September 7, 1995

MEMORANDUM TO: John Austin, Chief
Performance Assessment and
Hydrology Branch, DWM, NMSS

THRU: Norman Eisenberg, Section Leader
Performance Assessment and
Hydrology Branch, DWM, NMSS

FROM: Richard Codell, Sr. Hydraulic Engineer
Performance Assessment and
Hydrology Branch, DWM, NMSS

SUBJECT: PNEUMATIC PATHWAYS MODELING WITH
YUCCA MOUNTAIN PNEUMATIC DATA

I have completed some modeling studies with the pneumatic data provided by Nye County for boreholes NRG-4 and ONC-1. The purpose of this modeling study was to understand and report on the possible conceptual models operative at Yucca Mountain for the propagation of pneumatic pressure variations, and the possible effects of the Exploratory Tunnel Facility on the collection of data at the site. The results of the model were presented at the Pneumatic Pathways Technical Exchange on July 31, 1995, at DOE Headquarters in Washington D.C. I document my preliminary conclusions and the work leading up to these conclusions in the attached report. I also document the computer programs and input files in accordance with Technical Operating Procedure-18. I would be happy to brief you on any aspect of the study.

Attachment: As stated

CONTACT: Richard Codell, PAHB/DWM
415-8167

DISTRIBUTION:
Central File-w/ DWM r/f-w/ MFederline-w/color NMSS r/f-w JGreeves-w/color
PAHB-w/ JSurmeier-w/ PAHB Staff-w/ R Bagtzoglou (CNWRA)
B Sagar (CNWRA)-color

DOCUMENT NAME: S:\DWM\PAHB\RBC\NYEMEMO.RBC

OFC	PAHB	E	PAHB	E		CE				
NAME	RCodell/kv		NEisenberg		DBrooks					
DATE	9/6/95		9/7/95		9/7/95		/ /95		/ /95	

OFFICIAL RECORD COPY

ACNW: YES ✓ NO ___
IG : YES ___ NO ✓
LSS : YES ___ NO ___

Delete file after distribution: Yes ___ No ___

102

NH XT 1/1

9509190182 950907
NMSS SUBJ
102 CF

Hypothesis Testing for Yucca Mountain Pneumatic Models

by

Richard Codell
Sr. Hydraulic Engineer
Performance Assessment and Hydrology Branch
Division of Waste Management, NMSS

Introduction

The proposed Yucca Mountain site consists mainly of multiple layers of volcanic rock, some of which are highly fractured, with a deep unsaturated zone. Air moves through the unsaturated zone because of the influence of such phenomena as atmospheric temperature and pressure changes, wind, and density differences between the atmosphere and moist air in the ground. This exercise focused on the propagation of air pressure variations in the ground from changes in air pressure at the Earth's surface.

The purpose of this exercise was to determine if there is a consistent conceptual model of pressure propagation to explain the measured pressure variations in unsaturated-zone boreholes. This study was conducted on two boreholes only; NRG-4 and ONC-1 for the period March 26-April 20, 1995 prior to the penetration of the Calico Hills by the Tunnel Boring Machine (TBM). Further studies with additional boreholes, and for longer times prior to and after the penetration of the TBM may be conducted later.

Alternative Conceptual models

In this preliminary study, I proposed two highly idealized conceptual models for the propagation of pressure measured in the two boreholes. Pressure measurements are available at a number of packed-off intervals in each borehole, and at the surface.

Conceptual Model 1 - Vertical Permeation

Conceptual Model 1 assumes horizontally continuous layers of rock, with the pressure propagating vertically through all layers, as shown in Figure 1. Additional assumptions of this model are:

- The air in the rock acts as an ideal gas.
- Atmospheric pressure variations and those at depth vary only slightly from the mean pressure. This is a reasonable assumption, because natural pressure variations are less than a few percent, even during the most violent weather conditions.

- Gas flow follows Darcy's law, with flux proportional to the pressure gradient. There is no influence of the Klinkenberg effect, a phenomenon evident in low-permeability media where the porous openings are on the same order as the mean free path (Klinkenberg, 1941).

- Flux is dominated by frictional forces, and inertial forces are negligibly small.

- The pressure at the top of the column is that of the atmosphere. The bottom of the column is the water table, which is assumed to be a no-flow boundary.

For the assumed conditions, the equations of mass flux can be linearized, and the model attributed to Weeks (1978) would apply:

$$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial z}\left(D_z \frac{\partial \phi}{\partial z}\right) \qquad (1)$$

where D_z is the vertical diffusivity, ft^2/sec:

$$D_z = \frac{k\langle p\rangle}{\mu_a n_d} \qquad (2)$$

μ_a = viscosity of air, $lb.sec./ft^2$
k = intrinsic permeability, ft^2
$\langle p\rangle$ = mean pressure, lb/ft^2
ϕ = the pneumatic head, ft.,
n_d = porosity (dimensionless)
and t = time, sec.

The pneumatic head ϕ is defined:

$$\phi = \frac{p}{\rho} + z \qquad (3)$$

where ρ = the density of air, lb/ft^3 and z = height above the datum, ft, chosen in this case to be the water table. The pneumatic head is not actually calculated with Equation 3, but instead makes the assumption that the average head in the borehole is constant:

$$\langle \phi \rangle = \frac{\langle p\rangle}{\rho} + z = constant \qquad (4)$$

Therefore, the head relative to the reference constant head is:

$$\phi - \langle \phi \rangle = \frac{p}{\rho} + z - \frac{\langle p \rangle}{\rho} - z$$
$$= \frac{p - \langle p \rangle}{\rho} \tag{5}$$

Equation 1 is solved using the finite difference method for pressure head specified at the top boundary and no-flow conditions at the bottom boundary. The objective of the finite difference solution is to estimate the value of the unknown diffusion parameter D_z for the segment between two pressure measurement points. The initial calculation starts by specifying the head at the second measurement point, counting up from the bottom, and imposing no-flow conditions at the bottom of the column. The results of the finite difference solution at the location of the first measurement point, counting up from the bottom, are then compared to the measured head variations at that point. Values of the parameter D_z for the first layer are then adjusted iterativly to minimize the mean-squared error between calculated and measured head. Once the error has been minimized, the value of the parameters $D_{z,1}$ and $D_{z,2}$ for the bottom two layers, as shown in Figure 1, is set for the remainder of the calculations. The calculations now move up one layer, setting the pressure for the third point, with the second point now becoming the point at which to compare the measured versus calculated pressure. The Parameter $D_{z,3}$ is adjusted until the error between measured and calculated pressures is minimized. This procedure is repeated until the top layer is reached. The values of the diffusion parameters calculated by this procedure represent the average value between two measurement point, and not necessarily physical layers of rock.

The procedure differs somewhat from that of Weeks (1978) because the objective function for minimization is based on the equally weighted error for all measurement points, instead of only the point immediately below the excitation point. The minimization also uses the more-sophisticated Brent algorithm, with initial bracketing using a Golden ratio search (Press, 1992). A version of the code was also used during the development of these procedures that allowed the analyst to visually fit the results graphically on the computer screen instead of relying on the automatic minimization procedure. However, all results reported here are from the automatic minimization.

Conceptual Model 2 - Horizontal Permeation

Conceptual Model 2 assumes a radically different circumstance for pressure propagation. In this model, it is assumed that there is no propagation of pressure vertically through the rock layers, and that all pressure responses are a result of horizontal movement from extensive vertical fractures or faults a distance L from the measurement borehole, as shown in Figure 2. Under the same general assumptions of Conceptual Model 1, the pressure can be expressed by the PDE.:

$$\frac{\partial \phi}{\partial t} = \frac{D_x}{L^2} \frac{\partial^2 \phi}{\partial \left(\frac{x}{L}\right)^2} \tag{6}$$

where D_x is the horizontal pneumatic diffusivity for each layer:

$$D_x = \frac{k \langle p \rangle}{\mu_a n_d} \tag{7}$$

The boundary conditions for the model are assumed to be atmospheric pressure at x = L, and zero horizontal pressure gradient at x = 0. The parameters of this model are D_x/L^2 for each layer. The solution of the model is similar to that for Conceptual Model 1, except each layer is independent, and therefore the calculations are somewhat simpler.

Model hypothesis testing

The object of the exercise is to determine whether either, both, or neither model can be made to fit the data by adjusting the diffusivity parameters, and then to determine if the fitting parameters make sense in terms of data collected by independent means. The computer programs are exercised with the pressure data from the available boreholes, and the degree to which the two alternative models can be made to fit the data was determined by the squared difference between the measured and modeled pressure head response. Since the diffusivity terms are composed of permeability, porosity and in the case of Conceptual Model 2, distance between fractures and the borehole, it is possible to determine how well the diffusivity terms bracket the ranges of possible values determined independently from these other data. On a plot of permeability k versus porosity n_d, a fixed value of the diffusivity determined for each layer of the model would plot as a straight line, i.e., for Conceptual Model 1, the equation of the line would be:

$$k = \frac{D_z \mu_a n_d}{\langle p \rangle}$$

Likewise, Conceptual Model 2 (for a specified value of L):

$$k = \frac{\left(\frac{D_x}{L^2}\right) L^2 \mu_a n_d}{\langle p \rangle}$$

The data ranges for permeability and porosity would plot as rectangles or brackets in the same space, as illustrated in Figure 3.

Conclusions Based on Preliminary Data

Results of the hypothesis testing procedure for the preliminary data for NRG-4 and ONC-1 for the period March 26-April 20, 1995 (Montezar, 1995) are shown in Figures 4 through 7.

Results for NRG-4

Figure 4 shows the modeled versus predicted pressures for ports 2, 3, 5 and 7 of NRG-4 using Conceptual Model 1. The values of the diffusion parameter were determined using the automatic fitting procedure. The fit between modeled and measured pressures is very good. Figure 5 shows the same fit for Conceptual Model 2. Over much of the range, the fit is good, but it does not agree as well as Conceptual Model 1, especially at early times in the period. In this regard, Conceptual Model 1 appears to be a better choice.

Figure 6 shows a comparison between the modeled value of diffusivity and measured ranges of permeability from Lecain (1994) and air-filled porosity for nearby boreholes from Johnson (1994). Conceptual Model 1 correctly shows the apparent low-permeability layer between the Tiva Canyon and Topopah Springs layers, and is in reasonable agreement to measured values of permeability and porosity.

Figure 7 shows the same comparison for Conceptual Model 2 , for an assumed value of L = 500 ft. Agreement with measured values of permeability and porosity for this model are very good. The model is insensitive to contrasts between layers, but shows very good agreement to measured values within the layers. The preliminary conclusion from the hypothesis-testing exercise for the preliminary data from NRG-4 is that Conceptual Model 1 is slightly superior, but that neither model can be rejected.

The location of the measurement locations for NRG-4, and the estimated diffusivities from Conceptual Models 1 and 2 are shown on Figure 8.

Results for ONC-1

Conceptual Models 1 and 2 agreed much more closely for the ONC-1 data, as show in Figures 9 and 10. The time plots for the optimally fitted parameters from Conceptual Model 1 (Figure 9) and Conceptual Model 2 (Figure 10) gave nearly identical results. Although I have not yet completed a formal comparison similar to Figures 6 and 7 for the ONC-1 data, I compared several values of the parameter D_x/L^2 to measure values of permeability and porosity. On the basis of these comparisons, Conceptual Model 2 gave more reasonable results. Conceptual Model 1 predicted essentially infinite permeability for the top few layers of ONC-1, indicating preliminarily that pressure could propagate vertically through sizable conduits in the rock, a situation more consistent with Conceptual Model 2. Montezar (1995) reached a similar conclusion in his analysis of ONC-1 with the code AIRTOUGH.

The measurement locations and estimated diffusivities from Conceptual Models 1 and 2 are shown on Figure 11.

Plans for future work

This preliminary exercise acknowledged the importance of modeling and hypothesis testing to accept or reject alternative conceptual models, and to direct attention to the kinds of data collection and analyses that would further add to the confidence in model predictions. Logical extensions to this work include:

- Expand the coverage to other boreholes with data available, and longer periods of time to see if the preliminary models are robust.
- Examine pre and post-ESF penetration data to see which model fits the data better.
- Try additional data periods on NRG-4 and ONC-1 to see if parameter estimates are stable.
- G. Bodvarrson (1995) had success using the TOUGH code in a 3-dimensional modeling study to determine the parameters of the system. The approach was straightforward: find the set of laterally homogeneous parameters that best matched the pneumatic data. He was able to get reasonable matches for pneumatic responses in this way, but I am not comfortable with the assumption that the pneumatic parameters of the model were homogeneous over many kilometers spatial separation. It would be useful to try to match several of the boreholes using our much simpler one-dimensional models with the constraint that the layer are homogeneous to see if we would get equally good fits as Bodvarrson.

- Use the techniques of time-series and spectral analyses to extract information from the measured and computed pressures. For example, filtering the high-frequency variations from the pressure responses might allow closer inspection of the differences between the measured and modeled responses. The frequency spectra and phase of the time series might also be useful in discriminating differences between the results of alternative conceptual models. I have already used a variety of techniques such as Fast Fourier Transforms and digital filters (Newland, 1975, Press, 1992, StatSci, 1995) in preliminary stages of this project, although they were not reflected in the results presented here.

Any further work performed on the pneumatic pathways issue will be coordinated with the Vertical Slice on "Location and Characterization of Structural Features which Significantly Affect Water Vapor Movement".

References

Klinkenberg, L.J., "The permeability of porous media to liquids and gasses", American Petroleum Institute, Drilling and Production Practices, p200, 1941.

Weeks, E., "Field determination of vertical permeability to air in the unsaturated zone", U.S. Geological Survey Professional Paper 1051, 1978.

Press, W.H., B.P. Flanery, S.A. Teukolsky, and W.T. Vetering, Numerical Recipes, Cambridge University Press, 1992.

Montezar, P., "Interim report on results of instrumentation and monitoring of UE-25 ONC#1 and USW NRG-4 boreholes, Yucca Mountain, Nevada", Multimedia Environmental Technologies, Inc., Newport Beach California, July 1995.

Nelson, P. "Saturation Levels and Trends in the unsaturated zone, Yucca Mountain, Nevada", in High-Level Radioactive Waste Management, American Nuclear Society, LaGrange Park Ill, pp2774-2781, May 1994.

Lecain, G.D., and J.N. Walker, "Results of air permeability testing in a vertical borehole at Yucca Mountain Nevada" in High-Level Radioactive Waste Management, American Nuclear Society, LaGrange Park Ill, pp2782-2788, May 1994, also, unpublished results from author.

Newland, D.E., An Introduction to Random Vibrations and Spectral Analysis, Longman Group, Ltd., London, 1975.

Bodvarrson, G., model results presented at Pneumatic Pathways Technical Exchange, Department of Energy headquarters, Washington D.C., July 31, 1995.

Stat-Sci, S-Plus software.

Model 1 - Vertical Permeation of Pressure

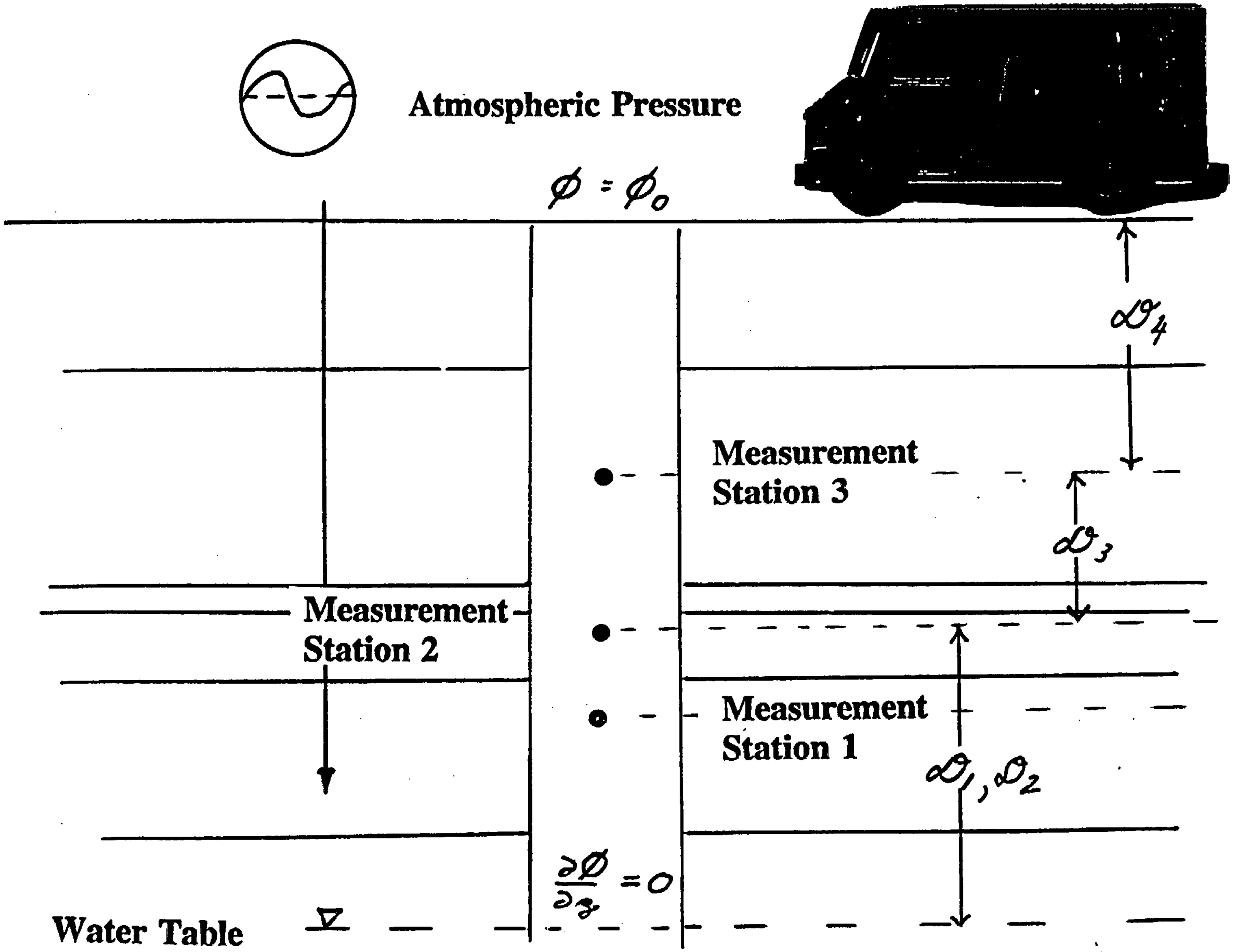

Figure 1 - Conceptual Model 1 for Vertical Permeation

Model 2 - Horizontal Permeation of Pressure

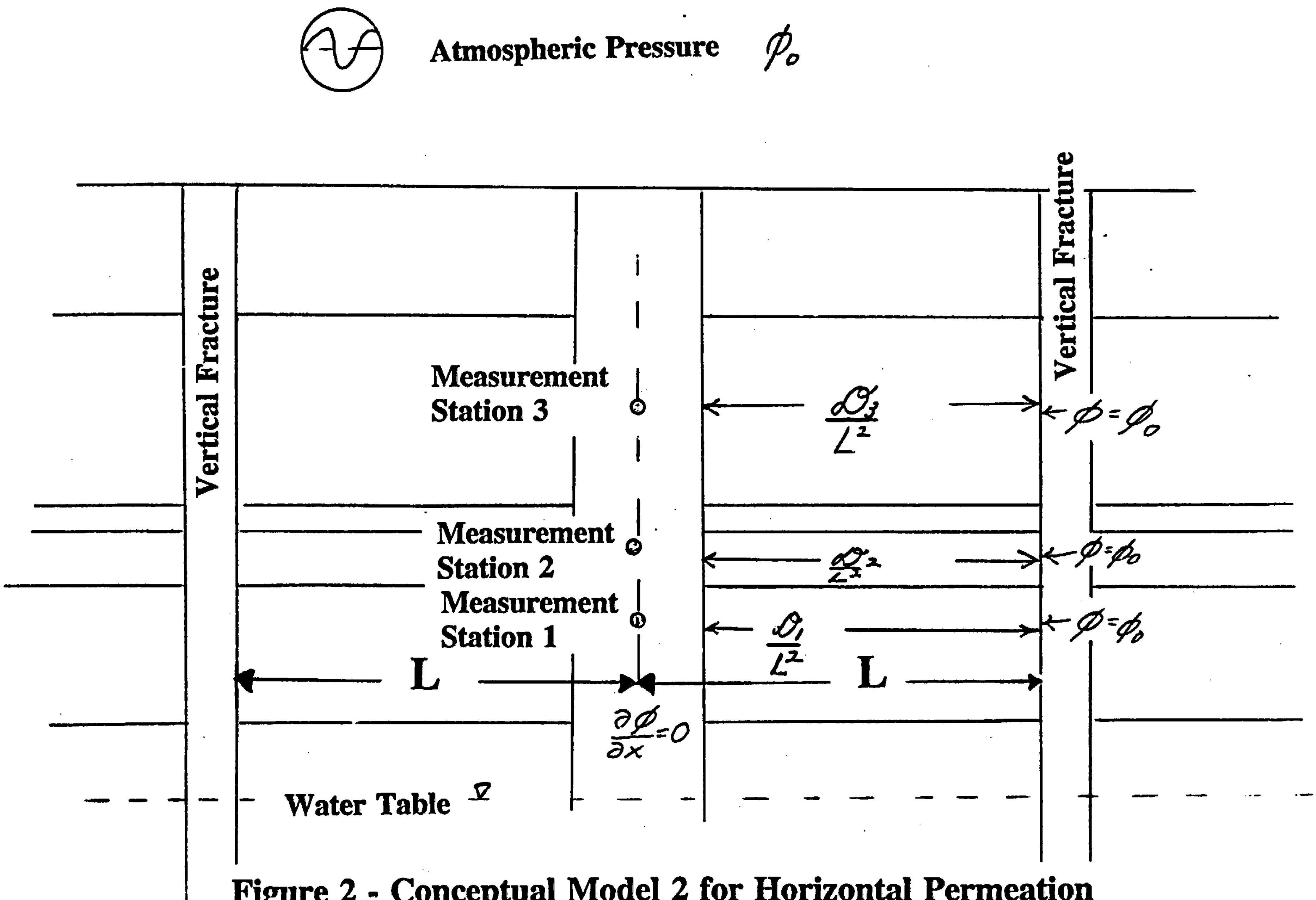

Figure 2 - Conceptual Model 2 for Horizontal Permeation

(example) Validation of Model Diffusivity

with field data on k and porosity

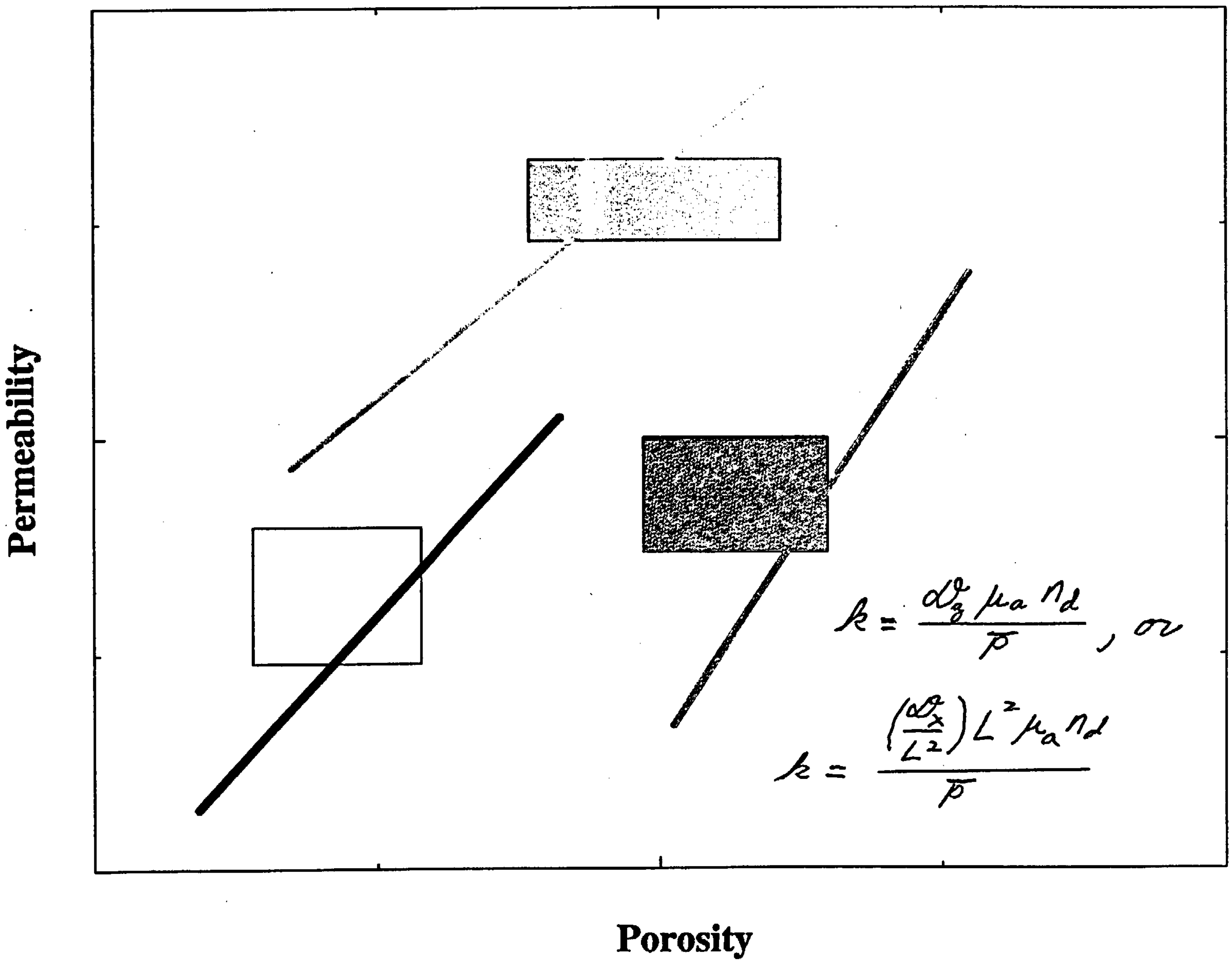

Figure 3 - Example Validation of Model Diffusivity

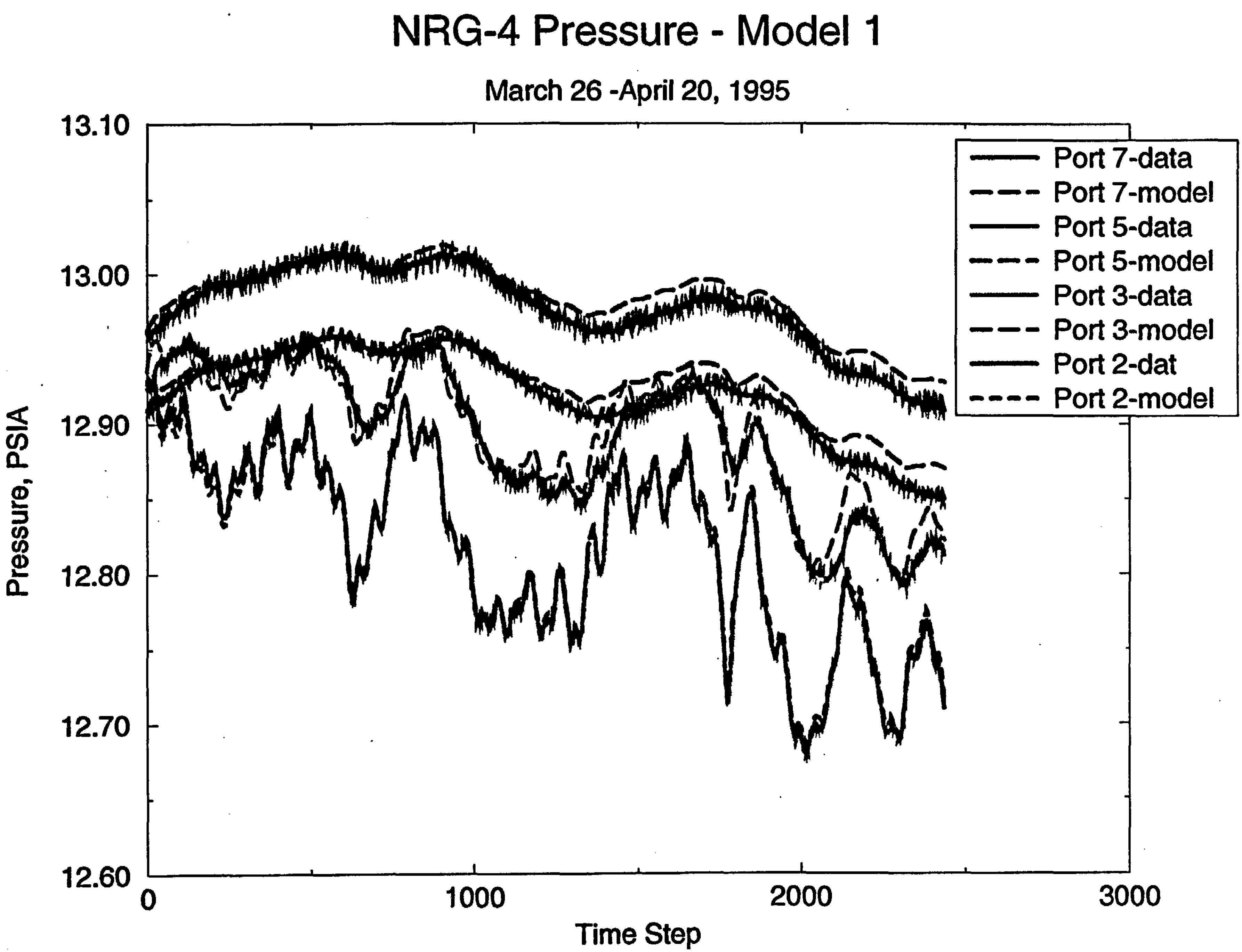

Figure 4 - Modeled and Measured Pressures for NRG-4 Conceptual Model 1

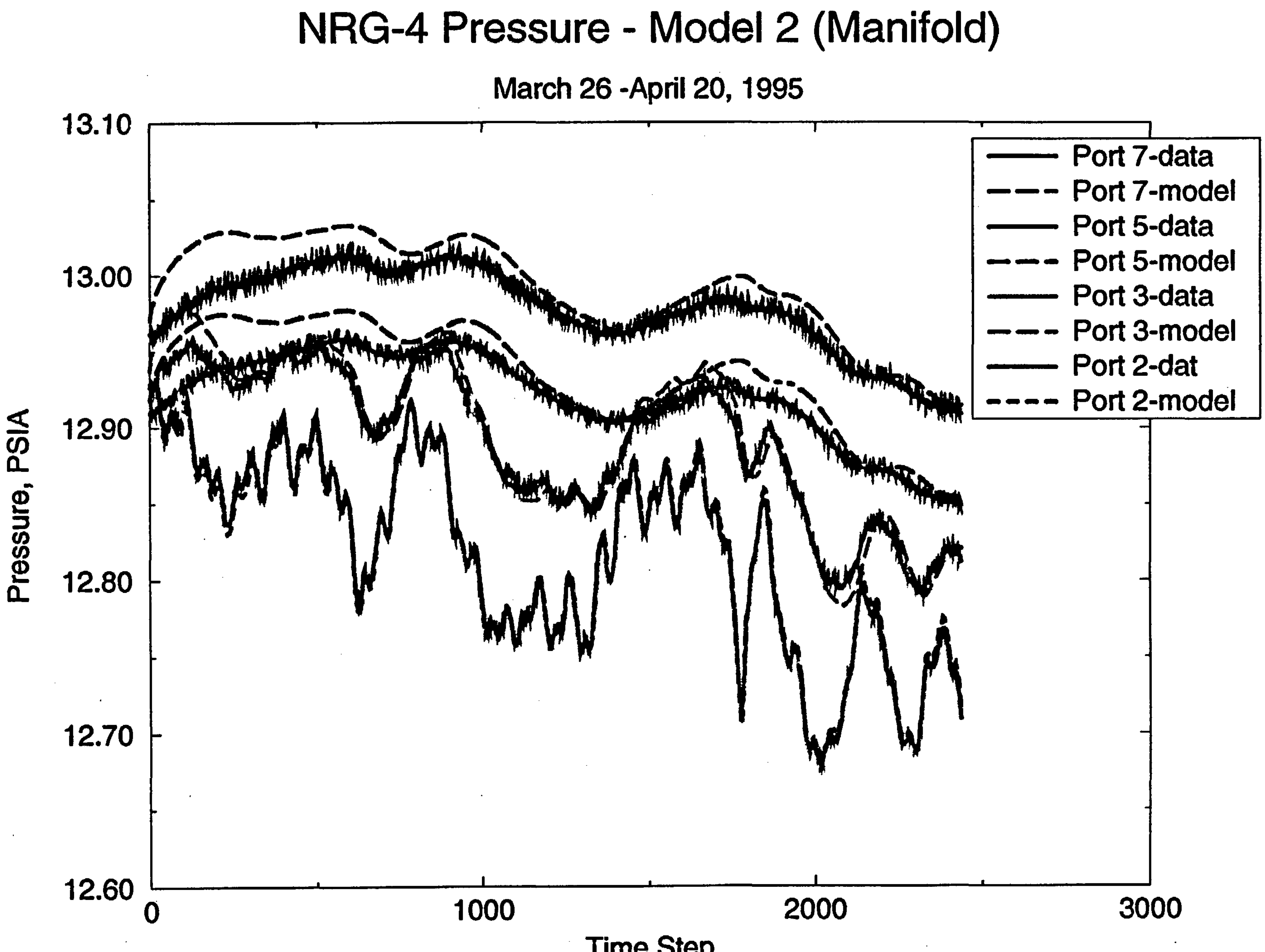

Figure 5 - Modeled and Measured Pressures for NRG-4 Conceptual Model 2

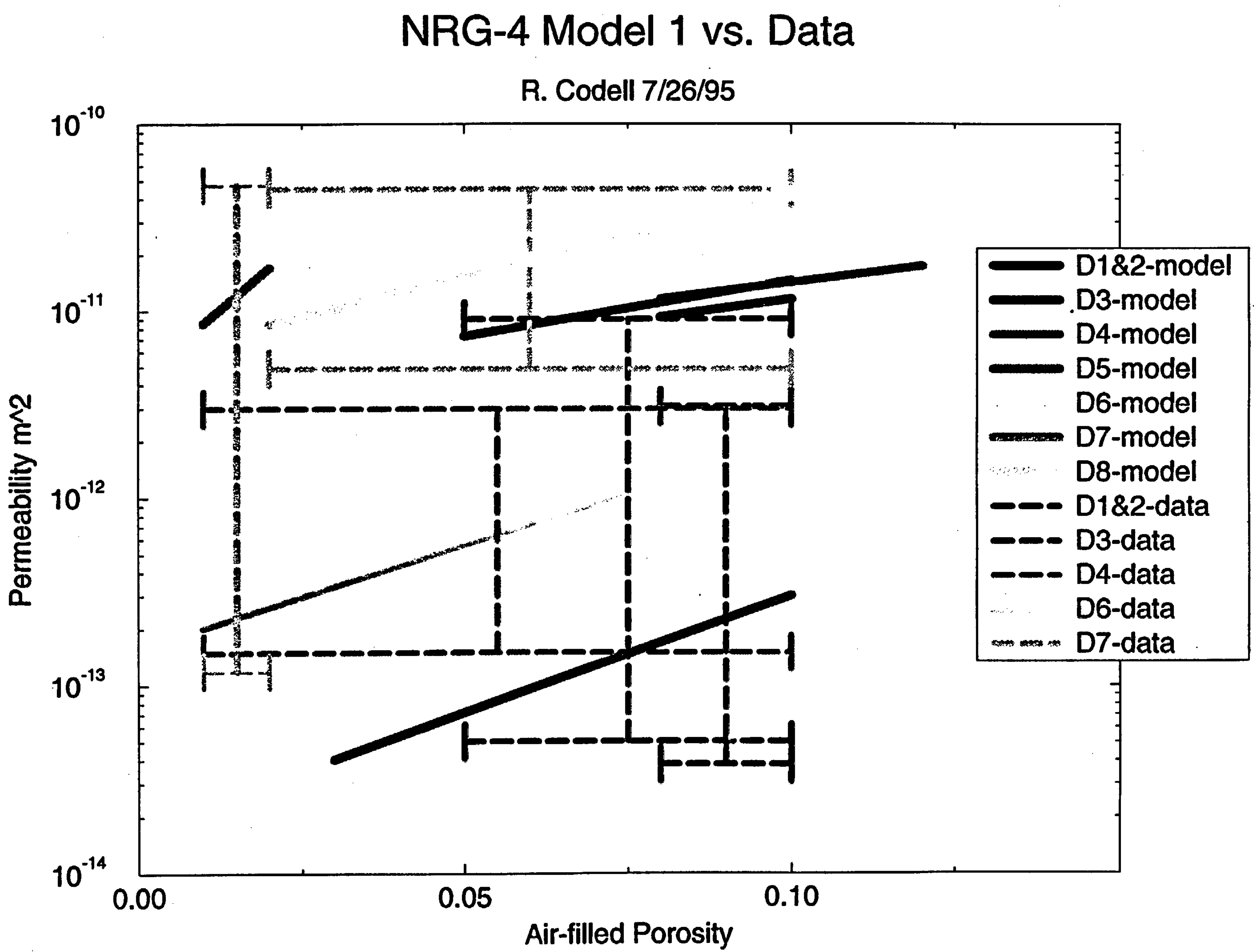

Figure 6 - Validation of Diffusivities from NRG-4 using Conceptual Model 1

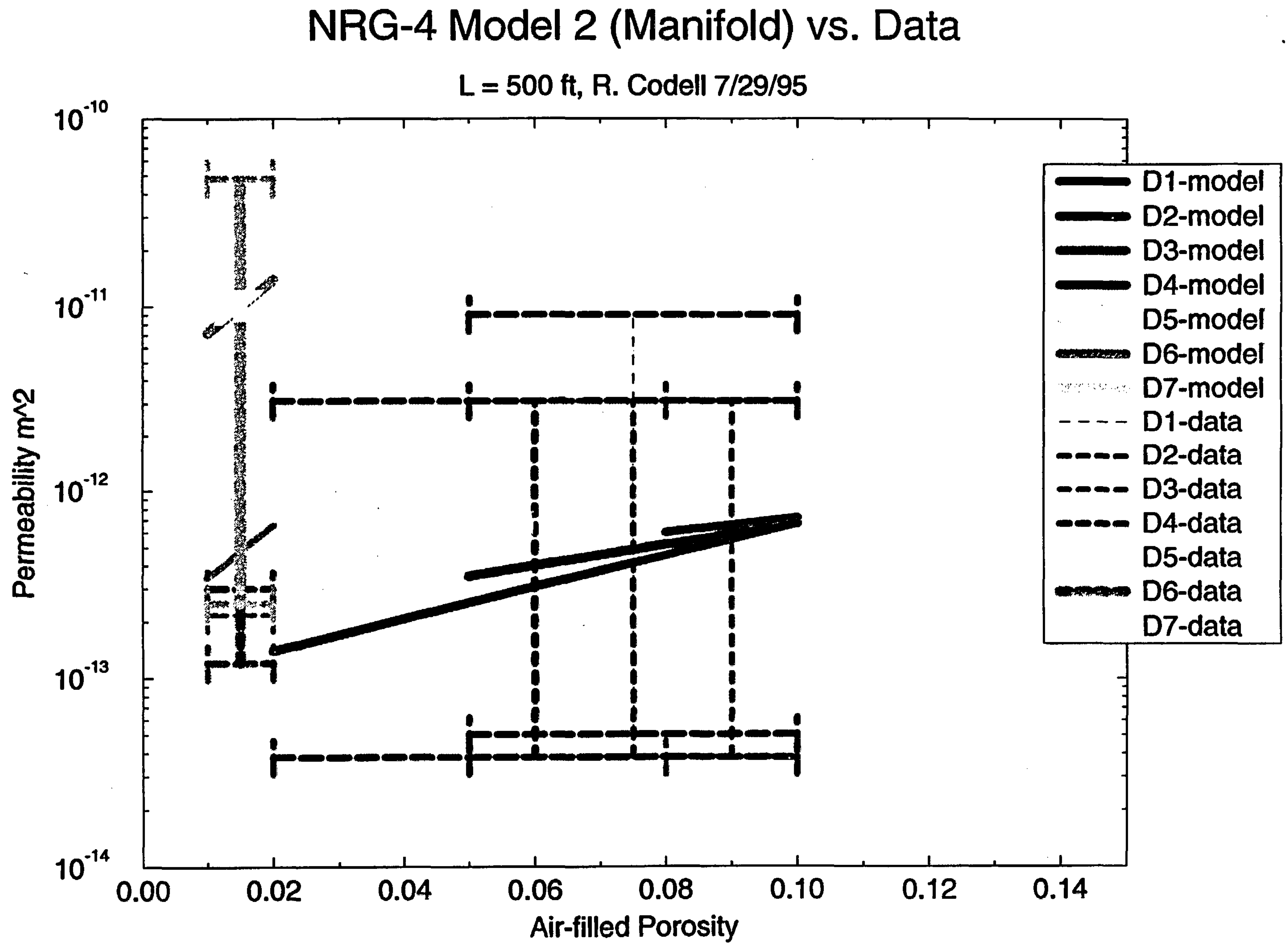

Figure 7 - Validation of Diffusivities from NRG-4 using Conceptual Model 2

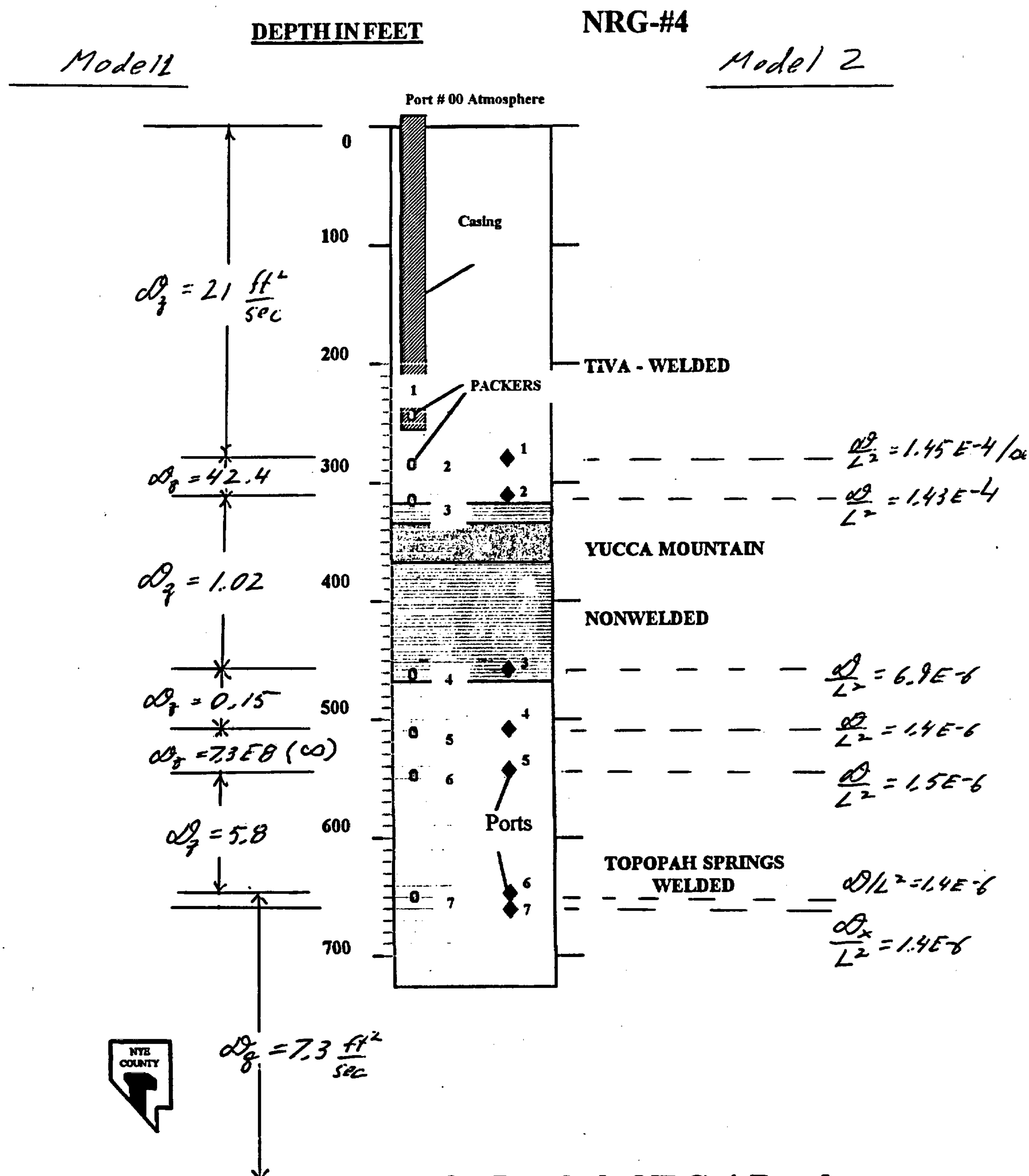

Figure 8 - Borehole NRG-4 Results

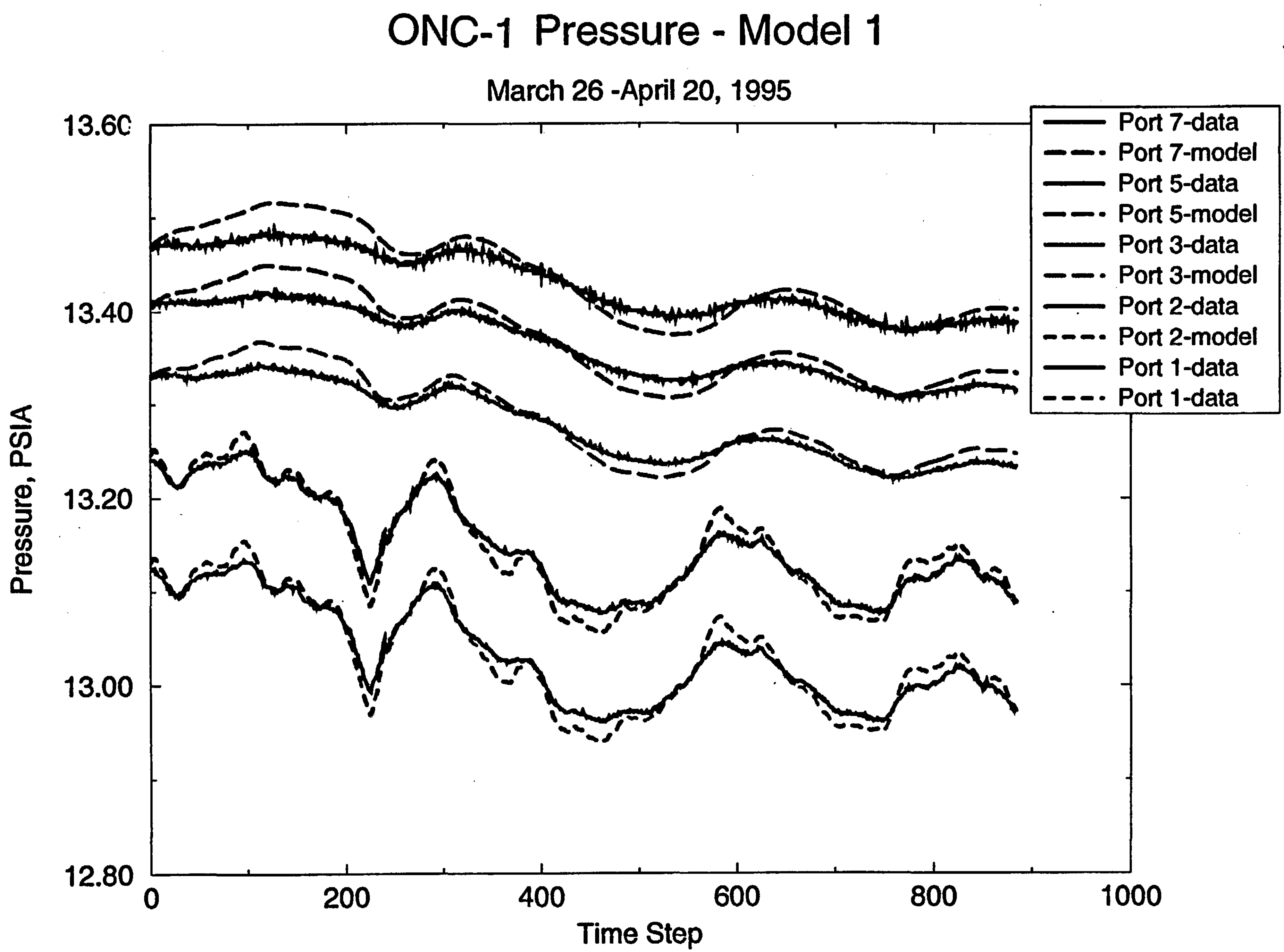

Figure 9 - Modeled and Measured Pressures for ONC-1 Conceptual Model 1

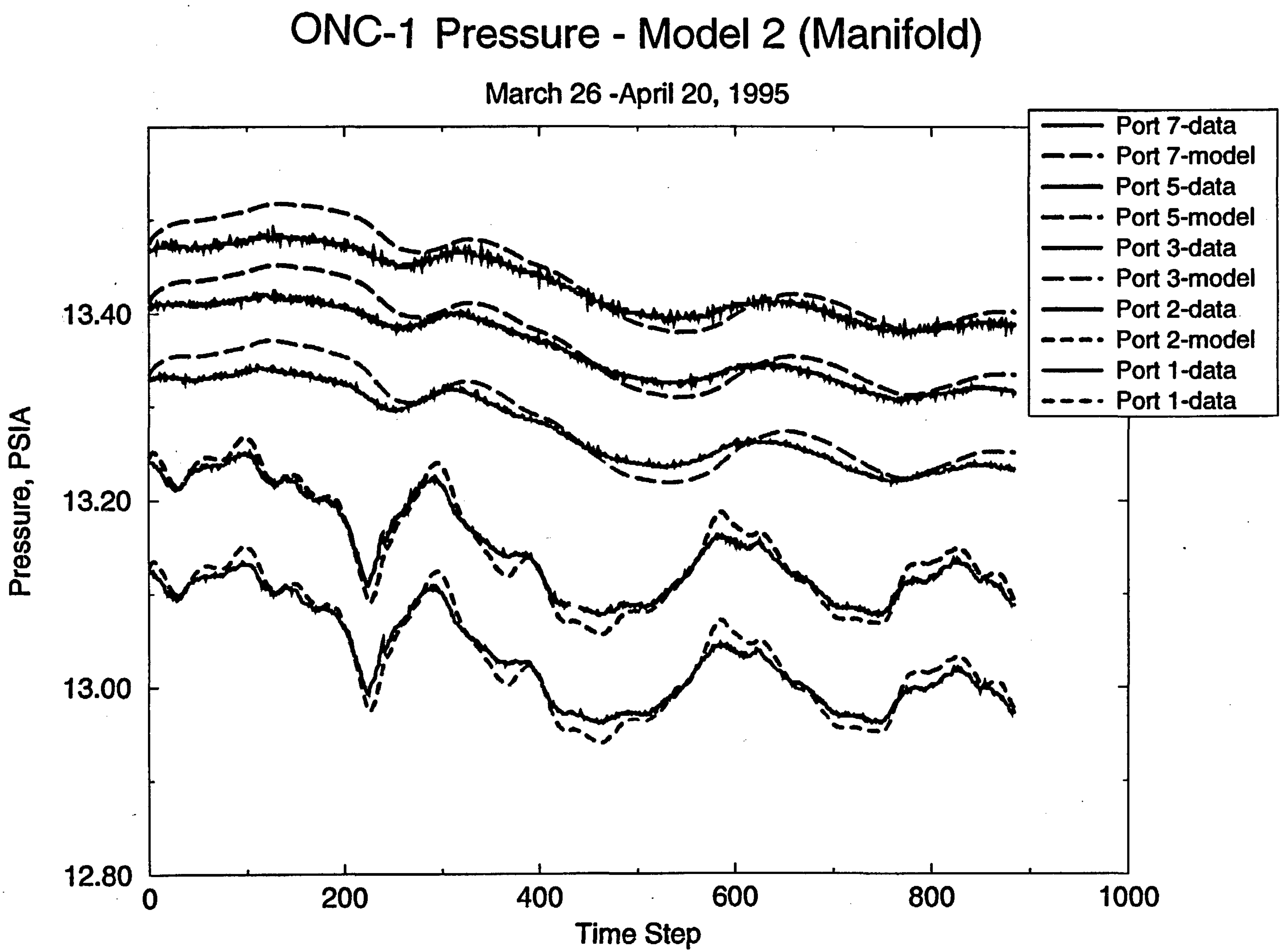

Figure 10 - Modeled and Measured Pressures for ONC-1 Conceptual Model 2

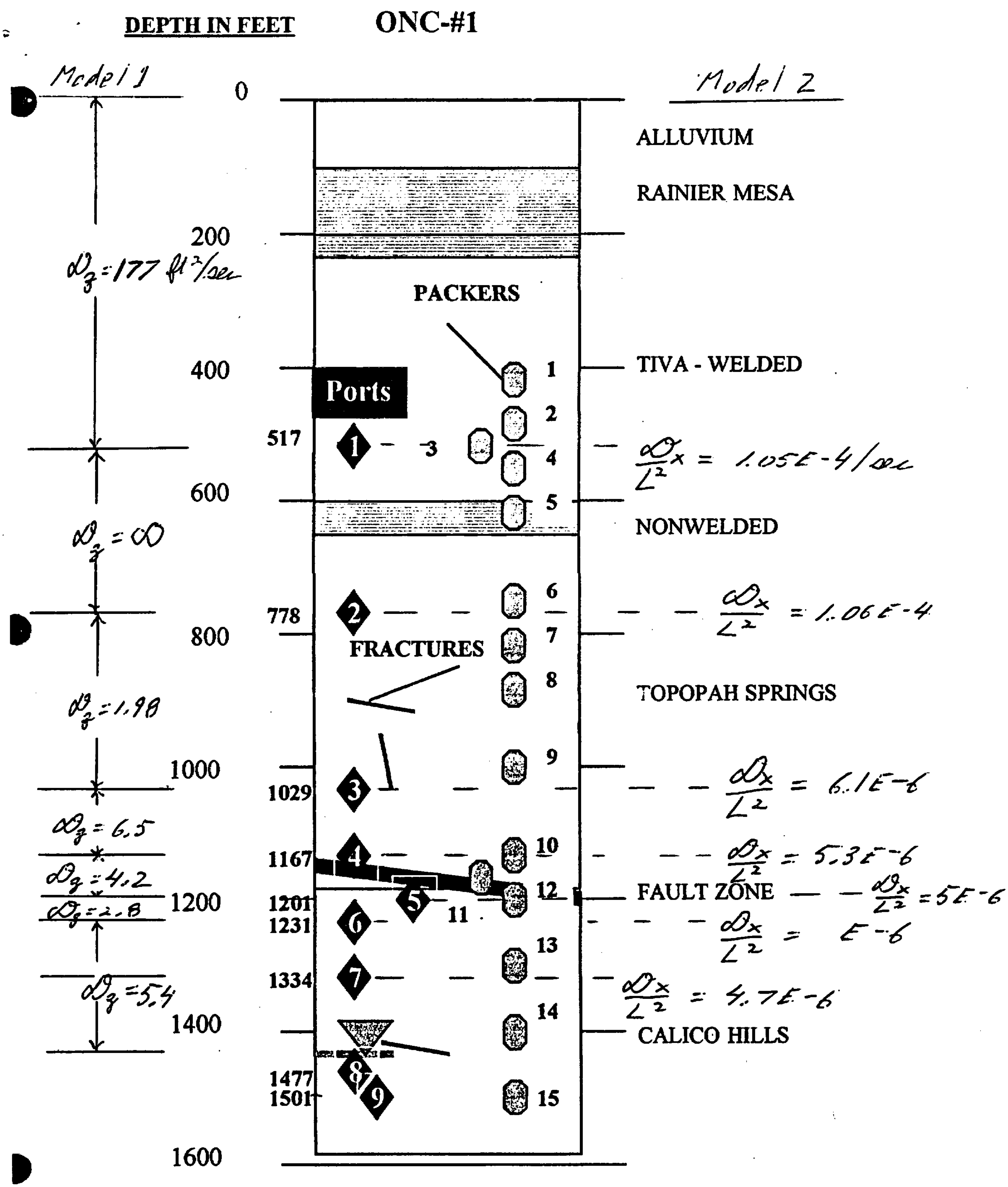

Figure 11 - Borehole ONC-1 Results

APPENDIX

Documentation for Programs AIRDIF and MANIFOLD

by

Richard Codell

Performance Assessment and Hydrology Branch

Division of Waste Management, NMSS

Introduction

Program AIRDIF is a program to estimate the diffusion coefficients for the propagation of air pressure variations from the surface through assumed layers of rock or soil. The model solves the partial differential equation (PDE) for the diffusion of pressure using the finite difference method, with a fully implicit backward-in-time integration. The calculated pressure heads from the model are compared to the measured values, and the differences minimized using a combination of a Golden ratio search and Brent's minimization (Press, 1992). Program MANIFOLD is similar in concept, but assumes that there is no vertical permeation of pressure, and that there are extensive fractures down to depth that allow the instantaneous pressure response to be felt to the lowest layers of rock. All pressure responses to the borehole therefore would permeate horizontally through each layer. The bases of these models are described in the body of this report.

Requirements of Computer Codes

Program AIRDIF is required to accomplish the following tasks:

1. Read an input file consisting of fundamental parameters including title, the name of the file containing measured pressures, number of measurement locations in the vertical borehole, the elevation with respect to the water table of each measurement location, the mean density of air in the borehole, the time increment of the measured data, and the target spatial interval between each grid point in the finite difference model.

2. Read an input file named in the first input file for the time series of pressure measurements at each of the specified measurement locations for equally spaced time interval specified in the first input file.

3. Convert measured pressures to head.

4. Set up a finite difference grid for the equation:

$$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial z}\left(D_z \frac{\partial \phi}{\partial z}\right) \tag{1}$$

where D_z is the vertical diffusivity:

$$D_z = \frac{k\langle p\rangle}{\mu_a n_d} \tag{2}$$

k = intrinsic permeability
μ_a = viscosity of air
<p> = mean pressure
ϕ = the pneumatic head,
n_d = porosity
t = time

Set the approximate spatial interval specified in the first input file, with adjustments made for the distance between measurement points in the vertical borehole. The boundary conditions for the finite difference model are:

- head specified at the top node of each calculational interval.

- no-flow specified at bottom, which is assumed to be the water table.

5. Solve the pressure response for intervals from bottom to top. The first calculation sets the measured pressure at the second node from the bottom, and uses the measured pressure from the bottom measured node to compare to the model results. Subsequent calculations step up the column, setting the measured pressure at the top node, and using the next node down to compare to the measured head, until the top layer is reached.

6. Each interval calculation described in 5 will be used to determine the optimal value of D_z that minimizes the squared difference for the entire measurement period between the modeled and measured head for all points below the top interval.

7. Upon completion of the iterations, the program will output the optimal values of D_z for each interval between measurement points.

8. Upon completion of the iterations, the program will convert calculated heads back to pressures, and output a file of measured and modeled pressures suitable for plotting with an external plotting program.

Program MANIFOLD is required to accomplish the following tasks:

1. Read an input file consisting of fundamental parameters of the run, including title, file names, number of measurement locations in the vertical borehole, the elevation with respect to the water table of each measurement location, the mean density of air in the borehole, the time increment of

the measured data, and the target spatial interval between each grid point in the finite difference model.

2. Read an input file named in the first input file for the time series of pressure measurements in PSIA at each of the specified measurement locations for equally spaced time interval specified in the first input file.

3. Convert measured pressures to head.

4. Set up a finite difference grid for the following PDE:

$$\frac{\partial \phi}{\partial t} = \frac{D_x}{L^2} \frac{\partial^2 \phi}{\partial \left(\frac{x}{L}\right)^2}$$

where D_x is the horizontal pneumatic diffusivity

This model assumes horizontal permeation of pressure from a vertical fracture L feet from the borehole. The boundary conditions for the finite difference model are:

- head specified at the vertical fracture.
- no-flow specified at the borehole.

5. Solve the pressure response for each interval, assuming it is independent computationally from each other interval.

6. Each interval calculation described in 5 will be used to determine the optimal value of the parameter D_x/L^2 that minimizes the squared difference for the measurement record between the modeled and measured head.

7. Upon completion of the iterations, the program will output the optimal values of D_x/L^2 for each interval between measurement points.

8. Upon completion of the iterations, the program will convert calculated heads back to pressures, and output a file of measured and modeled pressures suitable for plotting with an external plotting program.

<u>General data input file for AIRDIF and MANIFOLD</u>

Line 1 - Title - any title up to 80 characters
Line 2 - fn2 - the name of the file containing the pressure data
Line 3 - npmeas - The number of pressure measurement locations to be used from pressure file
Line 4 - form - The input format for the pressure file. The first column is the time (read but ignored - must be a

character format). Next npmeas columns in numerical format. First pressure measurement column should be atmospheric pressure, then numbered left to right, top to bottom measurement stations.

Lines 5 to 5+npmeas - elevations of measurement points with respect to water table, ft, starting at top measurement point.

Next line - Elevation of water table above datum, ft

Next line - rho, dt, dztarg*
rho = average density of air, lb/cubic feet
dt = equal time interval between pressure measurements, seconds
dztarg = approximate distance between grid cells, ft (*Note, dztarg is not used in program MANIFOLD, and may be left out).

Pressure input file for AIRDIF and MANIFOLD

Data in this file consists of measured pressures at the specified depths. The data are assumed to be in equal timestep increments dt seconds (specified in the general input file). The data are read in with the format specified in the general input file. The first column is the time, but is ignored. The next npmeas columns are assumed to be numbered from top (atmospheric pressure usually) to the bottom measurement point with the specified format. The programs read the data until they encounters an end-of-file. The programs are presently dimensioned for up to 2500 times in file, but can be increased by changing "npts" in the PARAMETER statements, and recompiled.

Example Input Files

The following files are examples of the input to AIRDIF and MANIFOLD for the NRG-4 data set. There are 8 stations for pressure measurements in the borehole. The station nearest to the surface is 1440 ft. above the water table. The average density of the air is assumed to be 0.08 lb/ft^3. The pressure measurements are 900 seconds apart. The target grid interval is 10 ft.

Parameter Data File for AIRDIF and MANIFOLD, file "nrg4.in". The same file is used for both, except MANIFOLD does not read the last entry on the last line, "dztarg".

```
nrg4 data
file1b.dat
8
(a14,8f10.0)
1440,1240,1128,970,930,896,793,778
0
0.080,900,10
```

Pressure Input File for AIRDIF and MANIFOLD, file "file1b.dat"
(First 6 lines only, lines are wrapped)

```
      09:18:00   12.8057   12.9165   12.9273   12.9292   12.8978
12.9092   12.9612   12.9616
      09:19:00   12.8057   12.9174   12.9282   12.9282   12.8978
12.9101   12.9597   12.9625
      09:20:00   12.8057   12.9162   12.9273   12.9291   12.8978
12.9101   12.9606   12.9613
      09:21:00   12.8084   12.9153   12.9265   12.9291   12.8978
12.9110   12.9591   12.9632
      09:22:00   12.8075   12.9171   12.9273   12.9281   12.8978
12.9094   12.9591   12.9622
      09:23:00   12.8066   12.9171   12.9282   12.9281   12.8960
12.9110   12.9599   12.9611
```

```
      . program airdif
c     Determine coefficients of diffusion by matching measured
c     versus calculated heads for Nye County and Yucca Mountain
c     pneumatic data
c     Vertical pressure propagation model 1
c     pressure data expected with atmospheric pressure first, and
c     then with increasing depth, and in equal time increments
c
c     R. Codell U.S. Nuclear Regulatory Commission
c     Washington D.C.
c     (301)415-8167
c     August 21, 1995
c
c     idim = dimensions of grid
c     nsta = number of measurement stations
c     npts = maximum number of time steps
c     p, pp = pressure in equation
c     dz = array of space steps between pressures
c     dzb = average space steps between pressures
c     d = diffusion coefficient
c     db = diffusion coefficient avergaged between pressures
c     zw = height of well measurement points, ft
c     n2 = location in grid of pressure measurement point at top
c     trid = matrix for finite difference backward in time solution
c     dt = time step, seconds
c     phi = head, ft
c
      implicit real*8(a-h,o-z)
c     func = function to minimize for lower layers
c     func1 = function to minimize for layers above
      external func,func1
      common p,pp,n2,trid,d,db,dz,dzb,dt,phi,klast,dzbig,zwt,
     1 lay,dstore,pn,npmeas
      parameter (idim=200,npts=2500,nsta=9)
      real*8 trid(idim,4),phi(npts,nsta),dz(idim),dzb(idim),db(idim),
     1 p(idim),pp(idim),d(idim),dzbig(nsta),zw(nsta),dtest(30),
     2 dstore(nsta),pn(npts,nsta)
      integer n2(nsta)
      character*14 tchar
      character*80 title,form
      character*20 fn1,fn2
c     file for basic constants
      write(6,*)' enter file name for input data '
      read(5,'(a)') fn1
      open(1,file=fn1)
      read(1,'(a)') title
      write(6,*) title
      read(1,'(a)') fn2
c     file for pressures in psig
      open(2,file=fn2)
c     read in the air pressure data
c     npmeas is the number of pressure measurement locations
      read(1,*) npmeas
c     form is the input format
      read(1,'(a)') form
c     read elevations of measurement points, feet
      read(1,*)(zw(l),l=npmeas+1,2,-1)
c     read in elevation of water table
      read(1,*) zwt
      zw(1)=zwt
```

```
c       dzbig = distance between measurement points, ft
c       first interval from bottom
        dzbig(1)=zw(2)-zwt
c       rest of intervals
        do i=2,npmeas
          dzbig(i)=zw(i+1)-zw(i)
        end do
c       read pressures psia (must be in equal time intervals)
        k=0
1       continue
        k=k+1
        read(2,form,end=2) tchar,(phi(k,l),l=npmeas,1,-1)
        go to 1
c       index of last value for presure points
2       klast=k-1
c       read in parameters of model
c       rho = density of air, lb/ft^3
c       dt = time step, seconds
c       dztarg = approximate distance step size, ft
c
        read(1,*) rho,dt,dztarg
c       pcon is conversion factor psia to head, ft
        pcon=144/rho
c       iz = index for grid numbering from bottom
        iz=0
c       dzbig = length of zone between two measurement points
c       n2 = location of top pressure in zone
        do l=1,npmeas
c         calculate the delta z between pressure stations
c         calculate number of grid steps in zone
          nz=dzbig(l)/dztarg+1
          dz1=dzbig(l)/nz
          do i=1,nz
            iz=iz+1
            dz(iz)=dz1
          end do
c         n2 = grid number index of each pressure measurement point
          n2(l)=iz+1
        end do
c       calculate the dzb, which is the average grid spacing centered on block e
        do j=2,iz
          dzb(j)=(dz(j-1)+dz(j))/2
        end do
c       convert pressure in psia to head in feet
        npm=npmeas
        call p2head(phi,pcon,zw,klast,npm)
c       start the pressure calculations with the lowest layer, specifying the
c       next to last pressure as a boundary condition and testing the goodness
c       of fit for the last pressure to variations in D
c
c       run through a set of diffusion values to determine starting points
        data dtest/0.0,.0001,.0003,.001,.003,.01,.03,.1,.3,1.,3.,10.,
     1  30.,50.,100.,200.,500.,1000.,3000.,11*10000./
        f2=func(dtest(1))
        do itest=1,20
          f1=f2
          f2=func(dtest(itest))
          delta=f2-f1
          if(delta.lt.0.0) then
c           slope has changed, pick these points
```

```
            diffa=dtest(itest-1)
            diffb=dtest(itest)
            go to 10
          end if
        end do
c       could not find suitable values
        write(6,*)' no suitable starting values for diffa and diffb '
10      continue
c       further bracket the diffusion values in interval using
c       a golden rule search (Numerical Recipes, 1992)
c
        call mnbrak(diffa,diffb,diffc,fa,fb,fc,func)
        write(6,*) 'diffa,diffb,diffc,fa,fb,fc'
        write(6,*) diffa,diffb,diffc,fa,fb,fc
        tol=1.0e-2
c       minimize function using Brent's algorithm
        zz=  brent(diffa,diffb,diffc,func,tol,xmin)
        write(6,*)' xmin = ',xmin
c       store the diffusivity for first two layers
        dstore(1)=xmin
        dstore(2)=xmin
c       cycle through the other layers above
        do lay=3,npmeas
c         run through set of diffusion values to determine starting points
c         find the minimum
          amin=1.e30
          do itest=1,20
            f2=func1(dtest(itest))
            if(f2.lt.amin) then
              amin=f2
              imin=itest
            end if
          end do
          diffa=dtest(imin-1)
          diffb=dtest(imin)
c         Bracket with Golden rule search
          call mnbrak(diffa,diffb,diffc,fa,fb,fc,func1)
          write(6,*) 'diffa,diffb,diffc,fa,fb,fc'
          write(6,*) diffa,diffb,diffc,fa,fb,fc
          tol=1.0e-2
c       minimize with Brent algorithm
          zz=  brent(diffa,diffb,diffc,func1,tol,xmin)
          write(6,*)' xmin = ',xmin
          dstore(lay)=xmin
        end do
        write(6,*)' diffusion coefficients '
        write(6,*) dstore
        open(7,file='pnorm.dat')
c       convert head back to pressure
        call head2p(phi,pcon,zw,klast,npm)
        call head2p(pn,pcon,zw,klast,npm)
        do i=1,klast
          write(7,'(i5,14f16.8)') i,(phi(i,j),pn(i,j),j=1,npmeas-1)
        end do
c
        stop
        end

        real*8 function func(diff)
c       function to minimize for lower layers
```

```
      implicit real*8(a-h,o-z)
      common p,pp,n2,trid,d,db,dz,dzb,dt,phi,klast,dzbig,zwt,
     1 lay,dstore,pn,npmeas
      parameter (idim=200,npts=2500,nsta=9)
      real*8 trid(idim,4),phi(npts,nsta),dz(idim),dzb(idim),db(idim),
     1 p(idim),pp(idim),d(idim),dzbig(nsta),dstore(nsta),
     2 pn(npts,nsta)
      integer n2(nsta)
      call bottom (diff,error)
      func=error
      return
      end

      subroutine bottom(diff,error)
      implicit real*8(a-h,o-z)
      parameter (idim=200,npts=2500,nsta=9)
      common p,pp,n2,trid,d,db,dz,dzb,dt,phi,klast,dzbig,zwt,
     1 lay,dstore,pn,npmeas
      real*8 trid(idim,4),phi(npts,nsta),dz(idim),dzb(idim),db(idim),
     1 p(idim),pp(idim),d(idim),dzbig(nsta),dstore(nsta),
     2 pn(npts,nsta)
      integer n2(nsta)
c     function for bottom layers, assume no flow at water table
c
c     set all tridiagonal nodes that stay fixed for run
c     make first node a no-flow, p(1)=p(2)
      trid(1,1)=0
      trid(1,2)=1
      trid(1,3)=-1
      trid(1,4)=0
c     set the node diffusion coefficients
      do i=1,n2(2)
        db(i)=diff
      end do
c     set the middle node coefficients that don't change
      do i=2,n2(2)-1
        trid(i,1)=db(i)/(dz(i-1)*dzb(i))
        trid(i,2)=-(db(i+1)/dz(i)+db(i)/dz(i-1))/dzb(i)-1.0/dt
        trid(i,3)=db(i+1)/(dz(i)*dzb(i))
      end do
c     top node in interval
      trid(n2(2),1)=0
      trid(n2(2),2)=1
      trid(n2(2),3)=0
c     set the initial pressures by linear interpolation
c     p(1)=phi(1,1)+dzbig(1)
      p(1)=phi(1,1)
      do i=2,n2(1)-1
        p(i)=p(1)
c       dzp=dzp+dz(i-1)
c       p(i)=p(1)+(dzp*(phi(1,1)-p(1))/dzbig(1))
      end do
      p(n2(1))=phi(1,1)
      dzp=0
      do i=n2(1)+1,n2(2)-1
        dzp=dzp+dz(i-1)
        p(i)=phi(1,1)+dzp*(phi(1,2)-phi(1,1))/dzbig(2)
      end do
      p(n2(2))=phi(1,2)
c     backward in time solution
```

```
      sumsq=0
      do it=1,klast
        trid(n2(2),4)=phi(it,2)
        do i=2,n2(2)-1
          trid(i,4)=-p(i)/dt
        end do
        call diag3(trid,n2(2),pp)
        do i=1,n2(2)
          p(i)=pp(i)
        end do
        pn(it,1)=p(n2(1))
c       error term = sum of squared difference between measured
c       and calculated head at bottom measurement point, with
c       excited pressure from measurement point above
c
        sumsq=sumsq+(p(n2(1))-phi(it,1))**2
      end do
      error=sumsq/klast
      write(6,*)' error = ', error
      return
      end
      real*8 function func1(diff)
c     function to be minimized for layers above bottom
      implicit real*8(a-h,o-z)
      common p,pp,n2,trid,d,db,dz,dzb,dt,phi,klast,dzbig,zwt,
     1 lay,dstore,pn,npmeas
      parameter (idim=200,npts=2500,nsta=9)
      real*8 trid(idim,4),phi(npts,nsta),dz(idim),dzb(idim),db(idim),
     1 p(idim),pp(idim),d(idim),dzbig(nsta),dstore(nsta),
     2 pn(npts,nsta)
      integer n2(nsta)
      dstore(lay)=diff
      call layer  (error)
      func1=error
      return
      end

      subroutine layer (error)
c     error from layers above bottom
      implicit real*8(a-h,o-z)
c     npts = maximum number of time steps in input data file
c     idim = maximum number of grid cells in column
c
      parameter (idim=200,npts=2500,nsta=9)
      common p,pp,n2,trid,d,db,dz,dzb,dt,phi,klast,dzbig,zwt,
     1 lay,dstore,pn,npmeas
      real*8 trid(idim,4),phi(npts,nsta),dz(idim),dzb(idim),db(idim),
     1 p(idim),pp(idim),d(idim),dzbig(nsta),dstore(nsta),
     2 pn(npts,nsta)
      integer n2(nsta)
c     set all tridiagonal nodes that stay fixed for run
c     make first node a no-flow, p(1)=p(2)
      trid(1,1)=0
      trid(1,2)=1
      trid(1,3)=-1
      trid(1,4)=0
c     set the node diffusion coefficients above bottom layer
      do  jz=n2(lay-1)+1,n2(lay)
        db(jz)=dstore(lay)
      end do
```

```
c       set the middle node coefficients that don't change
        do i=2,n2(lay)-1
          trid(i,1)=db(i)/(dz(i-1)*dzb(i))
          trid(i,2)=-(db(i+1)/dz(i)+db(i)/dz(i-1))/dzb(i)-1.0/dt
          trid(i,3)=db(i+1)/(dz(i)*dzb(i))
        end do
c       top node in interval
        trid(n2(lay),1)=0
        trid(n2(lay),2)=1
        trid(n2(lay),3)=0
c       set the initial pressures by linear interpolation
c       p(1)=phi(1,1)+dzbig(1)
        p(1)=phi(1,1)
c       dzp=0
        do i=2,n2(1)-1
          p(i)=p(1)
c         dzp=dzp+dz(i-1)
c         p(i)=p(1)+(dzp*(phi(1,1)-p(1))/dzbig(1))
        end do
        p(n2(1))=phi(1,1)
        dzp=0
        do i=n2(1)+1,n2(2)-1
          dzp=dzp+dz(i-1)
          p(i)=phi(1,1)+dzp*(phi(1,2)-phi(1,1))/dzbig(2)
        end do
        p(n2(2))=phi(1,2)
c       set pressures for layers above bottom
        do iz=1,lay-1
          i1=n2(iz)
          i2=n2(iz+1)
          p(i1)=phi(1,iz)
          p(i2)=phi(1,iz+1)
          dp=(p(i2)-p(i1))/(i2-i1)
          do i=i1+1,i2-1
            p(i)=p(i-1)+dp
          end do
        end do
c       backward in time solution
        sumsq=0
        do it=1,klast
          trid(n2(lay),4)=phi(it,lay)
          do i=2,n2(lay)-1
            trid(i,4)=-p(i)/dt
          end do
          call diag3(trid,n2(lay),pp)
          do i=1,n2(lay)
            p(i)=pp(i)
          end do
c         error term to minimize is sum of squared
c         errors from all measurement points
          do ilay=1,lay-1
            sumsq=sumsq+(p(n2(ilay))-phi(it,ilay))**2
          end do
          do il=1,npmeas-1
            pn(it,il)=p(n2(il))
          end do
        end do
        error=sumsq/klast
        write(6,*)' error = ', error
        return
```

```
      end

      subroutine p2head(phi,pcon,zw,klast,npmeas)
c     convert measured pressures to head, assuming that the
c     average pressure over the entire record is the benchmark
c
      implicit real*8(a-h,o-z)
      parameter (npts=2500,nsta=9)
      dimension phi(npts,nsta),zw(nsta)
      common/average/av(10)
c     normalize to each mean pressure over the entire record
      do m=1,npmeas
      sum=0
        do i=1,klast
          sum=sum+phi(i,m)
        end do
        av(m)=sum/klast
        do i=1,klast
          phi(i,m)=phi(i,m)-av(m)
        end do
      end do
      do l=1,npmeas
        do i=1,klast
c     pcon = conversion factor, psia to head, ft
          phi(i,l)=phi(i,l)*pcon
        end do
      end do
      return
      end

      subroutine head2p(phi,pcon,zw,klast,npmeas)
c     convert head back to pressure, using assumed averaged pressure
      common /average/ av(10)
      implicit real*8(a-h,o-z)
      parameter (npts=2500,nsta=9)
      dimension phi(npts,nsta),zw(nsta)
      do l=1,npmeas
        do i=1,klast
          phi(i,l)=phi(i,l)/pcon+av(l)
        end do
      end do
      return
      end

      subroutine diag3(a,n,x)
c     solve tridagonal matrix with Thomas algorithm
      implicit real*8(a-h,o-z)
      parameter (mc=200)
      real*8 a(mc,4),w(mc),b(mc),g(mc),x(mc)
      w(1)=a(1,2)
      g(1)=a(1,4)/w(1)
      do1 i=2,n
        im1=i-1
        b(im1)=a(im1,3)/w(im1)
        w(i)=a(i,2)-a(i,1)*b(im1)
        g(i)=(a(i,4)-a(i,1)*g(im1))/w(i)
    1 continue
      x(n)=g(n)
      np1=n+1
      do2 i=2,n
```

```
        j=np1-i
        x(j)=g(j)-b(j)*x(j+1)
 2    continue
 3    return
      end

      SUBROUTINE mnbrak(ax,bx,cx,fa,fb,fc,func)
c       bracket range with Golden rule search from Numerical Recipes(1992)
      implicit real*8(a-h,o-z)
      real*8 ax,bx,cx,fa,fb,fc,func,GOLD,GLIMIT,TINY
      EXTERNAL func
      PARAMETER (GOLD=1.618034, GLIMIT=100., TINY=1.e-20)
      real*8 dum,fu,q,r,u,ulim
      fa=func(ax)
      fb=func(bx)
      if(fb.gt.fa)then
        dum=ax
        ax=bx
        bx=dum
        dum=fb
        fb=fa
        fa=dum
      endif
      cx=bx+GOLD*(bx-ax)
      fc=func(cx)
1     if(fb.ge.fc)then
        r=(bx-ax)*(fb-fc)
        q=(bx-cx)*(fb-fa)
        u=bx-((bx-cx)*q-(bx-ax)*r)/(2.*sign(max(abs(q-r),TINY),q-r))
        ulim=bx+GLIMIT*(cx-bx)
        if((bx-u)*(u-cx).gt.0.)then
          fu=func(u)
          if(fu.lt.fc)then
            ax=bx
            fa=fb
            bx=u
            fb=fu
            return
          else if(fu.gt.fb)then
            cx=u
            fc=fu
            return
          endif
          u=cx+GOLD*(cx-bx)
          fu=func(u)
        else if((cx-u)*(u-ulim).gt.0.)then
          fu=func(u)
          if(fu.lt.fc)then
            bx=cx
            cx=u
            u=cx+GOLD*(cx-bx)
            fb=fc
            fc=fu
            fu=func(u)
          endif
        else if((u-ulim)*(ulim-cx).ge.0.)then
          u=ulim
          fu=func(u)
        else
          u=cx+GOLD*(cx-bx)
```

```
          fu=func(u)
        endif
        ax=bx
        bx=cx
        cx=u
        fa=fb
        fb=fc
        fc=fu
        goto 1
      endif
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software k#Q2$#1D[.

      real*8 FUNCTION brent(ax,bx,cx,f,tol,xmin)
c       Minimize function with Brent's algorithm from Numerical Recipes
      implicit real*8(a-h,o-z)
      INTEGER ITMAX
      real*8 ax,bx,cx,tol,xmin,f,CGOLD,ZEPS
      EXTERNAL f
      PARAMETER (ITMAX=100,CGOLD=.3819660,ZEPS=1.0e-10)
      INTEGER iter
      real*8 a,b,d,e,etemp,fu,fv,fw,fx,p,q,r,tol1,tol2,u,v,w,x,xm
      a=min(ax,cx)
      b=max(ax,cx)
      v=bx
      w=v
      x=v
      e=0.
      fx=f(x)
      fv=fx
      fw=fx
      do 11 iter=1,ITMAX
        xm=0.5*(a+b)
        tol1=tol*abs(x)+ZEPS
        tol2=2.*tol1
        if(abs(x-xm).le.(tol2-.5*(b-a))) goto 3
        if(abs(e).gt.tol1) then
          r=(x-w)*(fx-fv)
          q=(x-v)*(fx-fw)
          p=(x-v)*q-(x-w)*r
          q=2.*(q-r)
          if(q.gt.0.) p=-p
          q=abs(q)
          etemp=e
          e=d
          if(abs(p).ge.abs(.5*q*etemp).or.p.le.q*(a-x).or.p.ge.q*(b-x))
     *goto 1
          d=p/q
          u=x+d
          if(u-a.lt.tol2 .or. b-u.lt.tol2) d=sign(tol1,xm-x)
          goto 2
        endif
1       if(x.ge.xm) then
          e=a-x
        else
          e=b-x
        endif
        d=CGOLD*e
2       if(abs(d).ge.tol1) then
```

```
        u=x+d
      else
        u=x+sign(tol1,d)
      endif
      fu=f(u)
      if(fu.le.fx) then
        if(u.ge.x) then
          a=x
        else
          b=x
        endif
        v=w
        fv=fw
        w=x
        fw=fx
        x=u
        fx=fu
      else
        if(u.lt.x) then
          a=u
        else
          b=u
        endif
        if(fu.le.fw .or. w.eq.x) then
          v=w
          fv=fw
          w=u
          fw=fu
        else if(fu.le.fv .or. v.eq.x .or. v.eq.w) then
          v=u
          fv=fu
        endif
      endif
11    continue
      pause 'brent exceed maximum iterations'
3     xmin=x
      brent=fx
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software k#Q2$#1D[.
```