IWFM nodes and the intger value '1' for each node, indicating the entire model domain is one zone.

Template Parameter Overwrite File – This file is read by the program **REAL2IWFM**. It contains one line for each node and layer of the IWFM model. Contains a value of -1 if the value in the IWFM Parameter File is to be used, or the value that should be used in place of the value in the IWFM Parameter File. The format of the template file is identical to the Parameter Overwrite File produced by the program **REAL2IWFM**. The example file **`CVoverwriteIWFM.tpl`** contains a sample parameter overwrite file with '-1' values for the hydraulic conductivity values that are defined using horizontal flow barriers.

Stream Reach Groups File – This file is read by the program **STACDEP2OBS**. The program assigns the observation names used in the PEST Control File to individual river reaches or groups of reaches for which stream-groundwater flow observations are available. The Stream Reach Groups File lists the observation name followed by integer values for the stream reach numbers. The first two lines of the example file **`StAcDepGroups.in`** give the maximum number of reaches in a single group and the total number of stream reach observations described in this file. Each of the remaining lines contains a stream reach observation name followed by integer values indicating the Stream Reach Number form the IWFM model.