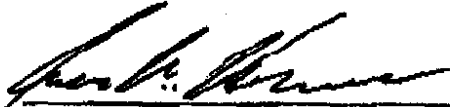# Analysis Report
Task 4 of AP-088
Conditioning of Base T Fields to Transient Heads

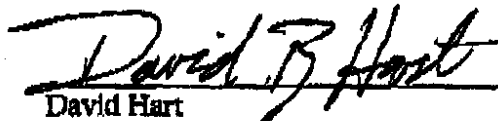(AP-088: Analysis Plan for Evaluation of the Effects of
Head Changes on Calibration of Culebra Transmissivity Fields)

Task Number 1.3.5.1.2.1

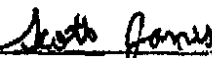Report Date: August 20th, 2003

Authors:

_____ Date: 8/21/03
Sean A. McKenna
PMTS, Geohydrology Department (6115)
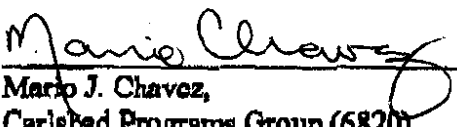
_____ Date: 8/21/03
David Hart
Student Intern, Geohydrology Department (6115)
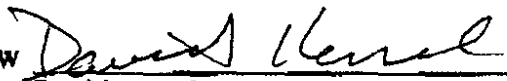
Technical Review:

_____ Date: 8/21/03
Scott James
SMTS, Geohydrology Department (6115)

QA Review:

_____ Date: 8/25/03
Mario J. Chavez,
Carlsbad Programs Group (6820)

Management Review _____ Date: 8/26/03
David Kessel
Manager, Performance Assessment and
Decision Analysis (6821)

WIPP:1.3.5.1.2.1:TD:QA-L:DPRP1:522085

INFORMATION ONLY

Table of Contents

Table of Figures

Table of Tables

1   Introduction

This document presents the methods, supporting data, and results of the stochastic inverse calibration of the Culebra T fields to both steady state heads obtained during calendar year 2000 and to a series of transient responses to various hydraulic tests over a period of 11 years. The calibration is done simultaneously to both the steady-state and the transient data for each of 150 different base transmissivity fields.

1.1   Background

The Waste Isolation Pilot Plant (WIPP) is located in southeastern New Mexico and has been developed by the U.S. Department of Energy (DOE) for the geologic (deep underground) disposal of transuranic (TRU) waste. Containment of TRU waste at the WIPP is regulated by the U.S. Environmental Protection Agency (EPA) according to the regulations set forth at Title 40 of the Code of Federal Regulations, Parts 191 and 194. The DOE demonstrates compliance with the containment requirements in the regulations by means of a performance assessment (PA), which estimates releases from the repository for the regulatory period of 10,000 years after closure.

In October 1996, DOE submitted the Compliance Certification Application (CCA; U.S. DOE, 1996) to the EPA, which included the results of extensive PA analyses and modeling. After an extensive review, in May 1998 the EPA certified that the WIPP met the criteria in the regulations and was approved for disposal of transuranic waste. The first shipment of waste arrived at the site in March 1999.

The results of the PA conducted for the CCA were subsequently summarized in a Sandia National Laboratories (SNL) report (Helton et al., 1998) and in refereed journal articles (Helton and Marietta, 2000).

The DOE is required to submit an application for re-certification every five years after the initial receipt of waste. The re-certification applications take into account any information or conditions that have changed since the original certification decision. Accordingly, the DOE is conducting a new PA in support of the Compliance Recertification Application (CRA).

1.2   Purpose

The purpose of these calculations is to calibrate the Culebra transmissivity fields to new steady-state, or "equilibrium," head data that have been collected since the CCA time period (i.e., the 2000 heads), and to incorporate the responses to transient hydraulic tests that were not included in the CCA calculations (e.g., the P-14 and WQSP-1 pumping test data). Additionally, these calculations incorporate recent updates in the geologic conceptual model and the influence of these updates on the spatial distribution of transmissivity within the Culebra. These recent updates in the geologic conceptual model have been used to produce the base transmissivity fields used in this study and are documented by Holt and Yarbrough (2003).

INFORMATION ONLY

1.3 Outline

This report documents the data, methods, and summary results of the work done as Task 4 of Analysis Plan 088 (Beauheim, 2002a). The sections of this report and a brief description of each subsection are:

1 Introduction
 1.1 Background: A brief background of the WIPP certification and recertification process
 1.2 Purpose: A concise statement of the purpose of this work
 1.3 Outline
 1.4 Model Setup: Definition of the spatial domain of the model and changes from the Task 3 model
 1.5 Observed Data: A description of the measured head and drawdown data used for the calibration of the base transmissivity fields and the references from which these measurements were obtained

2 Modeling Approach
 2.1 Boundary Conditions: The construction of the no-flow and fixed-head boundary conditions
 2.2 Spatial Discretization: The spatial discretization of the model domain into finite-difference cells
 2.3 Temporal Discretization: The discretization of the observed time period into stress periods and time steps within MODFLOW
 2.4 Weighting of Observed Data: Assignment of weights to each observation data set
 2.5 Pilot Point Calibration: The details of the numerical calibration process including details of the operation of a series of shells that do the parallel calculations
 2.6 Particle Tracking: A brief description of the particle-tracking setup
 2.7 File Naming Convention: A large table intended as a guide for understanding the run control process

3 Modeling Assumptions

4 Results

5 Summary

1.4 Model Setup

The model domain used for the stochastic inverse calibration of the Culebra T fields to steady-state and transient data is the same as that used in the steady-state calibrations (McKenna and Hart, 2003). This model domain is oriented with the compass directions and is 30.6 km in the north-south direction and 22.3 km in the east-west direction. The corners of the WIPP model domain are given in Table 1. These coordinates define the center of 100X100-m^2 model cells at the four corners of the model domain.

INFORMATION ONLY

Table 1. The UTM coordinates of the corners of the numerical model domain.

Domain Corner	X Coordinate (meters)	Y Coordinate (meters)
Northeast	624,000	3,597,100
Northwest	601,700	3,597,100
Southeast	624,000	3,566,500
Southwest	601,700	3,566,500

The WIPP land-withdrawal boundary, or the "WIPP site boundary" is an approximately 6.4 X 6.4 km area near the center of the model domain. The boundary of the WIPP site is defined by the coordinates shown in Table 2. For the calculations described in this report, the coordinates shown in Table 2 are used to determine when and where the particle tracks leave the WIPP site.

Table 2. The UTM coordinates of the WIPP site boundary.

Domain Corner	X Coordinate (meters)	Y Coordinate (meters)
Northeast	616,941	3,585,109
Northwest	610,495	3,585,068
Southeast	617,015	3,578,681
Southwest	610,567	3,578,623

The modeling approach used in these calculations is to employ the PEST software to adjust a residual transmissivity field at a number of selected pilot point locations. The addition of the calibrated residual field to a previously generated base transmissivity field produces the final calibrated transmissivity field. This approach is identical to that used in the steady-state calculations (McKenna and Hart, 2003). The base transmissivity fields used in the current calculations are somewhat different than those used in the steady-state calculations as additional geologic data used to create the base transmissivity fields became available after the steady-state calculations were completed. The creation of the base transmissivity fields used in these calculations and the major differences in these fields relative to the base transmissivity fields used in the steady-state calibrations are described by Holt and Yarbrough (2003). The most significant difference in the construction of the base transmissivity fields from the steady-state calibrations to the present transient calibrations is the change to the boundary of the high-transmissivity zone on the west side of the model and the change in the location of the no-flow boundary made to accommodate this change in location of the high-transmissivity zone boundary. These changes are shown in Figure 1.

INFORMATION ONLY
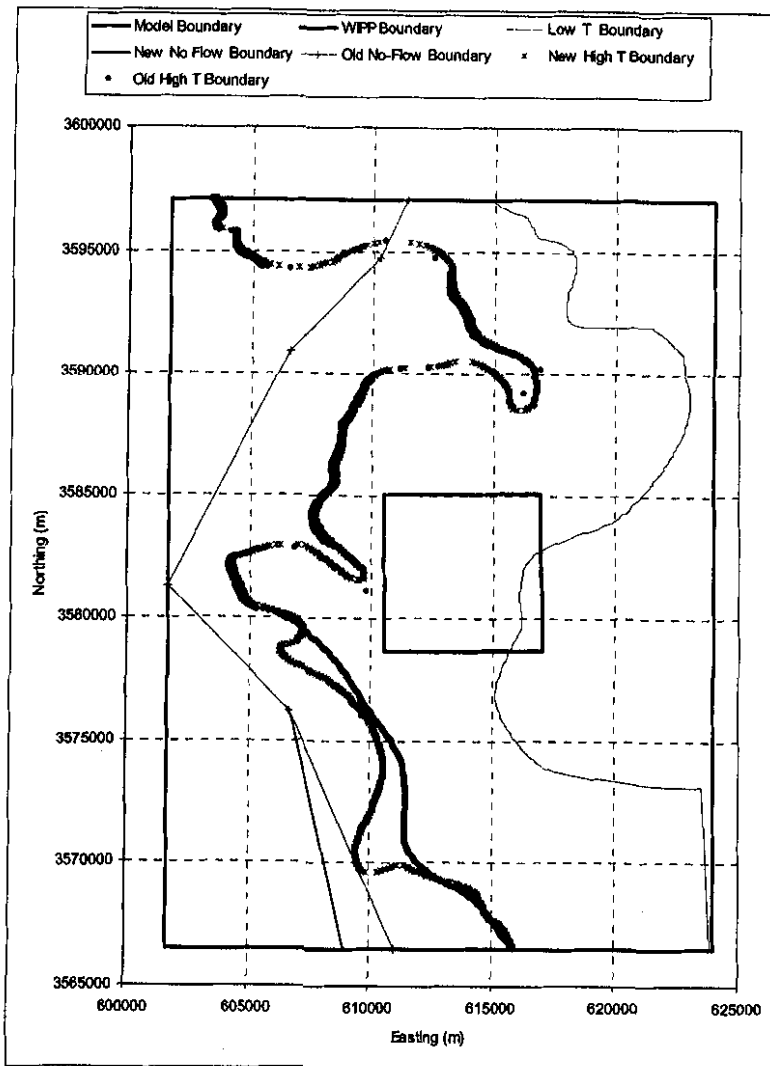
Figure 1. Model domain and zone boundaries for the steady-state and transient calibrations.

The major change in the high-transmissivity boundary from the steady-state calibrations to the transient calibrations is the much better definition of the shape and extent of the high-transmissivity reentrants on the west side of the model and the high-transmissivity boundary shift to the west near the southern end of the model domain. The no-flow boundary was adjusted to the west to maintain connectivity of the high-transmissivity zone all the way to the southern boundary of the model domain.

8

1.5 Observed Data

The observed data used for the transient calibrations are taken from a number of different sources. The steady-state data are those collected for the 2000 time period and used in the steady-state calibrations documented by McKenna and Hart (2003). The original source of the 2000 steady-state data is from Beauheim (2002b). For the 2000 time period, there are a total of 35 well locations with steady-state head measurements. The wells, their locations and the heads measured in the 2000 time period are given in Table 3.

Responses to seven different hydraulic tests are employed in the transient portion of the calibration (Table 4). Details on the original sources of the data shown in Table 4 are given in Beauheim (2003). Hydraulic responses for each of the seven tests are monitored in three to ten different observation wells depending on the hydraulic test.

A major change in the calibration data set from the CCA calculations is the exclusion of the hydraulic responses to the excavation of the shafts in the current calibration. The responses to the shaft excavations were excluded because:

1) Only 2 wells (H-1 and H-3) responded directly to the shaft excavations and the areas between the shafts and these wells are stressed by other hydraulic tests that are included in the calibration data set (H-3b2, WIPP-13 and H-19b0).

2) It was difficult to model both the flux and pressure changes accurately during the excavation of the shafts with MODFLOW. This difficulty is due to both the finite-difference discretization of MODFLOW that requires each shaft to be modeled as a complete model cell and some limitations of the data set.

3) The long-term effects of the shafts on site-wide water levels were important for the CCA modeling because that modeling sought to replicate heads over time. In the current CRA calibration effort, shaft effects are not important because drawdowns resulting from specific hydraulic tests are used as the calibration targets and shaft effects can be considered as second-order compared to the effects of the hydraulic tests that are simulated.

A small amount of processing of the observed data was necessary prior to using it in the calibration process. This processing included selecting the data values that would be used in the calibration procedure from the often voluminous measurements of head provided by the references given in Beauheim (2003). These data were chosen to provide an adequate description of the transient observations at each observation well across the response time without making the modeling too computationally burdensome in terms of the temporal discretization necessary to model responses to these observations. Scientific judgment was used in selecting these data points. This selection process resulted in a total of 1,332 observations for use in the transient calibration.

Additionally, the modeling of the pressure data is done here in terms of drawdown. Therefore, the value of drawdown at the start of any transient test must be zero. A separate perl script was written to normalize each set of observed heads to a zero value reference at the start of the test with the exception of the H-3 test that is only preceded by the steady-state simulation. The

INFORMATION ONLY

calculations are such that the resulting drawdown values are positive. These data normalization scripts are included as Appendix 1.

In addition to normalizing the measured head data, some of the tests produced negative drawdown values when normalized. These negative results are due to some of the observations having heads greater than the reference value. This occurs due to some hydraulic tests that were conducted at earlier times in the Culebra but were not included in the numerical model. If the drawdowns from one of these previous tests are still recovering to zero at the start of a simulation, they can cause negative drawdowns in the simulation as the recovery continues. Most of these effects were addressed through trend removal in initial data processing (Beauheim, 2003) but some residual effects remain.

The resultant transient calibration points are show in Figures 2 through 15. These figures show the time series of drawdown values for each observation well including the location of each hydraulic test and the locations of the observation wells for that test within the model domain. The values of drawdown are in meters where a positive drawdown indicates a decrease in the pressure within the well relative to the pressure before the start of the pumping (negative drawdown values indicate rises in the water level). For the WQSP-1 and WQSP-2 tests, well WQSP-3 showed no response. These results are used in the calibration process by setting the observed drawdown values to zero for WQSP-3. The maps in Figures 2 through 15 also show the locations of the pilot points used in the calibration (these are discussed later in this report).

Table 3. Well names and locations of the 35 steady-state data obtained during the 2000 measurement period and used in the simultaneous steady-state and transient calibrations.

Measurement Number	Well Name	Easting (X) Coordinate (m)	Northing (Y) Coordinate (m)	2000 Measured Head (m)
1	AEC-7	621126	3589381	933.19
2	DOE-1	615203	3580333	916.55
3	DOE-2	613683	3585294	940.03
4	ERDA-9	613696	3581958	921.59
5	H-1	613423	3581684	927.19
6	H-2b2	612661	3581649	926.62
7	H-3b2	613701	3580906	917.16
8	H-4b	612380	3578483	915.55
9	H-5b	616872	3584801	936.26
10	H-6b	610594	3585008	934.20
11	H-7b1	608124	3574648	913.86
12	H-11b4	615301	3579131	915.47
13	H-12	617023	3575452	914.66
14	H-14	612341	3580354	920.24
15	H-15	615315	3581859	919.87
16	H-17	615718	3577513	915.37
17	H-18	612264	3583166	937.22
18	H-19b0	614514	3580716	917.13
19	P-17	613926	3577466	915.20
20	WIPP-12	613710	3583524	935.30
21	WIPP-13	612644	3584247	935.17
22	WIPP-18	613735	3583179	936.08
23	WIPP-19	613739	3582782	932.66
24	WIPP-21	613743	3582319	927.00
25	WIPP-22	613739	3582653	930.96
26	WIPP-25	606385	3584028	932.70
27	WIPP-26	604014	3581162	921.06
28	WIPP-30	613721	3589701	936.88
29	WQSP-1	612561	3583427	935.64
30	WQSP-2	613776	3583973	938.82
31	WQSP-3	614686	3583518	935.89
32	WQSP-4	614728	3580766	917.49
33	WQSP-5	613668	3580353	917.22
34	WQSP-6	612605	3580736	920.02
35	H-9b	613989	3568261	911.57

Table 4. Transient hydraulic test and observation wells for the drawdown data.

Stress Point	Observation Well	Observation Start	Observation End	Observation Type
H-3b2	DOE-1	10/15/1985	3/18/1986	Drawdown
	H-1	10/15/1985	4/14/1986	Drawdown
	H-2b2	10/15/1985	4/2/1986	Drawdown
	H-11b1	10/15/1985	4/21/1986	Drawdown
WIPP-13	DOE-2	1/12/1987	5/15/1987	Drawdown
	H-2b2	1/12/1987	5/15/1987	Drawdown
	H-6b	1/12/1987	5/15/1987	Drawdown
	P-14	1/12/1987	5/15/1987	Drawdown
	WIPP-12	1/12/1987	5/15/1987	Drawdown
	WIPP-18	1/12/1987	5/15/1987	Drawdown
	WIPP-19	1/12/1987	5/15/1987	Drawdown
	WIPP-25	1/12/1987	4/2/1987	Drawdown
	WIPP-30	1/12/1987	5/15/1987	Drawdown
P-14	D-268	2/14/1989	3/7/1989	Drawdown
	H-6b	2/14/1989	3/10/1989	Drawdown
	H-18	2/14/1989	3/10/1989	Drawdown
	WIPP-25	2/14/1989	3/7/1989	Drawdown
	WIPP-26	2/14/1989	3/7/1989	Drawdown
H-11b1	H-4b	2/7/1996	12/11/1996	Drawdown
	H-12	2/6/1996	12/10/1996	Drawdown
	H-17	2/6/1986	12/10/1996	Drawdown
	P-17	2/7/1996	12/10/1996	Drawdown
H-19b0	DOE-1	12/15/1995	12/10/1996	Drawdown
	ERDA-9	12/15/1995	12/10/1996	Drawdown
	H-1	12/15/1995	12/10/1996	Drawdown
	H-14	2/7/1996	12/10/1996	Drawdown
	H-15	12/12/1995	12/10/1996	Drawdown
	H-2b2	2/7/1996	12/10/1996	Drawdown
	H-3b2	12/15/1995	12/10/1996	Drawdown
	WIPP-21	1/18/1996	12/9/1996	Drawdown
	WQSP-4	1/1/1996	12/10/1996	Drawdown
	WQSP-5	1/18/1996	12/10/1996	Drawdown
WQSP-1	H-18	1/25/1996	2/20/1996	Drawdown
	WIPP-13	1/25/1996	2/20/1996	Drawdown
	WQSP-3	1/15/1996	2/20/1996	Zero Response
WQSP-2	DOE-2	2/20/1996	3/28/1996	Drawdown
	H-18	2/20/1996	3/28/1996	Drawdown
	WIPP-13	2/20/1996	3/28/1996	Drawdown
	WQSP-1	2/20/1996	3/24/1996	Drawdown
	WQSP-3	2/20/1996	3/24/1996	Zero Response

12
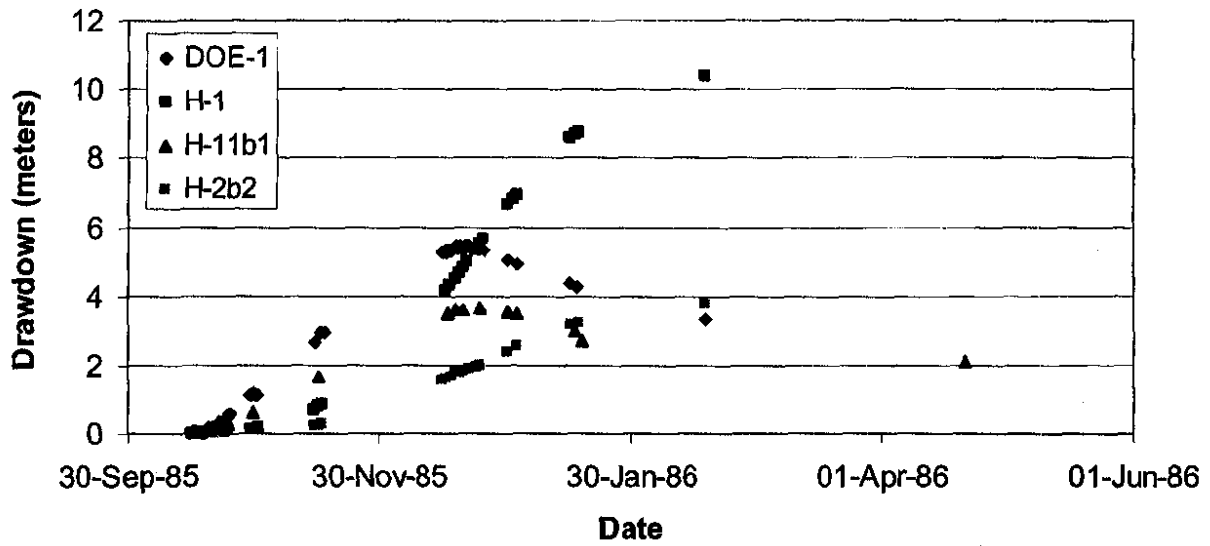
Figure 2. Observed drawdowns for the H-3b2 hydraulic test.
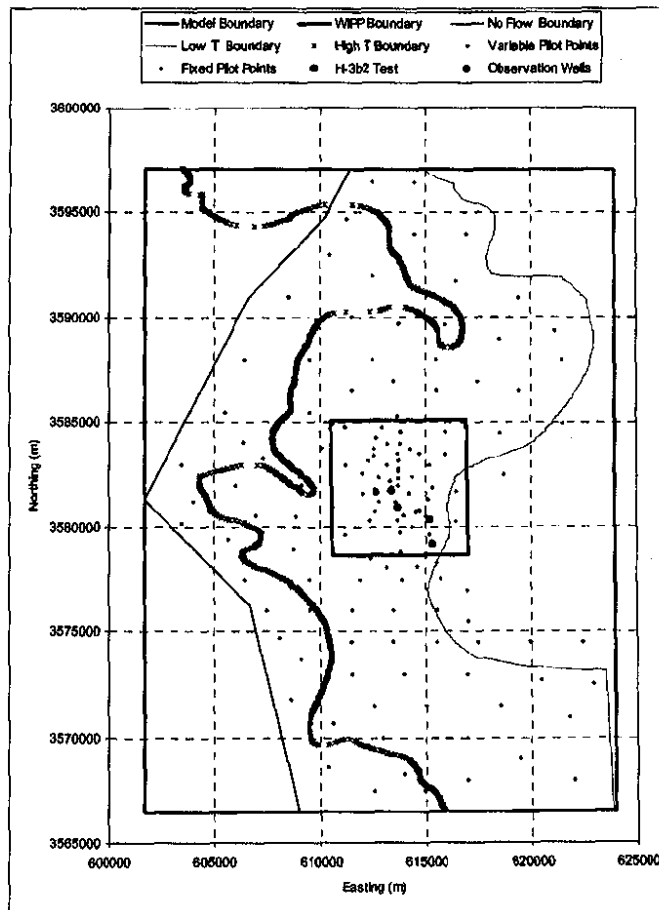
Figure 3. Locations of the H-3b2 hydraulic test and observation wells.
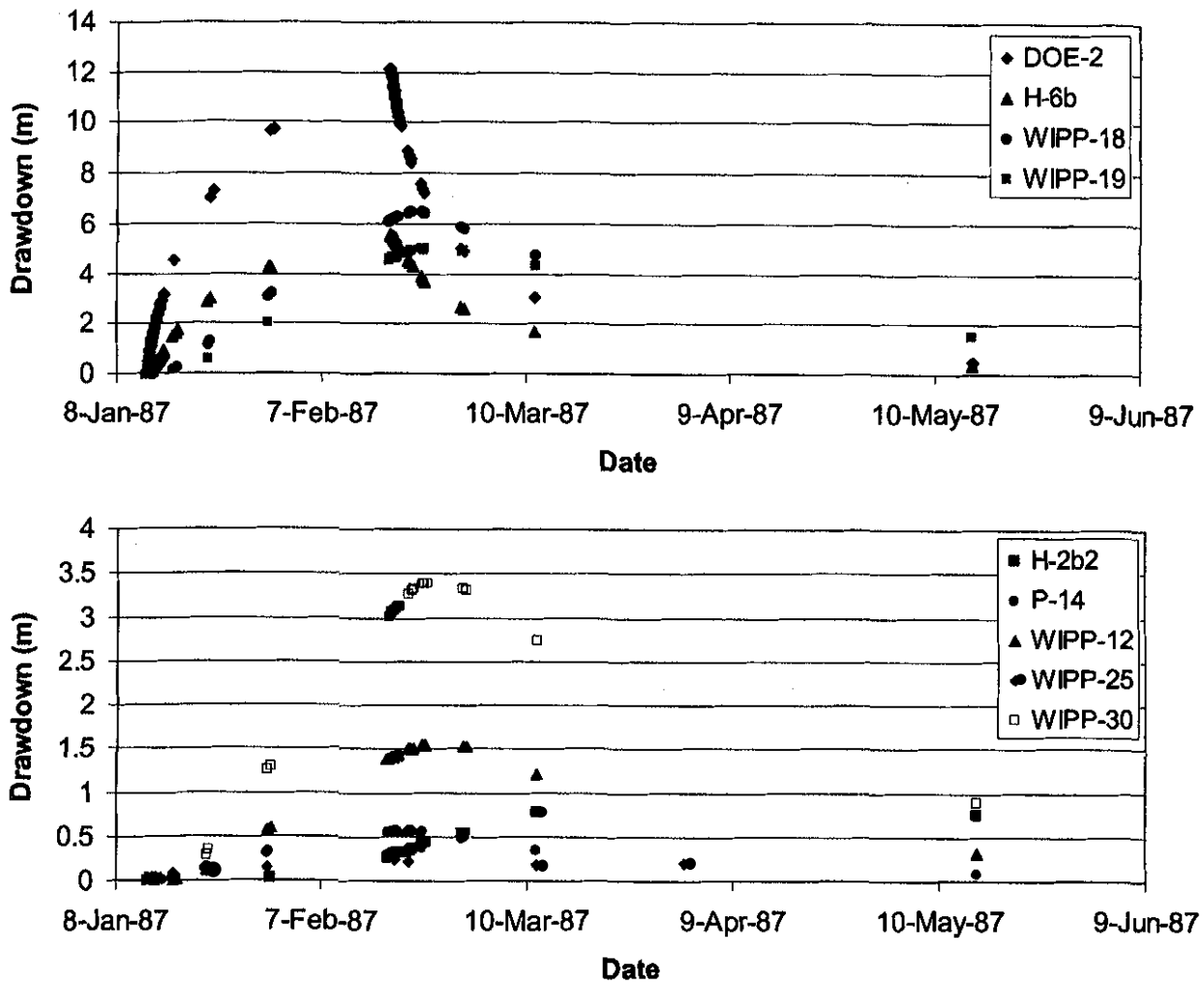
Figure 4. Observed drawdowns for the WIPP-13 hydraulic test. Note the change in the scale of the Y-axis from the upper to the lower image.
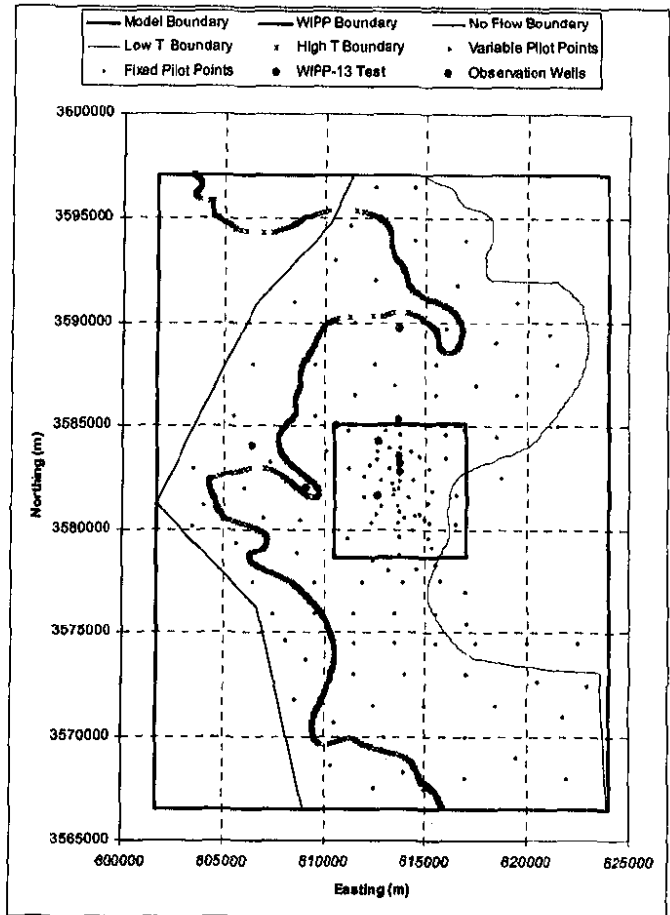
Figure 5. Locations of the WIPP-13 hydraulic test and observation wells.

15
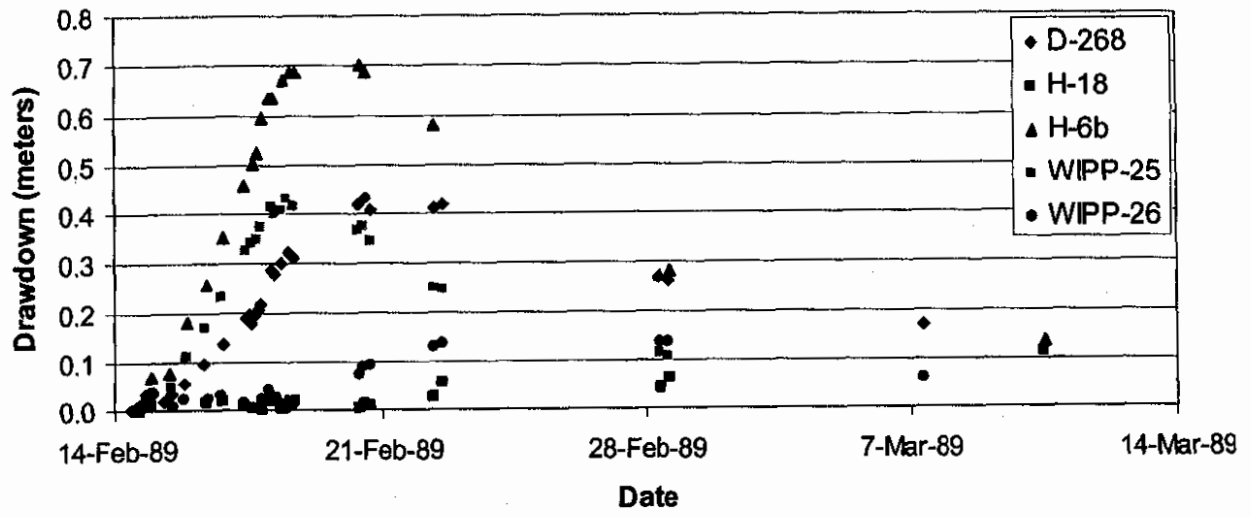
Figure 6. Observed drawdowns for the P-14 hydraulic test.
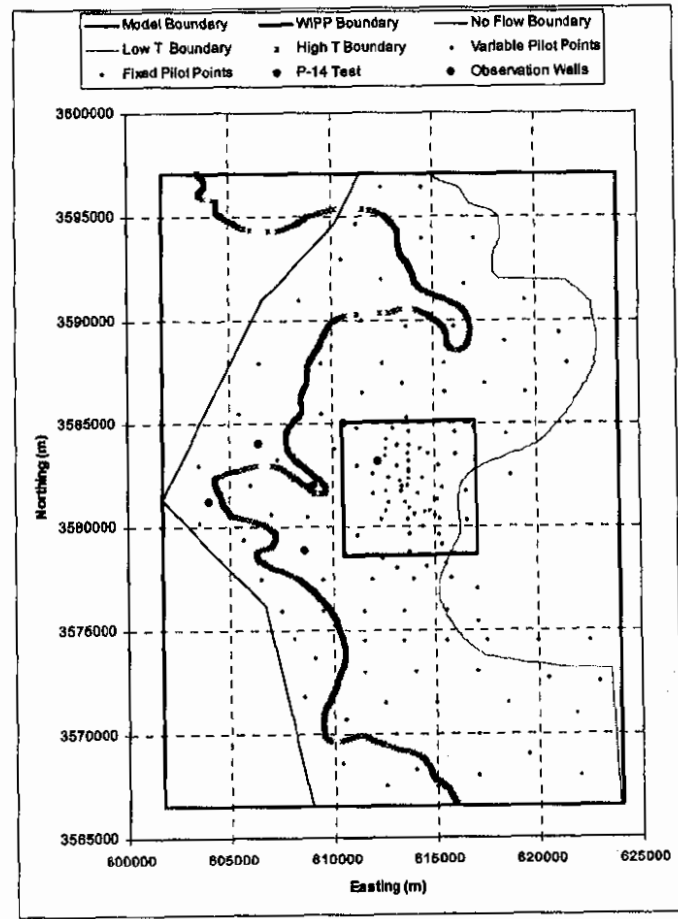
Figure 7. Locations of the P-14 hydraulic test and observation wells.

16
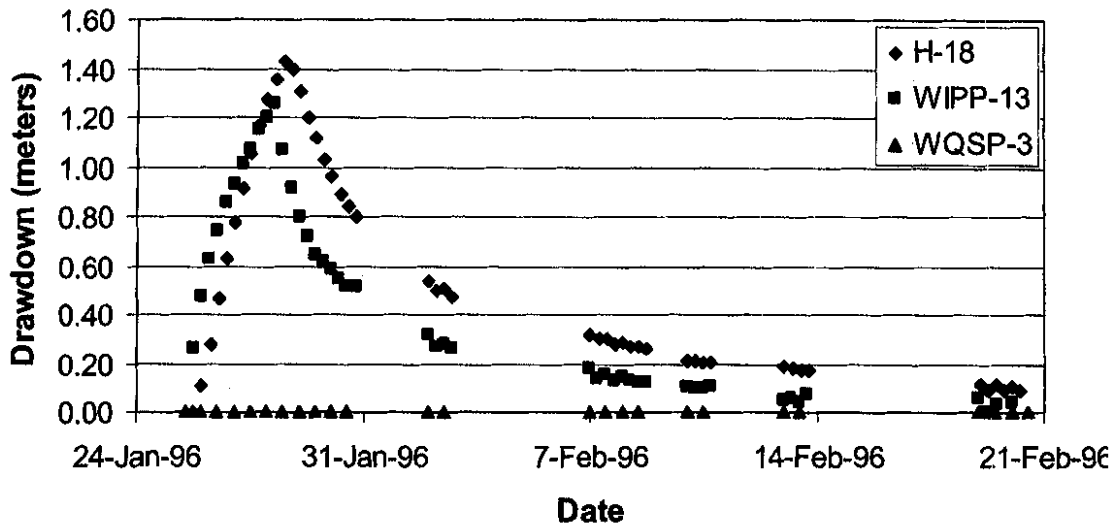
Figure 8. Observed drawdowns for the WQSP-1 hydraulic test.
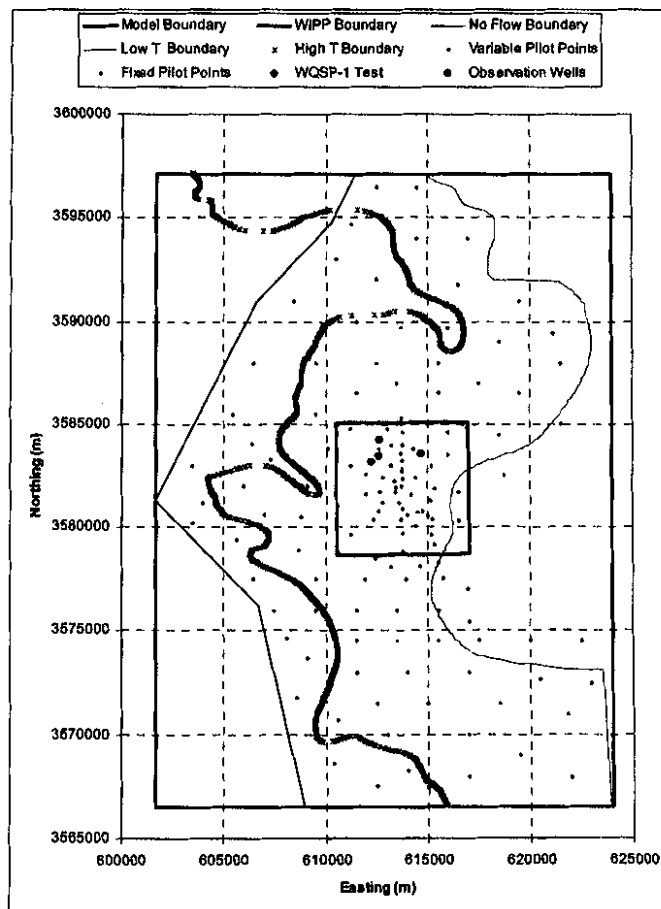
Figure 9. Locations of the WQSP-1 hydraulic test and observation wells.

17
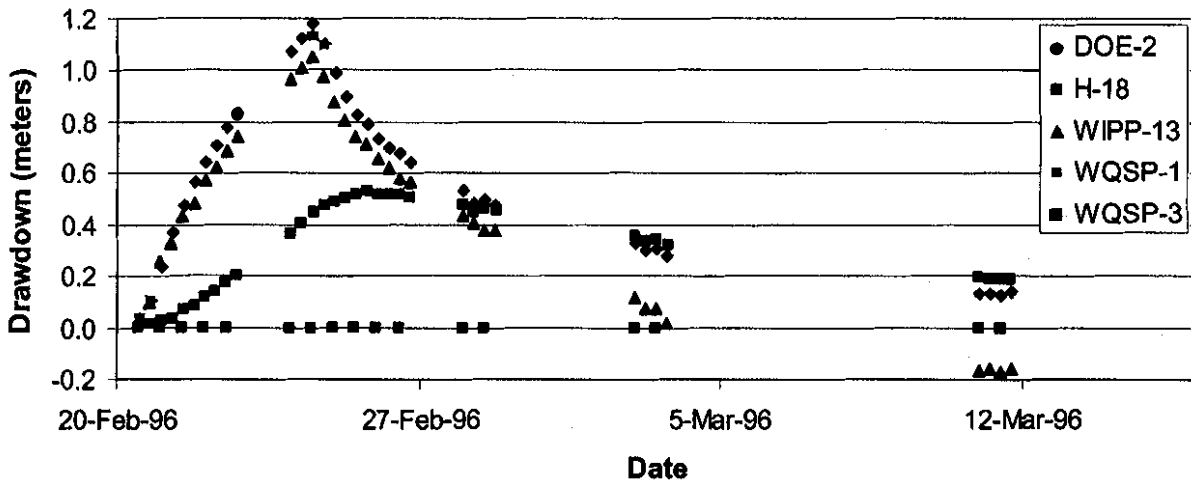
Figure 10. Observed drawdowns from the WQSP-2 hydraulic test.
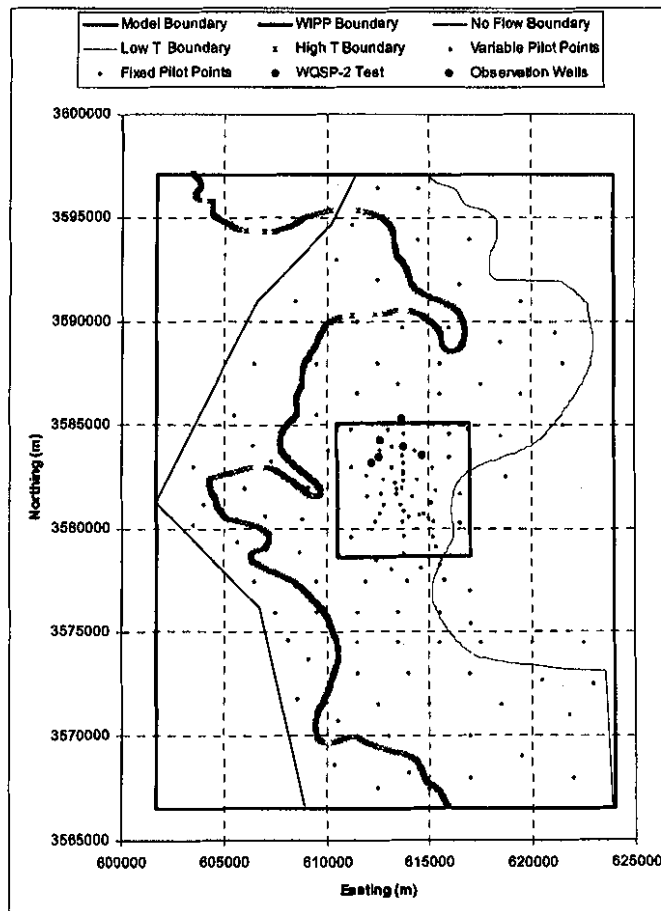
Figure 11. Locations of the WQSP-2 hydraulic test and observation wells.
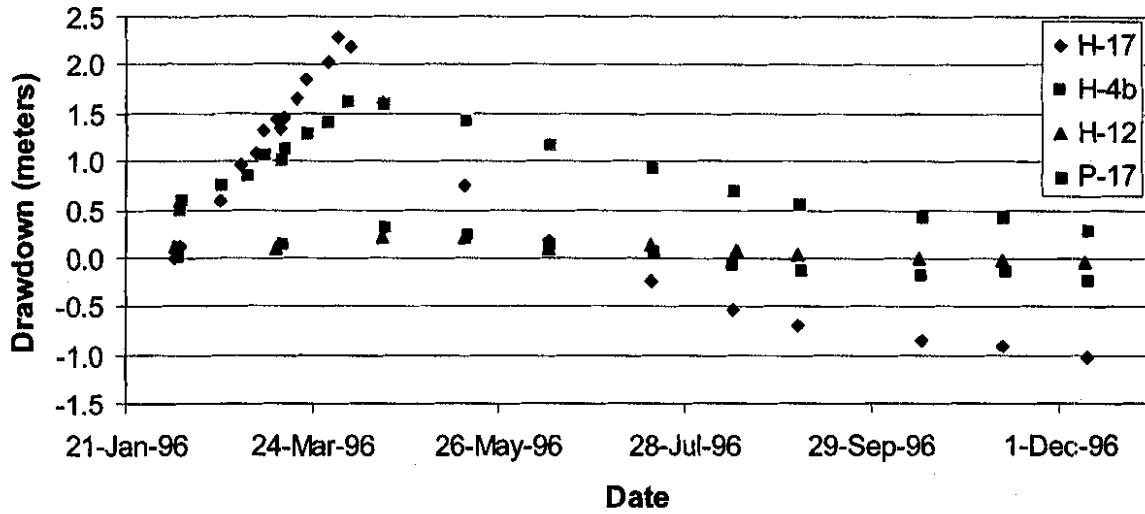
Figure 12. Observed drawdowns for the H-11 hydraulic test.
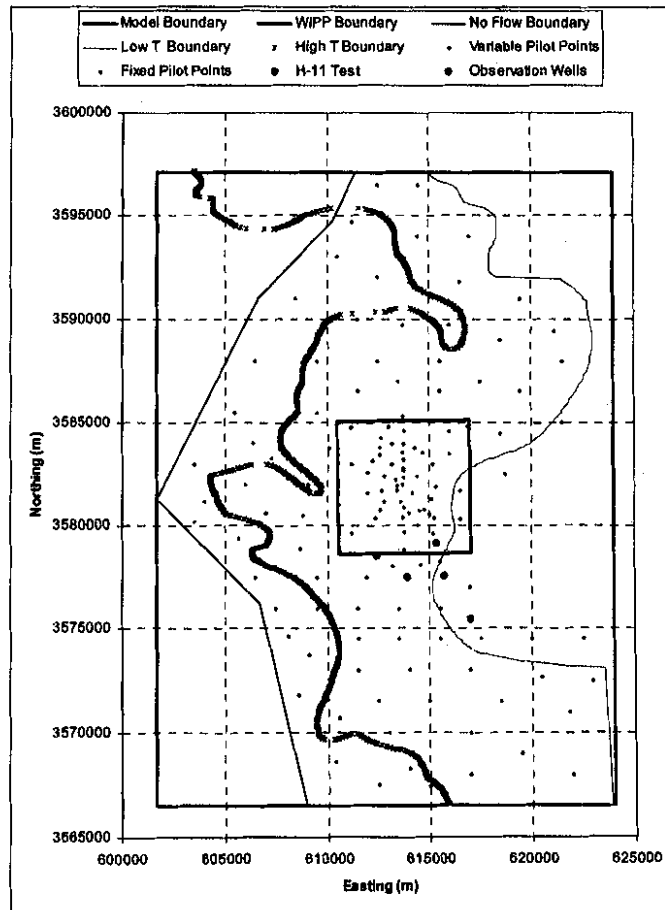
Figure 13. Locations of the H-11 hydraulic test and observation wells.

19
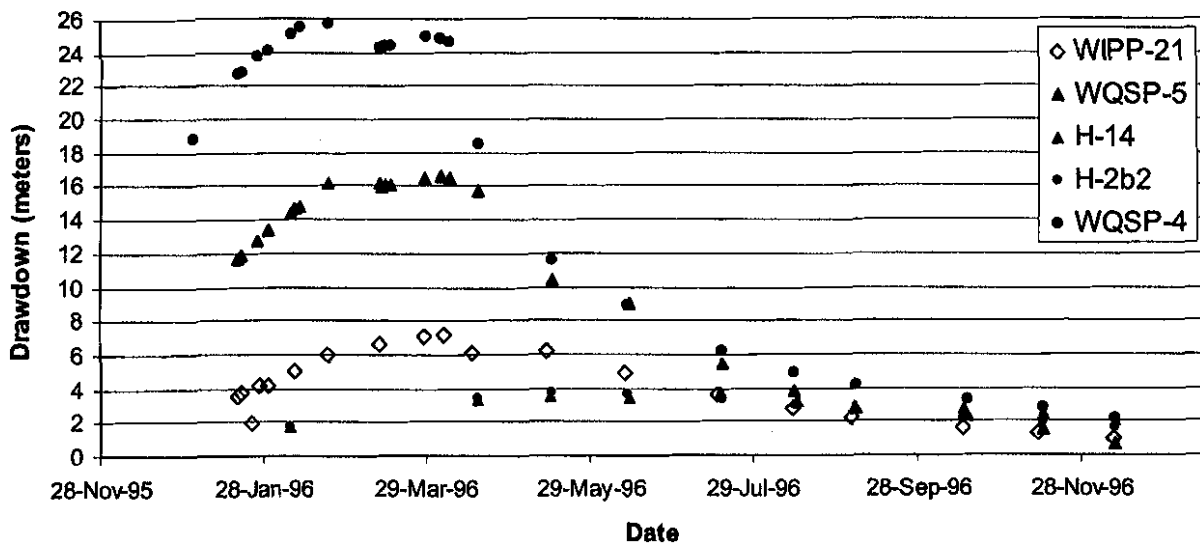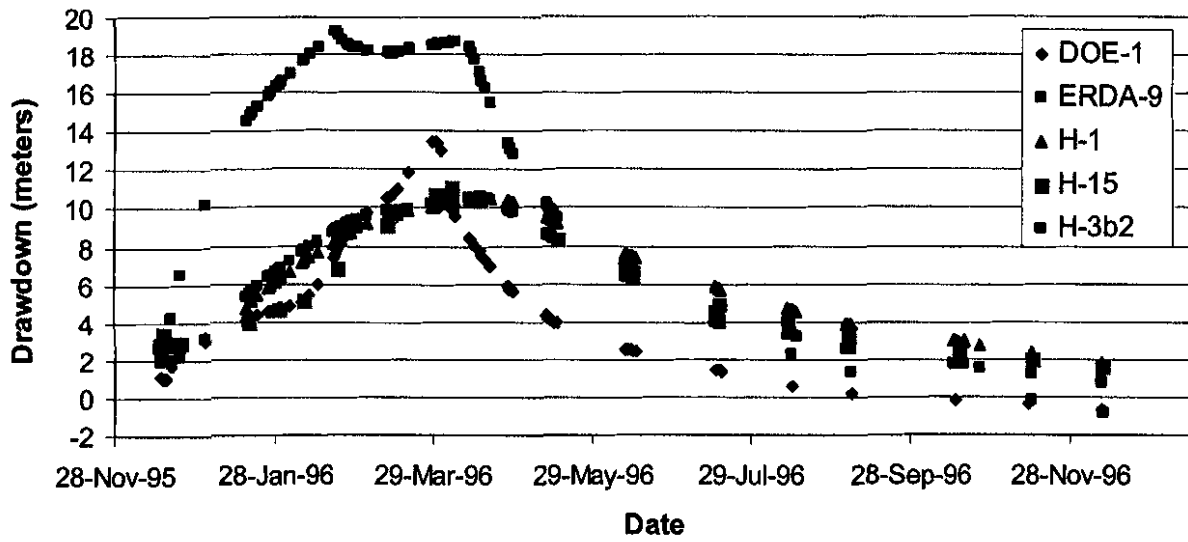
Figure 14. Observed drawdowns from the H-19 hydraulic test.

20
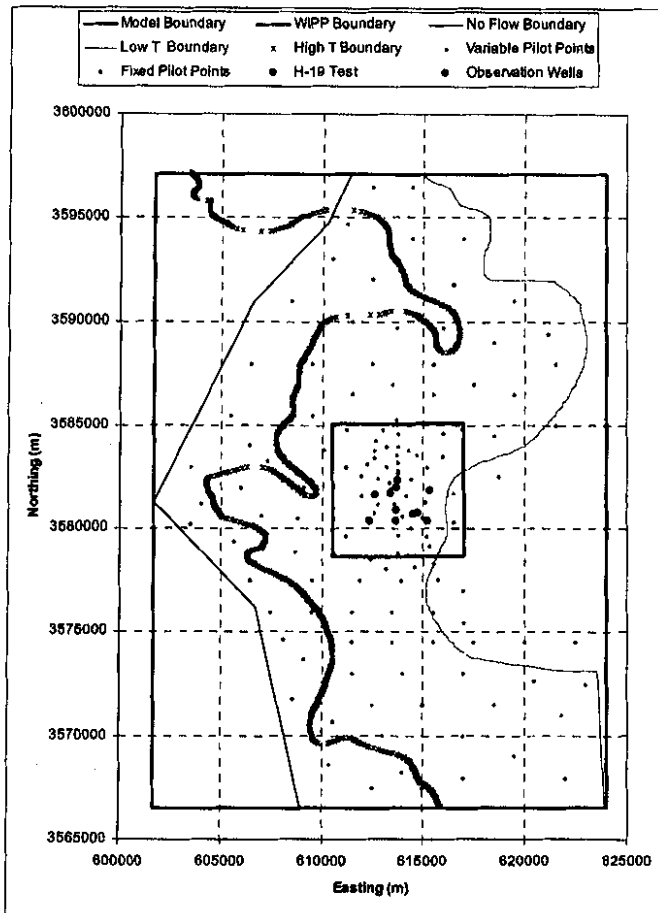
Figure 15. Locations of the H-19 hydraulic test and observation wells.

2   Modeling Approach

This section presents details on the modeling approach used to calibrate the transmissivity fields to both the 2000 steady-state heads and the 1,322 transient drawdown measurements. The assignment of boundary conditions, discretization of the spatial and temporal domain, weighting of the observations, and the use of PEST in combination with MODFLOW to calibrate the transmissivity fields are described. Changes in the modeling approach from that used to calibrate to the steady-state heads are discussed.

2.1   Boundary Conditions

The fixed-head boundary conditions are the same as those used in the calibration of the Culebra T fields to steady state head data for the 2000 time period (McKenna and Hart, 2003) with the exception of the change in the location of the no-flow boundary along the southern model domain boundary (see Section "1.4 Model Setup") that extends fixed heads further to the west than was necessary for the steady-state calibrations. The method used to create the fixed-head values involves fitting a bivariate Gaussian trend surface to the measured head data, determining the residuals between the measured heads and the trend surface, kriging the residuals throughout the domain, and then adding the trend back to the kriged residuals. These estimated heads are the initial heads and are the same for every base transmissivity field. Where these initial heads intersect a fixed-head boundary cell, the initial head value is maintained at that cell throughout the simulation. Details on this calculation were presented by McKenna and Hart (2003). The only change in the process for the transient calibrations is that the initial heads are mapped onto a grid with a 100X100 m^2 spatial discretization, whereas the spatial discretization of the grid for the steady-state simulations was 50X50 m^2.

A color scale map of the initial and boundary head values is shown in Figure 16. The fit of the bivariate Gaussian trend to the data is the same as that done for the steady-state calibrations. The results of fitting this trend are included as Appendix 2 for completeness. Appendix 2 is exactly the same as the final portion of Appendix 2 by McKenna and Hart (2003). The code **add_trend.c** (Appendix 3) is used to add the bivariate Gaussian trend back to the kriged residuals. This code was modified slightly from the steady-state calculations to accommodate the 100X100 m^2 spatial discretization of the grid used in the transient analyses. The **kt3d** software was used to krige the residuals throughout the model domain. The input file for **kt3d**, *kt3d.par*, is included as Appendix 4.

INFORMATION ONLY
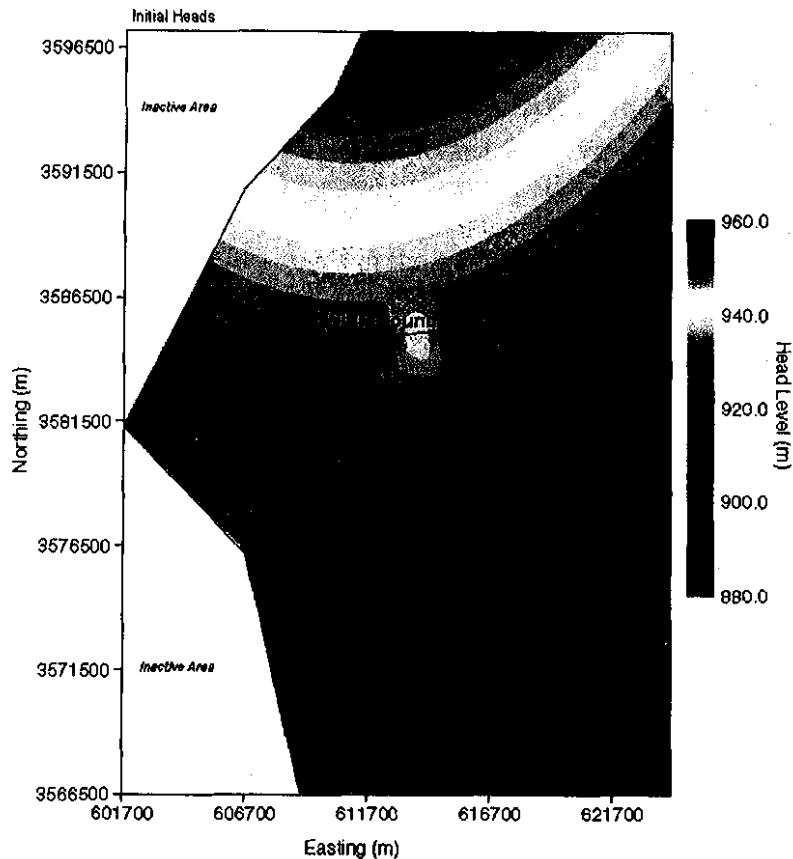
Figure 16. Map of initial heads created through kriging and used to assign fixed-head boundary conditions.

2.2 Spatial Discretization

The flow model is discretized into 68,768 regular, orthogonal cells each of which is 100X100 m^2. Details of the grid generation are described by Holt and Yarbrough (2003). A constant Culebra thickness of 7.75 m is used (U.S. DOE, 1996, Appendix TFIELD.4.1.1, Culebra:Thick). The 100-meter grid discretization was selected to make the finite-difference grid cell sizes considerably finer, on average, than those used in the CCA calculations, but still computationally tractable within the PA schedule. The cell size is a factor of 4 larger than the cells used to discretize the same model domain for the steady-state calculations (McKenna and Hart, 2003) and this increase in size is due to the increase in model run times in the transient calibration relative to the steady-state calibration. In the CCA calculations, a telescoping finite-difference grid was used with the smallest cell being approximately 100X100 m^2 near the center of the domain. The largest cells in the CCA flow model grid were approximately 800X800 m^2 near the edges of the domain (Lavenue, 1996).

INFORMATION ONLY

The elevation of the top of the Culebra was generated by Lance Yarbrough (University of Mississippi). The calculations performed to compute the top of the Culebra elevation surface are discussed by Holt and Yarbrough (2003).

Additional data on the extent of the Nash Draw high-transmissivity zone on the west side of the model have been added to the base transmissivity field construction (Holt and Yarbrough, 2003). These additional data have moved the southern edge of the no-flow boundary for these transient calculations to the west relative to the boundary location in the steady-state calculations (McKenna and Hart, 2003).

Of the 68,768 cells (224 east-west by 307 north-south), 14,999 (21.8%) lie to the west of the no-flow boundary, so the total number of active cells in the model is 53,769. This number is nearly a factor of 5 larger than the 10,800 (108X100) cells used in the CCA calculations.

2.3 Temporal Discretization

The time period of nearly 11 years and 2 months covered by the transient modeling begins October 15th, 1985 and ends December 11th, 1996 (Beauheim, 2002b). Additionally, a single steady-state calculation is run prior to the transient modeling. The length of this steady-state time period and the date at which it occurs are arbitrarily set to one day (86,400 s) occurring from October 14th, 1985 to October 15th, 1985. It is noted that these steady-state heads were measured in the year 2000 and are only set to these October dates to provide a steady-state solution prior to the start of any transient hydraulic events. The responses to the transient events are defined by the amount of drawdown relative to the initial steady-state solution. The discretization of this time interval is dictated by the pumping history of the different wells used in the hydraulic testing and the additional computational burden required for increasingly fine time discretization.

The groundwater flow model, **MODFLOW 2000 (MF2K)**, allows for the discretization of time into both "stress periods" and "time steps." A stress period is a length of time over which the boundary conditions and internal stresses on the system are constant. Even though these stresses are constant, this does not mean that the flow system is necessarily at steady state during the stress period. A time step is a subdivision of a stress period. System information such as the head or drawdown values are only calculated at the specified time steps. Each stress period must contain at least one time step. **MF2K** allows for the specification of the stress period length, the number of time steps in the stress period, and a time step multiplier. The time step multiplier increases the time between successive time steps geometrically. This geometric progression provides a nearly ideal time discretization for the start of a pumping or recovery period. To save on computational costs associated with calculating head/drawdown at each time step and with writing out the heads/drawdowns, the number of time steps in the model is kept to the minimum number possible that still adequately simulates the hydraulic tests. The time discretization in **MF2K** results in modeled heads calculated at times that may differ from the observation times. For this situation, the **PEST** utility, **mod2obs**, is used to interpolate the head, or drawdown, values in time from the simulation times to the observation times.

A summary of the time discretization is given in Table 5. There are five separate **MF2K** simulations for each complete forward simulation of the transient events. Each separate call to

INFORMATION ONLY 24

MF2K has its own set of input and output files. In Table 5, each call to MF2K is separated by a horizontal black line. The first call is the steady-state simulation. The second, third and fourth calls to MF2K (H-3, WIPP-13 and P-14) are all similar in that there is a single transient well that is pumped. For the H-3 and WIPP-13 calls, there are a total of 3 stress periods. In the first stress period, the well is pumping at a constant rate, in the second stress period, the pumped well is inactive and heads are recovering after the cessation of pumping, and the final stress period is simply a long time of no pumping activity used to advance the simulation time to be consistent with the calendar time. The first two stress periods are discretized using 8 time steps and the final stress period with no pumping activity is discretized using the minimum possible number of time steps, one.

The final MF2K call, the H-19 call, is considerably more complicated than the earlier calls to MF2K and simulates the hydraulic conditions during the H-11, H-19, WQSP-1 and WQSP-2 hydraulic tests. This final call contains 17 stress periods with as many as three different wells pumping during any single stress period. The pumping rates of the different wells in this call to MF2K and the stress periods are shown as a function of time in Figure 17. The first six stress periods in this call simulate pumping in the H-19 and H-11 wells without any observations (Table 5). These pumping periods are added to the model solely to account for the effects of these tests in observations of later hydraulic tests and therefore these tests can be modeled with a single time step. The pumping rates shown in Figure 17 are given as negative values to indicate the removal of water from the Culebra following the convention used in MF2K.

The MF2K simulations could be done using a single call to MF2K, but five separate calls were used here. Each of the five calls creates separate binary output files of drawdown and head that are much smaller and easier to manage than would be a single output file. Additionally, the simulated drawdowns at the start of each transient test must be zero (no drawdown prior to pumping). Because MF2K uses the resulting drawdowns and heads from the previous stress period as input to the next stress period, a single simulation would not necessarily start each transient test with zero drawdowns. Calling MF2K five times allows the initial drawdowns to be reset to zero each time using the shell scripts in Appendix 1. The heads simulated at the end of the final time step in each MF2K are used as the initial heads for the next call. The results of all five calls are combined to produce the 1332 model predictions prior to comparing them to the 1332 selected observation data thus ensuring that all steady-state and transient data are used simultaneously in the inverse calibration procedure.

Table 5. Discretization of time into 29 stress periods and 127 time steps with pumping well names and pumping rates.

Event Name	Global Stress Period No.	Internal Stress Period No.	Stress Period Length (secs)	No. of Time Steps	Start Date	Stop Date	Pumping Well(s)	Pumping Rate(s) (m3/s)
Steady	1	1	86400	1	10/14/85 9:00	10/15/85 9:00	0	0
H-3	2	1	5356800	8	10/15/85 9:00	12/16/85 9:00	H-3	3.03E-04
	3	2	10892700	8	12/16/85 9:00	4/21/86 10:45	None	0.00E+00
	4	3	22976100	1	4/21/86 10:45	1/12/87 9:00	None	0.00E+00
WIPP-13	5	1	3110400	8	1/12/87 9:00	2/17/87 9:00	WIPP-13	1.89E-03
	6	2	7539900	8	2/17/87 9:00	5/15/87 15:25	None	0.00E+00
	7	3	55359360	1	5/15/87 15:25	2/14/89 9:01	None	0.00E+00
P-14	8	1	44928	3	2/14/89 9:01	2/14/89 21:29	P-14	3.92E-03
	9	2	174612	8	2/14/89 21:29	2/16/89 22:00	P-14	3.64E-03
	10	3	50400	3	2/16/89 22:00	2/17/89 12:00	P-14	3.37E-03
	11	4	1820396	8	2/17/89 12:00	3/10/89 13:39	None	0.00E+00
	12	5	193212124	1	3/10/89 13:39	4/24/95 19:42	None	0.00E+00
H-19	13	1	148860	1	4/24/95 19:42	4/26/95 13:03	H-19b0	2.26E-04
	14	2	4399020	1	4/26/95 13:03	6/16/95 11:00	None	0.00E+00
	15	3	3614400	1	6/16/95 11:00	7/28/95 7:00	H-19b0	2.36E-04
	16	4	1168200	1	7/28/95 7:00	8/10/95 19:30	None	0.00E+00
	17	5	1292700	1	8/10/95 19:30	8/25/95 18:35	H11	2.44E-04
	18	6	9651300	1	8/25/95 18:35	12/15/95 11:30	None	0.00E+00
	19	7	2878200	8	12/15/95 11:30	1/17/96 19:00	H-19b0	2.71E-04
	20	8	670680	3	1/17/96 19:00	1/25/96 13:18	H-19b0	2.52E-04
	21	9	238980	3	1/25/96 13:18	1/28/96 7:41	H-19b0, WQSP-1	2.52E-04, 4.30E-04
	22	10	872340	3	1/28/96 7:41	2/7/96 10:00	H-19b0	2.52E-04
	23	11	1047000	8	2/7/96 10:00	2/19/96 12:50	H-19b0, H-11	2.52E-04, 2.23E-04
	24	12	81600	3	2/19/96 12:50	2/20/96 11:30	H-19b0, H-11	1.55E-04, 2.23E-04
	25	13	345600	3	2/20/96 11:30	2/24/96 11:30	H-19b0, H-11, WQSP-2	1.55E-04, 2.23E-04, 4.5E-04
	26	14	1395000	8	2/24/96 11:30	3/11/96 15:00	H-19b0, H-11	1.55E-04, 2.23E-04
	27	15	1445100	8	3/11/96 15:00	3/28/96 8:25	H-19b0, H-11	1.55E-04, 3.76E-04
	28	16	1220700	8	3/28/96 8:25	4/11/96 11:30	H-19b0	1.55E-04
	29	17	21074400	8	4/11/96 11:30	12/11/96 9:30	None	0.00E+00

26
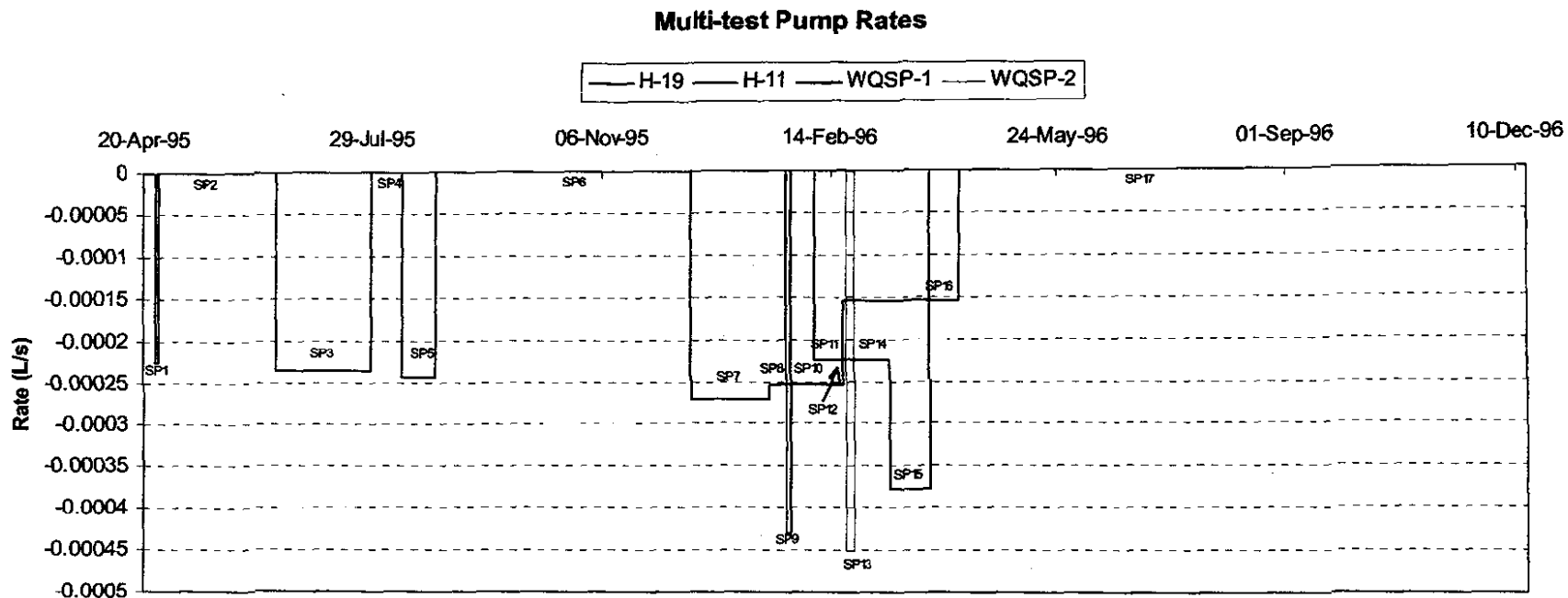
Figure 17. Temporal discretization and pumping rates for the fifth call to **MF2K**. A total of 17 stress periods ("SP") are used to discretize this model call.

27

2.4 Weighting of Observation Data

The observed data for each response to each transient hydraulic test are weighted to take into account the differences in the response across the different tests. The weights are calculated as the inverse of the maximum observed drawdown for each hydraulic test. This weighting scheme applies relatively less weight to tests with large drawdowns and relatively more weight to tests with smaller responses. This weighting scheme was used so that the overall calibration was not dominated by trying to reduce the very large residuals that may occur at a few of the observation locations with very large drawdowns. Under this weighting scheme, two tests that are both fit by the model to within 50 percent of the observed drawdown values would be given equal consideration in the calculation of the overall objective function even though one test may have an observed maximum drawdown of 10 meters and the other a maximum observed drawdown of 0.10 meters.

The weights assigned in this manner ranged from 0.052 to 20.19 with units of (1/m). The observed absence of a hydraulic response at WQSP-3 to pumping at WQSP-1 and WQSP-2 is also included in the calibration process by inserting "measurements" of zero drawdown that were given an arbitrarily high weight of 20. Through trial and error using the root mean squared error criterion of how well the modeled steady-state heads fit the observed steady-state heads, a weight of 2.273 is assigned to the 35 steady-state observations. This weight is near that of the average of all the weights assigned to the transient events and was found to be adequate to provide acceptable steady-state matches. It is noted that the steady-state data provide measurements of head while all of the transient events provide measurements of drawdown. However, the weights are applied to the residuals between the observed and modeled aquifer responses and because both heads and drawdowns are measured in meters, there was no need to adjust the weights to account for different measurement units.

The number of measurements made at individual wells during individual tests range from six to 104, and the number of measurements made at all wells during a single test range from 64 to 410. This means that different well responses and different tests carry different cumulative weights. Some areas of the modeling domain are covered by multiple well responses, while other areas of the domain have no transient response data. This means that some areas of the T field are most likely calibrated better than other areas and some areas of the domain are calibrated solely by the observed "steady-state" measurements.

The maximum observed drawdown, the weight assigned to all the observed test values for each test, and the total number of observations for each observation well are given in Table 6.

Table 6. Observation weights for each of the observation wells.

Test Well Observation Well	Maximum Drawdown (m)	Weight	Number of Observations
Steady	NA	2.273	35
H3-DOE1	5.426	0.184	57
H3-H1	10.396	0.096	26
H3-H11b1	3.622	0.276	19
H3-H2b2	3.781	0.265	20
W13-DOE2	12.138	0.082	104
W13-H2b2	0.781	1.281	23
W13-H6	5.545	0.180	93
W13-P14	0.570	1.755	38
W13-W12	1.553	0.644	27
W13-W18	6.481	0.154	26
W13-W19	5.048	0.198	22
W13-W25	0.246	4.062	11
W13-W30	3.391	0.295	24
P14-D268	0.432	2.317	38
P14-H18	0.113	8.850	21
P14-H6b	0.701	1.427	21
P14-W25	0.432	2.315	22
P14-W26	0.137	7.310	21
WQSP1-H18	1.431	0.699	47
WQSP1-W13	1.260	0.794	47
WQSP1-WQSP3	0.000	20.000	25
WQSP2-DOE2	1.178	0.849	34
WQSP2-H18	0.529	1.892	34
WQSP2-W13	1.053	0.949	34
WQSP2-WQSP1	1.132	0.884	6
WQSP2-WQSP3	0.050	20.000	18
H11-H17	1.030	0.971	23
H11-H4b	0.232	4.317	11
H11-H12	0.033	20.190	11
H11-P17	1.628	3.304	19
H19-DOE1	13.463	0.074	70
H19-ERDA9	10.571	0.095	80
H19-H1	10.618	0.094	80
H19-H15	11.110	0.090	22
H19-H3b2	19.283	0.052	69
H19-W21	7.153	0.140	19
H19-WQSP5	16.623	0.060	24
H19-H14	3.759	0.602	11
H19-H2b2	3.794	0.608	11
H19-WQSP4	25.721	0.462	24

29

2.5 Pilot Point Calibration

The calibration process proceeds in the same manner as for the steady-state calibration as described by McKenna and Hart (2003). This process creates a residual field that when added to the base transmissivity field reproduces the measured transmissivity values at the 43 measurement locations. The pilot points are then adjusted by **PEST** to update the residual field such that when the updated residual field is again added to the base transmissivity field, the fit to the observed head and drawdown data is improved relative to previous iterations of the model. The objective function to be minimized by PEST is the weighted sum of the squared errors (SSE) between the observed heads/drawdowns and the model predicted heads/drawdowns. This is the same objective function as that used in the steady-state calibrations (McKenna and Hart, 2003). In this transient calibration process, for each iteration a single steady-state solution is calculated and then multiple calls to **MF2K** are made, generally one solution for each transient pumping test. This combined set of steady state and transient runs allows for the simultaneous calibration of the transmissivity field to the steady-state heads observed in 2000 as well as to multiple pumping tests. The computational cost of calibrating to the multiple transient events is significant. For comparison, a single forward run of **MF2K** in steady-state takes on the order of 10-15 seconds whereas the run time for the combined steady-state and transient events is approximately 3 minutes (a factor of 12-18 times longer).

Due to these longer run times, two separate parallel PC clusters were employed. Each of these clusters consists of 16 computational nodes. One cluster is located in Albuquerque and the other is in the Sandia office in Carlsbad. Both clusters use the linux operating system. The total number of forward runs necessary to complete the calibration process can be estimated as:

Total Runs $\cong$ (# of parameters)X(#of PEST iterations)X(average runs per iteration)X(# of base transmissivity fields).

The maximum number of iterations used in these runs was set to 15, although not all fields went to the maximum number of iterations. Additionally, on average for the first 4 iterations, **PEST** used forward derivatives to calculate the entries of the Jacobian matrix and each entry only requires a single forward model evaluation. For the remaining 11 iterations, **PEST** uses central derivatives to calculate the Jacobian entries and each calculation requires two forward evaluations of the model (22 total). So, the average number of model evaluations is 1.733 = [(4+22)/15]. Therefore an estimate of the maximum possible total number of forward runs is equal to: 100X15X1.73X150 = 390,000. The total time necessary to complete these calculations in serial mode on a single processor would be 813 days, or 2.22 years. By employing parallel computation with 32 processors, this run time was cut to several months.

The model run times as well as the time necessary to read and write input output files across the cluster network were examined to determine the optimal number of client, or slave, nodes for each server, or master, node. The optimal number of clients per server was determined to be eight. More clients per server degraded overall performance due to increased communication between machines and fewer clients per server results in underutilization of the system. By combining the client and server activities on a single machine using a virtual server setup, a total

of 32 machines were necessary to calibrate four different base transmissivity fields simultaneously.

The initial residual fields are created using the geostatistical simulation code **sgsim**. The input parameters for this code, including the parameters defining the variogram of residuals between the measured transmissivity values and the values in the base transmissivity fields, are exactly the same as those used for the steady-state calibrations (McKenna and Hart, 2003, "Creation of Seed Transmissivity Fields" section of Subtask 2). The only change in the input parameter file for **sgsim** is the change from a $50X50$ m^2 grid to a $100X100$ m^2 grid. This change is made on lines 19 and 20 of the *sgsim.par* input file and an example of this file is shown in Appendix 5.

As was done for the steady-state calculations (McKenna and Hart, 2003), calculations of the residuals and the transmissivity fields are done in $\log_{10}$ space so that a unit change in the residual equates to a one order of magnitude change in the value of the transmissivity. The initial values of the pilot points are equal to the value of the initial residual field at each pilot point location. The pilot points are constrained to have a maximum perturbation of ± 3.0 from the initial value except for those pilot points within the high-transmissivity zone in Nash Draw and those in the low-transmissivity zone on the east side of the domain (Figure 18) (see Holt and Yarbrough, 2003) that are limited to perturbations of ± 1.0. These limits are employed to maintain the influence of the geologic conceptual model on the calibrated transmissivity fields.

A total of 100 pilot points are used in the calibration process. This is a slight decrease from the number used in the steady-state calibrations (115), and this decrease in the number of pilot points was made to improve computation time for the overall calibration process. The pilot point locations were chosen using a combination of a regular grid approach and deviations from that grid to accommodate specific pumping and observation well locations (Figure 18). The goal in these deviations from the regular grid was to put at least one pilot point between the pumping well and each observation well. Details of the pilot point locations relative to the pumping and observation wells in the WIPP site area are shown in Figure 19. This combined approach of a regular grid with specific deviations from that grid follows the guidelines for pilot point placement put forth by John Doherty as Appendix 1 in the work of McKenna and Hart (2003).

One change from the steady-state calibrations is that nine pilot points have been added to the east side of the low-transmissivity zone boundary (Figure 18). These points were added to allow **PEST** to adjust values within the low transmissivity zone. The zone option in **PEST** is employed to limit the influence of pilot points in any one zone to adjusting only locations that are in the same zone. This zone option was also used in the steady-state calibrations. Figure 19 shows, that to the extent possible, for each pumping well – observation well pair, at least one pilot point was located between the pumping and observation wells.

The variogram model for the residuals is the same as that used for the steady-state runs (McKenna and Hart, 2003; Figure 13). This variogram model has a range of 1,050 meters. Because the pilot point approach to calibration uses this range as a radius of influence, locations of the adjustable pilot points were as much as possible set to be at least 1,050 meters away from other pilot points (adjustable or fixed). For maximum impact, all pilot points should be at least

2100 meters away from any other pilot point but, given the existing well geometry, this distance is not always achievable.
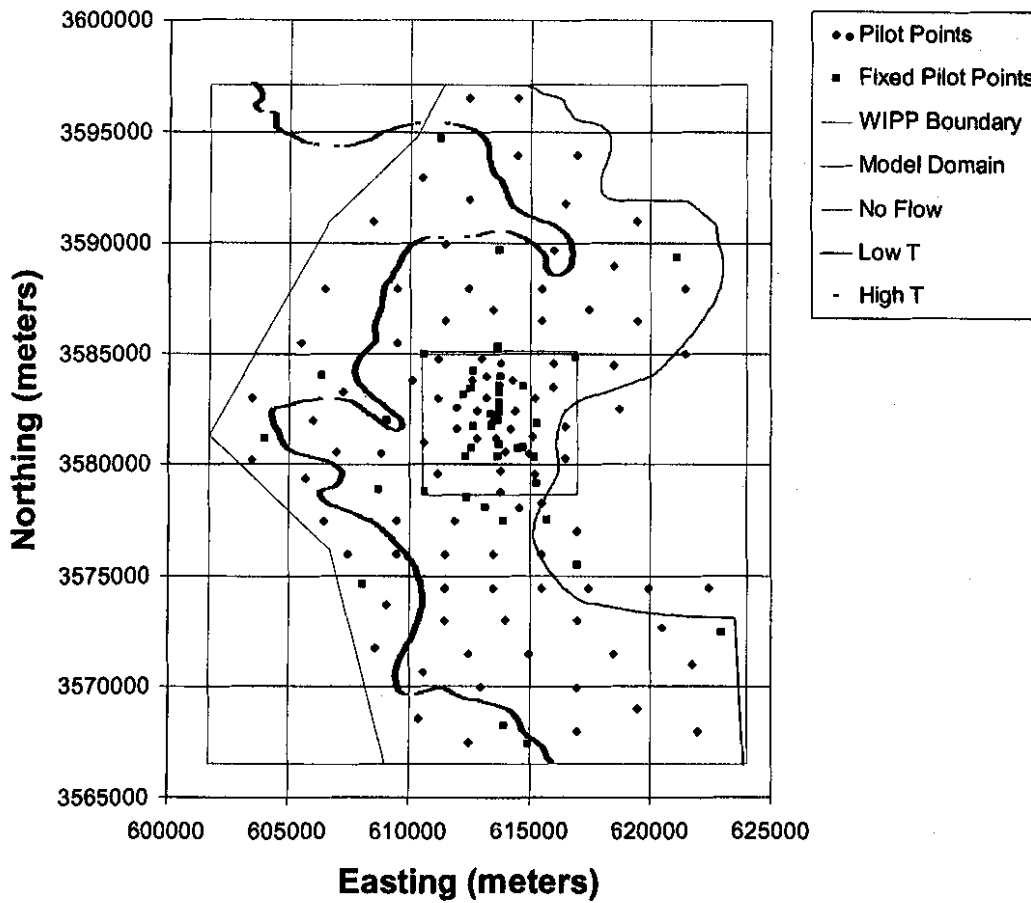
Figure 18. Location of the adjustable and fixed pilot points within the model domain.
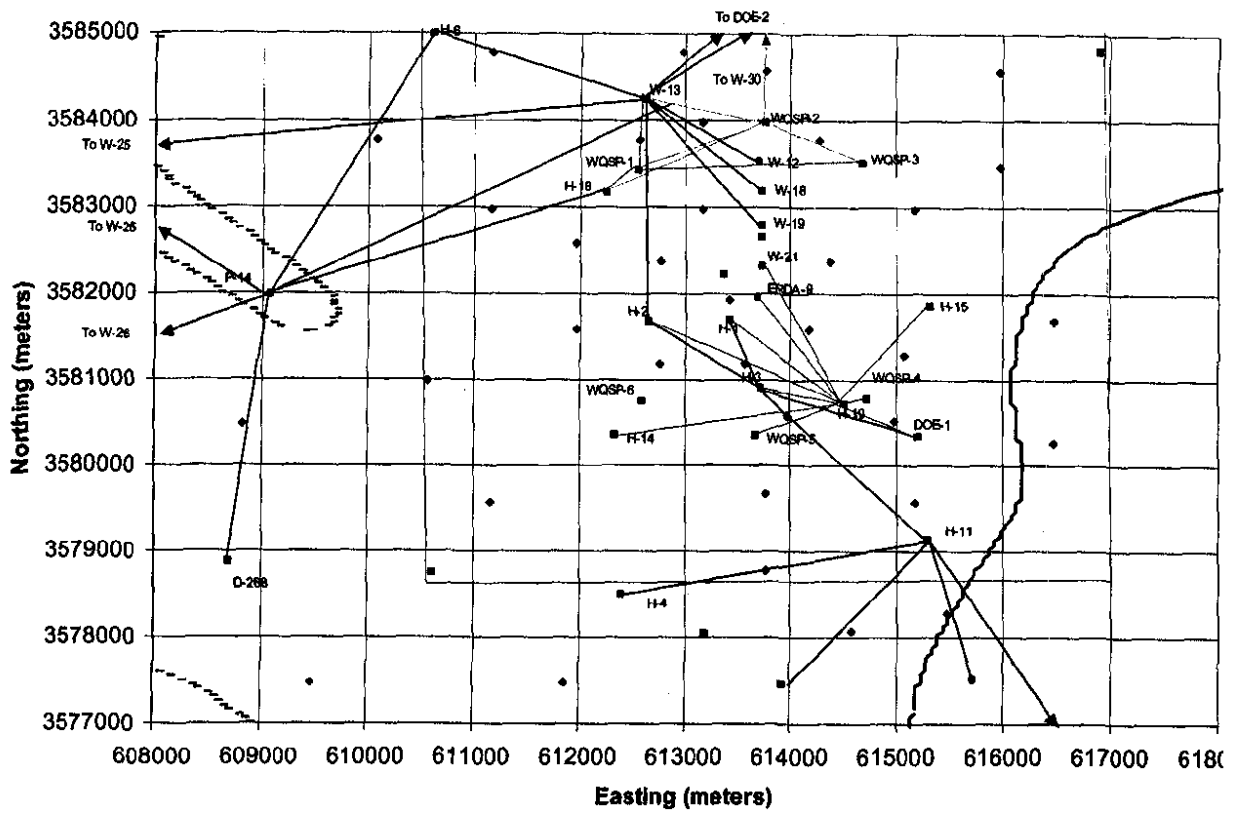
Figure 19. Close-up view of the pilot point locations in the area of the WIPP site. The colored lines connect the pumping and observation wells. The legend for this figure is the same as that used in Figure 18.

33

The mechanics of actually creating the calibrated transmissivity fields are controlled by a series of shells and small programs. All of these shells and programs are in the /h/wipp/data directory on both the Albuquerque and Carlsbad linux clusters. All calculations completed on the Carlsbad cluster were copied from there to the /h/wipp/data directory on the Albuquerque cluster so that the complete set of results exists on the Albuquerque cluster. The executable versions of the shells used in these calculations are contained in the /h/wipp/bin directory.

The first step in the calculations is to set up a subdirectory for each transmissivity field. This step is accomplished by running the *setupRealization* (Appendix 6) shell with the requested realization name(s) as an input argument. The realization names use the d##r## naming convention use by Holt and Yarbrough (2003) to name each base transmissivity field. The *setupRealization* shell is called from the /h/wipp/data directory and creates subdirectories, one for each base field, below this working directory. This shell also calls three other shells to complete other pieces of the model setup. The interactions between these different shells are shown schematically in Figure 20.
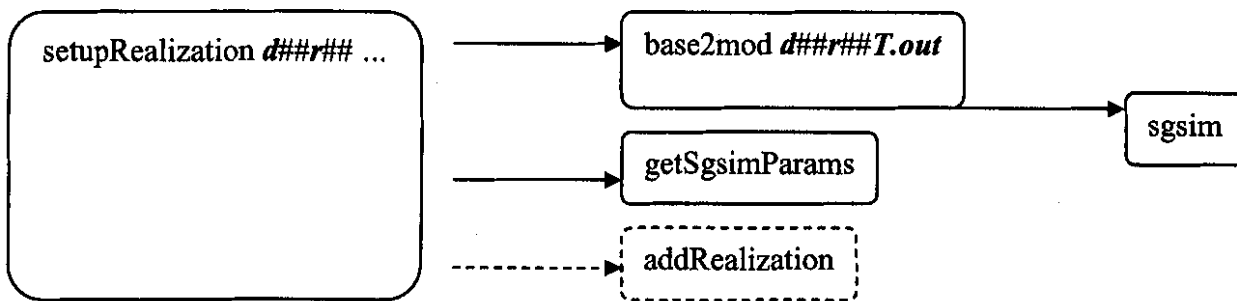
Figure 20. Schematic flowchart of the shells used to set up the run for a new base field.

The three additional shells called by *setupRealization* are: base2mod, getSgsimParams and addRealization. The functions of these three shells are:

1) base2mod (Appendix 7) reads the existing base transmissivity field that is currently formatted for viewing in ArcInfo and reformats the base transmissivity data into one that can be read by MF2K.

2) getSgsimParams (Appendix 8) creates a new *sgsim.par* file from the *sgsim.par.tpl* file. The only change to the template file is the value of the random number seed so that a unique residual field is created when **sgsim** is run. This shell makes a system call to **sgsim** to run it and saves the resulting *sgsim.out* file as the initial residual field. The newly created *sgsim.par* file is also saved. The getSgsimParams shell also creates a portion of the **PEST** control file, the **.pst* file, and saves it in the *ppoints.pcf_add* file. This file contains the initial value of the residual field at each of the pilot points, the transmissivity zones to which each pilot point belongs (high, middle, or low) and the bounds on the possible pilot point values. An initial residual field value of zero corresponds to the value obtained from the base transmissivity field. For the high and low transmissivity zones, the bounds on the possible pilot point values are set to −1.0 and +1.0 and for the middle transmissivity zone, the bounds are set to −3.0 and +3.0. The

34

input files to this shell that contain the pilot point locations and the zone definitions respectively are *ppoints.nodes* and *ppoints.zones*.

3) addRealization (Appendix 9) simply adds a realization number (i.e., d##r##) to the list of realizations waiting to be run. This shell is only involved in the queuing of future runs and does not affect the calibration process as indicated by the dashed lines in Figure 20.

After the subdirectories are set up for the calibration of a base transmissivity field, the main portion of the calibration is controlled by a series of shells as shown schematically in Figure 21. These shells are designed for the creation and maintenance of all the directories necessary for the calibration being done using parallel pest, or **ppest**. Use of parallel pest requires that both master and slave directories with corresponding master and slave computational nodes be assigned and that communications between the directories and nodes be maintained during the calibration process. The functions of these different shells are:

1) runWIPPTrans (Appendix 10) is a perl script that checks, once per hour, for the correct number of available idle slave nodes needed to be able to start a new calibration. If the necessary number of idle slaves is available then runWIPPtrans begins the next calibration. This check could be done by hand but would require someone to monitor the system 24-hours per day. The dashed lines in Figure 21 indicate that this shell does not directly influence the calibration process other than allowing it to continue.

2) runPest (Appendix 11) is a run management script that enters each slave subdirectory below the /d##r## level and starts a **PEST** slave, **pslave**, run, then starts a pmaster, **ppest**, run in the main directory. These activities are considered to be part of the model setup and do not directly affect the calibration process as shown by the dashed line in Figure 21.

3) pmaster.sh (Appendix 12) is a shell that automates running the master computational node and one slave. The slave controlled by this shell is actually a virtual slave as it uses the same computational node as does the master process. This shell also does the final post calibration forward model run and particle tracking on the results of that run. The particle-tracking code, **DTRKMF**, is called twice by this shell to track the particle to the boundary of the model domain (first call) and then to the WIPP site boundary (second call). This shell also takes care of renaming some of the output files from the generic names used in the calibration process (e.g., transient.*) to the final names that incorporate the realization name (e.g., d##r##). This renaming is done by using the *ln* ("hard link") command that is an intrinsic function in linux.

4) pslave (Appendix 13) is a shell that runs the **PEST** slave program on a slave computational node in a slave directory. The actual call to **MF2K** is made in the model.sh shell, discussed below, that is called by **pslave**. The **pslave** shell must be called within each of the 7 slave subdirectories. This shell has a similar function to that of pmaster but is not responsible for any of the final forward run functions nor any of the renaming of final files.

5) The shells and programs shown in the right column of Figure 21 are called from within model.sh and these are described in detail next.
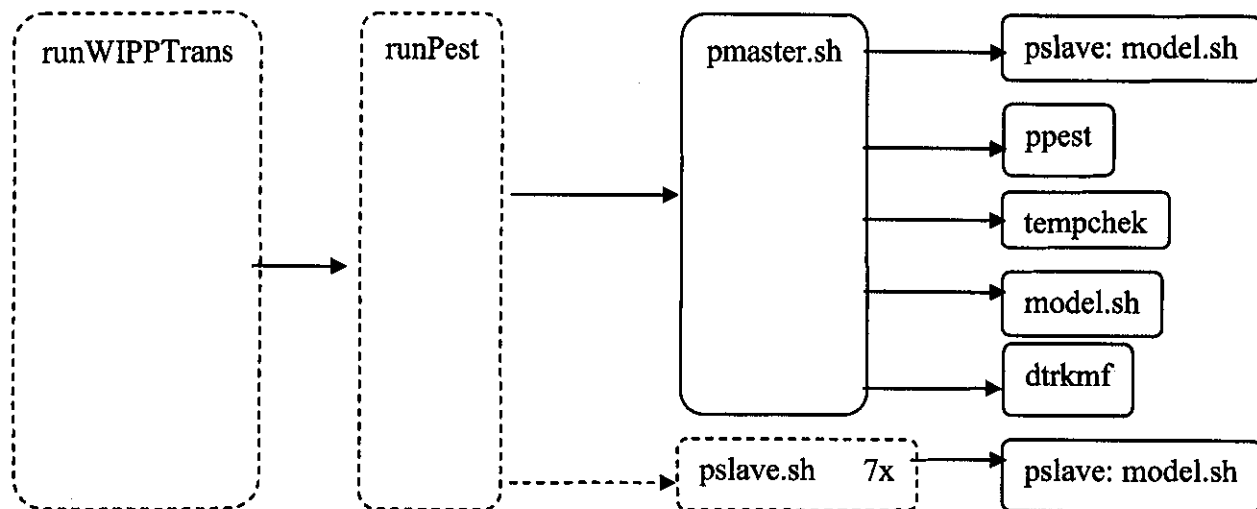
Figure 21. Schematic flow chart of the process used to setup and run the master and slave processes under control of **PEST**.

The calibration setup and initialized in the shells described above is completed by running **MF2K** under **PEST**. This modeling is controlled by the *model.sh* shell (Appendix 14). This shell file is actually composed of three separate functions. The modeling steps controlled by this shell and the outside calls to other programs or shells for each step are:

Function ResetTol: This function resets the tolerance in the **MF2K** *.lmg* file. At the start of every **MF2K** run, the tolerance on the solver is reset to the base level of $1X10^{-08}$. Each different input transmissivity field may prove more or less difficult to solve depending on the arrangement of the transmissivity values. The base tolerance of $1X10^{-08}$ is a relatively small tolerance and if **MF2K** can converge to this tolerance, the mass balance error in the flow model is always less than 0.01 percent.

Function RaiseTol: This function increases the solution tolerance in the *.lmg* file if **MF2K** was not able to converge to a solution with the current pilot point values. The maximum permissible tolerance is $1X10^{-02}$. If **MF2K** cannot converge with this maximum tolerance, then the calibration for this transmissivity field is terminated.

Function runMF2K (the main driver function):
    Step 0: Write the current value of the **MF2K** tolerance to the *.lmg* file.

    Step 1: Delete intermediate files from the previous **MF2K** run

    Step 2: Call the **PEST** utility code **fac2real** to create the current version of the residual field using the updated pilot point values.

Step 3: Call the **combine** code (Appendix 15) to add the updated version of the residual field created in Step 3 to the base transmissivity field and create the current version of the transmissivity field to be used in **MF2K**.

Step 4: Run **MF2K** six times, twice at steady state and then once for each transient test (H-3, WIPP-13 and P-14) and once for the WQSP-1, WQSP-2, H-11 and H-19 tests (all four of these are included in the "h19" run of **MF2K**). The reason that two steady-state solutions are calculated is due to different formats of output files that are needed in downstream calculations. The **PEST** utility code, **mod2obs**, requires **MF2K** output in binary format (the *steady.bin* run) while subsequent runs of **MF2K** require an initial head file that is in ascii format (the *steady* run).

Step 5: Call **mod2obs** 8 times to strip out the modeled drawdowns at the correct times and locations for comparison with the observed drawdown values.

Step 6: Use the intrinsic UNIX command *awk* to strip out the fourth column of the modeled heads files created in Step 5. The files created in Step 5 have additional columns for the well ID and the date and time and these columns are not needed in the next step.

Step 7: Call the correct perl shell (Appendix 1) to normalize the drawdown values to zero starting values.

Step 8: Collect the **MF2K** mass balance error information and write it to a file.

The "do" loop at the bottom of the model.sh file is called if **MF2K** fails to converge on the current transmissivity field. The tolerance in the **.lmg* file is raised by an order of magnitude and **MF2K** is called again. The tolerance continues to be raised by an order of magnitude until **MF2K** converges or until the tolerance reaches the maximum allowable tolerance value of 1.0E-02. If **MF2K** cannot converge with this maximum tolerance value, then a "could not converge" statement is printed to the screen and the calibration is over for this field.

This shell takes the current values of the pilot points and does the kriging to adjust the values surrounding the pilot points, then adds the kriged residual field to the base field to produce the current transmissivity field, runs **MF2K** using the current transmissivity field as input and parses the results of the **MF2K** run into the correct files, all while providing some measure of error checking for the current model.

Checks on the calibration process showed that the results were consistently insensitive to the value of pilot point 30 at location (615475, 3575975). The calculated sensitivity values for this pilot point were generally 10 orders of magnitude less than the sensitivity of the calibration to the other pilot points. This type of extreme parameter insensitivity can lead to numerical stability problems with the inverse solution. Therefore, partway through the calibration process, the value of pilot point 30 was fixed at its initial value. For realizations d04r04 through d04r10 and then all realizations from d06r01 forward, the value of pilot point 30 was fixed.

The reason the calibrations were so insensitive to pilot point 30 is not clear. However, pilot point 30 is located just inside the low-transmissivity boundary on the east side of the domain. The reason that the calibrations are so insensitive to the value of this pilot point may be due to the proximity of this zone boundary to the pilot point and the modeling set up that limits the influence of each pilot point to only other grid cells within the same zone. Given that the results were insensitive to the value of pilot point 30, fixing the value of this pilot point did not affect the subsequent calibrations.
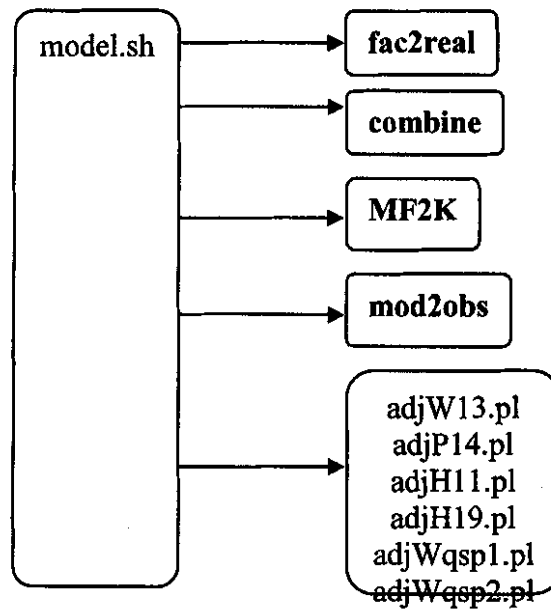
Figure 22. Schematic diagram of the flowchart showing the calls made by the model.sh program.

2.6 Particle Tracking

The final transmissivity field estimated through the calibration process is used as the basis for the calculation of travel time from the center of the repository to the WIPP boundary. The mechanics of this calculation within the overall calibration framework are discussed above. This travel time calculation is accomplished using a streamline particle-tracking algorithm as implemented in the **DTRKMF** software. For each calibrated T field, a final forward run of **MF2K** is done and the cell-by-cell fluxes from this run are stored in the *.bud file (the *budget* file). The *.bud and the model discretization (*.dis) file from **MF2K** are used as input to **DTRKMF** to calculate the travel time. For each calibrated T field, only a single particle is tracked providing a single travel time.

The starting UTM coordinates of the particle at the center of the repository are: X = 613,597.5 meters and Y = 3,581,385.2 meters (Ramsey et al., 1996, p. 9). The porosity used for the travel time calculations is 0.16, the same value used in the steady-state calibrations (McKenna and Hart, 2003). The particle is tracked to both the boundary of the WIPP site and to the boundary of the model domain.

2.7 File Naming Convention

A relatively large number of programs, shells, and files is needed to accomplish each transmissivity field calibration. Each transmissivity field calibration is completed within its own subdirectory. The general path for any of these subdirectories is: /h/wipp/data/d##r## where the d##r## is the original base transmissivity field naming convention (Holt and Yarbrough, 2003). All of the files that remain within each subdirectory are listed and described in Table 7. Additional intermediate files (e.g., each drawdown output array at each time step from **MF2K**) and intermediate subdirectories (e.g., the **PEST** slave subdirectories) are deleted at the end of the calibration process and are not included in Table 7. Table 7 is provided as an aid in understanding the different pieces of the calibration process.

Table 7. File listing and descriptions within a calibration subdirectory.

File Prefix/Suffix	File Definition
d##r##.mod	The final calibrated transmissivity field values in MODFLOW format.
d##r##T.out	Original base T field in 4 column ARC-INFO format (input to base2mod program)
d##r##.par	The final values of the estimated pilot points (output from PEST)
d##r##.rec	Output record file from PEST
d##r##.res	Residuals output file from PEST
*.ba6	MODFLOW input basic package file
*.bc6	MODFLOW input block centered-flow package file
base2mod.set	Input control file for the base2mod program
*.bud	MODFLOW cell by cell budget output file
combine.set	Input control file for the combine program
control.inp	DTRKMF input control file for particle track to model boundary

culebra.bot	Elevations of the bottom of the Culebra in MODFLOW format (input to MODFLOW)
culebra.ibd	MODFLOW input ibound array
culebra.ihd	MODFLOW input initial heads
culebra.spc	PEST utilities grid specification file (input to PEST utilities)
culebra.top	Elevations of the top of the Culebra in MODFLOW format (input to MODFLOW)
d##r##.pts.dat	Current value of pilot points in residual space (also includes X,Y coordinates and zone number). Same file as points.dat
*.dis	MODFLOW discretization input file
dtrk.dbg	DTRKMF debugging information file
fac2real.in	fac2real input file
files.fig	PEST utilities file name specification file (input)
*.hed	MODFLOW output head files
in_mod2obs.*	Input parameter file for the *mod2obs* code
*.inf	Inputs to ppk2fac defining the lower and upper bounds of the residual field and the zone values (all in MODFLOW matrix format)
jacob.runs	PEST output record of the Jacobian calculations
*.lmg	MODFLOW multigrid solver input file
*.log.mod	Log10 space transmissivity or residual field values in MODFLOW format.
*.lst	File containing the MODFLOW screen output
measured.*	The measured heads at a location (output). These files contain four columns: the observation well name, date, time, and modeled head and there is one file for each hydraulic test period.
modeled.*	The modeled heads at a location (output). These files contain four columns: the observation well name, date, time, and modeled head.
modeled.*.parsed	The same as the modeled.* files but with the first 3 columns (Well ID, Date and Time) removed.
*.mtt	PEST output file containing the statistical matrices
*.nam	MODFLOW name file (input)
obs_wells.*	Listing of the observation wells for each pumping test (input)
*.oc	MODFLOW output control file (input)
*.old	Results of the DTRKMF particle tracking with the incorrect starting point coordinates(not part of final results)
pcf.bot	Bottom portion of the PEST control file that does not change
pcf.top	Top portion of the PEST control file that does not change
pest.fnn	PEST intermediate output file (not used in calibration)
pest.*.ins	PEST instruction files that hold the PEST identification for each observation
pest.stp	PEST intermediate output file that tells current run status of PEST
points.dat	Current value of pilot points in residual space (also includes X,Y coordinates and zone number)
points.tpl	PEST input template file identifying the names, locations and zones for each pilot point

ppk2fac.in	ppk2fac input file
ppoints.nodes	Listing of the pilot point locations in vector notation (input to getSgsimParams shell).
ppoints.pcf_add	File created by getSgsimParams that contains the initial values of the residual field at the pilot points
ppoints.zones	Vector listing of zones for each pilot point (used as input to getSgsimParams shell)
reg.out	PEST regularization output
resid_ns.dat	Input data file for sgsim
sd.dat	Ppk2fac binary output file containing kriging variance information (not used in calibration process)
settings.fig	PEST utility input file specifying data column/row and date format
sgsim.console	Screen capture of the output generated while running sgsim
sgsim.dbg	Sgsim debug information file
sgsim.out	Sgsim output file containing the initial simulated residual field
sgsim.par	Sgsim input parameter file
sgsim.par.tpl	Template file used as the basis for each sgsim.par file created
sgsim.trn	Sgsim output containing the raw residual data and the normal-score transform data (not used in calibration process).
tolerance.log	Record of the MODFLOW lmg solver tolerance values used to achieve solutions.
transient.jac	PEST Jacobian matrix saved for restart (binary)
transient.jco	PEST Jacobian matrix for best parameters (binary
transient.jst	PEST Jacobian matrix from the previous iteration (binary)
tupdate.mod	The final calibrated transmissivity field values in MODFLOW format (same file as d##r##.mod)
transient.par	Final pilot point values estimated by PEST (same file as d##r##.par).
transient.pst	PEST control file (input driver file for PEST)
transient.rec	Output record file from PEST (same file as d##r##.rec)
transient.res	Residuals output file from PEST (same file as d##r##.res)
transient.rmf	The parallel PEST run management file (input)
transient.rmr	The parallel PEST run management record (output)
transient.rst	PEST intermediate output file that stores restart information at the beginning of each optimization iteration
transient.sen	PEST output file containing the parameter sensitivities
transient.seo	PEST output containing the observation sensitivities
*.trk	Results of the DTRKMF particle tracking
variogram.str	Input file to ppk2fac program that contains variogram model specifications
*.wel	MODFLOW well definition input file
wells.crd	Listing of well names and X,Y coordinates
wippctrl.inp	DTRKMF input control file for particle track to WIPP site boundary
YYMMDD_####.out	Screen capture of calibration run output. The file name contains the date and the batch queue job number
zones.inf	Input zone definition file for PEST

3 Modeling Assumptions

The major assumptions that apply to this set of model calculations are:

1) The boundary conditions along the model domain boundary are known and do not change over the time frame of the model. This assumption applies to both the no-flow boundary along the western edge of the domain as well as to the fixed-head boundaries that were created to be consistent with the 2000 head measurements in the model domain. Implicit in this assumption is that the fixed-head boundary conditions do not have a significant impact on the transient tests that were simulated in the interior of the model at times other than the 2000 period.

2) The fracture permeability of the Culebra can be adequately modeled as a continuum at the 100X100 m^2 grid block scale and the measured transmissivity values used to condition the model are representative of the transmissivity in the 100X100 m^2 grid block in which the well test was performed. Implicit in this assumption is the prior assumption that the hydraulic test interpretations were done correctly and used the correct conceptual model.

4 Results

A total of 150 base transmissivity fields were used as input to the calibration process. The base transmissivity field names and the resulting travel time to the WIPP boundary and the final value of the objective function (the weighted sum of squares) are shown in Table 8. The first 100 base fields were calibrated using both the Albuquerque and Carlsbad linux clusters by assigning different sets of 10 base fields (e.g., fields d03r01 through d03r10) to one of the clusters. All base transmissivity fields between 101 and 130, those with names starting at d11r01 and going through d13r10 were run on the Albuquerque cluster, while those starting at d21r01 and going through d22r10 were run on the Carlsbad cluster. Use of the two independent parallel clusters is the cause of the gap in the sequence of the base transmissivity fields (i.e., no fields between d13r10 and d21r01).

Not all base transmissivity fields yielded a resulting calibrated transmissivity field. The four fields highlighted in gray in Table 8 did not calibrate at all. The nine fields highlighted in yellow in Table 8 only made minimal progress and did not lower the SSE value below 10^5 m^2. Reasons for not producing a final calibrated transmissivity field include pilot point values proposed by **PEST** for which **MF2K** could not converge to a solution with the required tolerances and a numerically unstable inverse problem for which **PEST** could not find an optimal parameter set. Typically, these calibrations stopped after only a few iterations and resulted in values of the objective function that were greater than 10^5 m^2. It is possible that many of these base transmissivity fields could be calibrated with more effort and adjustment of some of the **PEST** input parameters; however, these parameters were set to work across the largest number of fields possible and the calibration process will not necessarily be able to make progress on every base field given the same set of parameters.

Table 8. Summary of transient calibrations for each base transmissivity field.

Base Field	Final Phi (m2)	Scaled Time (years)	Final Iteration	Base Field	Final Phi (m2)	Scaled Time (years)	Final Iteration
d01r01	1.60E+04	6.758E+04	6	d09r01	2.10E+04	2.916E+05	8
d01r02	8.73E+03	1.204E+04	15	d09r02	6.90E+03	2.005E+04	15
				d09r03	2.33E+03	4.095E+04	15
d01r04	4.70E+03	1.382E+04	7	d09r04	1.83E+03	1.286E+04	15
d01r05	2.29E+04	1.889E+04	6	d09r05	2.04E+03	1.073E+04	15
d01r06	1.64E+04	2.412E+05	8	d09r06	4.57E+03	1.051E+04	15
d01r07	6.04E+03	4.212E+04	6	d09r07	6.27E+03	9.472E+03	4
d01r08	1.02E+04	4.399E+03	9	d09r08	2.44E+03	1.774E+04	8
				d09r09	9.40E+03	4.359E+03	15
d01r10	2.57E+03	2.068E+04	9	d09r10	6.77E+03	5.079E+04	9
d02r01	1.41E+05	1.415E+05	6	d10r01	1.34E+05	2.978E+05	15
d02r02	5.66E+03	1.722E+04	6	d10r02	1.92E+03	3.111E+03	15
d02r03	2.66E+04	2.792E+05	4	d10r03	3.10E+03	1.253E+04	15
d02r04	1.78E+04	9.224E+04	4	d10r04	4.12E+03	3.799E+03	15
d02r05	1.28E+04	1.726E+04	15	d10r05	1.31E+04	2.839E+04	11
d02r06	1.21E+05	1.697E+05	4	d10r06	9.06E+03	9.210E+03	15
d02r07	1.33E+04	3.223E+04	7	d10r07	2.75E+03	1.007E+04	15
d02r08	1.28E+04	2.357E+04	7	d10r08	3.70E+03	1.909E+04	15
				d10r09	7.11E+03	6.805E+04	16
d02r10	1.22E+04	6.433E+03	5	d10r10	2.90E+03	2.837E+04	16
d03r01	1.01E+04	1.844E+04	8	d11r01	3.95E+03	1.702E+04	13
d03r02	1.68E+05	3.989E+05	7	d11r02	3.60E+03	1.468E+04	15
d03r03	1.10E+04	7.171E+03	7	d11r03	1.25E+04	1.601E+04	15
d03r04	1.26E+05	1.328E+05	8	d11r04	1.53E+04	8.166E+04	16
d03r05	1.15E+04	6.638E+03	5	d11r05	1.11E+04	1.900E+04	7
d03r06	3.11E+03	2.701E+04	10	d11r06	3.45E+03	3.840E+04	10
d03r07	9.14E+03	2.260E+04	5	d11r07	2.33E+03	7.363E+04	16
d03r08	9.84E+03	1.394E+04	11	d11r08	1.93E+03	4.520E+03	15
d03r09	3.15E+03	2.576E+04	8	d11r09	2.57E+03	7.199E+03	15
d03r10	1.66E+04	1.505E+04	5	d11r10	4.13E+03	1.438E+04	15
d04r01	7.11E+03	8.069E+04	5	d12r01	6.35E+03	2.394E+04	15
d04r02	5.82E+03	4.059E+04	6	d12r02	3.22E+03	2.692E+04	15
d04r03	4.98E+03	1.389E+04	8	d12r03	2.09E+03	1.678E+04	15
d04r04	7.06E+03	3.624E+04	15	d12r04	2.00E+04	1.562E+04	10
d04r05	3.04E+03	4.817E+04	12	d12r05	2.38E+03	5.655E+03	15
d04r06	8.77E+03	2.620E+04	15	d12r06	1.81E+03	3.940E+04	15
d04r07	3.62E+03	2.311E+04	15	d12r07	2.79E+03	1.828E+04	11
d04r08	3.32E+03	3.047E+04	6	d12r08	5.14E+03	7.981E+03	15
d04r09	1.60E+05	1.141E+06	2	d12r09	4.46E+03	9.414E+03	15
d04r10	4.99E+03	2.532E+04	14	d12r10	2.30E+04	3.206E+04	15
d05r01	1.01E+04	8.692E+04	5	d13r01	2.81E+03	2.103E+04	14
d05r02	1.26E+04	2.561E+04	8	d13r02	5.77E+03	2.564E+04	15
d05r03	6.24E+03	1.088E+04	13	d13r03	5.77E+03	1.149E+04	15
d05r04	1.75E+04	1.466E+04	8	d13r04	1.04E+04	4.060E+04	15
d05r05	2.27E+04	5.668E+03	5	d13r05	7.42E+03	3.425E+04	15
d05r06	1.18E+04	9.659E+04	10	d13r06	3.82E+03	4.140E+04	15
d05r07	7.90E+03	1.377E+04	7	d13r07	3.00E+03	2.421E+04	15
d05r08	2.24E+04	7.090E+04	4	d13r08	2.32E+03	2.031E+04	15
d05r09	1.37E+06	1.528E+05	1	d13r09	2.155E+03	3.626E+04	15
d05r10	1.36E+04	3.096E+04	5	d13r10	9.376E+03	2.204E+05	13
d06r01	1.56E+04	1.034E+05	15	d21r01	3.86E+03	1.004E+04	15
d06r02	3.91E+03	1.035E+04	15	d21r02	3.91E+03	9.023E+03	11
d06r03	2.75E+03	8.126E+04	16	d21r03	2.04E+03	1.167E+04	15
d06r04	5.05E+03	1.829E+04	9	d21r04	4.89E+03	1.572E+04	15
d06r05	8.68E+03	3.664E+04	15	d21r05	2.35E+03	2.375E+04	15
d06r06	4.21E+03	1.494E+04	13	d21r06	2.34E+03	2.072E+04	15
d06r07	1.76E+03	1.204E+04	15	d21r07	3.90E+03	2.014E+04	12
d06r08	1.49E+04	7.456E+04	15	d21r08	1.40E+04	1.953E+04	6
d06r09	1.50E+05	1.683E+05	2	d21r09	2.26E+04	3.331E+04	15
d06r10	3.26E+03	2.199E+04	15	d21r10	6.84E+03	7.384E+03	15
d07r01	5.23E+03	5.082E+03	7	d22r01	1.13E+05	4.756E+04	13
d07r02	4.60E+03	4.565E+04	15	d22r02	8.45E+04	1.012E+05	13
d07r03	5.58E+04	1.292E+04	3	d22r03	6.58E+03	7.067E+03	13
d07r04	1.33E+04	5.638E+03	6	d22r04	9.14E+03	1.054E+04	8
d07r05	7.65E+03	1.610E+04	6	d22r05	1.30E+04	1.438E+04	15
d07r06	1.03E+04	2.464E+04	5	d22r06	4.78E+03	4.431E+04	9
d07r07	3.96E+03	1.704E+04	15	d22r07	2.41E+03	2.159E+04	15
d07r08	3.93E+03	4.355E+03	14	d22r08	2.49E+03	3.077E+04	8
d07r09	2.87E+03	6.863E+04	8	d22r09	2.46E+03	1.587E+04	15
d07r10	1.99E+03	1.568E+04	15	d22r10	2.42E+03	3.912E+04	8
d08r01	4.19E+03	4.388E+03	15				
d08r02	2.49E+03	2.611E+04	15				
d08r03	2.99E+03	2.857E+04	13				
d08r04	5.64E+03	2.477E+04	15				
d08r05	4.15E+03	1.536E+04	15				
d08r06	2.61E+03	1.392E+04	15				
d08r07	2.15E+03	1.503E+04	11				
d08r08	3.48E+04	1.388E+04	4				
d08r09	1.41E+04	9.691E+03	7				

After removal of the base transmissivity fields that did not calibrate, the relationship between the final value of the objective function (the weighted sum of squares) and the travel time to the WIPP boundary is shown in Figure 23. Figure 23 shows that there is no relationship between the final value of the objective function and the travel time to the WIPP boundary. The travel times shown in Figure 23 (and in Figure 24) are the scaled travel times that have been multiplied by a factor of (4.0/7.75) to make them comparable to the CCA travel times where a Culebra thickness of 4 m was used (Meigs and McCord, 1996).
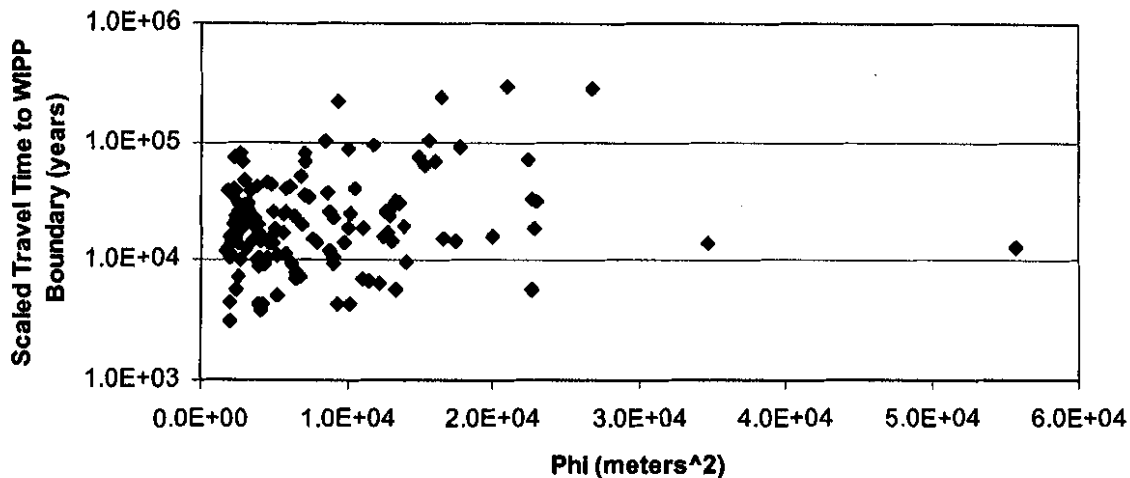
Figure 23. Relationship between the final value of the objective function and the particle travel time to the WIPP boundary. This figure includes the 137 base transmissivity fields for which a calibration was achieved.

The travel times calculated herein are compared to the travel times calculated for the CCA (Wallace, 1996) in Figure 24. Figure 24 shows the cumulative distribution functions for the two different sets of travel times. The travel times for these calculations are generally a factor of two to three longer than the travel times calculated for the CCA.

The current conceptual model of the geology contains a high-transmissivity pathway down the western side of the model that connects the northern and southern boundaries. In most of the calibrated fields, a significant amount of the groundwater flow occurs in this high-transmissivity pathway and therefore does not flow through the WIPP site. This high-transmissivity channel acts to divert water away from the site. The ability of this high-transmissivity channel to move water through the model is dependent on the boundary conditions and may have more or less impact on travel times for a different set of boundary conditions.

In the current set of calibrations, the pilot point values are constrained to limit the pilot point values to always be within ± three orders of magnitude of the value predicted by the conceptual model (or ± one order of magnitude in the high- and low-transmissivity zones). These constraints are added to preserve the influence of the geologic conceptual model implemented in these inversions. However, there may be locations within the model domain where constraining the inversion process to the conceptual model of the geology cannot reproduce the observed hydraulic behavior.
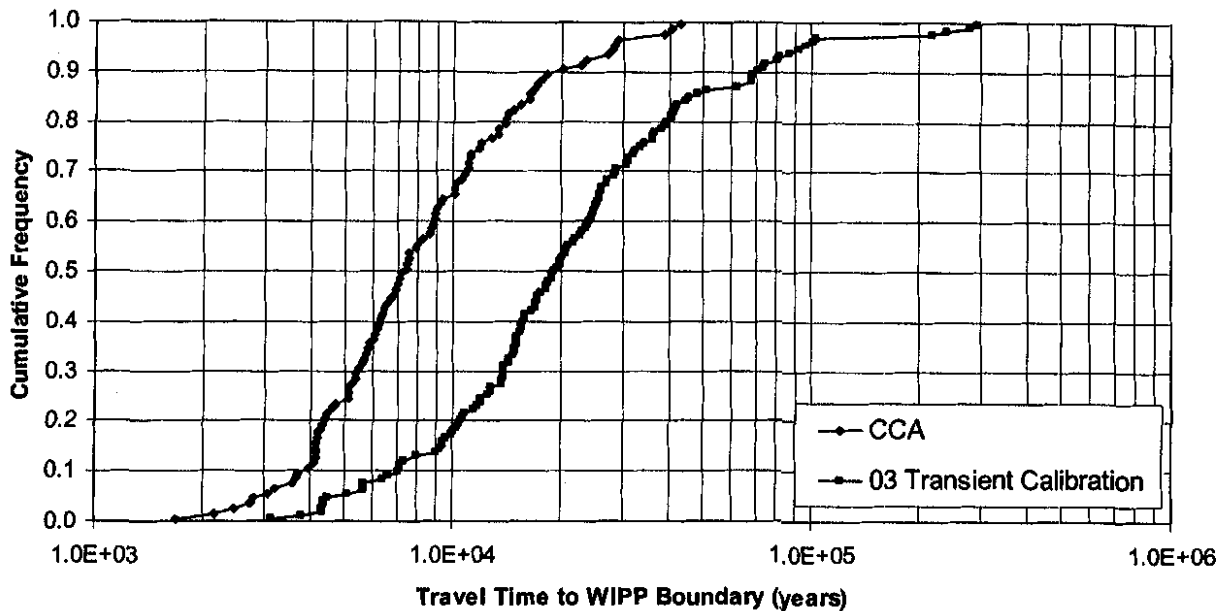
Figure 24. Comparison of cdfs for the current set of calculated travel times (137 calibrated fields) and the travel times calculated for the CCA.

The locations of all the particle tracks are show in Figures 25 and 26. In both Figures 25 and 26, the particle tracks are shown using only every 20th point along the track. This filtering was necessary to create Figures 25 and 26 in the graphing software and it leads to the particle tracks appearing less smooth than they actually are. Figure 25 shows a close-up view of the particle tracks within the WIPP site boundary. All but one of these particles exits the southern edge of the WIPP boundary and the majority of the particles exit the WIPP boundary to the southeast of the release point. Figure 26 shows the particle tracks within the entire model domain. The majority of the particles exit the domain nearly due south of the release point. The particles that migrate to the west tend to travel along the boundary of the high-transmissivity zone. This result is due to the large amount of groundwater flux within the high-transmissivity zone creating a streamline at the high-transmissivity zone boundary.

45
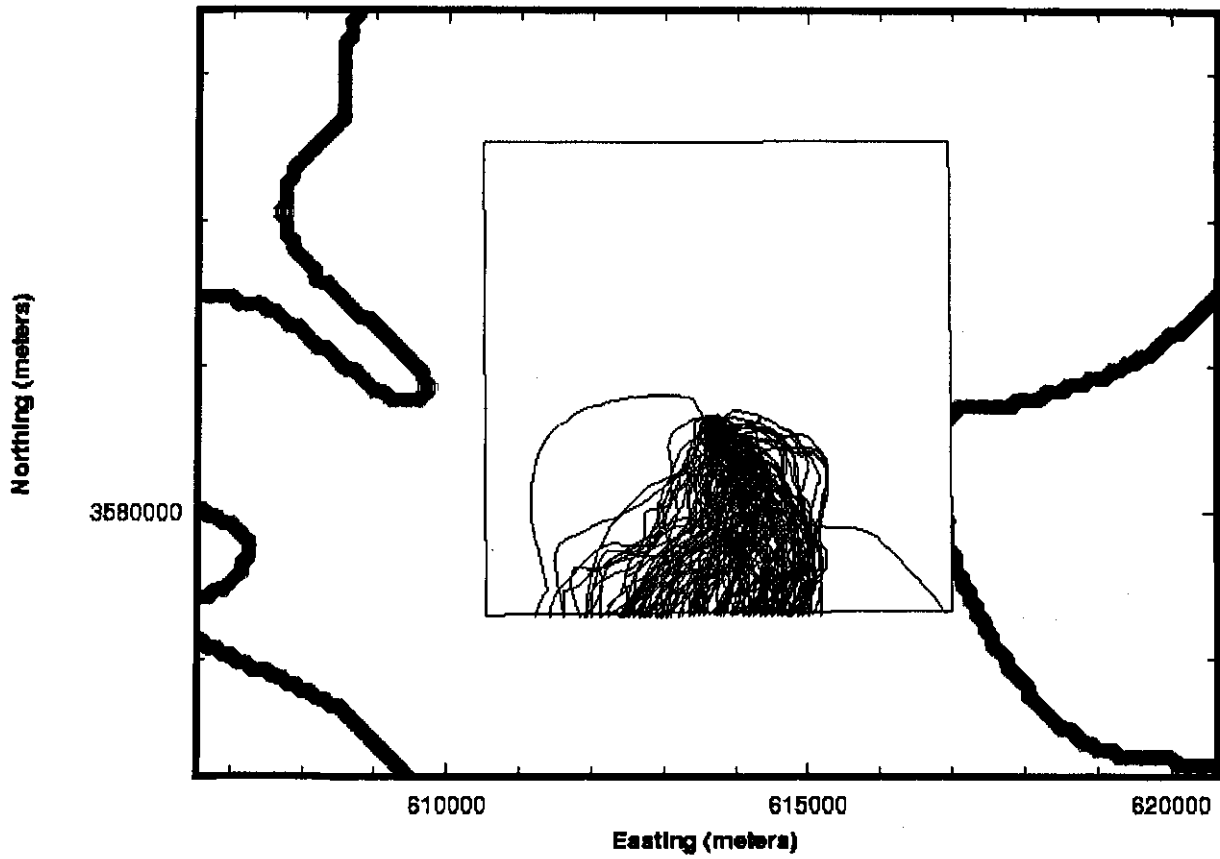
Figure 25. All particle tracks within the WIPP site boundary. The bold lines show the boundaries of the high- (left side) and low- (right side) transmissivity zones.

46
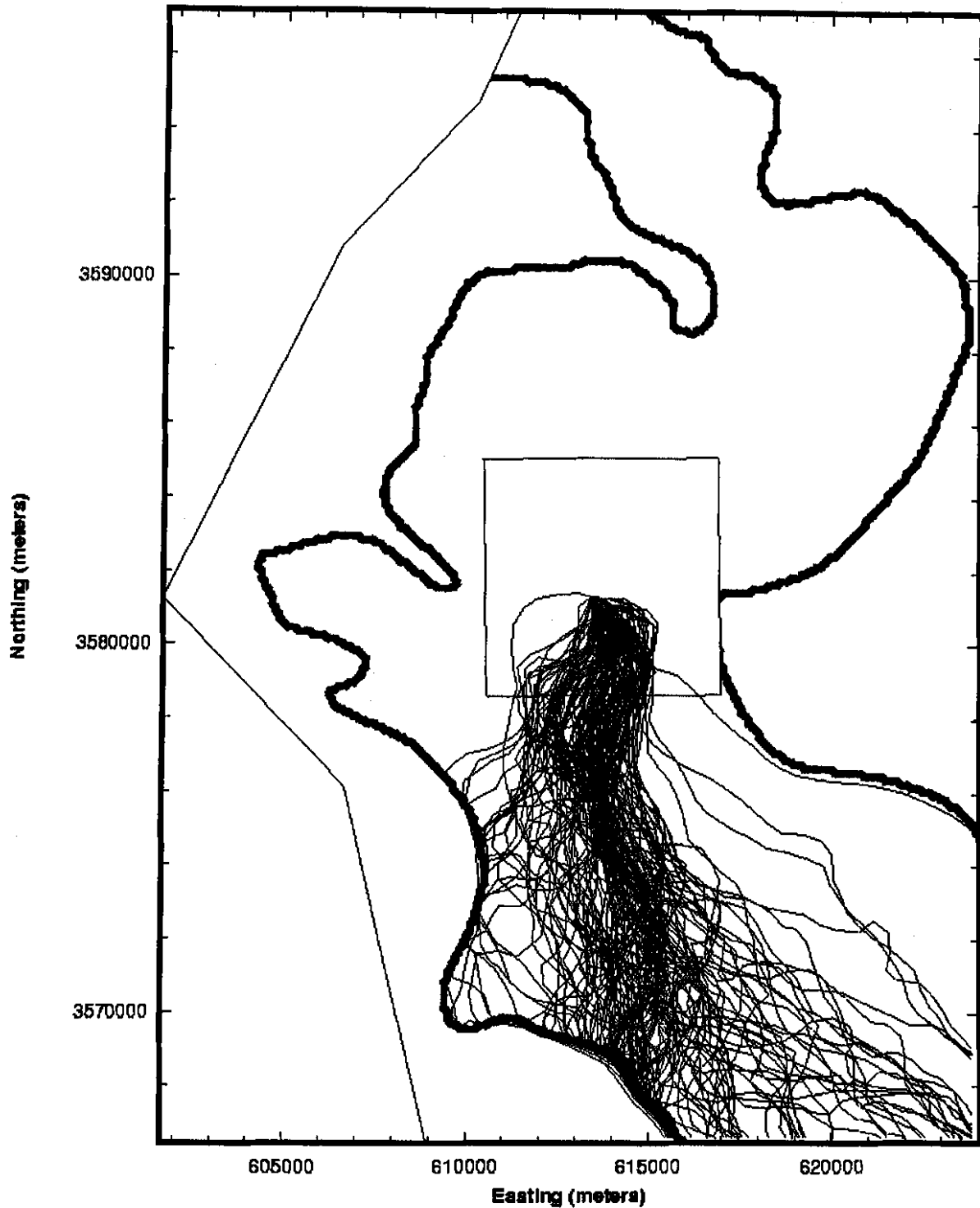
Figure 26. All particle tracks within the model domain. The bold lines show the boundaries of the high- (left) and low- (right) transmissivity zone boundaries. The no-flow and WIPP site boundaries are also shown.

INFORMATION ONLY

47

5 Summary

This analysis package provides details on the creation of the stochastic calibrations of the Culebra transmissivity fields for the WIPP compliance recertification calculations. The area modeled in this work is approximately 22.3 X 30.6 km^2 and is roughly centered on the WIPP land withdrawal boundary. Fixed-head and no-flow boundary conditions were assigned to the edges of the domain as they were for the calibrations to the steady-state data (McKenna and Hart, 2003). The model domain was discretized into uniform 100X100 m^2 cells. Observed data from seven different hydraulic tests collected over nearly eleven years were used along with a single set of steady-state observations collected in 2000.

The **PEST** and **MF2K** software packages were used to calibrate the flow model to the observed steady-state and transient head data. This calibration is done by using 100 pilot points to adjust the transmissivity values within the model domain to improve the fit to the observed heads. The pilot points are used to adjust a residual transmissivity field that is combined with a previously created base transmissivity field to yield the final calibrated transmissivity field. The updating of the pilot point values is done with a parallel version of **PEST** on two clusters of PC's, both running the linux operating system. A total of 150 base transmissivity fields was used as input to the calibration process and the resulting calibrated fields were used as input to track a single particle under steady-state conditions. Of these 150 fields, the calibration process was unsuccessful for four of them and was able to run but not produce a meaningful calibration on an additional nine base fields. The particle tracks on the 137 fields that were calibrated show travel times that are longer than those calculated for the CCA.

6 References

Beauheim, R.L. 2002a. Analysis Plan for the Evaluation of the Effects of Head Changes on Calibration of Culebra Transmissivity Fields, AP-088, Revision 1. ERMS#524785. Carlsbad, NM: Sandia National Laboratories.

Beauheim, R.L. 2002b. Routine Calculations Report In Support of Task 3 of AP-088, Calculation of Culebra Freshwater Heads in 1980, 1990, and 2000 for Use in T-Field Calibration. ERMS#522580.

Beauheim, R.L. 2003. Records Package for AP-088 Task 4, Conditioning of Base T Fields to Transient Heads: Compilation and Reduction of Transient Head Data. ERMS#527572.

Helton, J.C., J.E. Bean, J.W. Berglund, F.J. Davis, K. Economy, J.W. Garner, J.D. Johnson, R.J. MacKinnon, J. Miller, D.G. O'Brien, J.L. Ramsey, J.D. Schreiber, A. Shinta, L.N. Smith, D.M. Stoelzel, C. Stockman, and P. Vaughn. 1998. *Uncertainty and Sensitivity Analysis Results Obtained in the 1996 Performance Assessment for the Waste Isolation Pilot Plant.* SAND98-0365. Albuquerque, NM: Sandia National Laboratories.

Helton, J.C., and M.G. Marietta (editors). 2000. The 1996 Performance Assessment for the Waste Isolation Pilot Plant, Special Issue of: *Reliability Engineering and System Safety*, Vol 69, No. 1-3.

Holt, R.M., and L. Yarbrough. 2003. Addendum 2 to Analysis Report, Task 2 of AP-088, Estimating Base Transmissivity Fields. ERMS#529416.

Lavenue, A.M. 1996. Analysis of the Generation of Transmissivity Fields for the Culebra Dolomite. ERMS#240517.

McKenna, S.A., and D.B. Hart. 2003. Analysis Report, Task 3 of AP-088, Conditioning of Base T Fields to Steady-State Heads. ERMS#529633.

Meigs, L.C., and J.T. McCord. 1996. "Appendix A: Physical Transport in the Culebra Dolomite." In Analysis Package for the Culebra Flow and Transport Calculations (Task 3) of the Performance Assessment Analyses Supporting the Compliance Certification Application, Analysis Plan 019, Version 00, J.L. Ramsey, M.G. Wallace and H.-N. Jow, Sandia National Laboratories, Albuquerque, New Mexico, ERMS#240516.

Ramsey, J.L., M.G. Wallace, and H.-N. Jow. 1996. Analysis Package for the Culebra Flow and Transport Calculations (Task 3) of the Performance Assessment Calculations Supporting the Compliance Certification Application (CCA), AP-019. Albuquerque, NM: Sandia National Laboratories. (Copy on file in the Sandia WIPP Records Center, Sandia National Laboratories, Carlsbad, NM as ERMS#240516.)

INFORMATION ONLY

U.S. Department of Energy. 1996. *Title 40 CFR Part 191 Compliance Certification Application for the Waste Isolation Pilot Plant.* DOE/CAO-1996-2184. Carlsbad, NM: U.S. DOE, Carlsbad Area Office.

Wallace, M. 1996. Records Package for Screening Effort NS11: Subsidence Associated with Mining Inside or Outside the Controlled Area. ERMS#412918.

Appendix 1: Perl Scripts for Normalization of Drawdown Observation Data

WIPP-13 Observation Normalization Script

```perl
#!/usr/bin/perl
open(OBSFILE, "modelled.w13.parsed") or die "Can't open input file: $!\n";
open(OUTFILE, ">modelled.w13.parse2") or die "Can't open output file: $!\n";

#doe-2
$thisInit = 0.000;
$thisLim = 103;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}


#h-2
$thisInit = 0.000;
$thisLim = 22;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}



#h-6
$thisInit = 0.000;
$thisLim = 92;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}



#p-14
$thisInit = 0.000;
$thisLim = 37;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}


#w-12
```

```
    $thisInit = 0.000;
    $thisLim = 26;
    $start = <OBSFILE>;
    print OUTFILE "$thisInit","\n";
    for ( $i = 0 ; $i < $thisLim ; $i++) {
        $val = <OBSFILE>;
        printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
    }


    #w-18
    $thisInit = 0.000;
    $thisLim = 25;
    $start = <OBSFILE>;
    print OUTFILE "$thisInit","\n";
    for ( $i = 0 ; $i < $thisLim ; $i++) {
        $val = <OBSFILE>;
        printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
    }


    #w-19
    $thisInit = 0.000;
    $thisLim = 21;
    $start = <OBSFILE>;
    print OUTFILE "$thisInit","\n";
    for ( $i = 0 ; $i < $thisLim ; $i++) {
        $val = <OBSFILE>;
        printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
    }


    #w-25
    $thisInit = 0.000;
    $thisLim = 10;
    $start = <OBSFILE>;
    print OUTFILE "$thisInit","\n";
    for ( $i = 0 ; $i < $thisLim ; $i++) {
        $val = <OBSFILE>;
        printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
    }


    #w-30
    $thisInit = 0.000;
    $thisLim = 23;
    $start = <OBSFILE>;
    print OUTFILE "$thisInit","\n";
    for ( $i = 0 ; $i < $thisLim ; $i++) {
        $val = <OBSFILE>;
        printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
    }


close (OBSFILE);
close (OUTFILE);
```

P-14 Observation Normalization Script

```perl
#!/usr/bin/perl
open(OBSFILE, "modelled.p14.parsed") or die "Can't open input file: $!\n";
open(OUTFILE, ">modelled.p14.parse2") or die "Can't open output file: $!\n";

#d-268
$thisInit = 0.000;
$thisLim = 37;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}


#h-18
$thisInit = 0.000;
$thisLim = 20;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}



#h-6b
$thisInit = 0.000;
$thisLim = 20;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}



#wipp-25
$thisInit = 0.000;
$thisLim = 21;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}



#wipp-26
$thisInit = 0.000;
$thisLim = 20;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
```

```
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}

close (OBSFILE);
close (OUTFILE);
```

WQSP-1 Observation Normalization Script

```perl
#!/usr/bin/perl

open(OBSFILE, "modelled.wqsp1.parsed") or die "Can't open input file: $!\n";
open(OUTFILE, ">modelled.wqsp1.parse2") or die "Can't open output file:
$!\n";
$h18start = <OBSFILE>;
print OUTFILE "0.00000","\n";
for ( $i = 0 ; $i < 46 ; $i++) {
    $h18val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $h18val - $h18start);
}

$w13start = <OBSFILE>;
print OUTFILE "0.00000","\n";
for ( $i = 0 ; $i < 46 ; $i++) {
    $w13val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" ,$w13val - $w13start);
}

$wq3start = <OBSFILE>;
print OUTFILE "0.00000","\n";
for ( $i = 0 ; $i < 24 ; $i++) {
    $wq3val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $wq3val - $wq3start);
}

close (OBSFILE);
close (OUTFILE);
```

WQSP-2 Observation Normalization Script

```perl
#!/usr/bin/perl

open(OBSFILE, "modelled.wqsp2.parsed") or die "Can't open input file: $!\n";
open(OUTFILE, ">modelled.wqsp2.parse2") or die "Can't open output file:
$!\n";

$doe2start = <OBSFILE>;
print OUTFILE "0.00000","\n";
for ( $i = 0 ; $i < 33 ; $i++) {
    $doe2val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $doe2val - $doe2start);
}

$h18start = <OBSFILE>;
print OUTFILE "0.00000","\n";
for ( $i = 0 ; $i < 33 ; $i++) {
    $h18val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $h18val - $h18start);
}

$w13start = <OBSFILE>;
print OUTFILE "0.00000","\n";
for ( $i = 0 ; $i < 33 ; $i++) {
    $w13val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" ,$w13val - $w13start);
}

$wq1start = <OBSFILE>;
print OUTFILE "0.00000","\n";
for ( $i = 0 ; $i < 5 ; $i++) {
    $wq1val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $wq1val - $wq1start);
}

$wq3start = <OBSFILE>;
print OUTFILE "0.00000","\n";
for ( $i = 0 ; $i < 17 ; $i++) {
    $wq3val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $wq3val - $wq3start);
}

close (OBSFILE);
close (OUTFILE);
```

H-11 Observation Normalization Script

```perl
#!/usr/bin/perl

open(OBSFILE, "modelled.h11.parsed") or die "Can't open input file: $!\n";
open(OUTFILE, ">modelled.h11.parse2") or die "Can't open output file: $!\n";
$h17start = <OBSFILE>;
print OUTFILE "0.00000","\n";
for ( $i = 0 ; $i < 22 ; $i++) {
    $h17val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $h17val - $h17start);
}


$h4bstart = <OBSFILE>;
print OUTFILE "0.00000","\n";
for ( $i = 0 ; $i < 10 ; $i++) {
    $h4bval = <OBSFILE>;
    printf( OUTFILE "%.6E\n" ,$h4bval - $h4bstart);
}


$h12start = <OBSFILE>;
print OUTFILE "0.108962647","\n";
for ( $i = 0 ; $i < 10 ; $i++) {
    $h12val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , ($h12val - $h12start + 0.108962647) );
}


$p17start = <OBSFILE>;
print OUTFILE "0.504212966","\n";
for ( $i = 0 ; $i < 18 ; $i++) {
    $p17val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , ($p17val - $p17start + 0.504212966) );
}

close (OBSFILE);
close (OUTFILE);
```

H-19 Observation Normalization Script

```perl
#!/usr/bin/perl
open(OBSFILE, "modelled.h19.parsed") or die "Can't open input file: $!\n";
open(OUTFILE, ">modelled.h19.parse2") or die "Can't open output file: $!\n";

#doe-1   1.100 ; 70
$thisInit = 1.100;
$thisLim = 69;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}


#erda-9  2.0538; 80
$thisInit = 2.0538;
$thisLim = 79;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}


#h-1     2.8480; 80
$thisInit = 2.8480;
$thisLim = 79;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}


#h-15    2.7945; 22
$thisInit = 2.7945;
$thisLim = 21;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}

#h-3b2   1.871 ; 69
$thisInit = 1.871;
$thisLim = 68;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}
```

```perl
#wipp-21 3.5807; 19
$thisInit = 3.5807;
$thisLim = 18;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}


#wqsp-5 11.6845; 24
$thisInit = 11.6845;
$thisLim = 23;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}


#h-14    1.6613; 11
$thisInit = 1.6613;
$thisLim = 10;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}


#h-2b2    1.7481; 11
$thisInit = 1.7481;
$thisLim = 10;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}

#wqsp-4 18.7810; 24
$thisInit = 18.7810;
$thisLim = 23;
$start = <OBSFILE>;
print OUTFILE "$thisInit","\n";
for ( $i = 0 ; $i < $thisLim ; $i++) {
    $val = <OBSFILE>;
    printf( OUTFILE "%.6E\n" , $val - $start + $thisInit);
}

close (OBSFILE);
close (OUTFILE);
```

Appendix 2: Supplementary Material for Estimation of the Fixed-Head Boundary Values

Results of Fitting the Gaussian Trend Surface to the 2000 Heads

Nonlinear Regression

[Variables]
x = col(1)
y = col(2)
z = col(3)
[Parameters]
x0 = xatymax(x,z) "Auto {{previous: 611012}}
y0 = xatymax(y,z) "Auto {{previous: 3.78089e+006}}
a = max(z) "Auto {{previous: 1134.61}}
b = fwhm(x,z)/2.2 "Auto {{previous: 73559.4}}
c = fwhm(y,z)/2.2 "Auto {{previous: 313474}}
[Equation]
f=a*exp(-.5*(((x-x0)/b)^2 + ((y-y0)/c)^2))
fit f to z
[Constraints]
[Options]
tolerance=0.000100
stepsize=100
iterations=100

R = 0.84940930   Rsqr = 0.72149616   Adj Rsqr = 0.68668318

Standard Error of Estimate = 5.5471

	Coefficient	Std. Error	t	P
x0	611011.8967	1480.3846	412.7386	<0.0001
y0	3780891.5012	1052646.9742	3.5918	0.0011
a	1134.6135	1213.4826	0.9350	0.3568
b	73559.3533	12971.0833	5.6710	<0.0001
c	313474.4090	829108.9913	0.3781	0.7079

Analysis of Variance:

	DF	SS	MS	F	P
Regression	4	2550.8316	637.7079	20.7249	<0.0001
Residual	32	984.6434	30.7701		
Total	36	3535.4750	98.2076		

PRESS = 22345.6338

Durbin-Watson Statistic = 1.9526

Normality Test: Passed   (P = 0.2217)

Constant Variance Test: Passed   (P = 0.7532)

Power of performed test with alpha = 0.0500: 1.0000

Regression Diagnostics:

60

Row	Predicted	Residual	Std. Res.	Stud. Res.	Stud. Del. Res.
1	932.6014	0.5934	0.1070	0.2710	0.2670
2	923.1180	-6.5698 -	1.1844 -	1.2771 -	1.2903
3	933.2947	6.7384	1.2148	1.3332	1.3502
4	927.0594	-5.4669 -	0.9855 -	1.0519 -	1.0537
5	926.6641	0.5284	0.0953	0.1018	0.1002
6	926.8633	-0.2405 -	0.0433 -	0.0464 -	0.0457
7	925.0796	-7.9207 -	1.4279 -	1.5288 -	1.5629
8	920.9577	-5.4032 -	0.9741 -	1.0800 -	1.0829
9	930.0371	6.2218	1.1216	1.2514	1.2630
10	933.3632	0.8401	0.1514	0.1683	0.1657
11	913.0971	0.7597	0.1370	0.1853	0.1825
12	900.7080	10.8670	1.9591	2.5789	2.8519
13	920.7787	-5.3089 -	0.9571 -	1.0478 -	1.0494
14	912.2902	2.3718	0.4276	0.5457	0.5396
15	924.5063	-4.2673 -	0.7693 -	0.8310 -	0.8269
16	925.9052	-6.0349 -	1.0879 -	1.1654 -	1.1722
17	917.3946	-2.0229 -	0.3647 -	0.4152 -	0.4098
18	929.8066	7.4164	1.3370	1.4396	1.4652
19	924.2918	-7.1656 -	1.2918 -	1.3848 -	1.4058
20	918.4636	-3.2670 -	0.5890 -	0.6660 -	0.6601
21	929.9850	5.3120	0.9576	1.0276	1.0285
22	931.7313	3.4423	0.6206	0.6731	0.6673
23	929.3286	6.7544	1.2176	1.3036	1.3186
24	928.5840	4.0766	0.7349	0.7854	0.7806
25	927.7147	-0.7113 -	0.1282 -	0.1369 -	0.1348
26	928.3424	2.6150	0.4714	0.5036	0.4976
27	929.7111	2.9889	0.5388	0.6054	0.5993
28	921.9985	-0.9374 -	0.1690 -	0.1921 -	0.1892
29	944.4095	-3.4013 -	0.6132 -	3.9904 -	5.5410
30	904.9051	0.4562	0.0822	0.2244	0.2211
31	941.4068	-4.5235 -	0.8155 -	1.0984 -	1.1021
32	930.2232	5.4162	0.9764	1.0514	1.0532
33	930.7921	8.0232	1.4464	1.5579	1.5951
34	929.4395	6.4514	1.1630	1.2483	1.2597
35	924.2540	-6.7656 -	1.2197 -	1.3079 -	-1.3231
36	924.0527	-6.8350 -	1.2322 -	1.3245 -	1.3409
37	925.1606	-5.1385 -	0.9264 -	0.9965 -	0.9964

Influence Diagnostics:

Row	Cook'sDist	Leverage	DFFITS
1	0.0796	0.8442	0.6215
2	0.0531	0.1399 -	0.5204
3	0.0727	0.1698	0.6106
4	0.0308	0.1222 -	0.3932
5	0.0003	0.1238	0.0377
6	0.0001	0.1289 -	0.0176
7	0.0684	0.1277 -	0.5979
8	0.0535	0.1866 -	0.5186
9	0.0767	0.1967	0.6250
10	0.0013	0.1903	0.0803
11	0.0057	0.4540	0.1664
12	-3.6351	1.5771	(+inf)
13	0.0436	0.1656 -	0.4676
14	0.0375	0.3861	0.4279
15	0.0230	0.1430 -	0.3377

16	0.0401	0.1286 -	0.4502
17	0.0102	0.2286 -	0.2230
18	0.0661	0.1375	0.5850
19	0.0572	0.1299 -	0.5431
20	0.0247	0.2179 -	0.3484
21	0.0320	0.1316	0.4003
22	0.0160	0.1501	0.2804
23	0.0497	0.1276	0.5042
24	0.0175	0.1244	0.2942
25	0.0005	0.1224 -	0.0503
26	0.0072	0.1236	0.1869
27	0.0192	0.2078	0.3069
28	0.0022	0.2262 -	0.1023
29	-138.0564	1.0236	(+inf)
30	0.0650	0.8657	0.5614
31	0.1965	0.4488 -	0.9945
32	0.0353	0.1376	0.4208
33	0.0778	0.1381	0.6384
34	0.0474	0.1319	0.4911
35	0.0513	0.1303 -	0.5122
36	0.0545	0.1345 -	0.5286
37	0.0312	0.1358 -	0.3950

95% Confidence:

Row	Predicted	Regr. 5%	Regr. 95%	Pop. 5%	Pop. 95%
1	932.6014	922.2201	942.9827	917.2573	947.9454
2	923.1180	918.8917	927.3444	911.0544	935.1816
3	933.2947	928.6391	937.9502	921.0741	945.5152
4	927.0594	923.1093	931.0095	915.0898	939.0290
5	926.6641	922.6887	930.6394	914.6861	938.6420
6	926.8633	922.8069	930.9196	914.8582	938.8683
7	925.0796	921.0422	929.1170	913.0809	937.0783
8	920.9577	916.0772	925.8381	908.6496	933.2657
9	930.0371	925.0259	935.0482	917.6767	942.3975
10	933.3632	928.4346	938.2918	921.0360	945.6904
11	913.0971	905.4839	920.7103	899.4725	926.7217
12	900.7080	886.5185	914.8975	882.5694	918.8466
13	920.7787	916.1801	925.3773	908.5797	932.9777
14	912.2902	905.2695	919.3109	898.9876	925.5927
15	924.5063	920.2338	928.7788	912.4265	936.5861
16	925.9052	921.8540	929.9565	913.9019	937.9086
17	917.3946	911.9928	922.7964	904.8707	929.9185
18	929.8066	925.6171	933.9961	917.7559	941.8573
19	924.2918	920.2199	928.3638	912.2815	936.3022
20	918.4636	913.1893	923.7378	905.9942	930.9330
21	929.9850	925.8868	934.0832	917.9657	942.0043
22	931.7313	927.3535	936.1091	919.6138	943.8488
23	929.3286	925.2930	933.3642	917.3305	941.3267
24	928.5840	924.5993	932.5687	916.6029	940.5651
25	927.7147	923.7623	931.6672	915.7443	939.6851
26	928.3424	924.3696	932.3153	916.3653	940.3196
27	929.7111	924.5608	934.8615	917.2936	942.1286
28	921.9985	916.6246	927.3724	909.4866	934.5104
29	944.4095	932.9778	955.8411	928.3362	960.4828
30	904.9051	894.3920	915.4182	889.4716	920.3386
31	941.4068	933.8370	948.9765	927.8064	955.0071

32	930.2232	926.0312	934.4151	918.1716	942.2748
33	930.7921	926.5937	934.9905	918.7383	942.8459
34	929.4395	925.3356	933.5435	917.4183	941.4608
35	924.2540	920.1751	928.3329	912.2413	936.2668
36	924.0527	919.9090	928.1964	912.0179	936.0876
37	925.1606	920.9964	929.3249	913.1187	937.2026

Appendix 3: addtrend.c source code

```c
#include <stdio.h>
#include <math.h>
#include <string.h>

/*
    Sean A. McKenna                              June 2002
    Geohydrology Department
    Sandia National Laboratories
    Albuquerque, NM  87185-0735

    ph: 505 844-2450
    em: samcken@sandia.gov

    Code to read in a single GeoEAS Formatted output file from kt3d where the
    first column is a kriged residual field and the second column is the
    kriging variance.  This file then adds a trend surface to the residuals
    and writes a new file of the trend+residuals and the kriging variance in
    GeoEAS format.

*/
/*------------------------------------------------------------*/
char *read_line (fp)
FILE *fp;
{
    static char string[256];

    /*  This routine reads in a line of data from the given
        inout stream.  It however returns only lines that do
        not start with an '!', this symbol is used to denote a
        comment line.  The maximum line length is 256 characters.*/

    string[0] = '\0';
    do
      fgets (string, 256, fp);
      while ((string[0] == '!') && !feof (fp));

    return (string);
}


/*------------------------------------------------------------*/

main ()
{
    FILE *stream1,*stream2;
    char string[256],title[80],value_title[80],file1[80],file2[80];
    int i,j,nx,ny,data_col;
    double resid,krig_var,currx,curry,y0,x0,coeff_a,coeff_b,coeff_c;
    double delx,dely,o_x,o_y,trend,first,second;

    /* set constants */
    nx = 224;
    ny = 307;
    delx = 100.0;
```

```c
        dely = 100.0;
        o_x = 601700.0;
        o_y = 3566500.0;

    x0 = 611011.89;
        y0 = 3780891.50;
        coeff_a = 1134.61;
        coeff_b = 73559.35;
        coeff_c = 313474.40;


        /* open input and output files */
    printf ("Enter the name of the GeoEAS formatted residual file  \n");
    gets (file1);
    stream1 = fopen(file1,"r");

    printf ("Enter the name of the GeoEAS formatted output file  \n");
    gets (file2);
    stream2 = fopen(file2,"w");


    /* Read and Write file header information */
        sprintf (string, "%s", read_line (stream1));
    sscanf (string, "%s", &title);
    sprintf (string, "%s", read_line (stream1));
    sscanf (string, "%d", &data_col);
        sprintf (string, "%s", read_line (stream1));
    sscanf (string, "%s", &value_title);
        sprintf (string, "%s", read_line (stream1));
    sscanf (string, "%s", &value_title);

    fprintf (stream2,"Starting Head Field\n");
        fprintf (stream2,"2\n");
        fprintf (stream2,"Trend plus residual\n");
        fprintf (stream2,"Kriging Variance\n");


        /* read in residuals, calculate and add trend surface, write output */
        for (j=1;j<=ny;j++) {
          curry = (o_y+(float)j*dely)-(dely/2.0);
          for (i=1;i<=nx;i++) {
            currx = (o_x+(float)i*delx)-(delx/2.0);
          fscanf (stream1,"%lf %lf",&resid, &krig_var);
                if (resid < 1.0E-09) resid = 0.0;
                first  = ((currx-x0)/coeff_b)*((currx-x0)/coeff_b);
                second = ((curry-y0)/coeff_c)*((curry-y0)/coeff_c);
          trend = coeff_a*exp(-0.5*(first+second));
                if ((i==1)&&(j<=10)) printf ("j = %3d, trend = %7.2f X = %9.1f  Y
= %9.1f\n", j, trend,currx,curry);
                fprintf (stream2," %7.2lf   %7.3lf\n", (trend+resid),krig_var);
          }
        }

    fclose (stream1);
        fclose (stream2);


    }
```

Appendix 4: kt3d input file

```
                    Parameters for KT3D
                    *******************

START OF PARAMETERS:
bnd_00.dat                          \file with data
1    2    0    4    0               \    columns for X, Y, Z, var, sec var
-1.0e21    1.0e21                   \    trimming limits
0                                   \option: 0=grid, 1=cross, 2=jackknife
xvk.dat                             \file with jackknife data
1    2    0    3    0               \    columns for X,Y,Z,vr and sec var
1                                   \debugging level: 0,1,2,3
kt3d_rsd_00.dbg                     \file for debugging output
kt3d_rsd_00.out                     \file for kriged output
224    601700.    100.0            \nx,xmn,xsiz
307    3566500.    100.0           \ny,ymn,ysiz
1     0.5     1.0                   \nz,zmn,zsiz
1     1     1                       \x,y and z block discretization
0     8                             \min, max data for kriging
2                                   \max per octant (0-> not used)
20000.0  20000.0  20.0             \maximum search radii
 0.0    0.0    0.0                  \angles for search ellipsoid
0     0.000                         \0=SK,1=OK,2=non-st SK,3=exdrift
0 0 0 0 0 0 0 0 0                   \drift: x,y,z,xx,yy,zz,xy,xz,zy
0                                   \0, variable; 1, estimate trend
extdrift.dat                        \gridded file with drift/mean
4                                   \    column number in gridded file
1     4.5                           \nst, nugget effect
3     22.0  0.0    0.0    0.0       \it,cc,ang1,ang2,ang3
3000.0  3000.0  10.0               \a_hmax, a_hmin, a_vert
```

Appendix 5: Example sgsim.par input file

Parameters for SGSIM
****

```
START OF PARAMETERS:
resid_ns.dat                        \file with data
1  2  0  4  0  0                    \columns for X,Y,Z,vr,wt,sec.var.
-100.0        100.0                 \trimming limits
1                                   \transform the data (0=no, 1=yes)
sgsim.trn                           \file for output trans table
0                                   \consider ref. dist (0=no, 1=yes)
histsmth.out                        \file with ref. dist distribution
1  2                                \columns for vr and wt
-1.0    1.0                         \zmin,zmax(tail extrapolation)
1       0.0                         \lower tail option, parameter
1      15.0                         \upper tail option, parameter
1                                   \debugging level: 0,1,2,3
sgsim.dbg                           \file for debugging output
sgsim.out
1                                   \number of realizations to generate
224    601650.    100.0             \nx,xmn,xsiz
307    3566450.   100.0             \ny,ymn,ysiz
1     0.5     1.0                   \nz,zmn,zsiz
68729
0       8                           \min and max original data for sim
16                                  \number of simulated nodes to use
1                                   \assign data to nodes (0=no, 1=yes)
1       3                           \multiple grid search (0=no, 1=yes),num
4                                   \maximum data per octant (0=not used)
1100.0  1100.0  3.0                 \maximum search radii (hmax,hmin,vert)
0.0    0.0    0.0                   \angles for search ellipsoid
1       0.60    1.0                 \ktype:0=SK,1=OK,2=LVM,3=EXDR,4=COLC
../data/ydata.dat                   \file with LVM, EXDR, or COLC variable
4                                   \column for secondary variable
1     0.20                          \nst, nugget effect
1     0.80   0.0    0.0    0.0      \it,cc,ang1,ang2,ang3
1050.0 1050.0    15.0               \a_hmax, a_hmin, a_vert
```

Appendix 6: SetupRealization shell

```bash
#!/bin/bash
export PATH=/h/wipp/bin:$PATH
REALIZATIONS=$*
SLAVES="slave1 slave2 slave3 slave4 slave5 slave6 slave7 slave8"
DATADIR=/h/wipp/data
ROOTDIR=`pwd`
MFILES=`cat $DATADIR/settings/filelist.master`
SFILES=`cat $DATADIR/settings/filelist.slaves`

for Realiz in $REALIZATIONS
do
    mkdir $ROOTDIR/$Realiz
    chmod 777 $ROOTDIR/$Realiz
    cd $ROOTDIR/$Realiz
    cp $DATADIR/realizations/${Realiz}T.out $ROOTDIR/$Realiz

    # Load the files necessary in the realization main dir
    for File in $MFILES
    do
      cp $DATADIR/$File $ROOTDIR/$Realiz
    done #File in $MFILES

    /h/wipp/bin/base2mod ${Realiz}T.out
    /h/wipp/bin/getSgsimParams > sgsim.console
    cat pcf.top > transient.pst
    cat ppoints.pcf_add >> transient.pst
    cat pcf.bot >> transient.pst

    chmod 664 $ROOTDIR/$Realiz/*

    # Load the files necessary in each slave subdirectory
    for Slave in $SLAVES
    do
      mkdir $ROOTDIR/$Realiz/$Slave
      chmod 777 $ROOTDIR/$Realiz/$Slave
      cp $ROOTDIR/$Realiz/meanT.log.mod $ROOTDIR/$Realiz/$Slave
      # Copy all the files in the SFILES list
      for File in $SFILES
      do
          cp $DATADIR/$File $ROOTDIR/$Realiz/$Slave
      done #File in $SFILES
      chmod 664 $ROOTDIR/$Realiz/$Slave/*
    done #Slave in $SLAVES

    cd $ROOTDIR/$Realiz
    /h/wipp/bin/addRealization $ROOTDIR/$Realiz
    cd $ROOTDIR

done #Realiz in $REALIZATIONS
```

Appendix 7: base2mod source code

```perl
#!/usr/bin/perl
# This program converts a base-T-field to a modflow compatible array file
# This file reads input from the file 'base2mod.set'
# To use individually, add the file you wish to convert to the end of
settings
# or type in the file when prompted. You can also enter input on the command
# line.                             --- dbh, May 5, 2003

open SETTINGS, "base2mod.set" or die "No settings file base2mod.set : Error";

chomp ( $nx = <SETTINGS> );
chomp ( $ny = <SETTINGS> );
chomp ( $fout = <SETTINGS> );

if ( @ARGV[0] ne "" )
{
    $fin = @ARGV[0];
} else {
    chomp ( $fin = <SETTINGS> );
}

if ($fin eq "")
{
    print "Please enter the input file: ";
    chomp ( $fin = <STDIN> );
}

close SETTINGS;

open OUTPUT, ">$fout" or die "Can't open file $fout : Error";
open INPUT, "$fin" or die "Can't open file $fin : Error";

print "Num X in Grid: $nx\n";
print "Num Y in Grid: $ny\n";
print "Reading from: $fin\n";
print "Writing to: $fout\n";

for ( $node = 0; $node < $nx * $ny ; $node++ )
{
    chomp ( $linein = <INPUT> ) ;
    ($blank, $linein) = split / +/, $linein, 2;
    ($xcoor, $linein) = split / +/, $linein, 2;
    ($ycoor, $linein) = split / +/, $linein, 2;
    ($trans, $linein) = split / +/, $linein, 2;
    @trans[$node] = $trans;
}

for ( $node = ( ( $nx * $ny ) - $nx ) ; $node >= 0 ; $node++)
{
    $trans = @trans[$node];
    print OUTPUT "$trans ";
    if ( ($node + 1) % $nx == 0 )
    {
```

69

```perl
        $node -= 2 * $nx;
        print OUTPUT "\n";
    }
}

close INPUT;
close OUTPUT;
exit;

__END__
```

Appendix 8: getSgsimParams shell

```perl
#!/usr/bin/perl
chomp ($realiz = `ls *T.out`);
$realiz =~ s/T\.out//;
$dnum = $realiz;
$rnum = $realiz;
$dnum =~ s/d//;
$dnum =~ s/r[0-9][0-9]//;
$rnum =~ s/d[0-9][0-9]r//;

$realnum = ($dnum - 1) * 10 + $rnum;
$randseed = 69069 - (4 * $realnum);

open SGSIMTPL, "sgsim.par.tpl";
open SGSIMPAR, ">sgsim.par";

$linein = <SGSIMTPL>;
while ($linein ne "")
{
    $linein =~ s/OUTPUT_NAME_HERE/sgsim.out/;
    $linein =~ s/RANDOM_SEED_HERE/$randseed/;
    print SGSIMPAR $linein;
    $linein = <SGSIMTPL>;
}

close SGSIMTPL;
close SGSIMPAR;

system "/h/WIPP/sgsim/bin/sgsim_release <<EOF
sgsim.par
EOF";

open SGSIMOUT, "sgsim.out";

chomp ($linein = <SGSIMOUT>);
chomp ($linein = <SGSIMOUT>);
chomp ($linein = <SGSIMOUT>);
chomp ($linein = <SGSIMOUT>);
@resid[0] = 0;
$node = 0;

while ($linein ne "")
{
    @resid[$node] = $linein;
    chomp ($linein = <SGSIMOUT>);
    $node++;
}

close SGSIMOUT;

open POINTSDAT, "ppoints.nodes";
open ZONESDAT, "ppoints.zones";
open PCFADD, ">ppoints.pcf_add";
```

```
chomp ($linein = <POINTSDAT>);
chomp ($zone = <ZONESDAT>);
$ppoint = 1;

while ($linein ne "" && $ppoints <= 100)
{
    $resid = @resid[$linein] + 3.0;
    if ($ppoint == 30)
    {  # This fixing of pp030 was done at 10:00am on May 30, 2002
      printf PCFADD
"pp%.3d\tfixed\trelative\t%.4f\t2.0000\t4.0000\tzone1\t1.00\t-
3.00\t1\n",$ppoint,$resid;
    }
    elsif ($zone == 1)
    {
      printf PCFADD
"pp%.3d\tnone\trelative\t%.4f\t2.0000\t4.0000\tzone1\t1.00\t-
3.00\t1\n",$ppoint,$resid;
    }
    else
    {
      printf PCFADD
"pp%.3d\tnone\trelative\t%.4f\t0.0001\t6.0000\tzone2\t1.00\t-
3.00\t1\n",$ppoint,$resid;
    }
    chomp ($linein = <POINTSDAT>);
    chomp ($zone = <ZONESDAT>);
    $ppoint++;
}

exit;
```

72

Appendix 9: addRealizations shell

```
#!/bin/bash
REALDIR=$1
ISQUEUED=`cat /h/wipp/data/runs/waiting | grep $REALDIR`
if [[ -z "$ISQUEUED" ]]
then
    echo "$REALDIR" >> /h/wipp/data/runs/waiting
    echo "$REALDIR will be executed asap."
else
    echo "$REALDIR is already queued to run."
fi
```

Appendix 10: runWIPPTrans shell

```perl
#!/usr/bin/perl

system "touch /h/wipp/data/runs/.manager";
chomp ($stop = `cat /h/wipp/data/runs/.manager`);
while ( $stop eq "" )
{
    chomp ($SlavesFree = `CFree | wc -l | awk '{print $1}'`);
    while ( $SlavesFree < 8 && $stop eq "" ) {
      system ("sleep","3600s");
      chomp ($SlavesFree = `CFree | wc -l | awk '{print $1}'`);
      chomp ($stop = `cat /h/wipp/data/runs/.manager`);
    }
    if ($stop eq "")
    {
      open REALS, "/h/wipp/data/runs/waiting";
      chomp ($Realiz = <REALS>);
      close REALS;
      if ($Realiz eq "") {
          $stop = "stop";
      } else {
          system "cat /h/wipp/data/runs/waiting | grep -wv $Realiz >
/h/wipp/data/runs/waiting";
          open RUNNING, ">>/h/wipp/data/runs/running";
          print RUNNING "$Realiz\n";
          close RUNNING;
          system "at -m now <<EOF
/h/wipp/bin/runPest $Realiz
EOF";
      }
    }
    system ("sleep","60s");
}
exit;
```

74

Appendix 11: runPest shell

```bash
#!/bin/bash
TOPDIR=$1
cd $TOPDIR
SLAVES=`ls -d slave* | grep -v slave1`
for Slave in $SLAVES
do
    cd $Slave
    CQue /h/wipp/bin/pslave.sh
    cd $TOPDIR
done
CQue /h/wipp/bin/pmaster.sh
```

Appendix 12: pmaster.sh shell

```
#!/bin/bash
MASTERDIR=`pwd`
FILE=`ls 0* | tail -n 1`
REALIZ=${MASTERDIR##*/}

cd slave1
/h/WIPP/pest/bin/pslave_5.51_release < model.in &> ${FILE}.a &

cd $MASTERDIR

/h/WIPP/pest/bin/ppest_5.51_release transient
wait

/h/WIPP/pest/bin/tempchek_5.5_release points.tpl points.dat transient.par
/h/wipp/bin/model.sh
#dtrk goes here
ln culebra.top fort.33
ln culebra.bot fort.34
/h/WIPP/dtrkmf/bin/dtrkmf_v0100 control.inp steady.bud $REALIZ.trk dtrk.dbg
/h/WIPP/dtrkmf/bin/dtrkmf_v0100 wippctrl.inp steady.bud ${REALIZ}-wipp.trk
dtrk.dbg

ln transient.par  $REALIZ.par
ln transient.rec  $REALIZ.rec
ln transient.res  $REALIZ.res
ln steady.bud     $REALIZ.bud
ln points.dat     $REALIZ.pts.dat
ln Tupdate.mod    $REALIZ.mod

cat /h/wipp/data/runs/running | grep -wv $MASTERDIR >
/h/wipp/data/runs/running
echo $MASTERDIR >> /h/wipp/data/runs/finished
rm /tmp/$REALIZ.jacob.runs
```

Appendix 13: pslave.sh shell

```
#!/bin/bash
/h/WIPP/pest/bin/pslave_5.51_release < model.in
```

Appendix 14: model.sh shell

```bash
#!/bin/bash
export PATH=/h/wipp/bin:$PATH

# Function to set original tolerance levels
ResetTol() {
  echo '3.0      2.2     5.4      0
2     50      1.0E-8    1.0     1 ' > culebra.lmg
}

# Function to raise tolerance levels
RaiseTol() {
    NewTol=$(awk <culebra.lmg 'NR==2 {
      NT = $3*10;
      if (NT <= MaxTol) printf("%6.1E", NT)
    }' MaxTol=0.01)
    [ "$NewTol" ] || { echo mf2k could not converge ; exit ; }
    cat > culebra.lmg <<EOF
3.0      2.2     5.4      0
2     50      $NewTol    1.0     1
EOF
}

# Function to run model: runMF2K()
runMF2K() {

    # Step 0: Write tolerance to a file
    MAIN=$NETDIR
    awk <culebra.lmg 'NR==2 {printf("Slave= %s Tol= %s\n",Dir,$3)}' \
Dir=${NETDIR##*/} >> $MAIN/Tolerance.log


    # Step 1:  Clean up output files from the previous run.
    rm modelled.* *.drw *.hed *.bud *.lst *.parsed


    # Step 2:  Run FAC2REAL to get the residual field
    echo -n 'F'
    /h/WIPP/pest-util/bin/fac2real_release < fac2real.in > /dev/null


    # Step 3:  Add the residual field to the log10()Transmissivity field
    #          to get the t-update field
    echo -n 'C'
    /h/wipp/bin/combine meanT.log.mod residT.log.mod Tupdate.mod


    # Step 4:  Run modflow-2000 on the updated field
    # TESTS are:    steady steady.bin shafts h3 w13 p14 h19
    #               shafts removed march 5, 2003
    TESTS="steady steady.bin h3 w13 p14 h19"
    echo -n 'M'
    for Test in $TESTS
```

```
do
    /h/WIPP/modflow/bin/mf2k_1.6.release $Test > /dev/null
done


# Step 5:   Strip out the heads
#OBSWELLS are: steady shafts h3 w13 p14 wqsp1 wqsp2 h19 h11
#              shafts test removed from list march 5, 2003
OBSWELLS="steady h3 w13 p14 wqsp1 wqsp2 h11 h19"
echo -n 'O'
for ObsWell in $OBSWELLS
do
    /h/WIPP/pest-util/bin/mod2obs_release < in_mod2obs.$ObsWell > /dev/null
done


# Step 6:   Parse the output from mod2obs to contain only modelled values
#           rather than incuding well and time information.  This step is
#           necessary for pest to read observations correctly
echo -n 'P'
for Well in $OBSWELLS
do
    # grab the fourth column
    cat modelled.$Well | awk '{print "   "$4}' > modelled.$Well.parsed
done

# Step 7:   Adjust those observations that do not start at zero drawdown
/h/wipp/bin/adjH19.pl
mv modelled.h19.parse2 modelled.h19.parsed
/h/wipp/bin/adjP14.pl
mv modelled.p14.parse2 modelled.p14.parsed
/h/wipp/bin/adjW13.pl
mv modelled.w13.parse2 modelled.w13.parsed

#           Adjust observations to start at zero drawdown in H-11 test
/h/wipp/bin/adjH11.pl
mv modelled.h11.parse2 modelled.h11.parsed
#           Adjust observations to start at zero drawdown in WQSP tests
/h/wipp/bin/adjWqsp1.pl
/h/wipp/bin/adjWqsp2.pl
mv modelled.wqsp1.parse2 modelled.wqsp1.parsed
mv modelled.wqsp2.parse2 modelled.wqsp2.parsed

# Step 8:   Gather modflow water budjet error data
for Test in $TESTS
do
    TESTERR=`grep PERCENT $Test.lst | tail -n 1 | awk '{print $4}'`
    echo $TESTERR
done > mfPercentErr.parsed

    echo -n ". "
}

NETDIR=`pwd`

#  Make SURE that the desired AMG solver file is good to go
ResetTol
```

```
#   Run the model
time runMF2K

# Part B:   Error handling if modflow crashes
while [ 1 ]
do
#   Runtime error handling for convergence failure
    CONVGFAIL=`grep -i "FAILED TO CONVERGE" *.lst`
    if [ -n "$CONVGFAIL" ]
    then
      # Get the more flexible solver rules to finish run
      RaiseTol
      # Re-run the model
      time runMF2K
    else
      # Put back the good rules when done
      ResetTol
      break
    fi
done
```

Appendix 15: combine source code

```perl
#!/usr/bin/perl
# This command takes two log10 input files and adds them together, putting
# the output into unlog format

$baseField = @ARGV[0];   #perl starts cmdline arguments at 0, not 1
$residField = @ARGV[1];
$outField = @ARGV[2];

open SETTINGS, "combine.set" or die "No settings file combine.set : Error";

chomp ( $nx = <SETTINGS> );
chomp ( $ny = <SETTINGS> );

if ($baseField eq "" || $residField eq "" || $outField eq "")
{
    chomp ($baseField = <SETTINGS>);
    chomp ($residField = <SETTINGS>);
    chomp ($outField = <SETTINGS>);
}

close SETTINGS;

open BASE, "$baseField" or die "Can't open $baseField. Error";
open RESID, "$residField" or die "Can't open $residField. Error";
open OUT, ">$outField" or die "Can't open $outField. Error";

print "Combining $nx x $ny arrays $baseField,$residField into $outField\n";

use POSIX;

$node = 0;
@trans[0] = 0;

$node =( $nx * $ny )- $nx;
$linein = <BASE>;
while ($linein ne "" )
{
    chomp $linein;
    ($trans, $linein) = split / +/, $linein, 2;
    if ($trans eq "") {
       ($trans, $linein) = split / +/, $linein, 2;
    }
    while ($trans ne "" && $trans ne " ")
    {
      @trans[$node] = $trans;
      $node++;
      ($trans, $linein) = split / +/, $linein, 2;
    }
    if ( ($node) % $nx == 0)
    {
      $node -= 2 * $nx;
    }
    $linein = <BASE>;
```

81

```perl
    }
$node =( $nx * $ny )- $nx;
$linein = <RESID>;
while ($linein ne "" )
{
    chomp $linein;
    ($trans, $linein) = split / +/, $linein, 2;
    if ($trans eq "") {
       ($trans, $linein) = split / +/, $linein, 2;
    }
    while ($trans ne "" && $trans ne " ")
    {
      @trans[$node] += $trans;
      $node++;
      ($trans, $linein) = split / +/, $linein, 2;
    }
    if ( ($node) % $nx == 0)
    {
      $node -= 2 * $nx;
    }
    $linein = <RESID>;
}

for ($node = ($nx * $ny) - $nx ; $node >= 0; $node++)
{
    $ntrans = @trans[$node];
#     $ctrans = $ntrans;
    $ctrans = POSIX::pow(10, $ntrans);
    $trans  = sprintf "%.5E",$ctrans;
    print OUT "$trans ";
    if ( ($node + 1) % $nx == 0 )
    {
      print OUT "\n";
      $node -= 2 * $nx;
    }
}

close OUT;
close RESID;
close BASE;
```