CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

Goodluck Ofoegbu
(210) 522 6641

March 31 1999

Continuation of project work described on page 1 of CNWRA Scientific Notebook Number 263 including modifications to the project reported on page 77 of the same notebook (Number 263).

[signature]

G. I. Ofoegbu

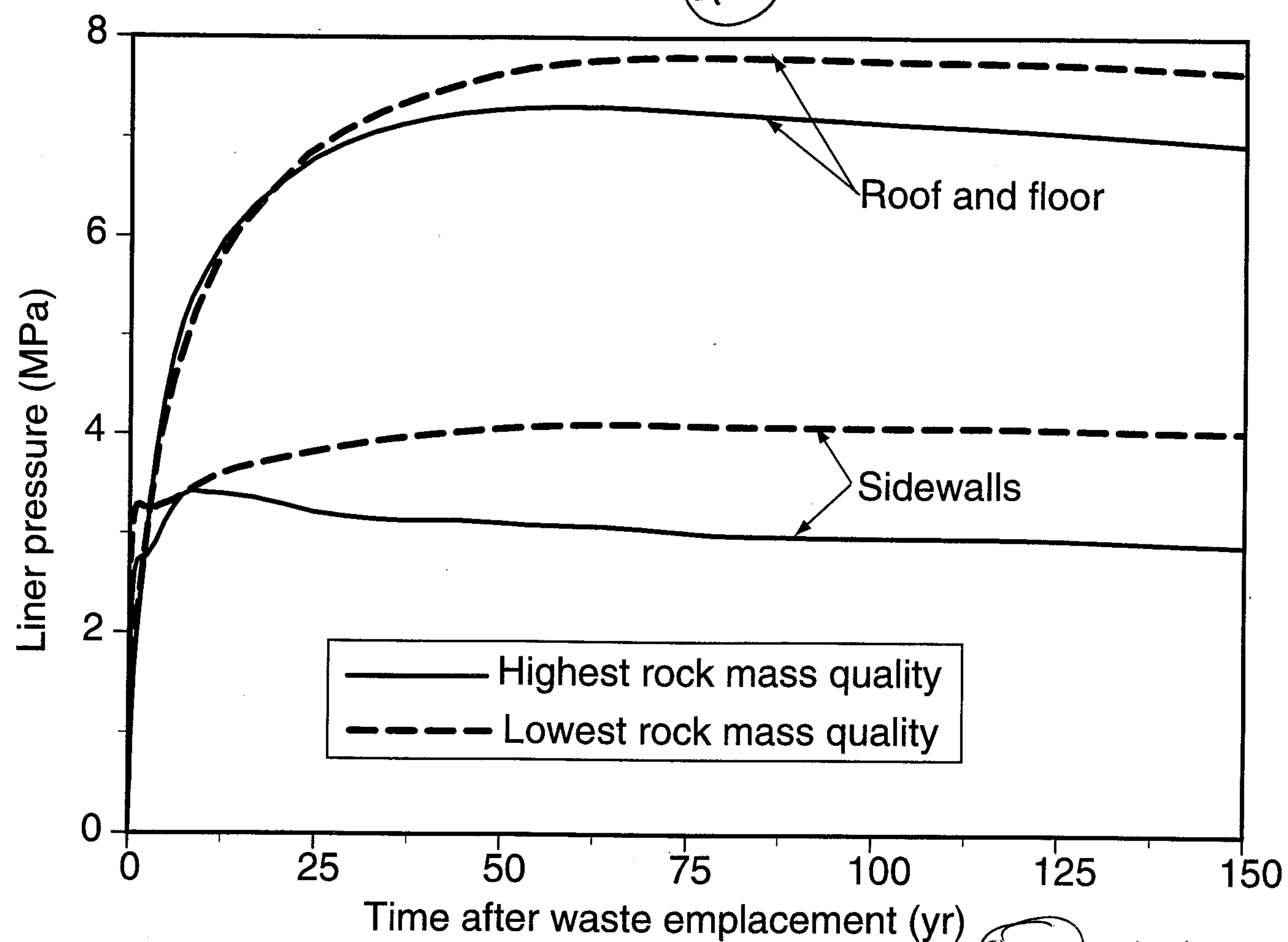
Table of Contents

Table of Contents Contd.

March 31 1999

Pages 4-7 entries by GWO 3/31/99

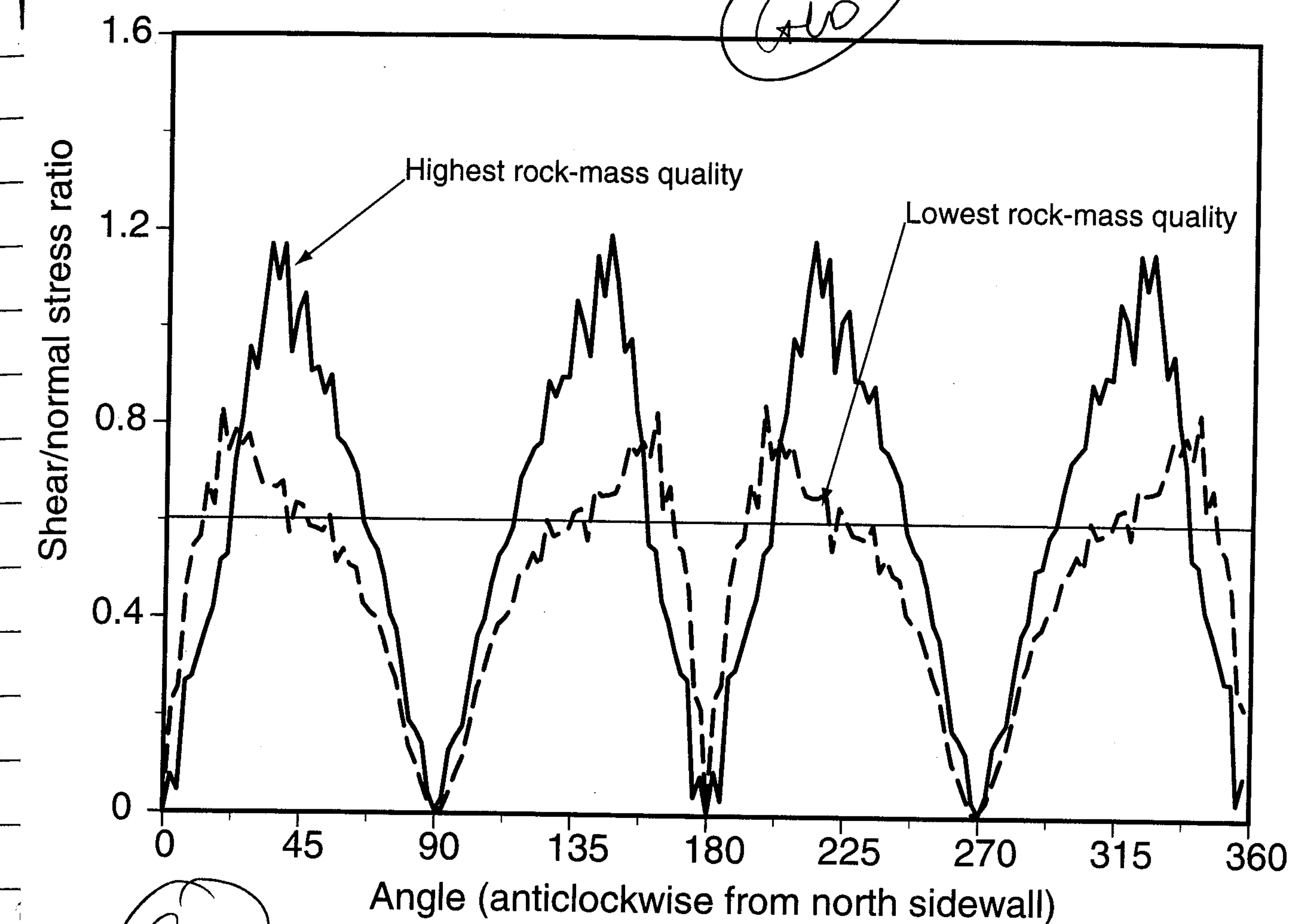
GWO



GWO 3/31/99

Histories of liner pressure at points on the roof, floor, and north and south sidewalls of the drifts, from model d1 (highest rock mass quality) and model d2 (lowest rock mass quality). The curves further illustrate the ~~noh~~ nonhydrostatic nature of the liner loads computed from the two models. The values of liner load from models d1 and d2 may have been influenced by the frictional properties applied to the liner-rock interface in both models. The liner was

GWO



GWO

modeled as fully bonded to rock. The ratio τ_L/P_L (where τ_L shear stress at liner-rock interface and P_L is liner pressure) at 150 yr is shown in the above figure. Values of τ_L/P_L may exceed the ~~maximum~~ GWO 3/31/99 allowable value for the concrete-rock interface, under which condition the calculated liner pressure may be in error. Additional analysis will now be performed with the frictional coefficient for the liner-rock interface set to 0.6 (equal to a friction angle of about 30°). The applicable friction angle on the concrete-rock interfaces is expected to be greater than 30°. So the value of 0.6 for the frictional coefficient ~~is expected to~~ and the case of fully bonded interface analysed earlier should cover the applicable range of values of frictional behavior for ~~an~~ concrete-rock interfaces.

Two models are setup as follows:

d11 :	Model d1 modified to include frictional liner-rock interface with frictional coefficient = 0.6
d21 :	Model d2 modified in the same way.

Only the mechanical analysis aspect is modified. ~~The input files for model d11 are: (only the mech.~~ (GO 3/31) The mechanical analysis input files are:

d11m.inp	Input file for d11 mechanical analysis.
d11mProps.def	Mechanical property definitions for model d11

d21m.inp	Input file for model d21 mechanical analysis
d21mProps.def	Mechanical properties definitions for model m21
fricContacts.def	Definitions of liner-rock contacts (non-bonded version)

In addition, these models use the following files listed on p. 83 of Notebook number 263:

Files	Used by
2.3 — 2.4	Both d11 and d21
2.6	✓
2.9	✓
2.11	✓
2.13	d11
2.15 — 2.17	d11
2.2	d11
2.19 — 2.20	d21
2.22 — 2.24	d21

Pages 4-7 entries by GO 3/31/99

Pages 7-10 entries by GO 4/7/99

April 7 1999

Models d12 and d22, which are similar to d11 and d21, respectively, except that the liner-rock interface are assigned a friction coefficient of 0.3. The input files associated with d12 and d22 are as follows:

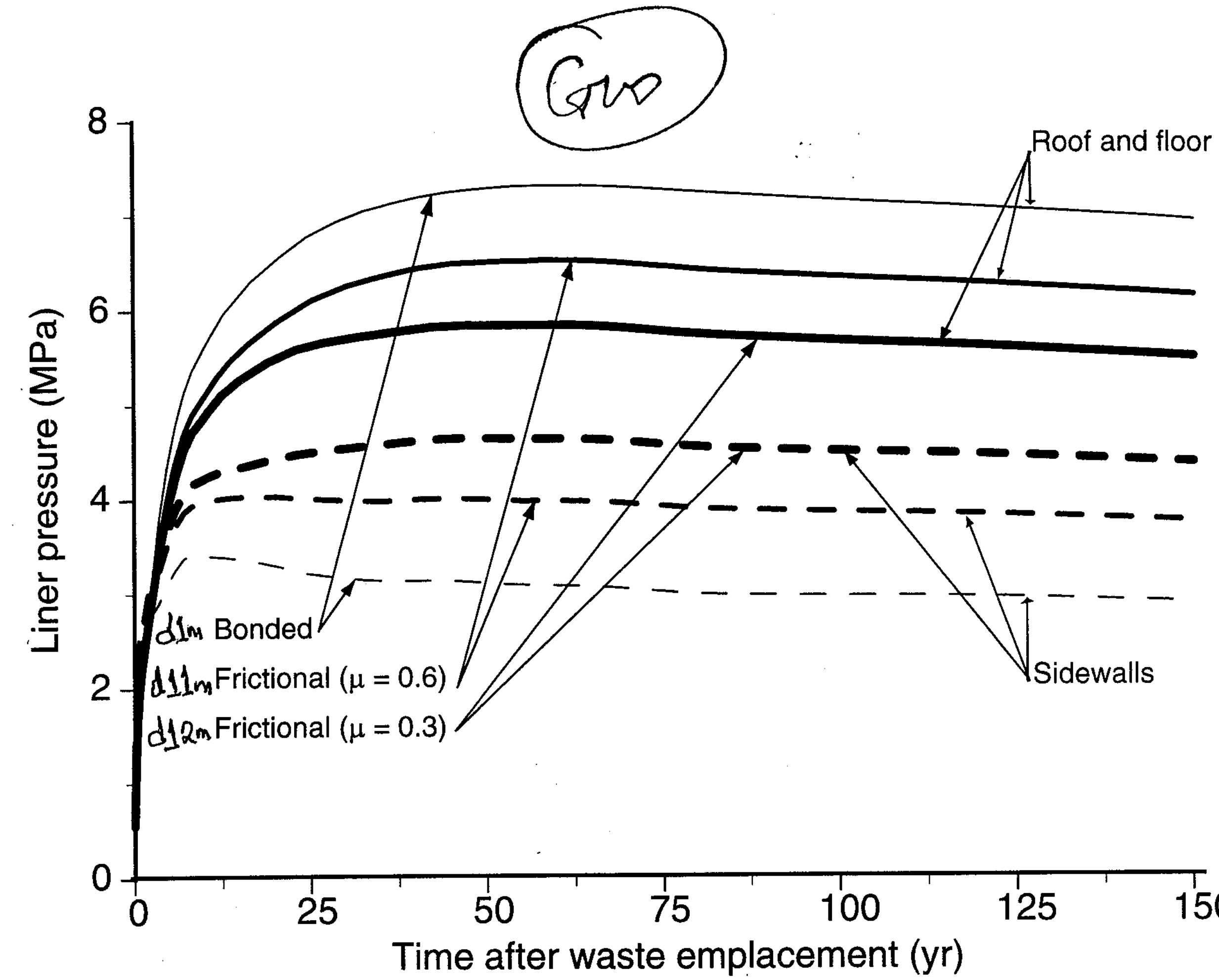
d12m.inp, d22m.inp	Input files for analysis
d12mProps.def, d22mProps.def	Mechanical properties

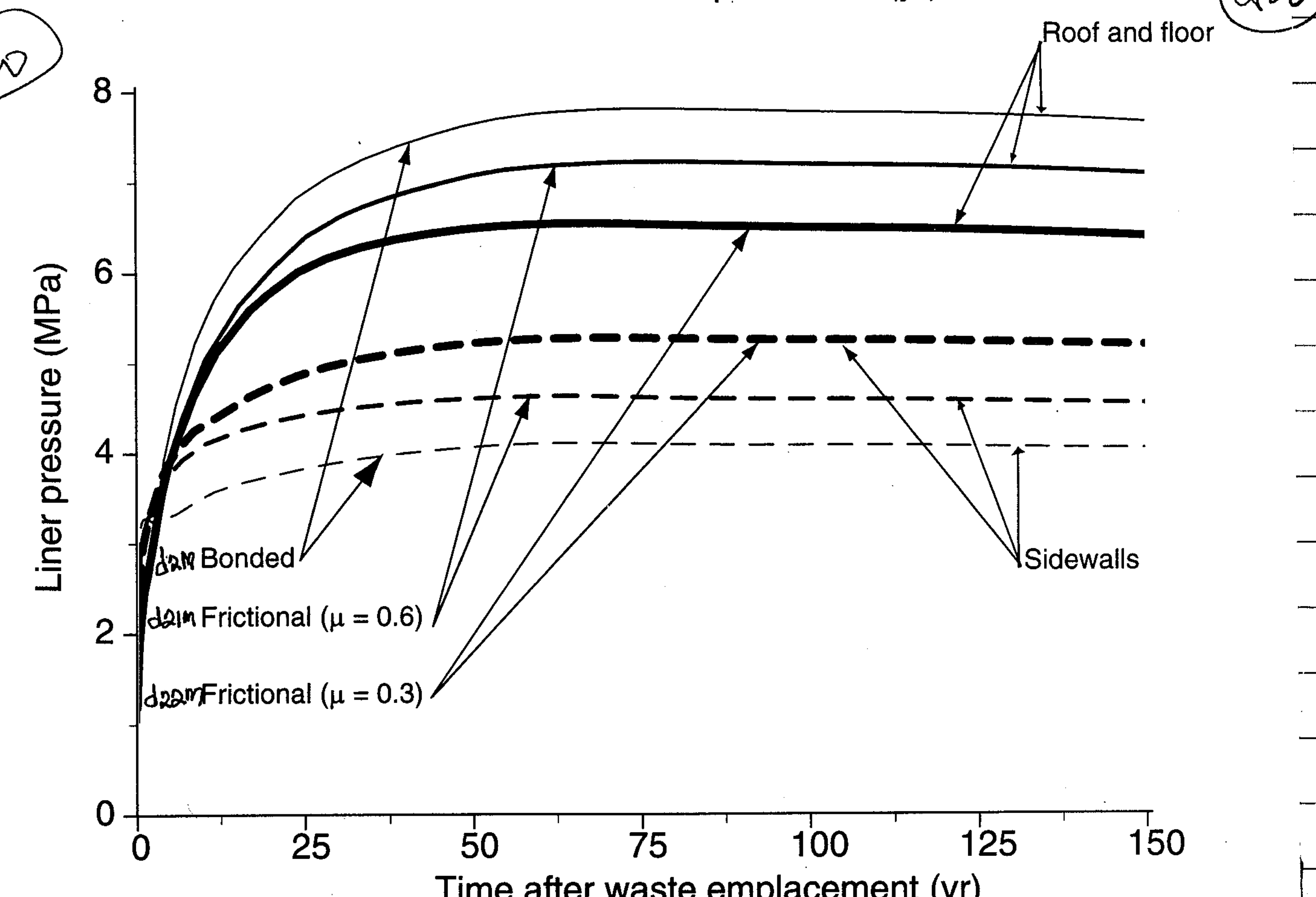
All other input files are the same as those listed for d11 and d21 models. The models used to investigate the effects of liner-rock interface properties on liner load are summarized as follows (μ stands for coefficient of friction at the liner-rock interface).

d1m	Highest-Q area;	Bonded liner
d2m	Lowest-Q area;	Bonded liner
d11m	Highest-Q area;	μ = 0.6
d21m	Lowest-Q area;	μ = 0.6
d12m	Highest-Q area;	μ = 0.3
d22m	Lowest-Q area;	μ = 0.3

The results are presented on p. 9 in terms of liner-load histories at ~~the~~ GVO points on the roof, floor, and north and south sidewalls of the openings. The top plots give the results from the highest-Q models (d1m, d11m, and d12m) and the bottom plots give the results from the lowest-Q models (d2m, d21m, and d22m). The results indicate the following:

(1) The liner load approaches a hydrostatic (equal-allround) distribution as the liner-rock interface approaches frictionless. Liner load decreases in the roof and floor areas and





increases in the sidewall areas as the liner-rock friction coefficient is decreased.

(2) For nonzero values of friction coefficient the liner load in the roof and floor areas exceeds the liner load in the sidewall areas by an amount that increases as the friction coefficient increases.

(3) Models that assume fully ~~bonded~~ liner ~~not~~ would give maximum values for the roof and floor area liner load and minimum values for the sidewall-area liner load.

Pages 7-10 entries by GW 4/7/99

April 8 1999

GW

Pages 10-17 entries by GW 4/8/99

Unsupported Openings

Models d13m and d23m are developed to examine the behavior of unsupported openings in the highest-Q area (d13m) and in the lowest-Q area (d23m). The two models are adopted from models d1m and d2m, respectively (which were described on p. 83-84 of Notebook #263), by assigning to the Liner zone the same properties as

are assigned to the Elastic-plastic zone (both zones are defined on p. 81 of Notebook #263).

The interface between liner and ~~drift~~ rock is ~~maintained~~ retained but modeled as fully bonded as in d1m and d2m. Since both the liner and surrounding rock are assigned the same properties and the liner is fully bonded to the rock, the ~~liner~~ opening is essentially a nonsupported 5-m diameter opening. (Recall from p. 80 of Notebook #263 that the liner has internal and external diameters of 5.0 and 5.4 m, respectively). GW 4/8/99

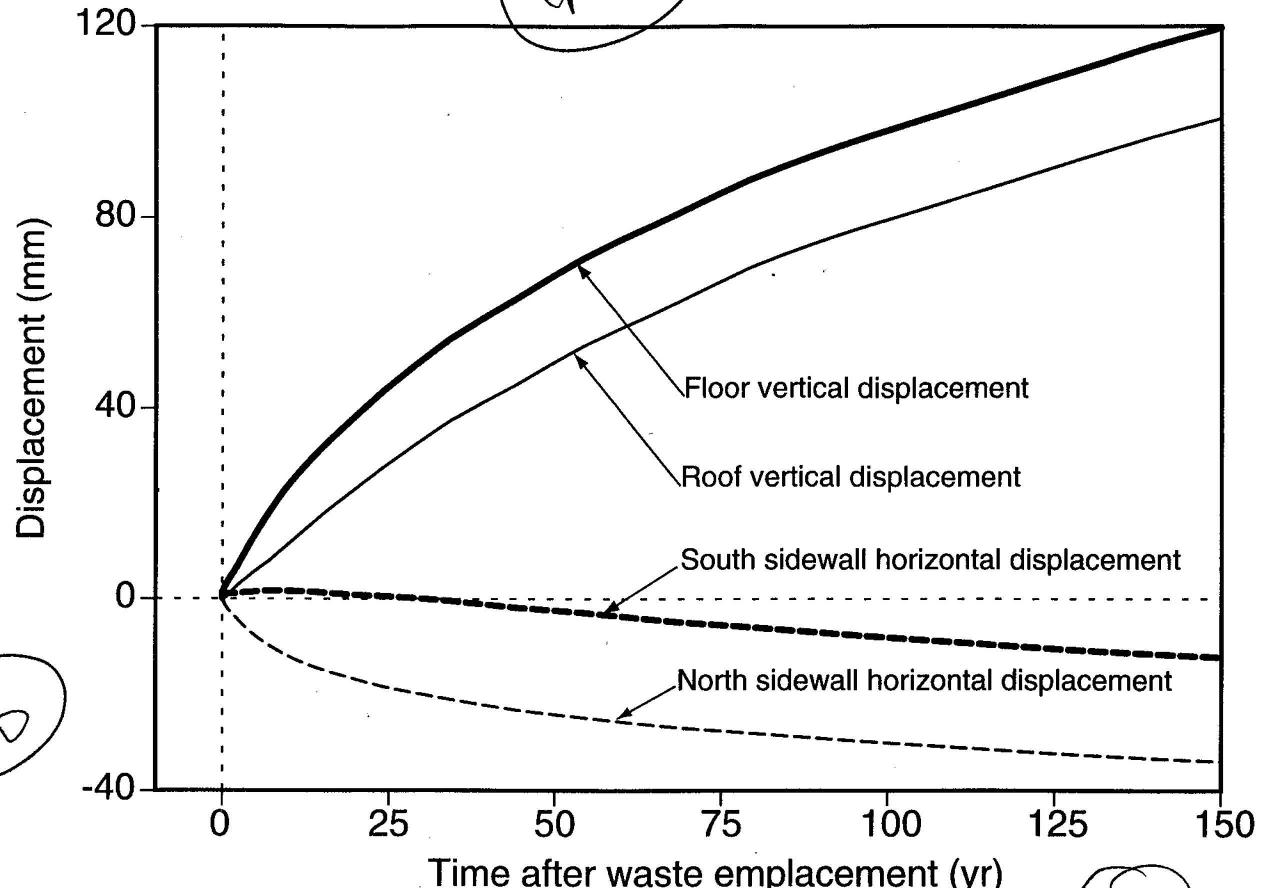
The input files are:

d13m.inp , d23m.inp	Main input files
d13mProps.def, d23mProps.def	Mechanical properties
contacts.def	Contact definition (same as for d1m & d2m)

along with the other files listed by number on p.7 of this notebook.

Model d13m has been executed ~~Succef~~ successfully to the desired simulation time of 150 yr. However, model d23m became numerically unstable after about a simulation time of 1 yr.

GWO



GWO GWO

This figure shows the displacement history at the roof, floor, and north and south sidewalls of the opening from model d13m (i.e., highest-Q area). Vertical displacements are positive upward and horizontal displacements are positive northwards. Recall that the simulation times are ~~a~~:

1 x 10^{-6} yr	Initial static equilibrium
2x10^{-6} yr	End of excavation and instantaneous waste emplacement
~~2x10^{-6} — 150 yr~~ GWO	
150 yr	End of simulation of thermal loading from waste empl.

GWO 4/8/99

Step in Analysis	Time at End of Step (yr)	Process simulated
1	1 x 10^{-6}	Initial static equilibrium under gravitational loading, initial stress and boundary restraint. No drift.
2	2 x 10^{-6}	Drift excavated
3	150	Temperature history ~~due to~~ from simulated repository thermal load applied in this step. GWO 4/8/99

(see p. 25 of Notebook #263 for more information about the analysis sequence). The above table indicates that the displacement ~~at~~ GWO 4/8/99 caused by mechanical excavation corresponds to the displacement at time of 2x10^{-6} yr. All subsequent displacement is caused by thermal loading. The values of displacement at 2x10^{-6} yr are

-0.0012 mm (horizontal) at North sidewall
0.18 mm (horizontal) at South sidewall
-0.36 mm (vertical) at Roof
1.4 mm (vertical at Floor

These displacements are negligible (essentially equal to zero) relative to the thermally induced displacements.

The distributions of Γ^N (see p. 71 of Notebook #263 for the definition of Γ^N) from models d1m and d13m are compared on p. 15. The three plots are, from left to right:

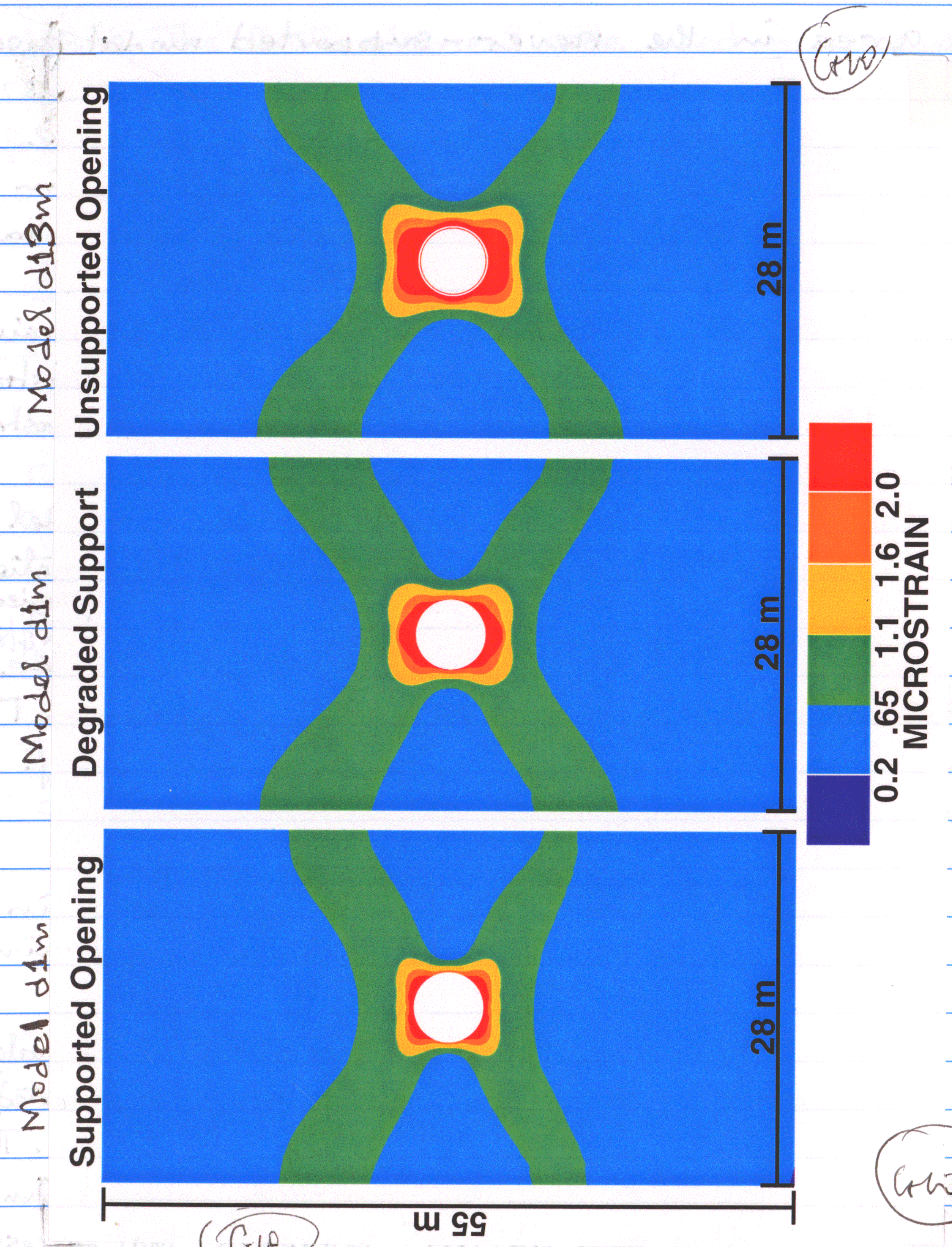
(1) Model d1m with concrete liner;
(2) Model d1m after concrete liner was removed; and
(3) Model d13m (unsupported opening at all times)

The 0.65 – 1.1 microstrain zones are essentially the same in all three plots, indicating that the occurrence of this relatively high Γ^N zone away from the opening is not influenced by the concrete lining. Note, however, that the lining has lower stiffness than the surrounding rock in the d1m model. The stiffness of the support system relative to that of the rock may be important. This issue will be investigated further.

GW 4/8/99

The ~~extent of~~ high-Γ^N zone close to the opening ($\Gamma^N > 1.1$ microstrain zone) has the same shape in all three plots but covers the largest

GW



GW

GW

area in the never-supported model case (d13m). The maximum Γ^N value is also ~~&~~ greatest in the d13 model. The maximum value of Γ^N (greatest value in the $\Gamma^N > 2.0$ microstrain zone) is as follows:

d1m with liner:	3.78 microstrain
d1m after liner is removed:	6.01 microstrain
d13m (never supported):	13.5 microstrain

The large Γ^N value from the d13m model suggests that excessive plastic deformation may be the reason the d23m model ~~was~~ experienced GW 4/8/99 numerical instability and could not ~~be~~ executed as a result. The maximum Γ^N values from the d2m model (see p. 91 of Notebook 263 for the plots) are:

d2m with liner:	4.5 microstrain
d2m after liner is removed:	24.2 microstrain

Model d2m is the with-liner-support equivalent of model d23m (intended to be never supported). Both models are for the lowest-Q area. The relationship between maximum Γ^N values in models d1m and d13m suggests that excessive plastic deformation ~~woul~~ would occur in model

d23m. The finite element code used in the analysis could not handle such large strain and the model failed to execute as a result.

Page 10-17 entries by GW 4/8/99

April 19 1999

Page 17-19 entries by GW 4/19/99

Two models are setup to investigate the effects of liner stiffness on the calculated Γ^N distributions. The value of Young's modulus for concrete liner is set to 2×10^5 MPa, which is high compared to the 2.76×10^4 MPa used in all previous drift-scale models that include concrete-liner support (p. 82 of Notebook #263). The input files for the current two models are (in addition to other files listed by number on p. 7):

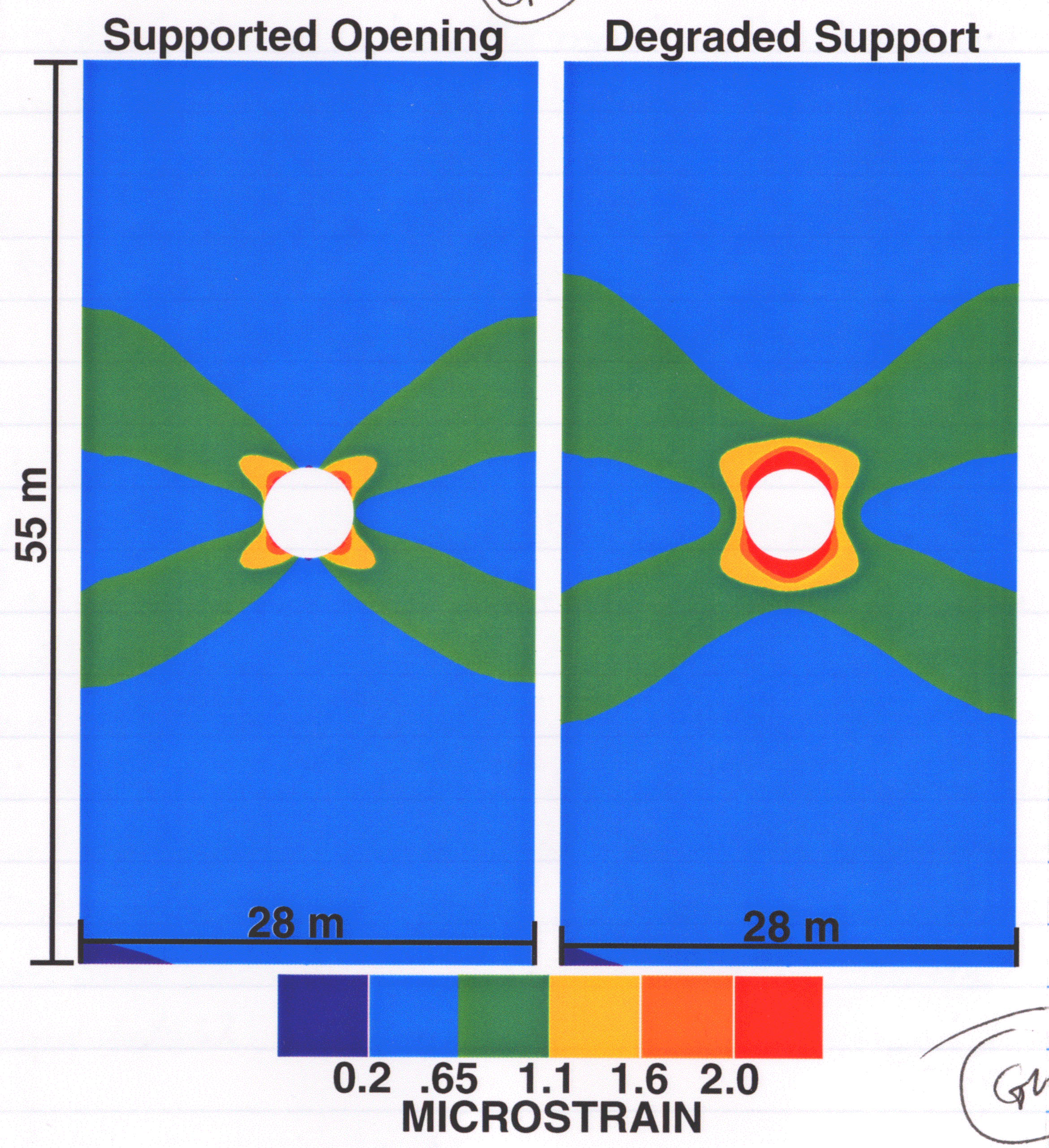
d14m.inp	Main input file
d14mProps.def	Mechanical properties

for the highest-Q area, and

d24m.inp	Main input file
d24mProps.def	Mechanical properties

for the lowest-Q area.

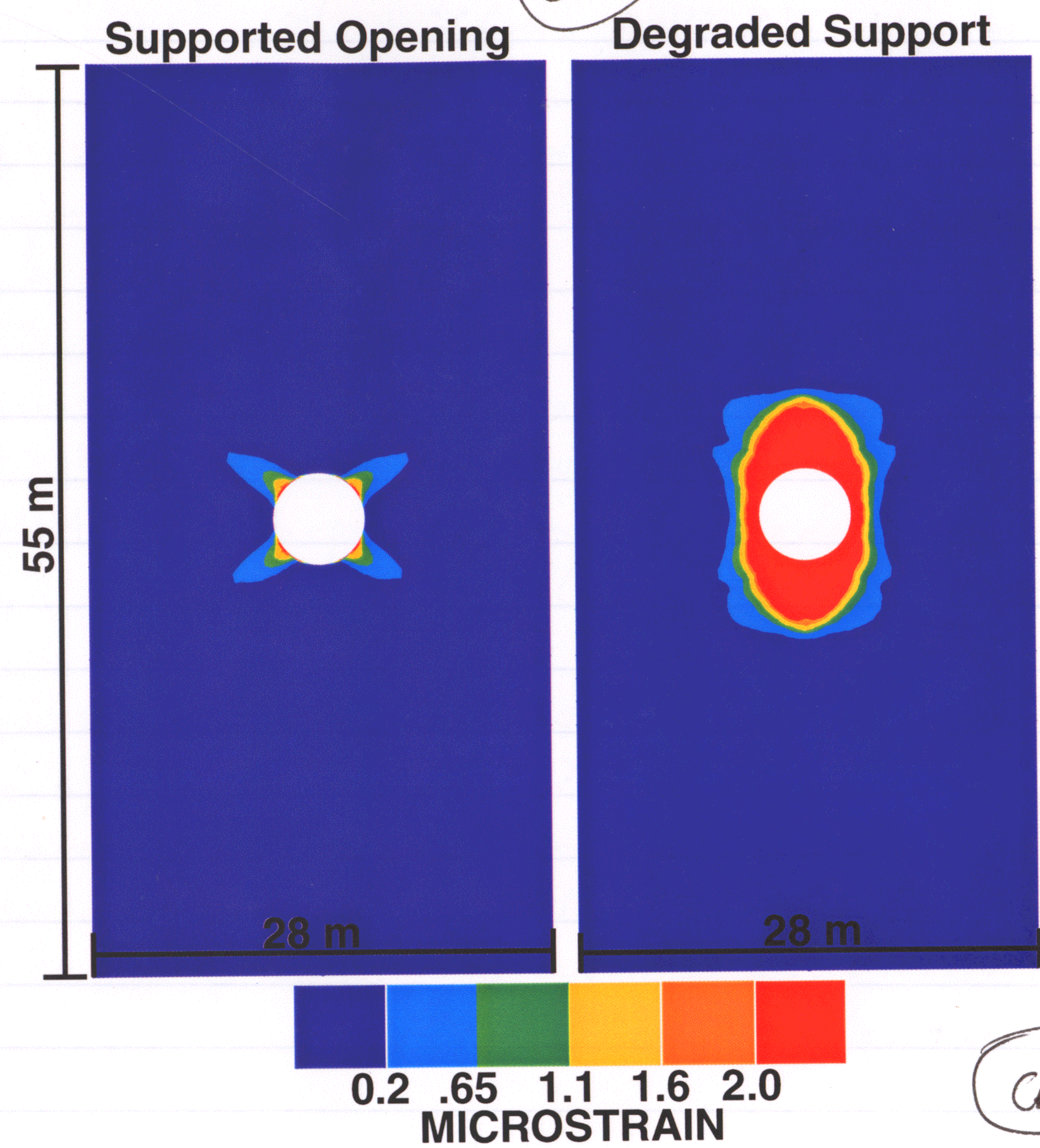
GWO



GWO

Distributions of inelastic strain (Γ^N) from models d14 (this page) and d24 (next page). These figures compare well with the figures on p. 90-91 of Notebook #263. The results indicate that the Γ^N distribution close to the opening is significantly affected by the stiffness of the support system, both in the highest-Q models (#d14) and in the

GWO



GWO

lowest-Q models (d24).

Pages 17-19 entries by GWO 4/19/99

April 20 1999 Pages 20-27 entries by GW 4/20/99

Three areas of investigation were identified following an external review of the repository-scale model (see p. 1ff of Notebook 263):

(1) Develop the explanation for the occurrence of high-Γ^N zones in the middle of inter-drift pillars (see p. 71 of Notebook 263, for example).

(2) Conduct analyses with site-specific (Yucca Mt) E-vs-Q relationships (p. 22 of Notebook 263) to emphasize the uncertainties in these relationships and how such uncertainties would affect thermal-mechanical calculations.

(3) Conduct analyses in which ~~A~~ the time-variation of Q is incorporated in ~~such a~~ a way that permits ~~way as to~~ both rock-mass stiffness and strength parameters to vary simultaneously with time. Simultaneous temporal variation of stiffness and strength parameters may eliminate the apparent inconsistency between the stiffness and strength parameters currently applied in the models. GW 4/20/99

A mechanically homogeneous, linear-elastic model (ug10) was developed to see if the stress distributions from such a model would explain the distributions of inelastic strain (Γ^N) from earlier models.

The input files for the model are

ug10.inp	Main input file
allNodes.def	Node definitions (same as the file with the same name on p. ~~32~~ 27 of Notebook 263). GW 4/20/99
ug10Elem.def	Element connectivity and and element and material properties.
iniTemp.def	Initial temperature (also on p. 27 of Notebook 263)

The model was adopted from the repository-scale model described on pp. 4-15 of Notebook 263 and is similar to the constant-Q models (Cases 2 and 3 on p. 60 of Notebook 263). The mechanical-property values are:

Young's modulus = 36.5 GPa

Poisson's ratio = 0.21

Thermal expansivity varies with temperature following table on p.15 of Notebook 263. The model does not require strength parameters.

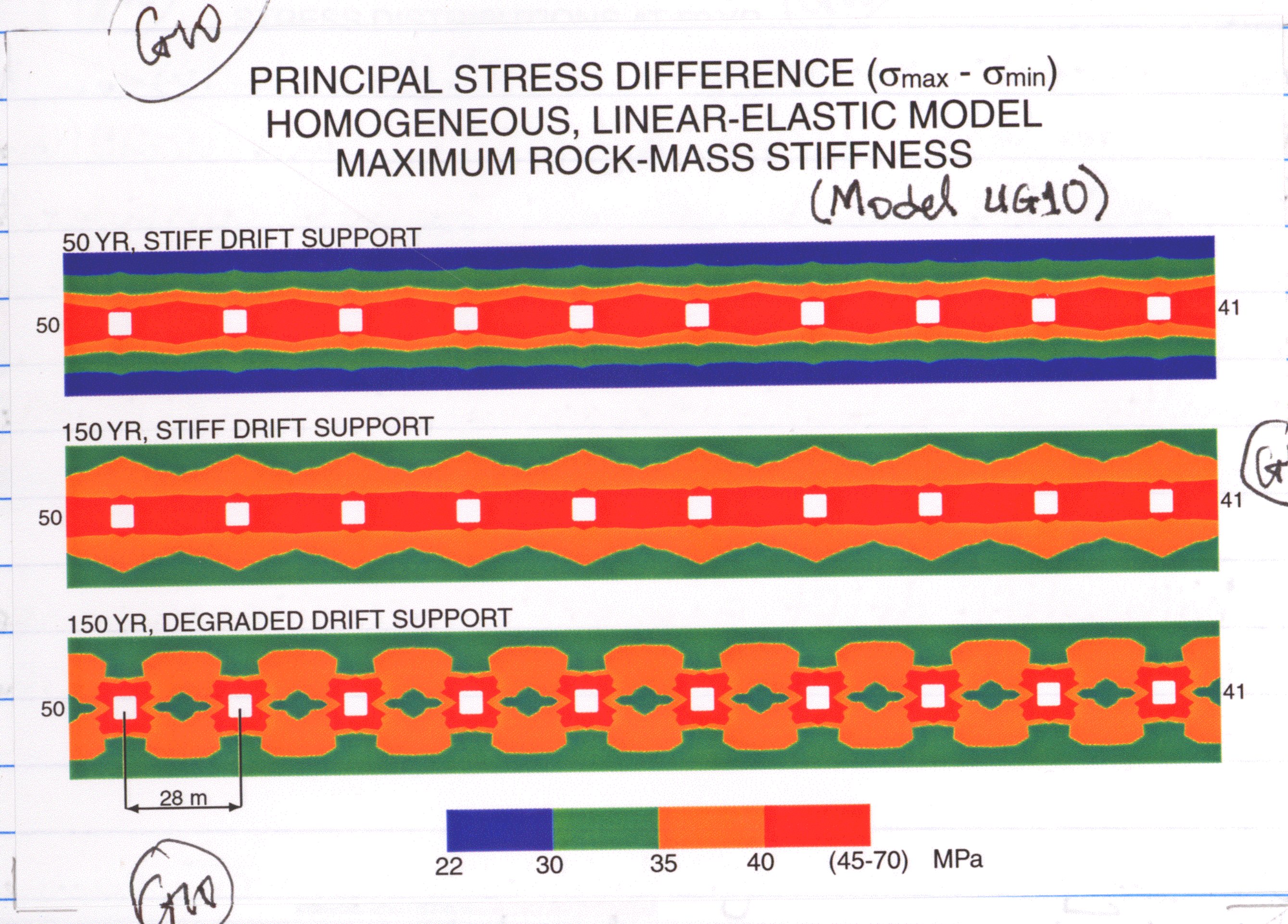
A second mechanically homogeneous, and linear elastic model (ug20) was also developed to provide information on the effects of Young's modulus (which may allow issue #2 on p.20 to be addressed through models ug10 and ug20). The model is the same as model ug10 except that the value of Young's modulus ~~is~~ is set to 8.02 GPa in ug20. The input files for ug20 that are different from those for ug10 are

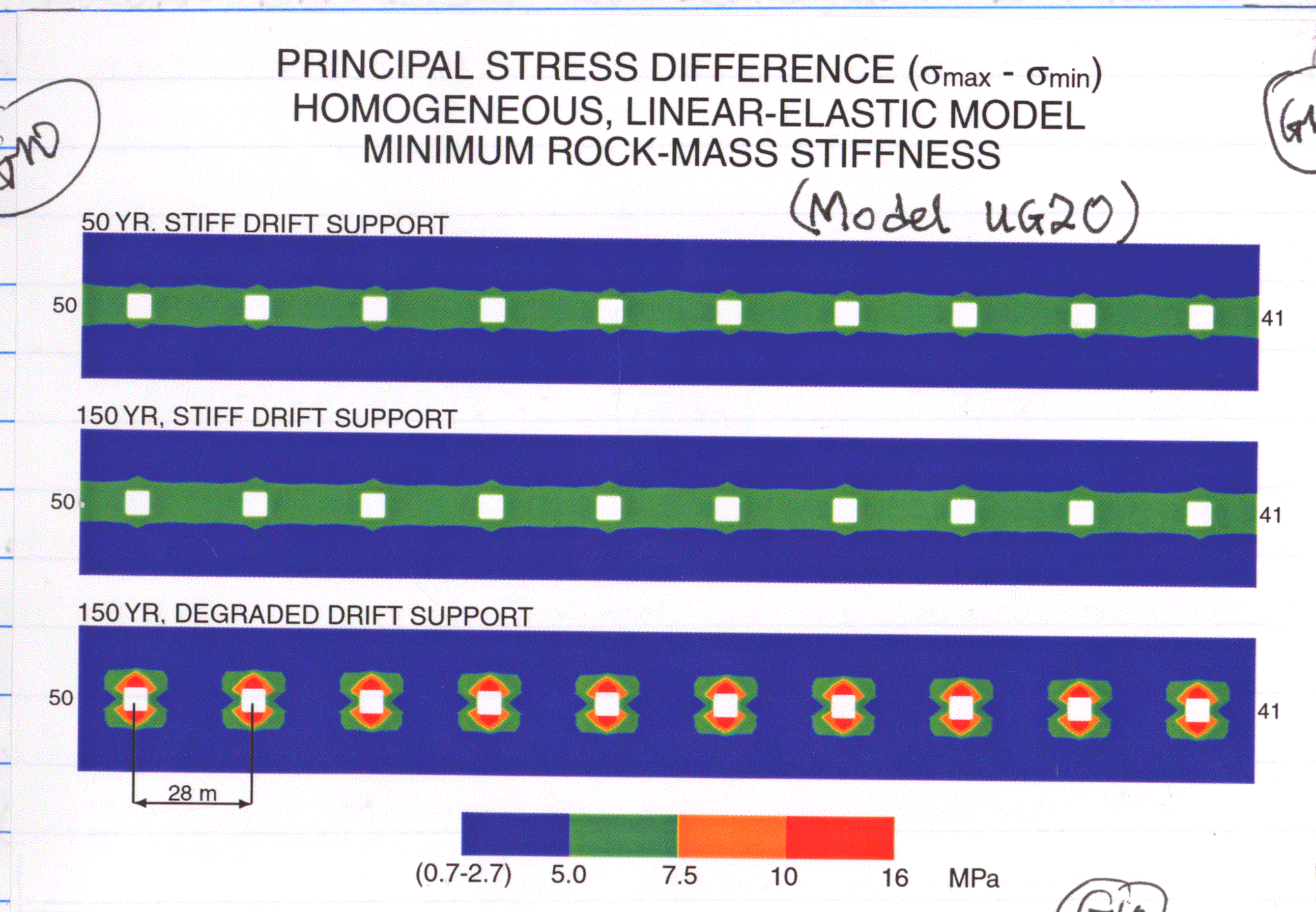
ug20.inp	Main input file
ug20Elem.def	Element connectivity and element and material properties.

The results from models ug10 and ug20 are presented on p.23 in terms of the principal stress difference $\sigma_{max} - \sigma_{min}$ (referred to hereafter as P_d). The occurrence of inelastic response depends on ~~whether~~ (GNO 4/8/99) the value of P_d at a point relative ~~its~~ smaller to the strength S at the given point.

$P_d < S$ indicates elastic conditions

$P_d > S$ indicates occurrence of inelastic response





The correct values for s differ between the two models ug10 (highest Q) and ug20 (lowest Q) but the models do not include consideration of inelastic response. So the results on p. 23 do not include the effects of s. But these results can be interpreted to indicate where inelastic response is ~~most~~ more likely.

GWO 4/18/99

(1) With stiff drift support, inelastic response is more likely in the pillars than in the roof and floor areas.

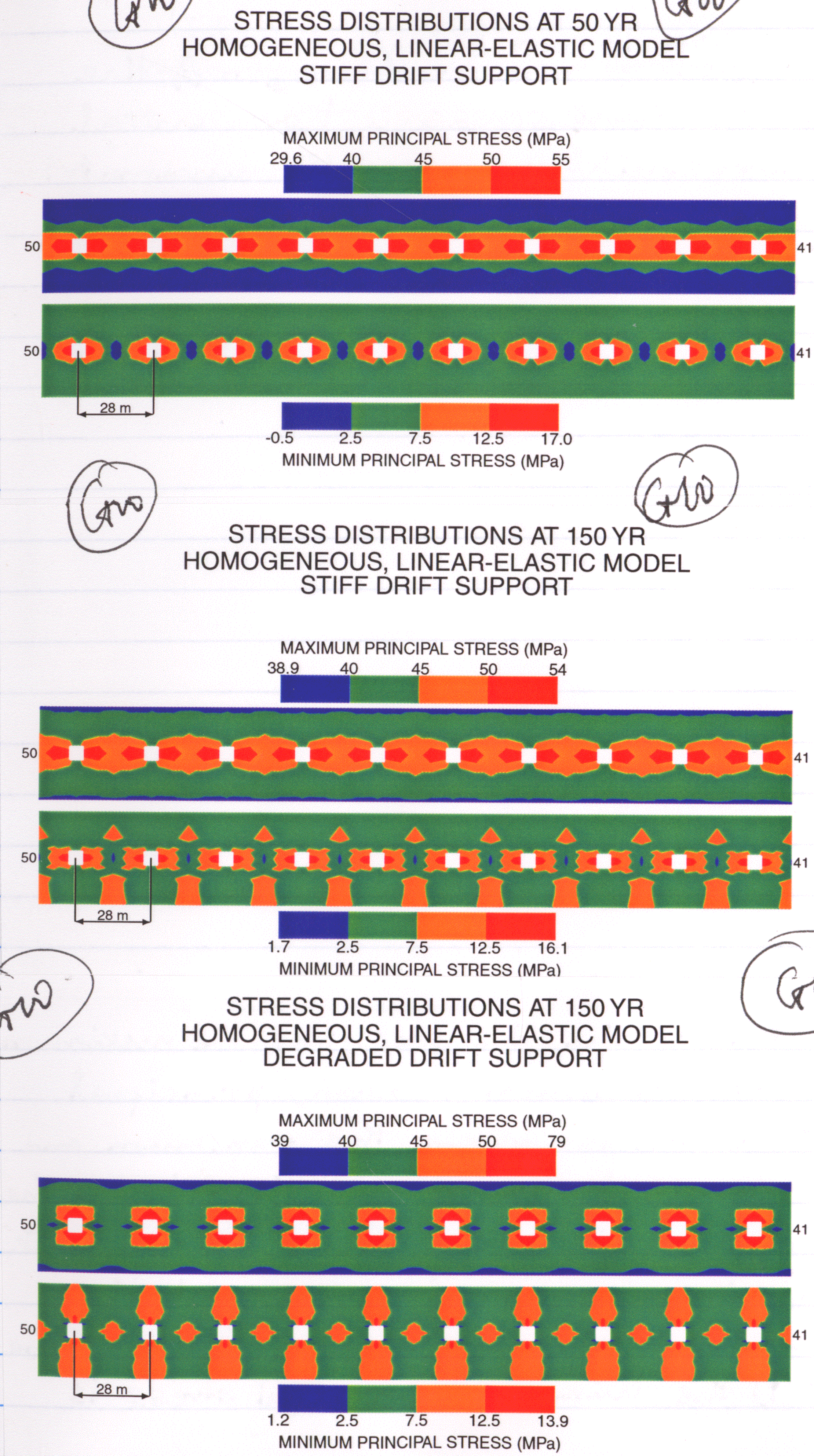
(2) Degradation of drift support causes inelastic response to become more likely in the roof and floor areas than in the pillars

(3) The effect of support degradation is more severe in the lowest-Q case (ug20) than in the highest-Q case (ug10).

AHC 8-30-2000 AHC 8-30-2000

(4) P_d values in the highest-Q case are much higher than in the lowest-Q case.

The reason for the P_d patterns on p. 23 is illustrated on p. 25 using principal-stress (σ_{min} and σ_{max}) distributions from model ug10.



Principal stress distributions from model ug10.

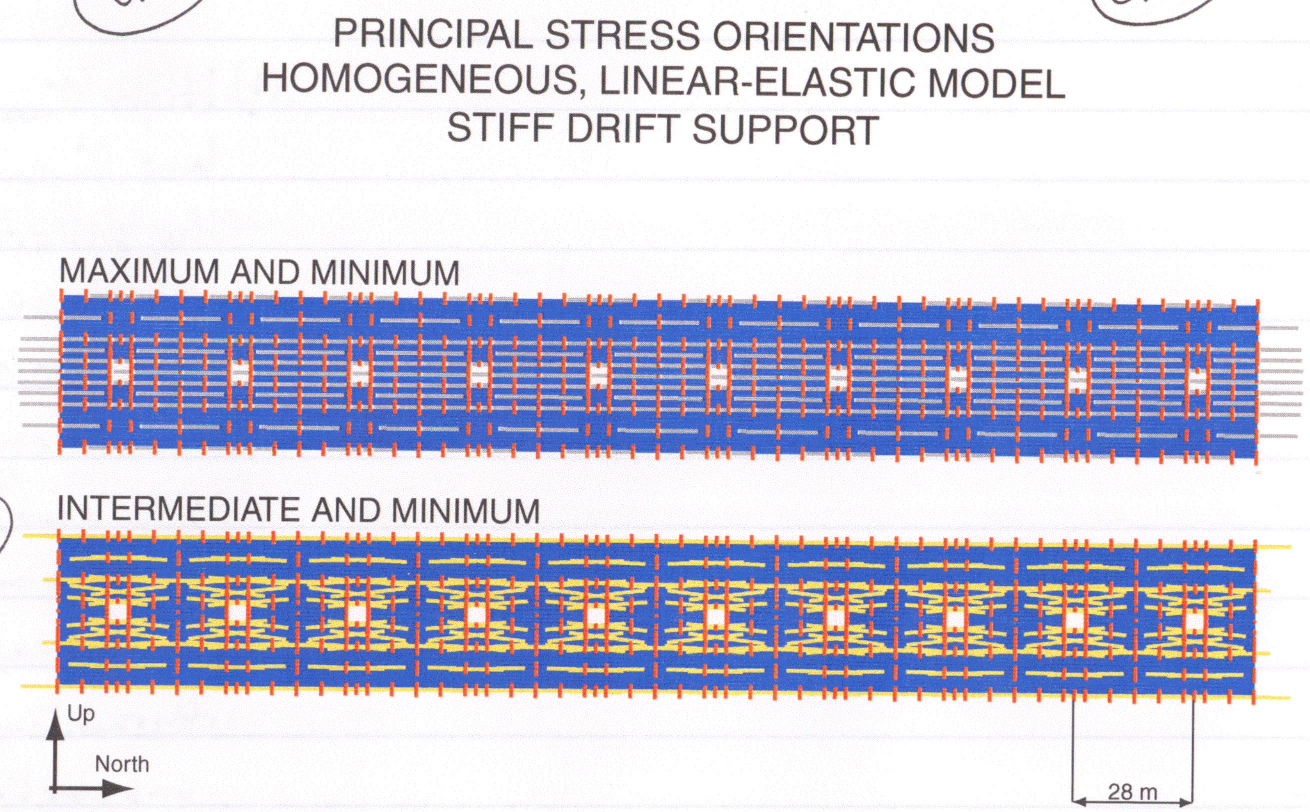
(1) High σ_{max} and low σ_{min} in the pillars in the presence of stiff support

(2) Relatively high σ_{max} and σ_{min} in the roof and floor areas following degradation of support.

(3) Relatively low σ_{max} and high σ_{min} in the pillars following degradation of support.

In all cases (see figure below ~~not shown in the figures~~ GW 4/20/99), the maximum principal stress is horizontal or near-horizontal whereas the minimum principal stress is vertical or near-vertical.

GW GW



GW

These plots show the orientations of in-plane principal stresses. The top plot gives the maximum ~~an~~ (grey) and minimum (reddish) principal stresses at locations where the maximum and minimum principal stresses are in-plane. The bottom figure gives the ~~maximum~~ GW 4/8/99 intermediate (yellow) and minimum principal stresses at the locations where they are in-plane. The figure shows that (i) the minimum principal stress is

in-plane and vertical everywhere, (ii) the maximum principal stress is in-plane in the pillars and out-of-plane in the roof and floor areas, and (iii) the intermediate principal stress is out-of-plane in the pillars and in-plane in the roof and floor areas.

The principal-stress orientations illustrated in the figure suggest the following failure modes: (i) ~~A~~ Failure in the roof and floor areas would be controlled by slip on gently dipping (~30° dip) fractures that strike normal to the drift orientation. (ii) In the pillars, failure would be controlled by slip on gently dipping (~30° dip) fractures that strike parallel to the drift orientation.

4/20/99 GW

Pages 20-27 entries by GW 4/20/99

November 02 1999 Pages 28-30 Entries by GVO 11/02/99

Work was initiated to develop information for a paper that will be presented at the 4th North American Rock Mechanics Symposium. The symposium is scheduled for 7/31 - 8/03 2000 at Seattle, Washington. The abstract for the proposed paper is shown on this page.

Estimating Long-Term Thermal-Mechanical Effects on Permeability at Yucca Mountain

Goodluck I Ofoegbu and Rui Chen.
Center for Nuclear Waste Regulatory Analyses, Southwest Research Institute, 6220 Culebra Road, San Antonio, TX 78238-5166. Phone: (210) 522-6641; Fax: (210) 522-6081; e-mail: gofoegbu@swri.org.

Information potentially subject to copyright protection was redacted from this location. The redacted material is an abstract for a paper that was published.

GVO

Changes in Fracture Porosity and Permeability from Mechanical Deformation

Changes in fracture porosity and permeability can result from both elastic and inelastic deformations. The following discussion concerns changes due to inelastic deformations. Elastic changes will be considered later. These notes replace the notes on pages 54–55 of Scientific Notebook #263 (though only the material on page 55 is actually modified).

Relationship Between Fracture Porosity Change and Inelastic Volumetric Strain

Definitions

v_f = fracture volume
v_t = total volume (sum of fracture volume and solid-rock volume)
ϕ_f = fracture porosity = v_f/v_t

Consider a small rock body subjected to a complete cycle of loading and unloading such that all recoverable deformations caused by the loading is recovered during unloading. Let the rock body undergo inelastic (i.e., non-recoverable) deformation as a result of the loading and unloading cycle, such that its volume changes from v_{to} at the beginning of the cycle to v_{t1} at the end. The change in volume is accounted for entirely by a change in fracture volume from v_{fo} at the beginning of the cycle to v_{f1} at the end.

The inelastic volumetric strain ε^N is given by

$$\varepsilon^N = \frac{v_{t1} - v_{to}}{v_{to}} = \frac{v_{f1} - v_{fo}}{v_{to}} = \frac{\Delta v_f}{v_{to}} \qquad (1)$$

where $\Delta v_f = v_{f1} - v_{fo}$. The initial fracture porosity ϕ_{fo} is

$$\phi_{fo} = \frac{v_{fo}}{v_{to}} \qquad (2)$$

and the final fracture porosity ϕ_{f1} is given by

$$\phi_{f1} = \frac{v_{f1}}{v_{t1}} = \frac{v_{fo} + \Delta v_f}{v_{to} + \Delta v_f} \qquad (3)$$

which, dividing both numerator and denominator by v_{to} and using the result obtained earlier for ε^N and ϕ_{fo}, gives

$$\phi_{f1} = \frac{\phi_{fo} + \varepsilon^N}{1 + \varepsilon^N} \qquad (4)$$

Because $1 + \varepsilon^N \approx 1$, the foregoing equation implies

$$\phi_{f1} = \phi_{fo} + \varepsilon^N \qquad (5)$$

which, using $\Delta\phi_f = \phi_{f1} - \phi_{fo}$, gives

$$\Delta\phi_f = \varepsilon^N \qquad (6)$$

That is, the change in fracture porosity resulting from inelastic deformation is equal to the inelastic volumetric strain.

Fracture Porosity and Fracture Aperture

Consider a hypothetical rock body with a fracture spacing s that has the same value in three perpendicular directions (i.e., the principal directions). Then the linear fracture density (number of fractures per unit length) in any principal direction is $1/s$ and the volumetric fracture density is $3/s$. It is further assumed that the fracture aperture b is the same for every fracture. The fracture aperture changes from b_o to $b_o + \Delta b$ as a result of inelastic deformation during a loading cycle such as was defined previously. The corresponding change in fracture porosity is from ϕ_{fo} to $\phi_{fo} + \Delta\phi_f$, where

$$\phi_{fo} = \frac{3b_o}{s} \quad \text{and} \quad \Delta\phi_f = \frac{3\Delta b}{s} \qquad (7)$$

Fracture Permeability, Fracture Porosity, and Inelastic Volumetric Strain

The fracture permeability κ_f of the rock body is related to the fracture aperture through the equation (cf. Elsworth and Mase, 1993: Comprehensive Rock Engineering edited by J.A. Hudson, volume 1:201–226)

$$\kappa_f = \frac{b^3}{\cancel{12s}\ 6s} \qquad (8)$$

GVO 11/2/

Therefore, the fracture permeabilities κ_{fo} and κ_{f1}, at the beginning and end, respectively, of the loading cycle are:

$$\kappa_{fo} = \frac{b_o^3}{\cancel{12s}\ 6s} \quad \text{and} \quad \kappa_{f1} = \frac{(b_o + \Delta b)^3}{\cancel{12s}\ 6s} \qquad (9)$$

GVO 11/2/99

Using these relationships, the fracture permeability at the end of a given deformation sequence (such as the hypothetical load-unload cycle), considering only the effects of inelastic deformation, can be related to the initial fracture permeability and porosity as follows:

$$\kappa_{f1} = R_\kappa \kappa_{fo} \quad \text{where} \quad R_\kappa = \left(1 + \frac{\Delta b}{b_o}\right)^3 = \left(1 + \frac{\Delta\phi_f}{\phi_{fo}}\right)^3 = \left(1 + \frac{\varepsilon^N}{\phi_{fo}}\right)^3 \qquad (10)$$

$$R_\phi = \frac{\phi_{fo} + \Delta\phi_f}{\phi_{fo}} = (R_\kappa)^{1/3}$$

GVO

First, the procedure for calculating fracture-permeability change from a continuum elastoplastic model, which was documented on pp. 54-55 of Notebook #263, was revised as is documented above. In the earlier model it was assumed that all the fractures in a rock element can be consolidated into a single fracture. This assumption has been replaced with a less-restrictive assumption that the fractures consist of three orthogonal sets with a spacing of s in each of the three principal directions.

The revised ABAQUS user subroutine UVARM that ~~implicit~~ implements the procedure on p. 29 is in file KFrac02.f, which is reproduced on this page (Page 58 of Notebook #263 shows the previous ~~initial~~ version of the subroutine). GWO 11/2/99

```
D:\ThermMech\KFrac02.f

      SUBROUTINE UVARM(UVAR,DIRECT,T,TIME,DTIME,CMNAME,ORNAME,
     1           NUVARM,NOEL,NPT,LAYER,KSPT,KSTEP,KINC,NDI,NSHR)
C
      INCLUDE 'ABA_PARAM.INC'
      CHARACTER*8 CMNAME,ORNAME,FLGRAY(15)
      DIMENSION UVAR(NUVARM),DIRECT(3,3),T(3,3),TIME(2)
      DIMENSION ARRAY(15),JARRAY(15)
C
C     Code KFRAC02 (Version 2 of Code KCHANGE)
C
C     Computes changes in fracture permeability
C     following procedure documented in CNWRA Scientific Notebook
C     #321, p. 29.
C
C     Values of inelastic strain required for the calculation are
C     obtained through ABAQUS user-interface subroutine GETVRM
C
C     Externally supplied input parameter:
C
C     PHIOF       Value of fracture porosity at the strain-free state;
C                 fracture porosity is defined as
C                 total fracture volume per unit bulk volume
C
C     The calculated permeability ratio is stored in vector UVAR
C     at location UVAR(1)
C
C     Location in UVAR               Stored Variable
C     ----------------               ---------------
C
C          1                    Ratio (Rk) of current permeability to the initial
C                               (strain-free state) permeability
C
      PHIOF = 1.0D-4
C
C     Obtain current values of inelastic (plastic) strain components
C
      JRCD = 0
      CALL GETVRM('PE',ARRAY,JARRAY,FLGRAY,JRCD)
C     IF (JRCD .NE. 0) THEN
C       WRITE(6,1000) NOEL,NPT,TIME(2)
C       RETURN
C     END IF
C
      DPHIF = ARRAY(1) + ARRAY(2) + ARRAY(3)
      RPHI = 1.0 + DPHIF/PHIOF
      UVAR(1) = RPHI*RPHI*RPHI
      RETURN
C
C----67--1---------2---------3---------4---------5---------6---------7-*
C
 1000 FORMAT(/,'ERROR IN UVARM-CALL FOR VARIABLE IE',/,
     1       10X,'FOR ELEMENT NUMBER = ',I5,/,
     2       10X,'INTEGRATION POINT  = ',I5,/,
     3       10X,'AT TIME            = ',E12.3)
      END
```

Revised to also store $\log_{10}(R_k)$ as UVAR(2). GWO 11/8/99

GWO

Pages 28-30 entries by GWO 11/02/99

November 3 1999

Rock Strength Parameters

Pages 31-33 entries by GWO 11/3/99

The rock-mass strength was previously modeled using the Drucker-Prager yield criterion (p. 23 of Notebook #263), which required transforming the strength parameters from the Mohr-Coulomb model to their Drucker-Prager equivalent. The ~~use of~~ (GWO 11/3/99) the Drucker-Prager model was adopted because the Mohr-Coulomb model was not available in ABAQUS (Version 5.6) at the start of the project. The Mohr-Coulomb model has now been implemented in ABAQUS (starting from Version 5.8) and will be used ~~for~~ in subsequent analyses.

The Mohr-Coulomb strength criterion defines rock strength in terms of the equation

$$\tau = c_s + \sigma_n \tan \phi_s$$

where τ is the shear stress and σ_n is the normal stress on a potential failure surface, and c_s and ϕ_s are the cohesion and friction angle. ~~This~~ (GWO 11/3/99) (subscript s stands for "standard formulation"). This equation can be re-written in terms of major ~~pri~~ and minor principal compressive stresses, σ_1 and σ_3, respectively, as follows:

$$\sigma_1 = 2c_s \tan\alpha + \sigma_3 \tan^2\alpha \quad ---- (1)$$

where

$$\alpha = \pi/4 + \phi_s/2 \quad \text{(radians)}$$

$$\alpha = 45 + \phi_s/2 \quad \text{(degrees)}.$$

The ABAQUS formulation of the Mohr-Coulomb model is given in terms q and p, as follows

$$q = c_a + p\tan\phi_a$$

where

$$q = \sigma_1 - \sigma_3$$

$$p = \tfrac{1}{3}(\sigma_1 + 2\sigma_3)$$

} for cylindrical stress states (σ_1 and σ_3 are compressive stresses: Note that ABAQUS sets tension positive).

and c_a and ϕ_a are the ABAQUS equivalents of the cohesion and friction angle. The strength equation may be recast as follows using the expressions for p and q:

$$\sigma_1 = \frac{c_a}{1 - \frac{1}{3}\tan\phi_a} + \left(\frac{1 + \frac{2}{3}\tan\phi_a}{1 - \frac{1}{3}\tan\phi_a}\right)\sigma_3 \quad --- (2)$$

Comparing the corresponding terms of equations (1) and (2), the following results:

$$\tan\phi_a = \frac{3(\tan^2\alpha - 1)}{2 + \tan^2\alpha}, \text{ and}$$

$$c_a = (2c_s \tan\alpha)(1 - \tfrac{1}{3}\tan\phi_a)$$

which give the ABAQUS friction angle and cohesion, ϕ_a and c_a, in terms of the standard friction angle and cohesion, ϕ_s and c_s. Both ϕ_s and c_s are obtained using procedures described on p. 20–22 of Notebook #263.

Example:

For $\phi_s \doteq 10°$, $\phi_a = 20.24°$ and $c_a = 2.09c_s$.
For $\phi_s = 20°$, $\phi_a = 37.67°$ and $c_a = 2.12c_s$.
For $\phi_s = 40°$, $\phi_a = 58.57°$ and $c_a = 1.95c_s$.

For the Mohr-Coulomb failure criterion, ABAQUS requires the user to specify:

p-q friction angle ϕ_a (degrees)
p-q dilation angle (degrees)
standard-formulation cohesion c_s (stress unit)

The p-q cohesion c_a is calculated internally by ABAQUS using the specified ϕ_a and c_s.

Pages 31–33 entries by GW 11/3/99

November 4 1999

Pages 34-36 entries by GLO 11/4/99

A simulated unconfined compression test was conducted (using ABAQUS) to test the implementation of the Mohr-Colomb failure criterion. The result of the test (calculated unconfined compressive strength) suggests that the input values of c and ϕ were interpreted (by ABAQUS) as c_s and ϕ_s (standard formulation) instead of c_s and ϕ_a as was indicated on ~~p. 34~~ p. 33. (GW 11/4/99) The problem was discussed with HKS technical support, and it was determined that the values of ϕ and c for ~~the~~ ABAQUS input should be based on the standard formulation, that is,

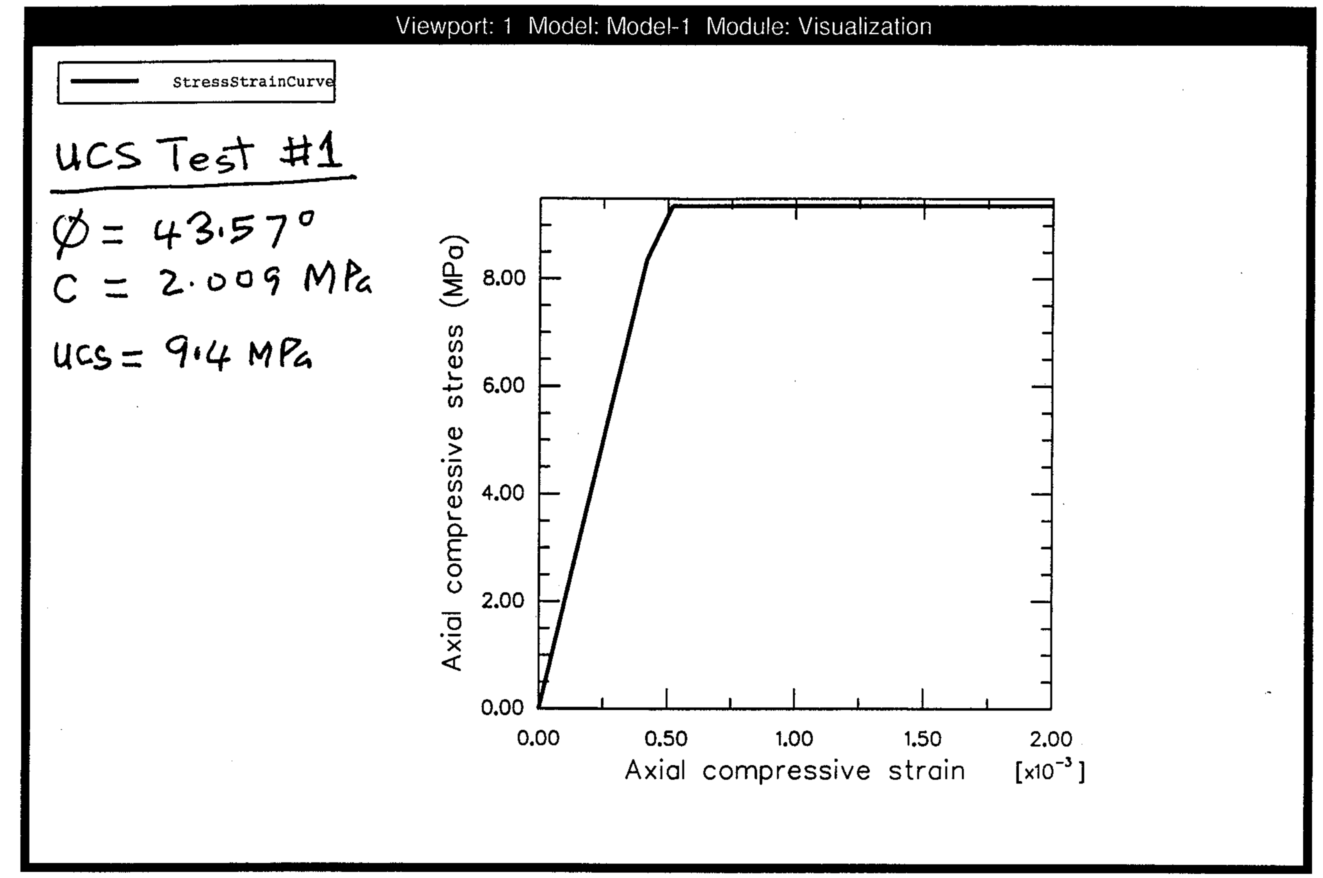
$$\phi = \phi_s \qquad c = c_s$$

} See p. 31-32.

This was verified through three simulated unconfined compression tests, the results of which are reproduced on p. 35-36.

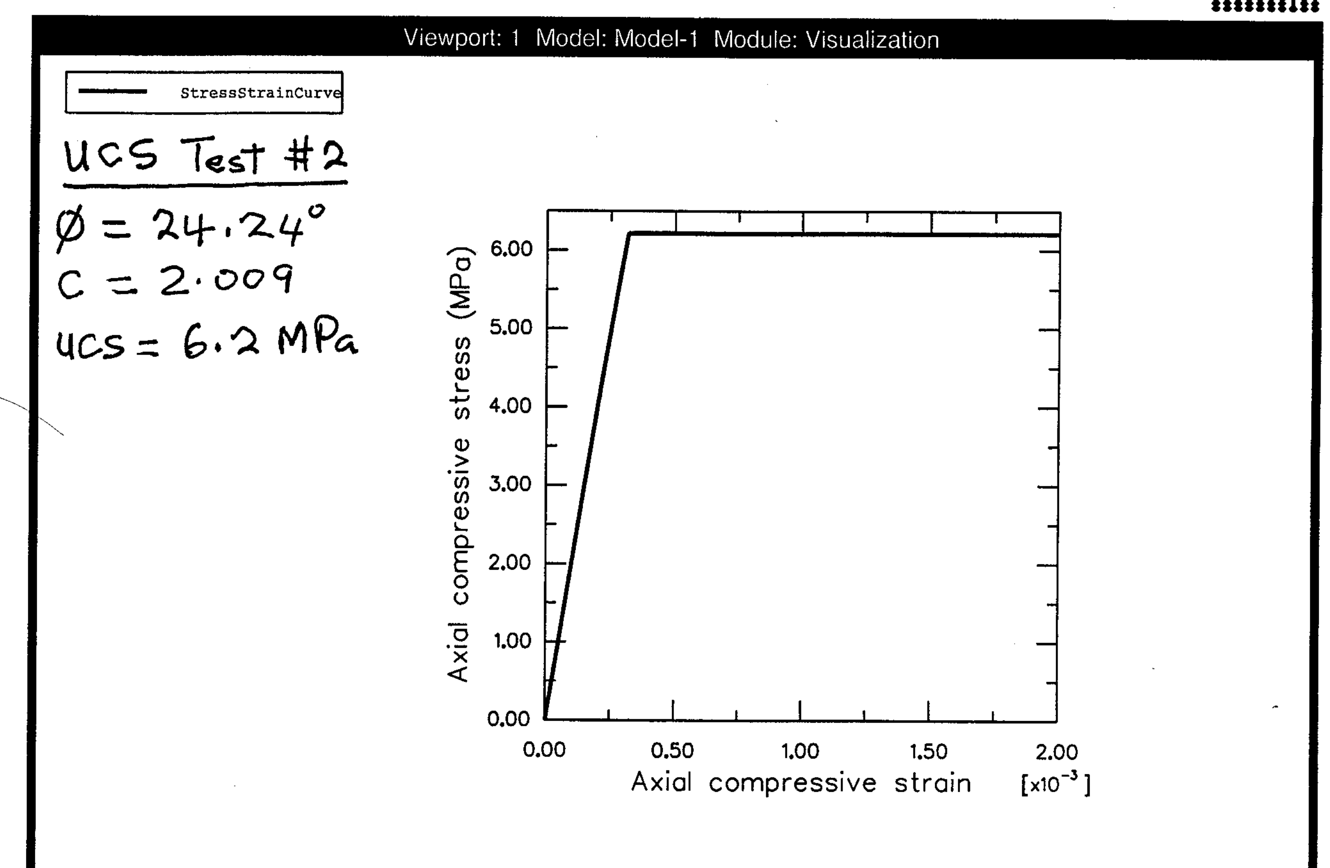
Printed on: Wed Nov 03 14:07:11 CST 1999

ABAQUS



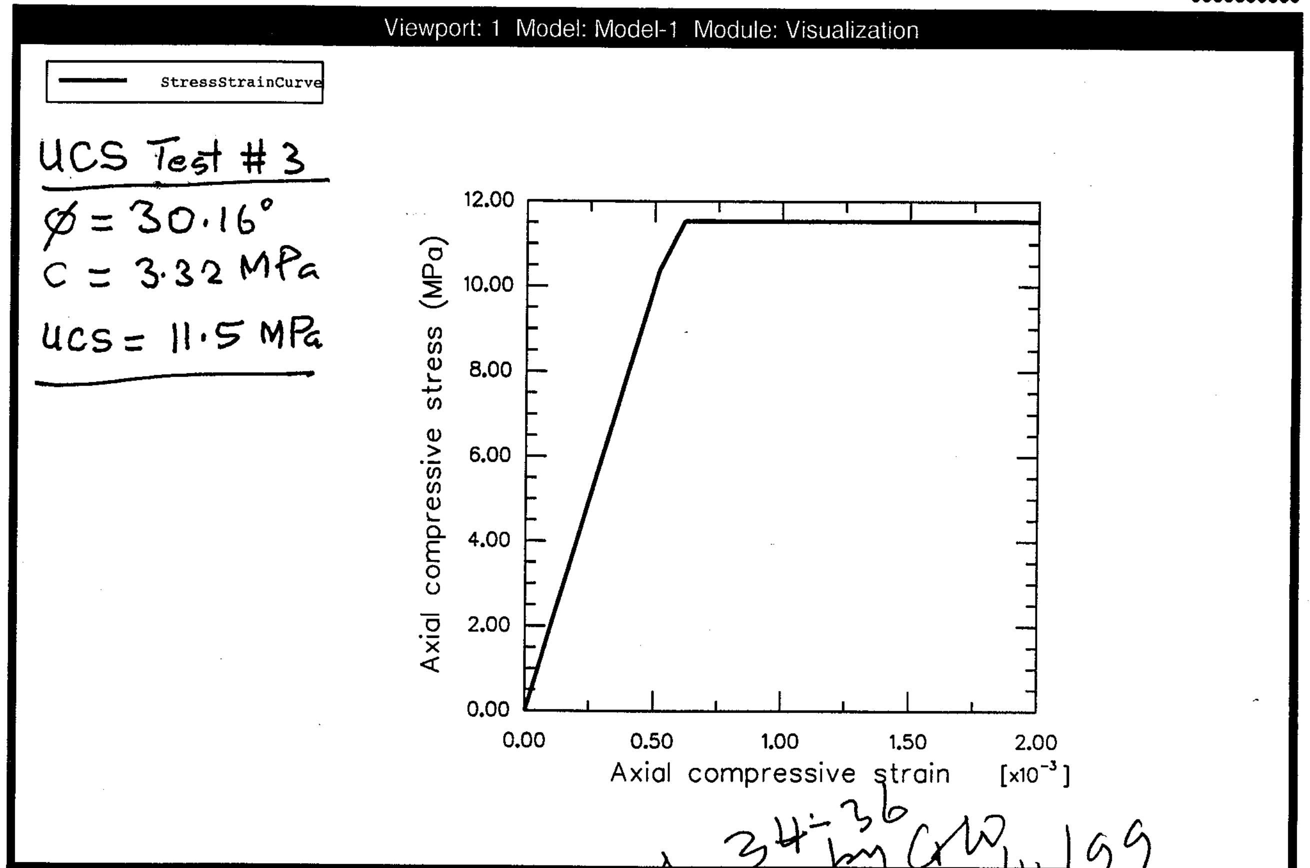
Printed on: Thu Nov 04 09:37:04 CST 1999

ABAQUS



Printed on: Thu Nov 04 09:57:27 CST 1999

ABAQUS



Pages 34-36 entries by GW 11/4/99

November 5 1999

Pages 36-39 entries by GW 11/5/99

Analyze model m01 (described on p. 61 of Notebook #263) again with the following changes:

(1) Use the Mohr-Coulomb failure criterion, instead of Drucker-Prager, for rock-mass strength (see p. 31-36).

(2) Use the revised equations (p. 29-30) for permeability change.

The revised model (sm01) uses the following input files:

sm01.inp	Main mechanical-analysis input file.
allNodes.def	Node definitions (not changed; see p. 61 of notebook #263).
iniTemp.def	Initial temperature (not changed).
mdfld1.def	Defines a field variable assigned the value of x-coordinate at every node (not changed)
sm01Elem.def	Element and property-type definitions.
sm01Emat.def	Elastic properties file
sm01Fric.def	Friction and dilation angles
sm01Cpar.def	Cohesion parameter
t0150.fil	Temperature-history file from thermal analysis.
KFrac02.f	FORTRAN code on p.30.

The results of the analyses were processed by running the following command files from ABAQUS/POST:

KContPlots.jnl	Main command file, which invokes the other command files listed on the next page.

lKContPars.def — Defines contour parameters

grpsOf25.def — Sets up element groups that divide the model domain into groups of 25 drifts.

g25kKContours.def — Contour-plotting commands.

The resulting contour plots is contained in four PostScript files as follows:

g251SmK.ps — First group of 25 drifts (Drifts #100–76).

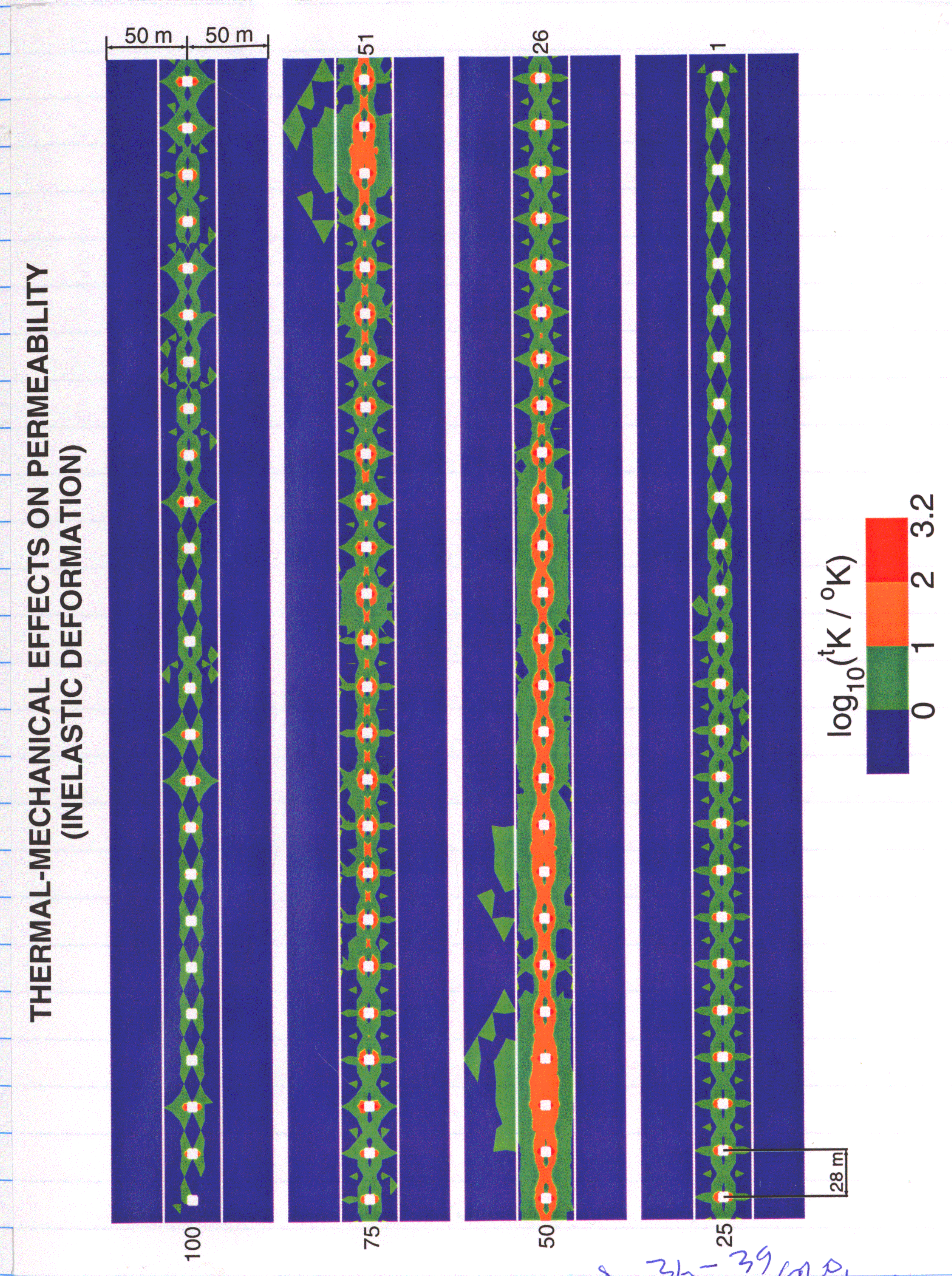
g252SmK.ps — Drifts #75 – 51

g253SmK.ps — Drifts #50 – 26

g254SmK.ps — Drifts #25 – 1

These files were assembled using Adobe Illustrator to obtain the combined plot on p. 39. The plot suggests the following observation:

~~The Increase~~ (GRD 11/5/99) The rock-mass permeability in the ~~dri~~ emplacement-drift area is likely to increase within a tabular zone centered on the ~~emplace~~ drift (GRD 11/5/99) drift-array axial plane. The tabular-zone thickness, generally a few drift diameters, would be maximum in the middle area and in areas of high thermal stress. Values of R_k may be highest in some areas of the tabular zone.



Pages 36–39 entries by GRD 11/5/99

November 09 1999 Pages 40-52 entries by GW 11/9/99

Setup drift-scale model for the investigations referred to on p. 28.

Basic Data

EDA-II Design of the Emplacement Area.
Drift geometry: Circular section of diameter 5.5 m; horizontal axis.
Drift spacing: 81 m center-to-center
Area Mass Loading: 60 MTU/acre

$$= \frac{60}{4047} \text{ MTU/m}^2$$

(1 acre = 4047 m^2)

Total No. of MTU for YM Site: 70,011
Total heat output from all waste packages: Given as function of time (see p. 13 of Notebook 263).

Drift Heat Source (per unit volume)

No. of MTU per drift length = MPD_L

$$= \left(81 \frac{m^2}{m}\right)\left(\frac{60}{4047} \frac{\text{MTU}}{m^2}\right)$$

= 1.20089 MTU/m of drift.

Heat output of all waste packages = Q (function of time from p. 13 of Notebook 263).
Number of MTU for YM Site = N = 70,011
Therefore, ~~# of~~ heat output of 1 MTU = Q/N GW 11/9/99

(Note: Both Q and N may need to be changed to account for the fact that Defence HLW is not included in the calculation of area mass loading).

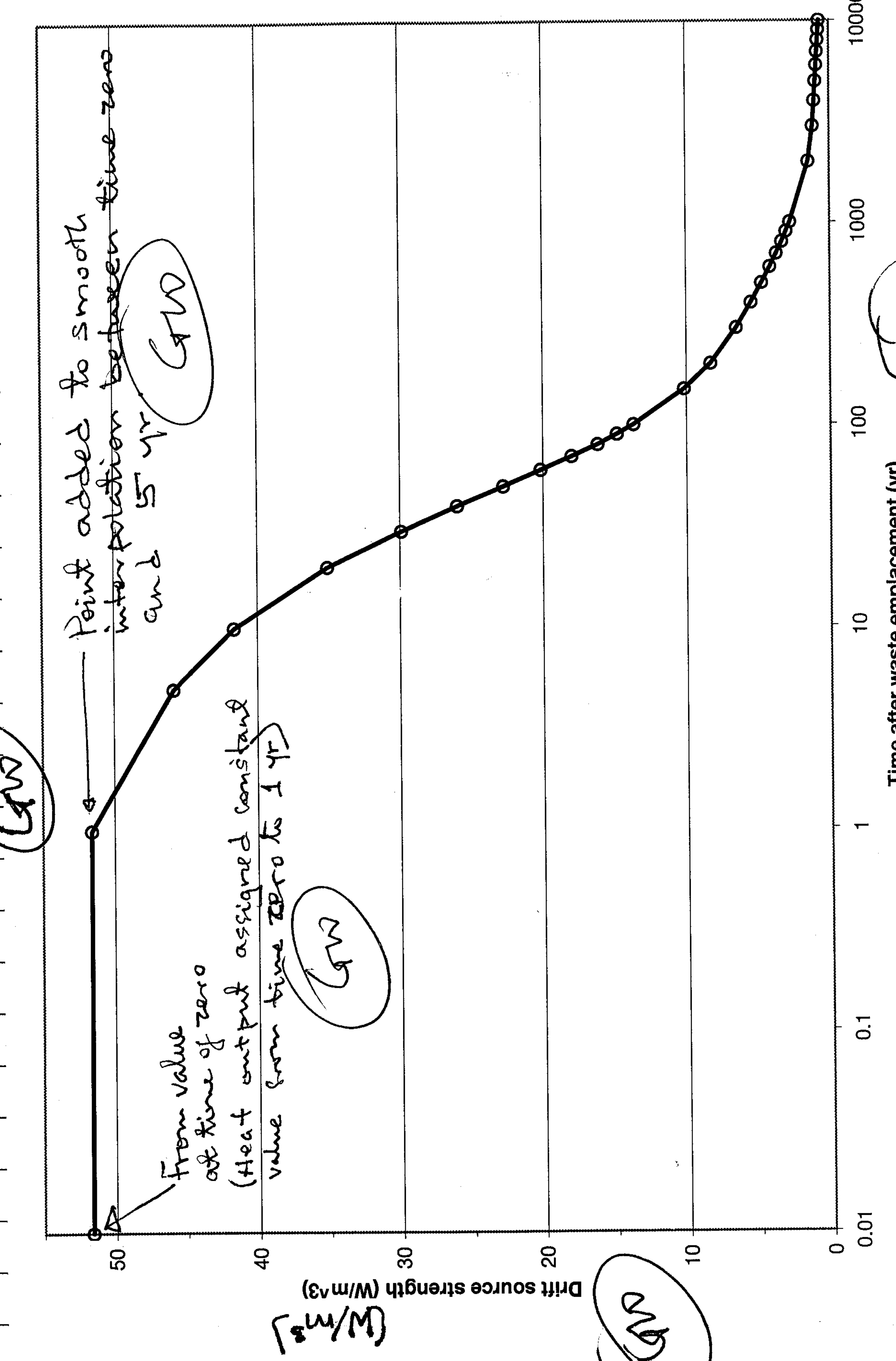
Volume of drift = A_{cd} m^3/m
where A_{cd} = drift cross-sectional area.

∴ Volumetric heat source = q

$$= \frac{(1.20089)(Q)}{(A_{cd})(N)} \quad W/m^3$$

q is a function of time. The computation of q is executed using the ~~C~~ C++ code on page 42. The resulting heat source history is shown on p. 43. (Note that the heat source in the ~~ABQU~~ ABAQUS input file is given in units of J/yr/m^3 whereas the one plotted on p. 43 is in units of J/s/m^3). GW 11/9/99

```
/***************************************************************************
Eda2DrftSrc

This code calculates the heat source per unit volume of emplacement
drift for the EDA-II thermal loading option: 60 MTU/acre with 81 m
(center-to-center) drift spacing.

The input data are:

 1.          Drift geometry
 2.          Total number of MTU to be disposed at YM repository
 3.          Total heat output from this number of MTU
 4.          Area mass loading (# of MTU per unit area)
 5.          Drift spacing

The calculated source-strength history
(J/yr/m^3) is output following ABAQUS input format.

Set value of variable theSection as follows:

  theSection = CIRCLE for a drift with circular section (drift-scale model)
  theSection = SQUARE for a drift with square section (site-scale model)

Author:                       G. I. Ofoegbu
Date:                         November 1999
System:                       C++ compiler
                              (Code developed using Borland C++ Builder 4)
***************************************************************************/

#include <stdio>
#include <iomanip>
#include <fstream>
#include <strstrea>
#include <math>
#include <string>
#include <conio>

#pragma hdrstop
#include <condefs>

using namespace std;

enum SectionType {CIRCLE, SQUARE};

//---------------------------------------------------------------------------
#pragma argsused
int main(int argc, char **argv)
{
  SectionType theSection = CIRCLE;     // Circular section
//  SectionType theSection = SQUARE;     // Square section

  char buf[151];
  char* inFileName = "d:\\ThermMech\\allWPSrc.txt";
  char* outFileName;
  if (theSection == SQUARE)
    outFileName = "d:\\ThermMech\\NARMS2000\\SiteScale\\sDrftSrc.def";
  else
    outFileName = "d:\\ThermMech\\NARMS2000\\DrftScale\\cDrftSrc.def";
  char* plotFileName = "d:\\ThermMech\\NARMS2000\\drftSrc.txt";
  int onThisLine = 0;
  float year,wpQ;
  float srcStrength;
  size_t nBuf = sizeof(buf);

  float megaJoule = 1.0e6;
  float pi = 4.0*atan(1.0);
  float mtuTotal = 70011.;
  float amL = 60.0;
  float driftSpacing = 81.0;
  float driftHeight = 5.5;
  float secPerYear = 365.25*24.0*60.0*60.0;
  float acre = 4047.0;
  float zeroTime = 2.0e-6;

  float xSecArea = driftHeight*driftHeight;
  if (theSection == CIRCLE) xSecArea *= (pi/4.0);

  ifstream Fin(inFileName);
  if (!Fin){
    cerr << "Unable to open file " << inFileName << endl;
    cout << "Press return to end ";
    getch();
    return (1);
  }
  ofstream Fout(outFileName);
  if (!Fout){
    cerr << "Unable to open file " << outFileName << endl;
    cout << "Press return to end ";
    getch();
    return (1);
  }
  ofstream Fplot(plotFileName);
  if (!Fplot){
    cerr << "Unable to open file " << plotFileName << endl;
    cout << "Press return to end ";
    getch();
    return (1);
  }

  Fout << "***\n";
  if (theSection == SQUARE)
    Fout << "*** Heat source strength function for a 5.5x5.5 m square drift\n";
  else
    Fout << "*** Heat source strength function for a 5.5-m diameter drift\n";
  Fout << "*** t1,q1,t2,q2 ... etc\n";
  Fout << "*** t = time in yr; q = Joules/m^3/yr\n";
  Fout << "***\n";

  Fplot << setw(12) << "Year" << setw(12) << "W/m^3" << endl;

  float mpd = driftSpacing*amL/acre;
  while (Fin){
    Fin.getline(buf,150);
    istrstream inpLine(buf,nBuf);
    inpLine >> year >> wpQ;
    if (inpLine.good()){
      srcStrength = mpd*(wpQ/mtuTotal)*megaJoule*secPerYear/xSecArea;
      Fplot << setiosflags(ios::fixed) << setprecision(1)
            << setw(12) << year
            << resetiosflags(ios::fixed)
            << setiosflags(ios::scientific) << setprecision(3)
            << setw(12) << (srcStrength/secPerYear)
            << resetiosflags(ios::scientific)
            << endl;
     if (onThisLine == 3){
       Fout << setiosflags(ios::fixed) << setprecision(3)
            << setw(12) << (year + zeroTime) << ","
            << resetiosflags(ios::fixed)
            << setiosflags(ios::scientific)
            << setprecision(5)
            << setw(12) << srcStrength
            << resetiosflags(ios::scientific)
            << endl;
       onThisLine = 0;
     }
     else{
       Fout << setiosflags(ios::scientific) << setprecision(3)
            << setw(12) << (year + zeroTime) << ","
            << resetiosflags(ios::fixed)
            << setiosflags(ios::scientific)
            << setprecision(5)
            << setw(12) << srcStrength << ","
            << resetiosflags(ios::scientific);
       onThisLine++;
     }
     if (year < zeroTime){  // Keep flux constant thru first year
       Fout << setiosflags(ios::scientific) << setprecision(2)
            << setw(12) << (year + 1.0 + zeroTime) << ","
            << resetiosflags(ios::fixed)
            << setiosflags(ios::scientific)
            << setprecision(5)
            << setw(12) << srcStrength << ","
            << resetiosflags(ios::scientific);
       onThisLine++;
     }
    }
  }

  cout << "Done ... Press return to end: ";
  getch();
  return 0;
}
```

Thermal Properties

Thermal conductivity K and density ρ are obtained from the reference reproduced below:

Table 4-3. Lithostratigraphic Units and Their Bulk Density, Thermal Conductivity, and Specific Heat (References 5.12, 5.54, 5.55, 5.60, 5.61, and 5.63)

T/M Unit	SNL Unit	Thickness (m)	Bulk Density (kg/m^3)	Thermal Conductivity (W/m-K) T<=100C	T>100C	Specific Heat (J/kg-K) T<=94C	94C<T<=114C	T>114
TCw	Tpcrv							
	Tpcm							
	Tpcpul							
	Tpcpmn							
	Tpcpll	28.4	2300	1.92	1.69	883.17	4076.00	912.13
	Tpcpln	60.9	2300	1.88	1.28	883.17	4076.00	912.13
	Tpcpv	0.0						
PTn	Tpcpv1	13.8	1460	1.07	0.51	1526.44	20076.03	1043.56
	Tpbt4	1.4	1310	0.50	0.35	1701.22	22374.81	1163.05
	Tpy	1.4	1790	0.97	0.44	1245.03	16374.86	851.17
	Tpbt3	0.4	1390	1.02	0.48	1603.31	21087.05	1096.12
	Tpp	15.0	1130	0.82	0.35	1972.21	25938.94	1348.32
	Tpbt2	5.5	1200	0.67	0.23	1857.17	24425.83	1269.67
	Tptrv	4.7	1200	1.00	0.37	1814.58	19624.25	1410.00
TSw1		2.4						
	Tptrn	41.1	2380	1.62	1.06	872.90	5153.57	849.54
	Tptrl	10.9	2130	1.68	1.27	975.35	5758.45	949.25
	Tptpul	62.5	2130	1.97	1.07	975.35	5758.45	949.25
TSw2	Tptpmn	29.7	2250	2.33	1.56	951.73	4657.16	970.62
	Tptpll	96.5	2210	2.13	1.50	968.96	4741.45	988.19
	Tptpln	56.6	2270	1.84	1.43	943.35	4616.12	962.07
TSw3	Tptpv	19.1	2270	2.08	1.74	1018.63	8229.30	1008.33
CHn1		16.2						
	Tpbt1	3.6	1660	1.31	0.68	1577.11	21311.27	1190.87
	Tac(v)	71.8	1520	1.18	0.59	1722.37	23274.14	1300.56
	Tac(z)							
CHn2	Tacbt	11.8	1790	1.34	0.70	1424.53	15131.62	1316.82
TSw2*		182.8	2235	2.06	1.49	958.12	4688.41	977.14

* Note: The thermal conductivity and bulk density values are calculated based on the weighted average over three combining units, Tptpmn, Tptpll, and Tptpln. The specific heat values are from Reference 5.12, Section 1.1326a, Table 5, p. 10.

Title: Repository Ground Support Analysis for Viability Assessment
Document Identifier: BCAA00000-01717-0200-00004 REV 01 Page: 20 of 140

GNO

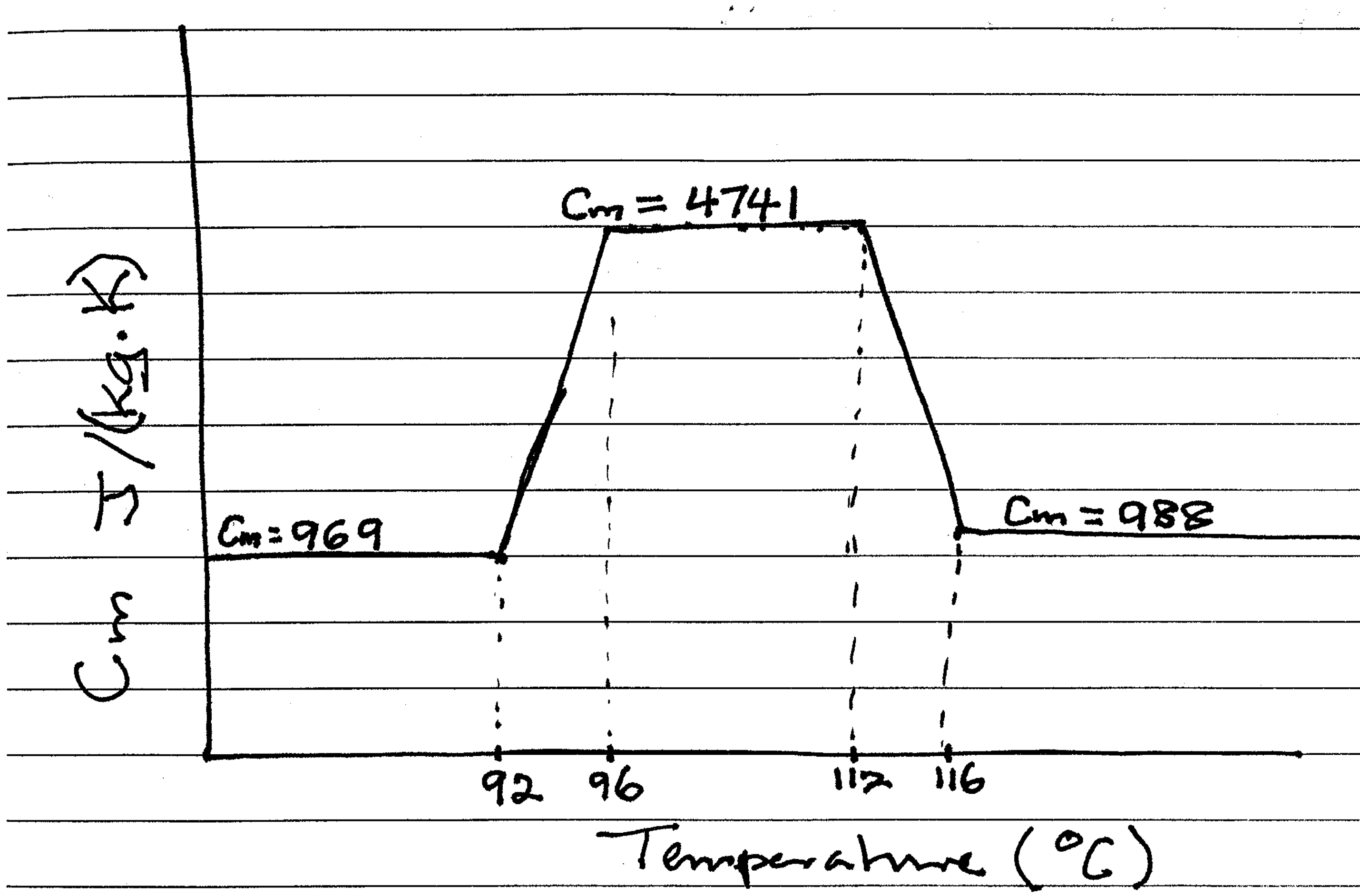
$\rho = 2210 \text{ kg/m}^3$ (replaces ρ given on p. 12 of Notebook #263).

GNO

$$K = 2.13 \text{ J/(s.m.K)}$$
$$= 2.13 \frac{\text{J}}{\text{s.m.K}} \times 3.1558\text{E}7 \frac{\text{s}}{\text{yr}}$$
$$= 6.7218 \times 10^7 \text{ J/(yr.m.K)}$$

The change in K at 100°C is to account ~~of~~ for assumed dry-out of rock at 100°C. This assumption is not made in the model. Therefore K is assigned constant value as above.

Values of specific heat capacity C_m are also taken from the table on p. 44. C_m is modeled as a function of temperature as follows to simulate the effects of evaporation in a conduction-only heat transfer analysis:



Analyses are also conducted with a constant C_m of 969 J/(kg.K).

Mechanical Properties

Analyses will be conducted for low-Q and high-Q areas, represented by rock mass quality categories 1 and 5, respectively, as given in the reference reproduced below:

Title: Repository Ground Support Analysis for Viability Assessment
Document Identifier: BCAA00000-01717-0200-00004 REV 01 **Page:** 22 of 140

Table 4-7. Rock Mass Mechanical Properties for TSw2 Unit.

Parameter	Rock Mass Quality Category		Source
	1	5	
Elastic Modulus *E* (GPa)	~~7.76~~ 8.0 (GW 11/5/99)	32.61	Reference 5.62, Table 4.
Poisson's Ratio *ν*	0.21	0.21	Reference 5.17, Table 7-6.
Cohesion *c* (MPa)	1.5	5.2	Reference 5.62, Table 5.
Friction Angle *ϕ* (degrees)	43	46	Reference 5.62, Table 6.
Shear Modulus *G* (GPa)	3.21	13.48	Calculated using $G=E/[2(1+\nu)]$, Reference 5.52, p. 173.
Bulk Modulus *B* (GPa)	4.46	18.74	Calculated using $B=E/[3(1-2\nu)]$, Reference 5.52, p. 173.
Tensile Strength q_t (MPa)	1.32	4.21	Reference 5.62, Table 8.

Poisson's ratio, ν, is set to the intact-rock value of 0.21 and the value of Young's modulus, E, is set to ~~7.76~~ 8.0 GPa (GW 11/5/99) for Category 1 (low-Q) and 32.6 GPa for rock-mass quality Category 5 (high-Q). The values of thermal expansivity α are assigned as follows using information from

Table 4-8 of the reference on p. 46.

Temperature (°C)	α 10^{-6}/K
0	7.14
50	7.14
100	9.07
125	9.98
150	11.74
175	13.09
200	15.47
225	18.0

Analyses will also be conducted using a constant α of 10×10^{-6}/K.

The values of rock strength parameters (friction angle and cohesion) given in the table on p. 46 are not supported by the Q values for the corresponding rock-mass quality categories (Q = 0.47 for Category 1 and Q = 9.30 for Category 5). Therefore, the values of friction angle ϕ and cohesion c were obtained using these Q values in the Hoek and Brown (1997) procedure described on pp. 20–22 of Notebook 263. The resulting c and ϕ values are:

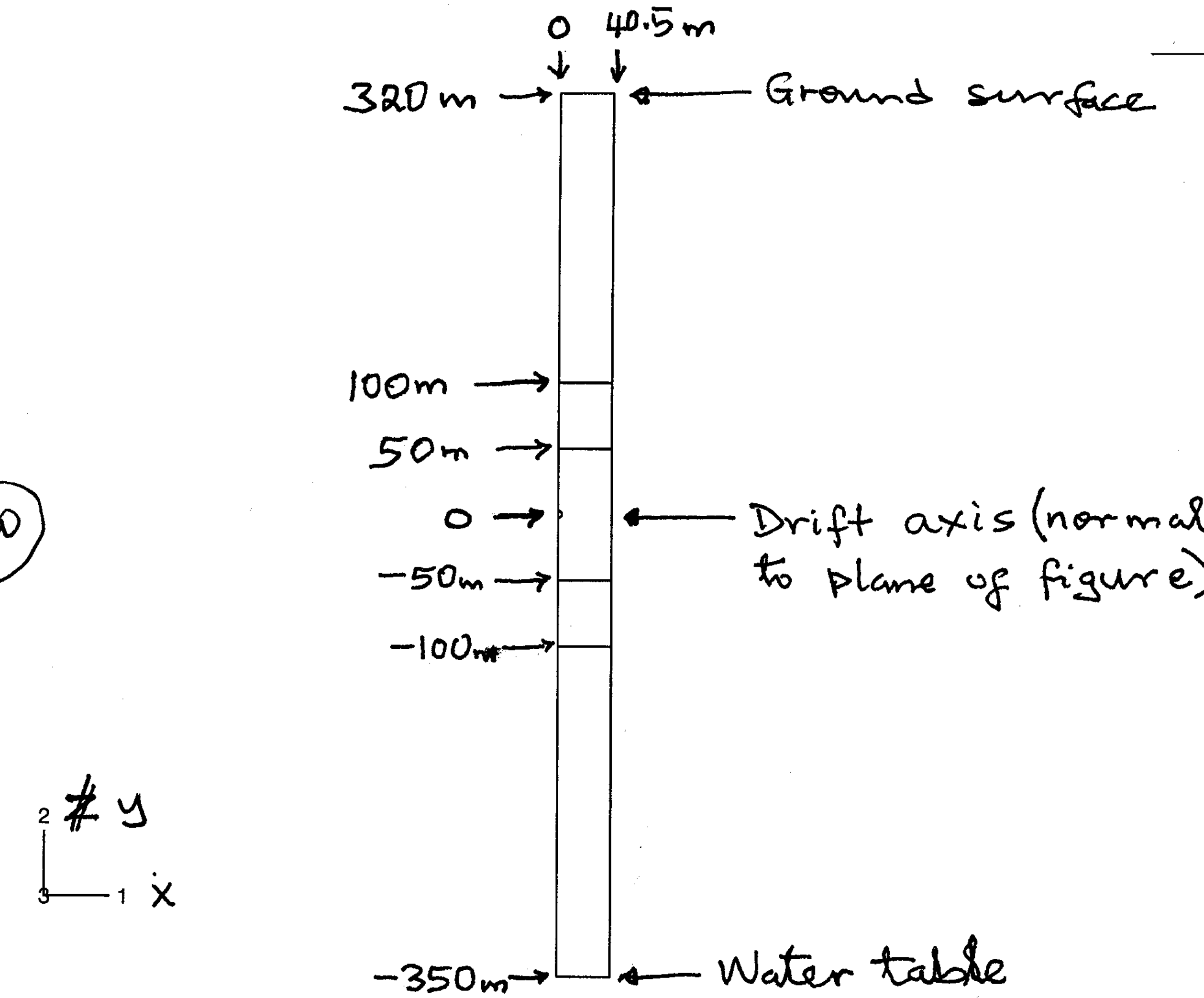
Rock-mass Strength Parameters

Q value	50% of intact-rock UCS (MPa)	Ø (degrees)	C MPa
0.47	84	27.5	2.82
9.30	84	34.4	5.08

UCS = Unconfined compression strength.
The intact rock UCS is reduced to 50% of its value to account for the effect of sustained loading on intact-rock strength.

Model Geometry

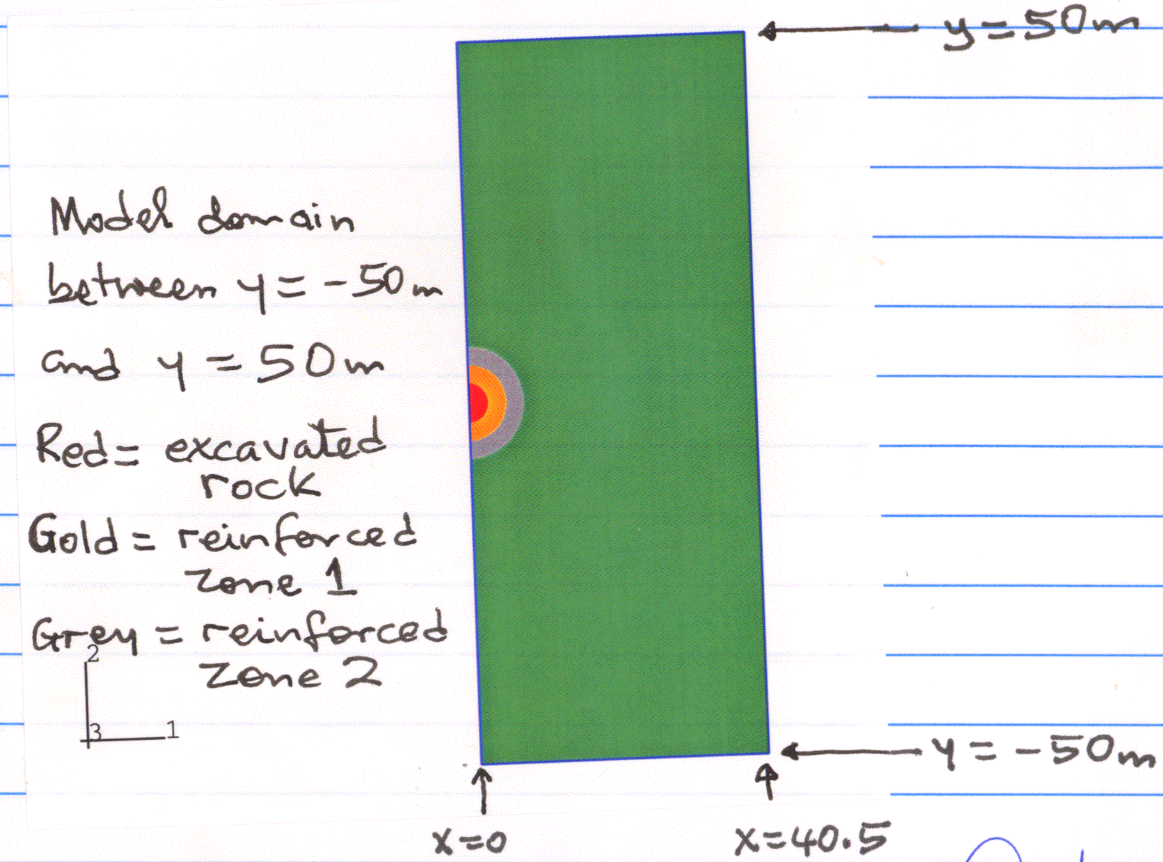
The model domain is a vertical rectangle 40.5 m wide by 670 m high that extends from the water table at y = −350 m to the ground surface at y = 320 m. The drift axis is horizontal and at y = 0 (see figure on p. 49).



(Elastoplastic zone: y = −100 to y = 100 m ~~Z = −100 to Z = 100 m~~)

Geometry of Entire Model Domain

The zones above y = 100 m and below y = −100 m is modeled as linear-elastic. The rest of the model domain (−100 m ≤ y ≤ 100 m) is modeled as elastic-plastic using the Mohr-Coulomb strength criterion. Material within the drift perimeter, which represents the excavated rock, is modeled as linear-elastic.



GRO 11/9/99

The drift is represented by a ~~sc~~ semicircular section of radius 2.75m. The two reinforced zones are each 2.5m thick. The reinforced zones are place holders for a bolt-reinforced zone that may be represented in the model at a later stage to examine issues related to reinforcement. However, in majority of the models the reinforced zones will be assigned the same properties as the rest of the elastic-plastic zone.

Boundary Conditions

Constant temperatures at y=320m and at y=-350m. Zero flux on the vertical boundaries for heat flow analysis. The mechanical boundary conditions are: zero horizontal displacement on the vertical boundaries and zero vertical displacement at the base. The top of the model is a stress-free surface.

Initial Conditions

Temperature of 18.7°C at the ground surface (y=320m) and temperature increasing with depth following the geothermal gradient defined on p. 24 of Notebook 263.

The initial vertical stress is based on a depth gradient of 0.02168 MPa/m (from density of 2210 kg/m^3 and gravitational acceleration of 9.81 m/s^2). The initial horizontal stress is calculated from a horizontal-to-vertical stress ratio of 0.2658 (from $\nu/1-\nu$, where $\nu = 0.21$).

Forcing Functions

The thermal load described on p. 40-43 is applied to the drift section (red zone on p.50) ~~Tempera~~ GWD 11/9/99 in the heat conduction analysis. Then the temperature history calculated from the heat conduction analysis is applied in the thermal-mechanical analysis. ~~The tunnel~~ The drift section is removed to simulate excavation GWD 11/9/99 before the application of the temperature history.

Pages 40-52 entries by GWD 11/9/99

December 22 1999

Pages 52-61 entries by GWD 12/22/99

Analysis Cases and Results

Thermal Analysis Cases

Two thermal analysis cases:

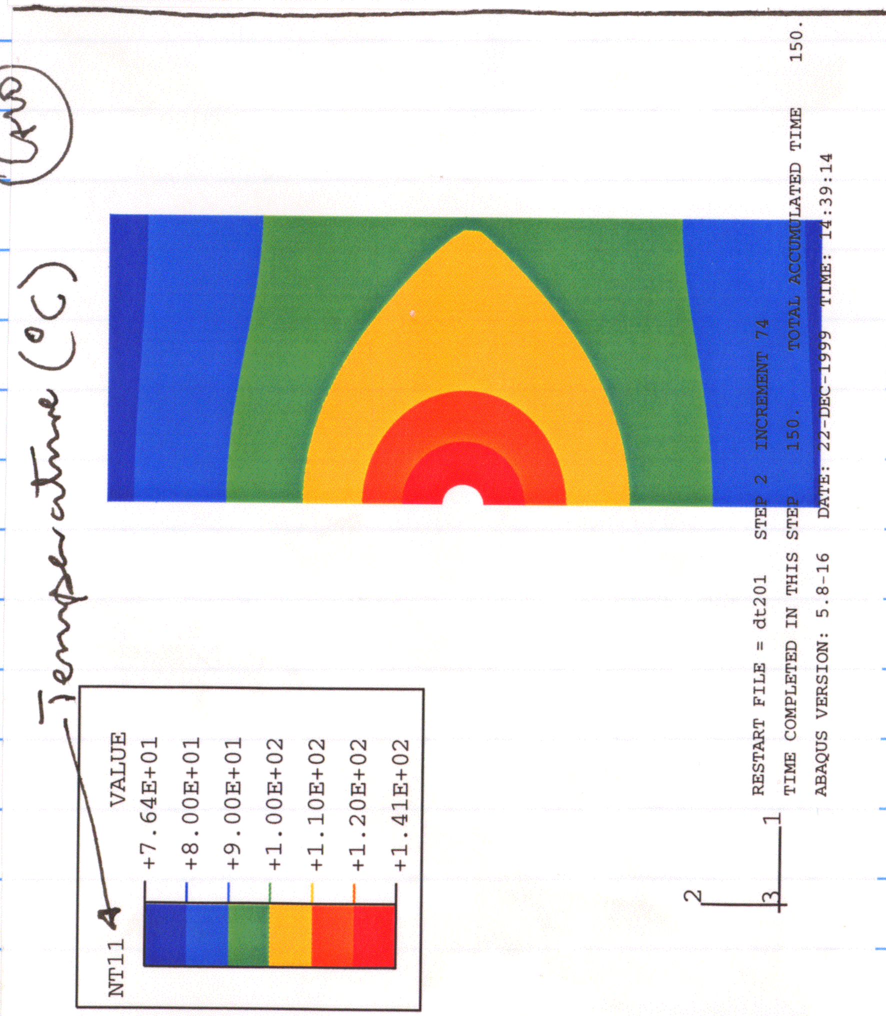
dt201 and dt301

Both are the same in every respect, except as follows:

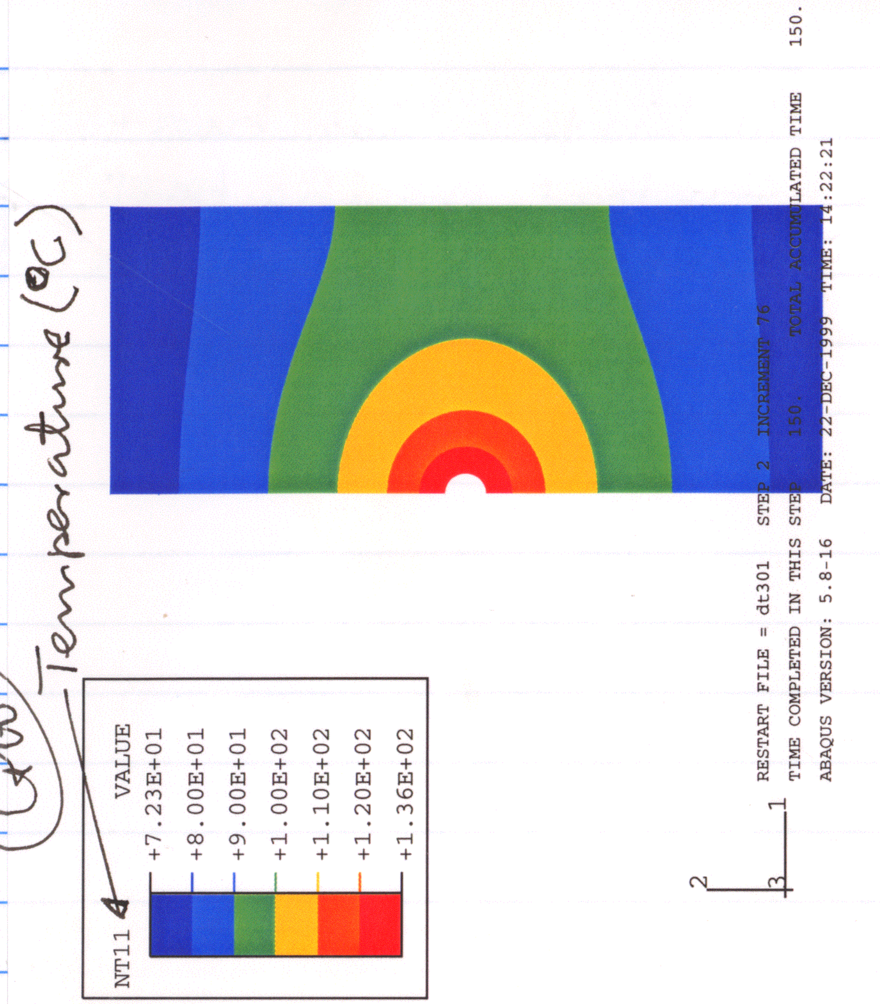
dt201: Constant specific heat of 969 J/(kg·K)

dt301: Specific heat varies with temperature as defined on p. 45.

Temperature distributions at 150 yr for zone between y=-50m and y=50m GWD



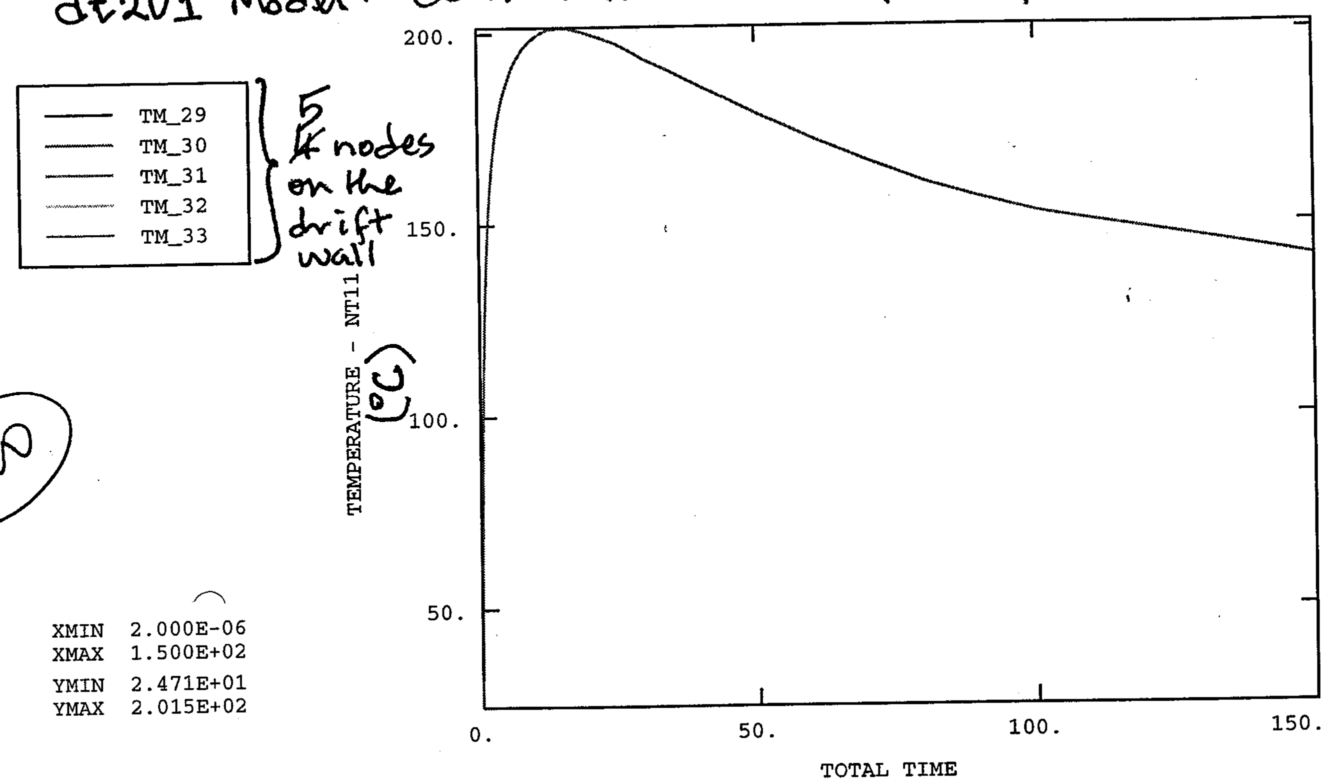
Case 201: Constant heat capacity



Case 301: Temperature-dependent heat capacity (as described on p. 45).

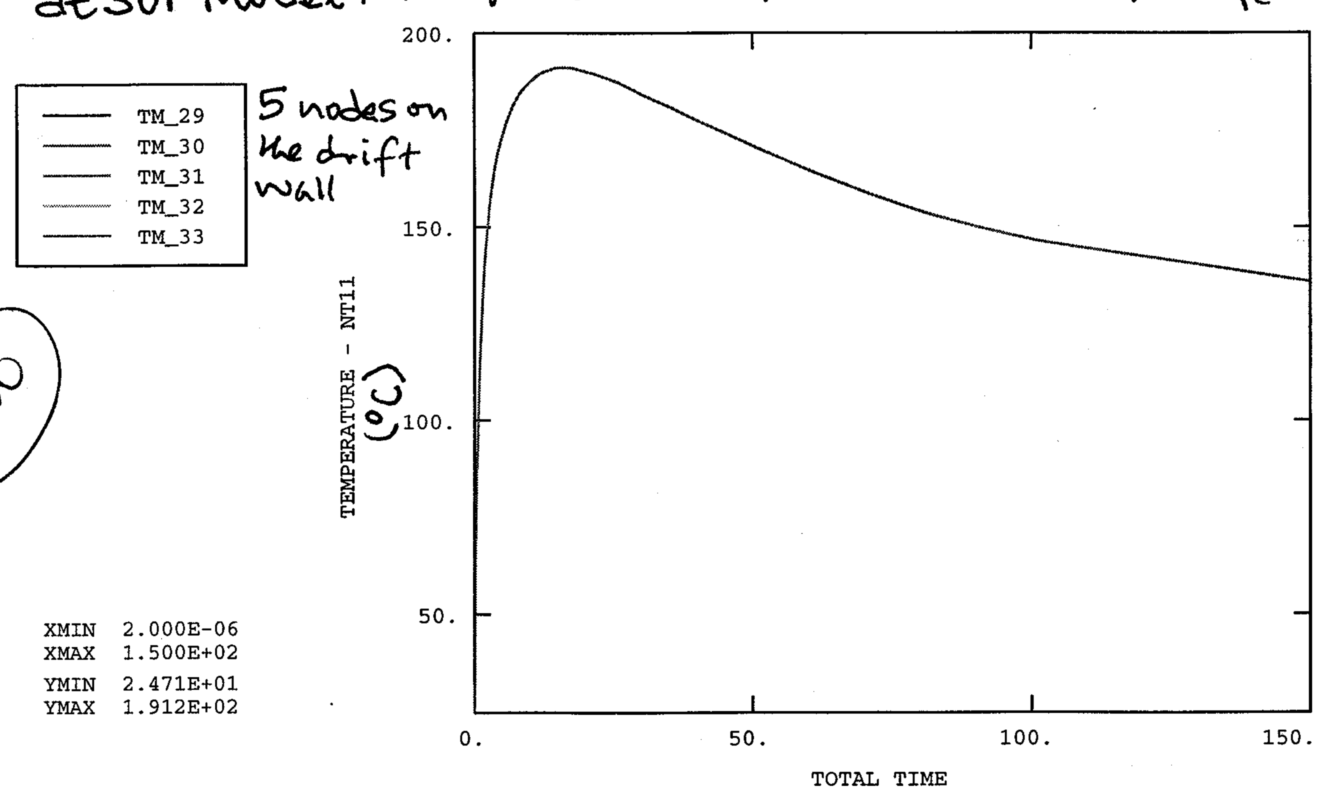
Drift-wall temperature histories from Case 201 (constant specific heat) and Case 301 (temperature-dependent specific heat).

dt201 Model: Constant heat capacity (Case 201)



GWO

dt301 Model: Temperature-dependent heat capacity (case 301)



GWO

Thermal-Mechanical ~~Mechanical~~ Analysis Case dm201:

GWO 12/22/99

Mechanical model: dm201
Constant thermal expansivity

Thermal model: dt201
Constant specific heat.

TM Analysis Case dm202

GWO 12/22/99

Mechanical model: dm202
Temperature-dependent thermal expansivity

Thermal model: dt201
Constant specific heat

TM Analysis Case dm301

Mechanical model: dm301 (Same as dm201)

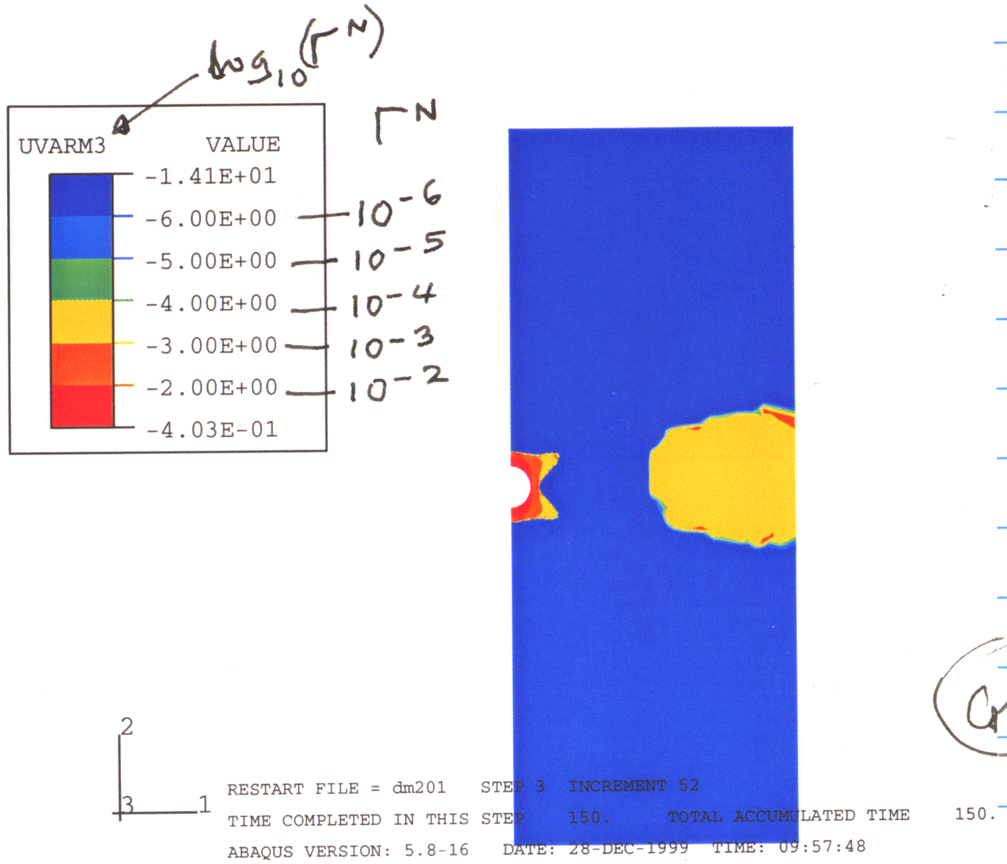
Thermal model: dt301
Temperature-dependent specific heat

TM Analysis Case dm302

GWO 12/22/99

Mechanical model: dm302 (same as dm202)

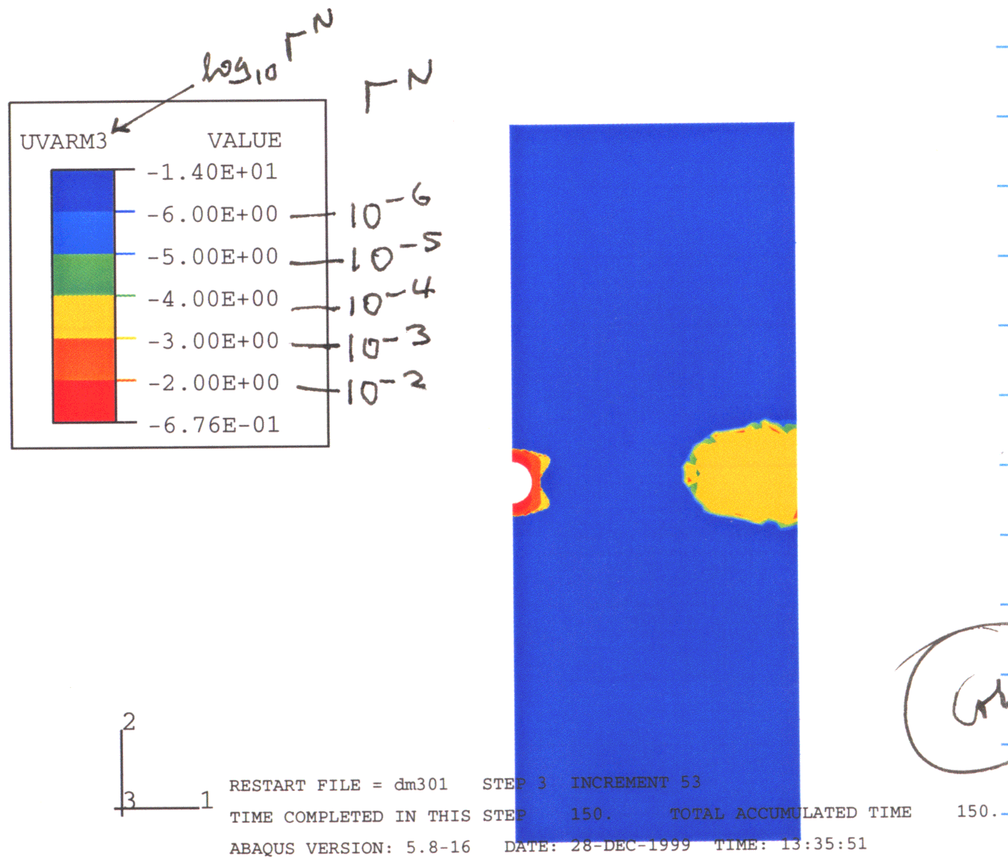
Thermal model: dt301
Temperature-dependent specific heat.



dm201:
Constant specific heat;
constant thermal expansivity

C_m = constant
α = constant
RMQ5

GHO

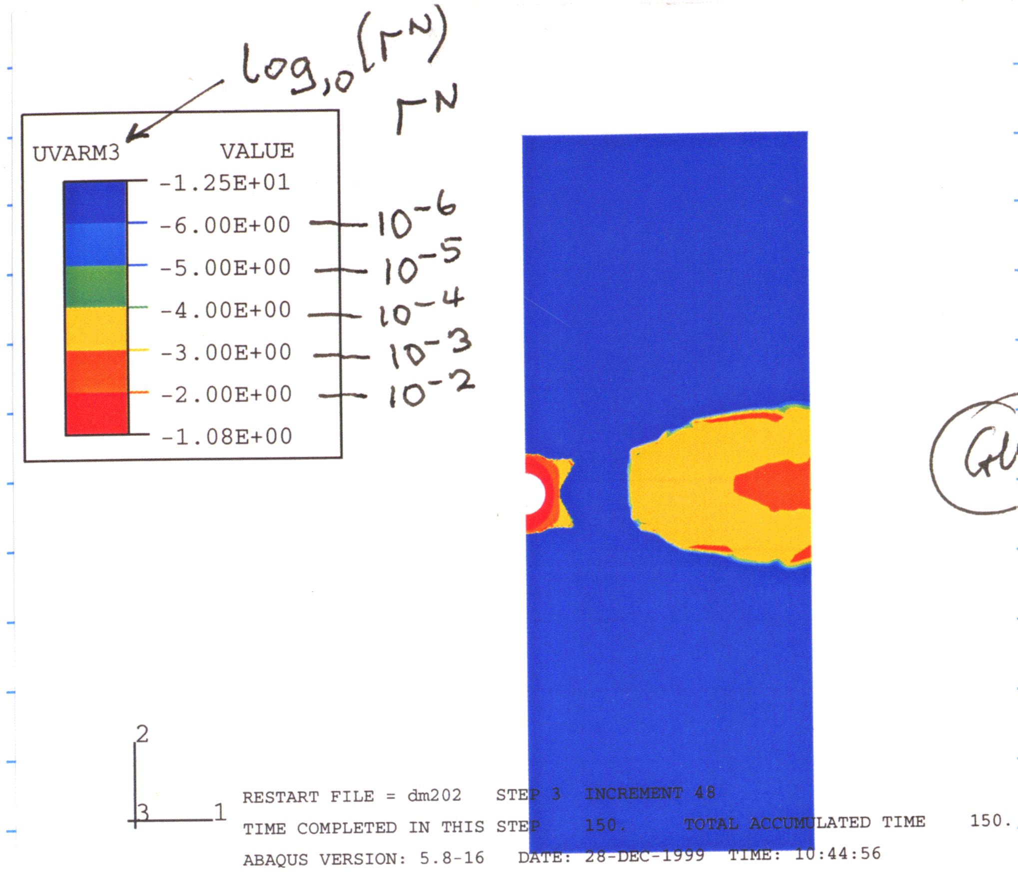


dm301:
Temperature-dependent specific heat;
Constant thermal expansivity.

$C_m = C_m(\theta)$
α = constant
RMQ5

GHO

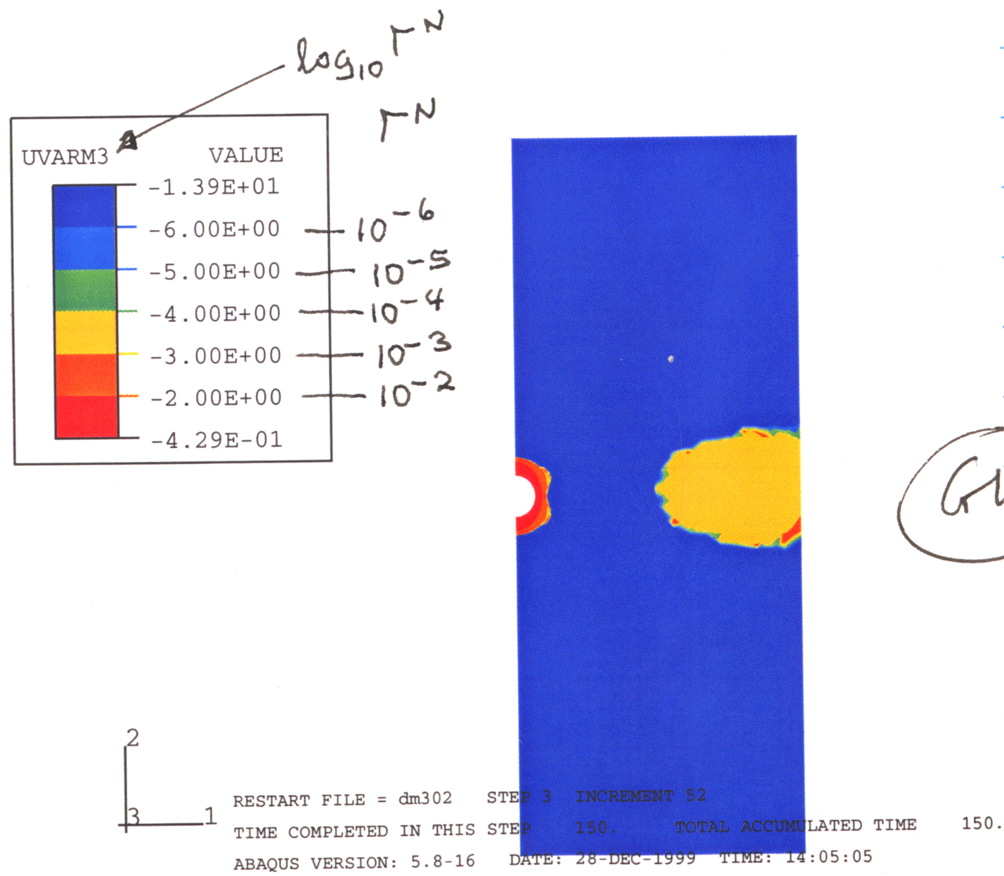
Γ^N = equivalent plastic strain (i.e., ABAQUS output variable PEEQ).



dm202:

C_m = constant
$\alpha = \alpha(\theta)$
RMQ5

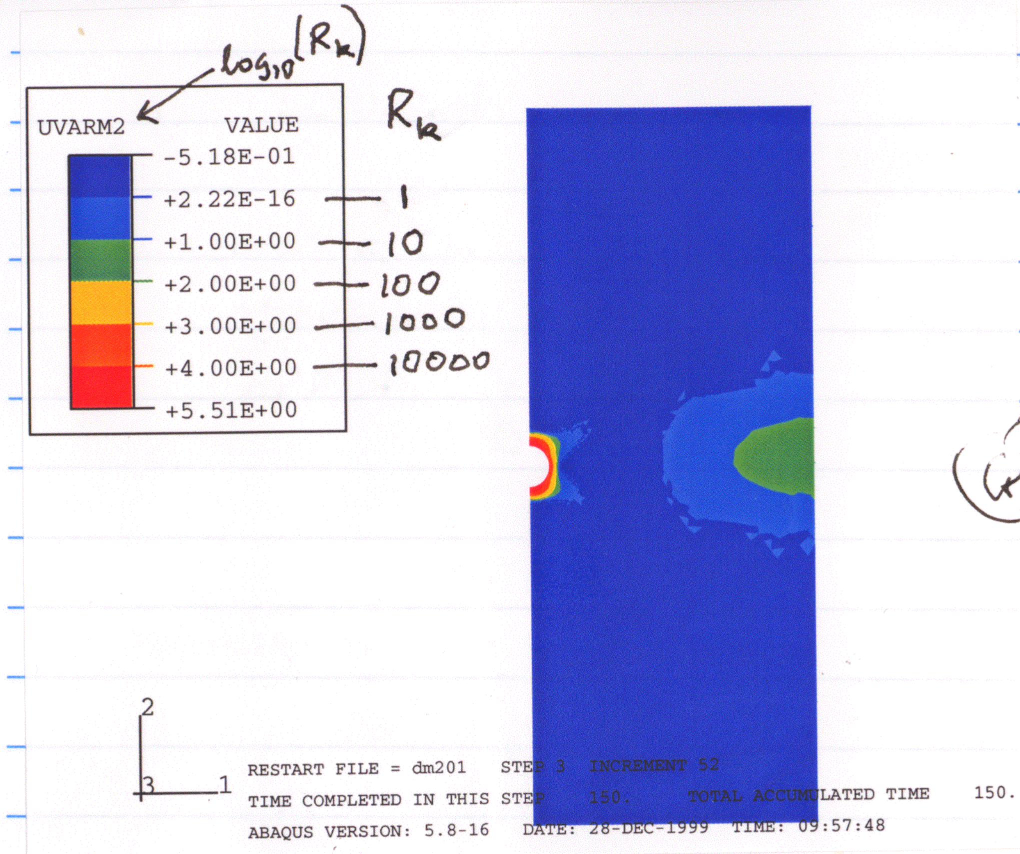
GHO



dm302:
$C_m = C_m(\theta)$

$\alpha = \alpha(\theta)$
RMQ5

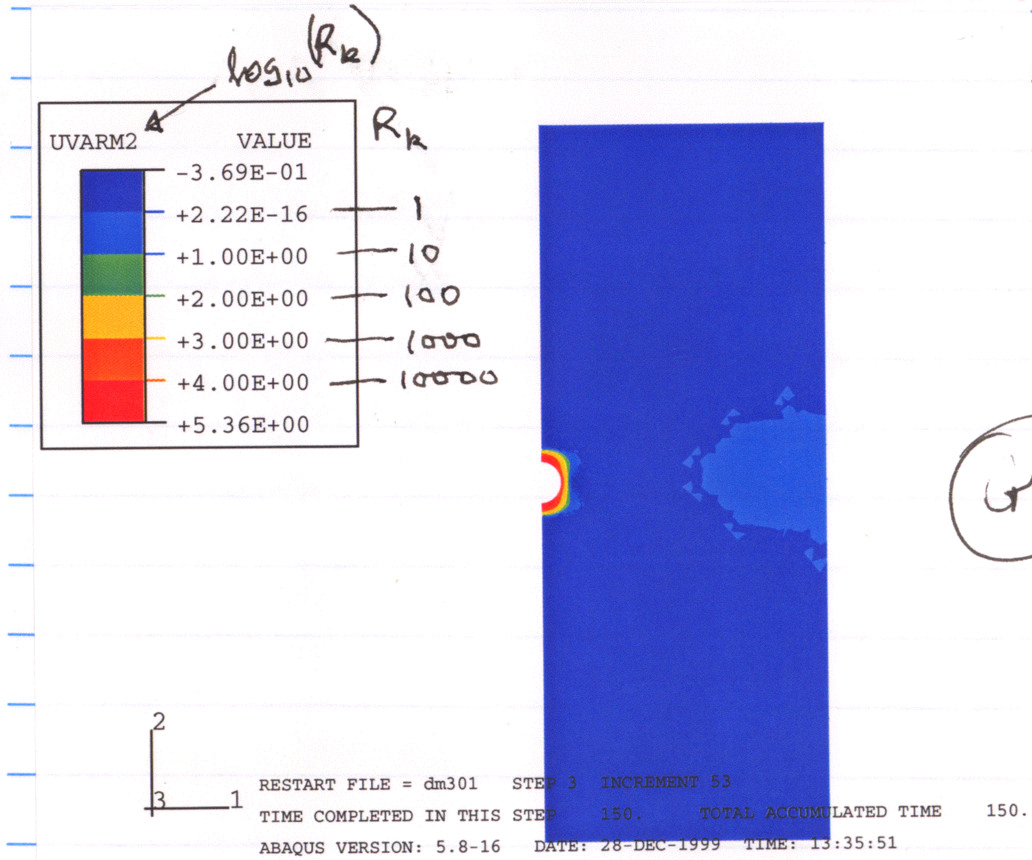
GHO



dm201:

C_m = constant
α = constant
RMQ5

GW

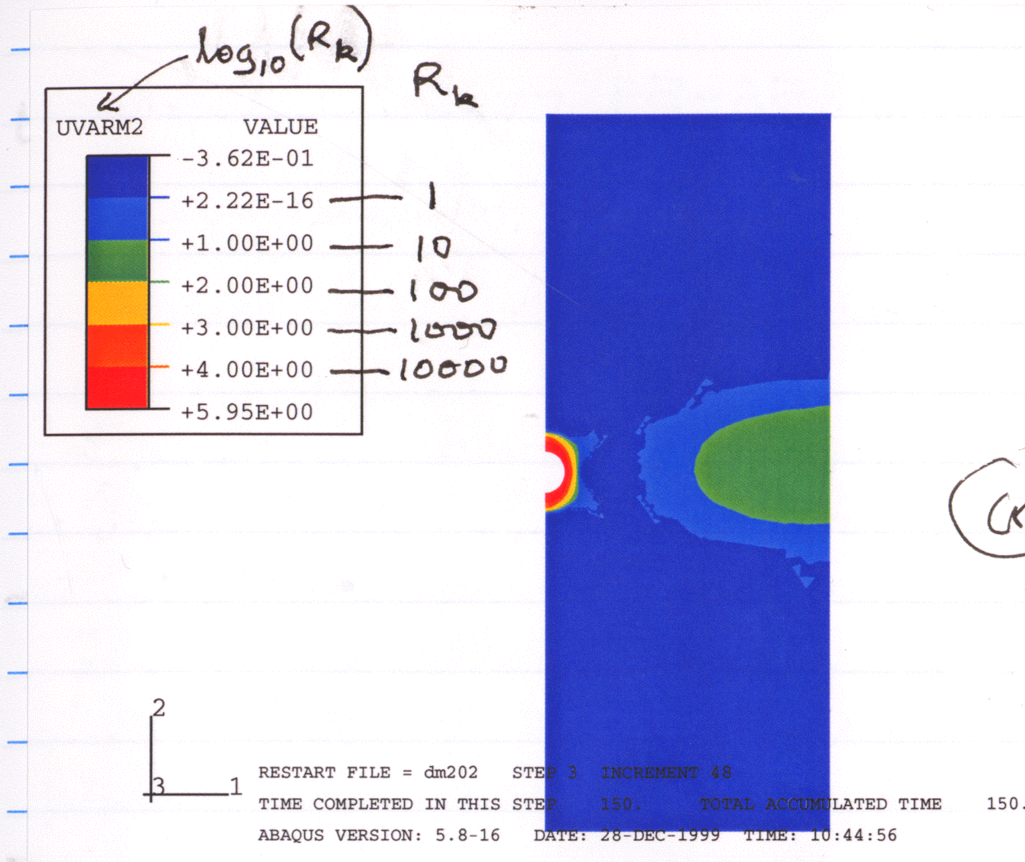


dm301:

$C_m = C_m(\theta)$
α = constant
RMQ5

GW

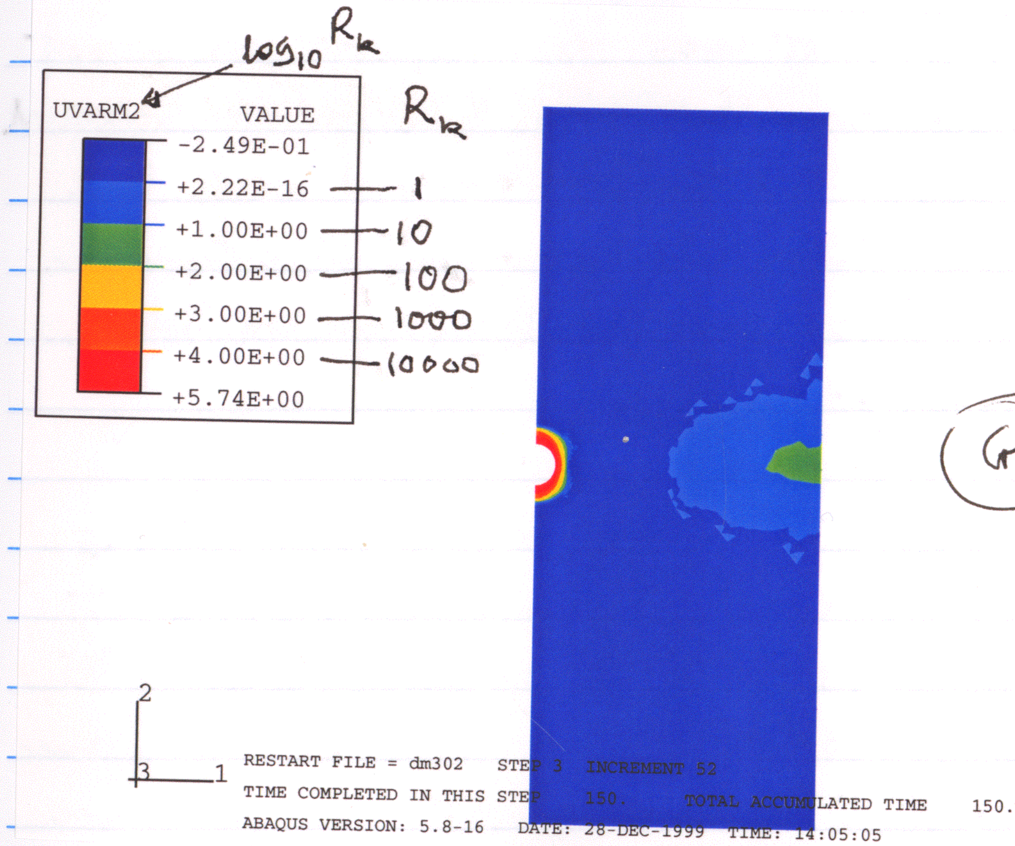
Permeability ratio, R_k, which ~~was~~ is defined on p. 29.



dm202:

C_m = constant
$\alpha = \alpha(\theta)$
RMQ5

GW



dm302:

$C_m = C_m(\theta)$
$\alpha = \alpha(\theta)$
RMQ5

GW

The results on p. 56-59 raise the following issues:

(1) DOE's models ~~are based on~~ use the temperature-dependent functions (GW 12/22/99) $C_m = C_m(\theta)$ and $\alpha = \alpha(\theta)$. As a result, DOE's predictions of inelastic response would be close to the results from dm302 (p. 57 and 59). However, the temperature-dependent function $C_m = C_m(\theta)$, which was introduced by DOE to account for the effect of evaporation in conduction-only models, may need to be re-examined to ensure that the quantity of heat loss to evaporation is not over-assigned. Results from thermal-hydrological models need to be compared with the results on p. 53 to determine whether the $C_m = C_m(\theta)$ function proposed by DOE is acceptable.

(2) The values of α used in these models (and in DOE models) are for intact rock. Rock-mass α's are expected to be smaller. However, rock-mass α's are also expected to vary more with temperature than intact rock α. As the models dm202 and dm302 show, (GW) ~~an increase in $\partial\alpha$~~ higher values of $\partial\alpha/\partial\theta$ lead to more ~~intense~~

inelastic response. Therefore, the use of intact rock α's may not necessarily lead to more inelastic response than the unknown rock-mass α as may often be assumed.

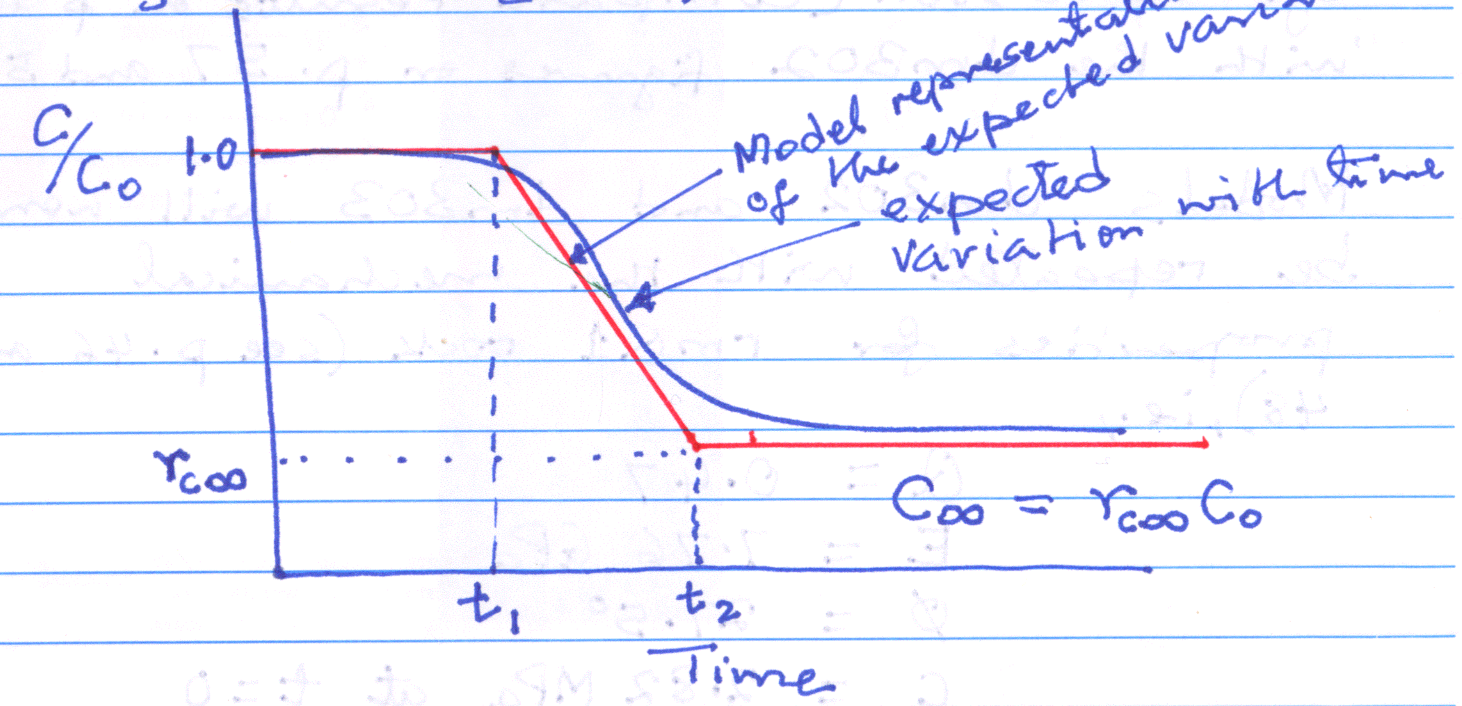
Pages 52–61 entries by GW 12/22/99

Dec 29 1999

Pages 61–63 entries by GW 12/29/99

TM Analysis Case dm303

Same as analysis case dm302 (p. 55) except for the following change: the cohesion parameter c is varied with time as follows ($c_0 = 5.08$ MPa as given on page 48)



For model dm303:

$c_0 = 5.08$ MPa, $r_{c\infty} = 0.5$

$t_1 = 50$ yr, $t_2 = 100$ yr.

As a result, the degradation ~~variation~~ of c is represented in model dm303 as follows:

GVO 12/29/99

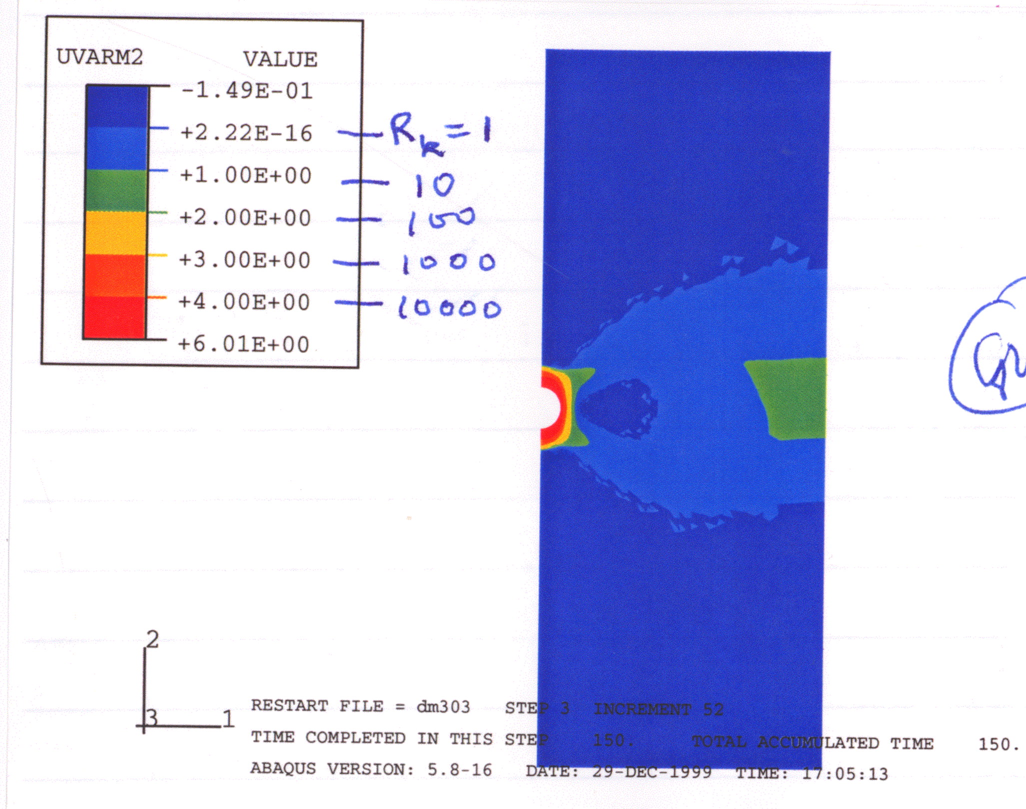
Time (yr)	c (MPa)
0	5.08
50	5.08
100	2.54
150	2.54

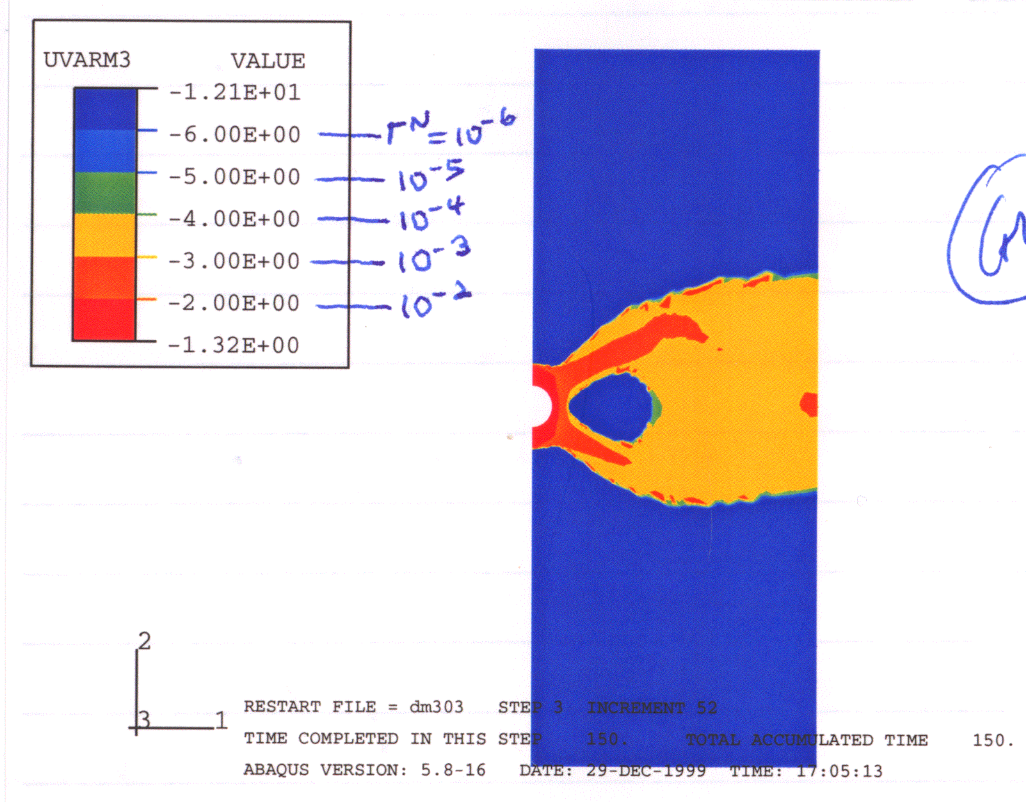
The results (p. 63) show a large increase in the zone of inelastic response owing to the degradation of cohesion (compare results on p. 63 with the dm302 figures on p. 57 and 59).

Models dm302 and dm303 will now be repeated with the mechanical properties for rmQ1 rock (see p. 46 and 48), i.e.,

$$Q = 0.47$$
$$E = 7.76 \text{ GPa}$$
$$\phi = 27.5^\circ$$
$$c = 2.82 \text{ MPa at } t = 0$$

and with $t_1 = 50$ yr, $t_2 = 100$ yr and $r_{c\infty} = 0.5$ as on p. 61.





dm303:

$$C_m = C_m(\theta)$$
$$\alpha = \alpha(\theta)$$
$$c = c(t)$$

RMQ5

GVO

Degradation of rock-mass cohesion parameter causes large increase in the zone of inelastic response.

GVO

Pages 61–63 entries by GVO 12/29/99

January 5 2000

Pages 64–68 entries by GW 1/5/2000

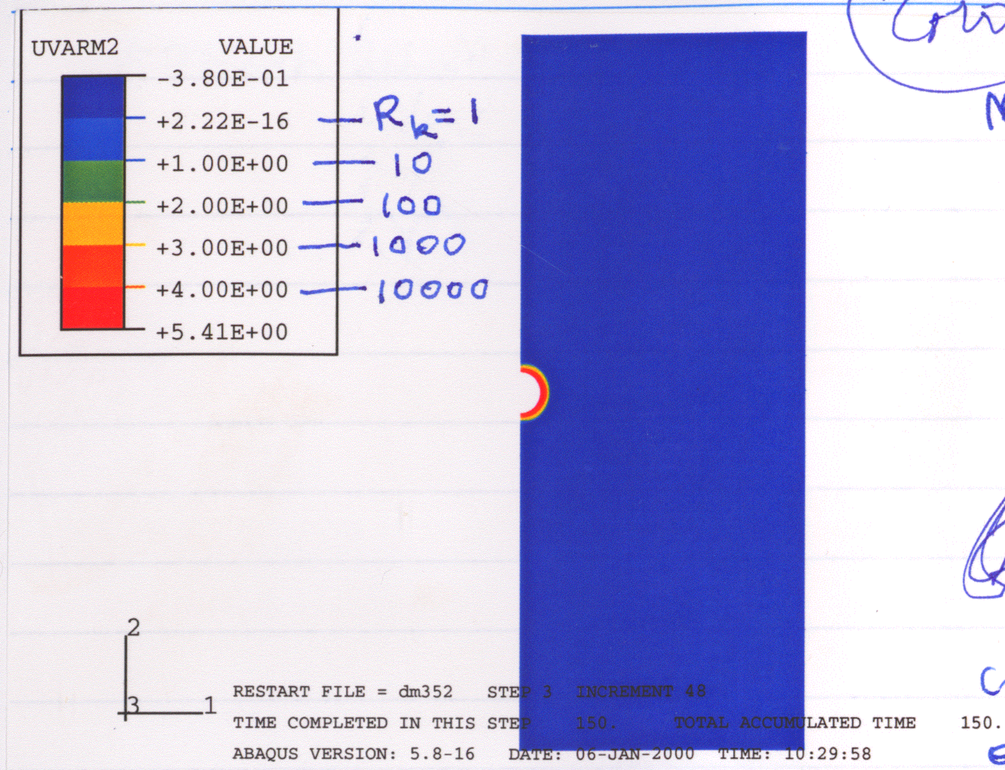
Model dm352:
Same as dm302 but with ~~rm05~~ RMQ1 (GW 1/5/00) mechanical properties as described on p. 62.

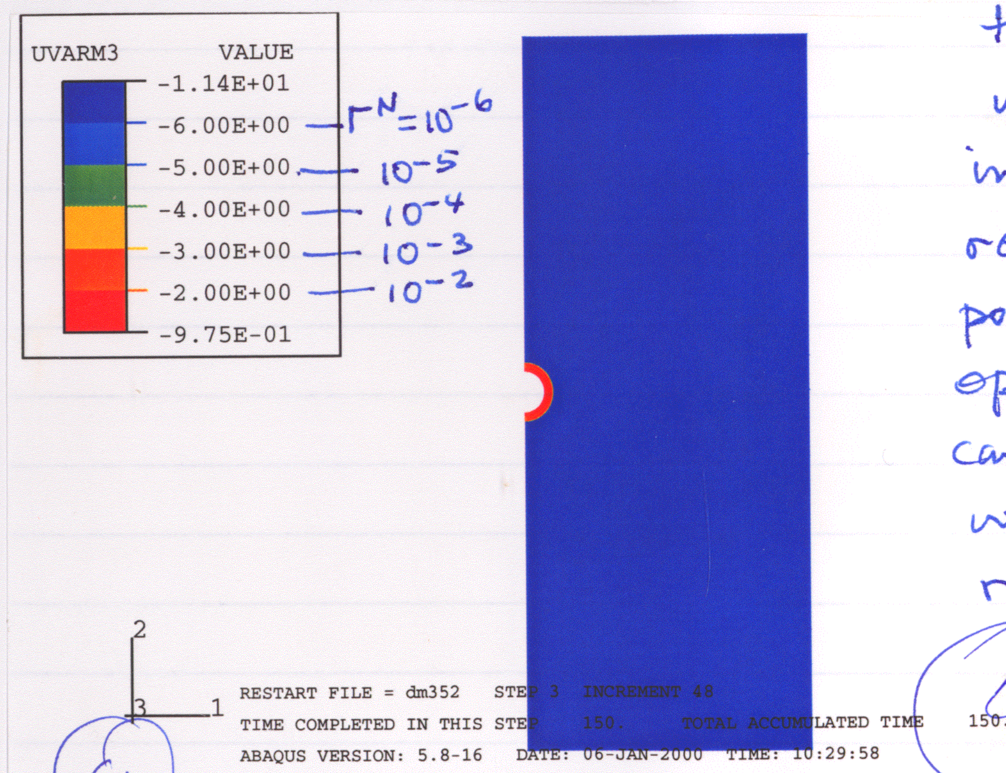
Model dm353:
Same as dm303 but with ~~rm05~~ RMQ1 mechanical properties as described on p. 62. (GW 1/5/00)

These two models produced low inelastic response (p. 65–66). Two additional models were also analyzed to examine the effects of increasing the value of rock-mass Young's modulus (for RMQ1 rock) by a factor of 2.5.

Model dm354: same as dm352 with $E = 20$ GPa, i.e., 2.5×8 GPa.

Model dm355: Same as dm353 with $E = 20$ GPa.





GW 1/5/00

GW

Model dm352:

$C_m = C_m(\theta)$

$\alpha = \alpha(\theta)$

$E = 8.0$ GPa

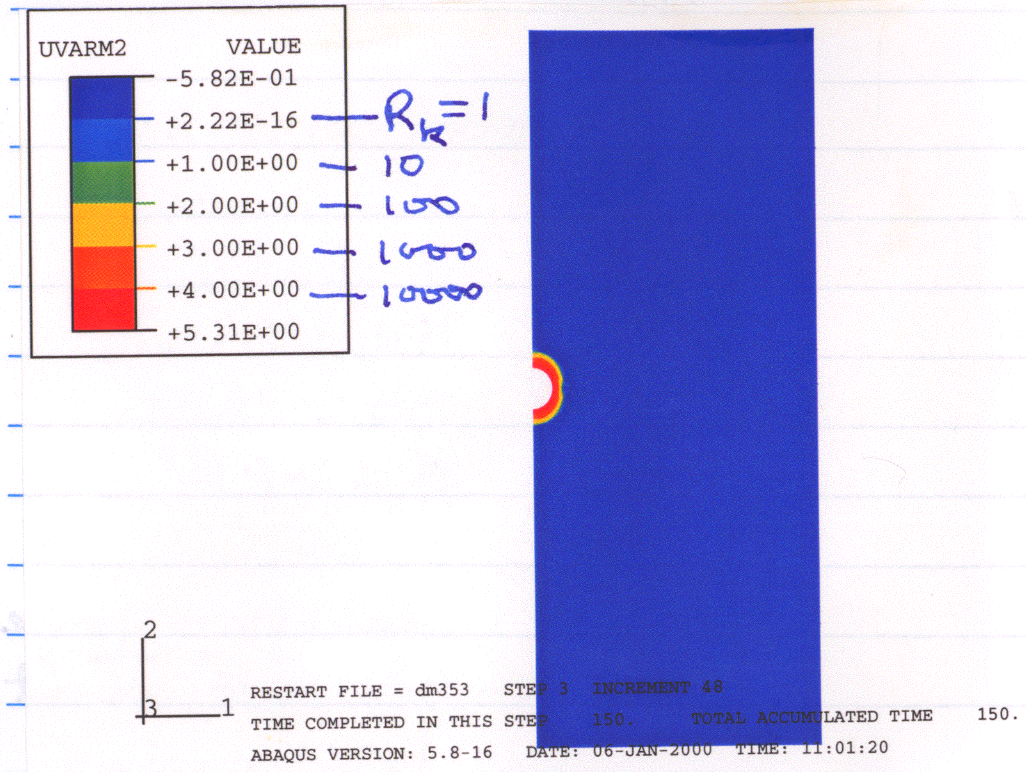
$C = 2.82$ MPa

$\phi = 27.5°$

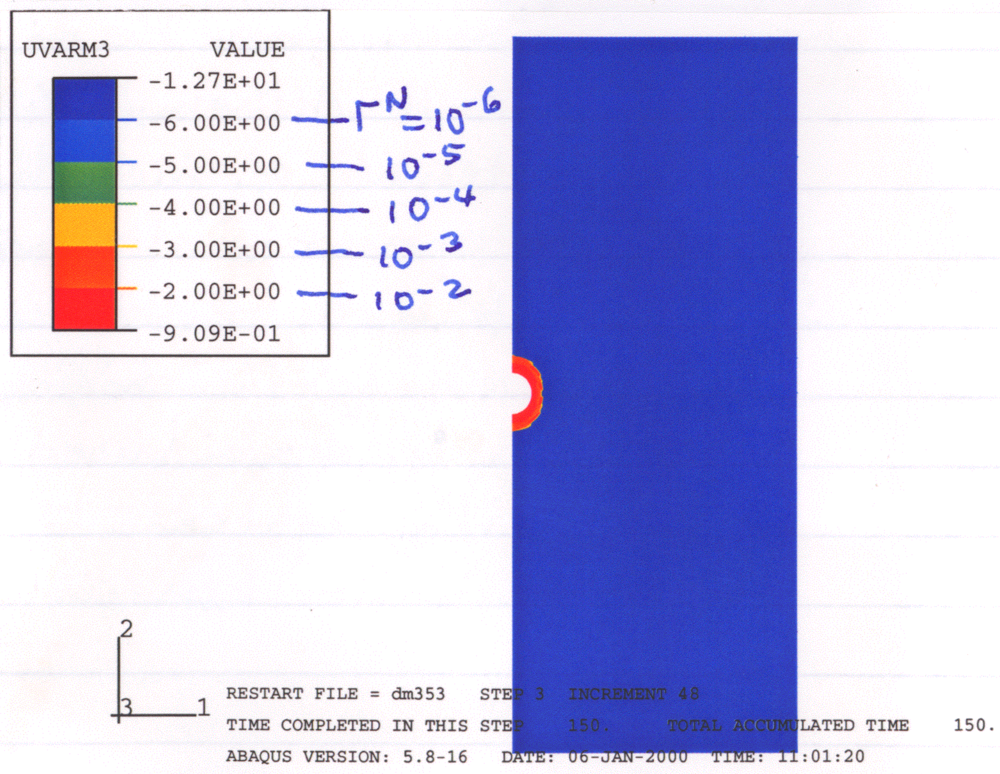
GW

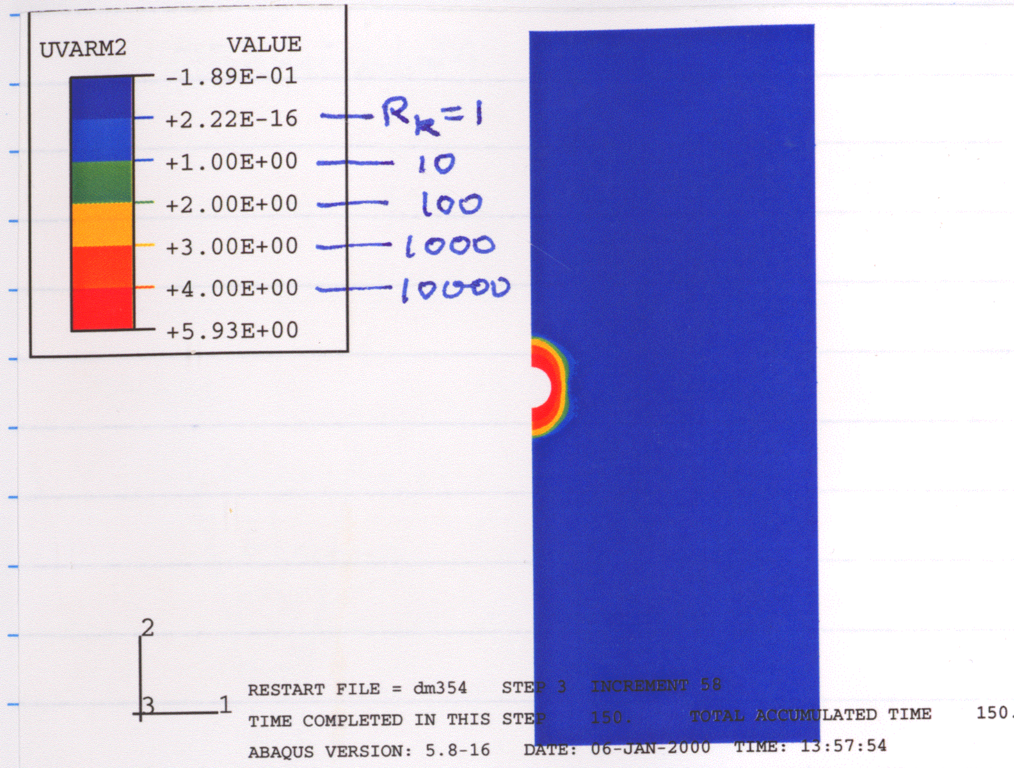
Inelastic response confined to a zone of about one drift radius thick from the drift wall. Although the intensity of inelastic response is high (indicating potentially unstable openings) the drifts can be stabilized with systematic reinforcement.

GW

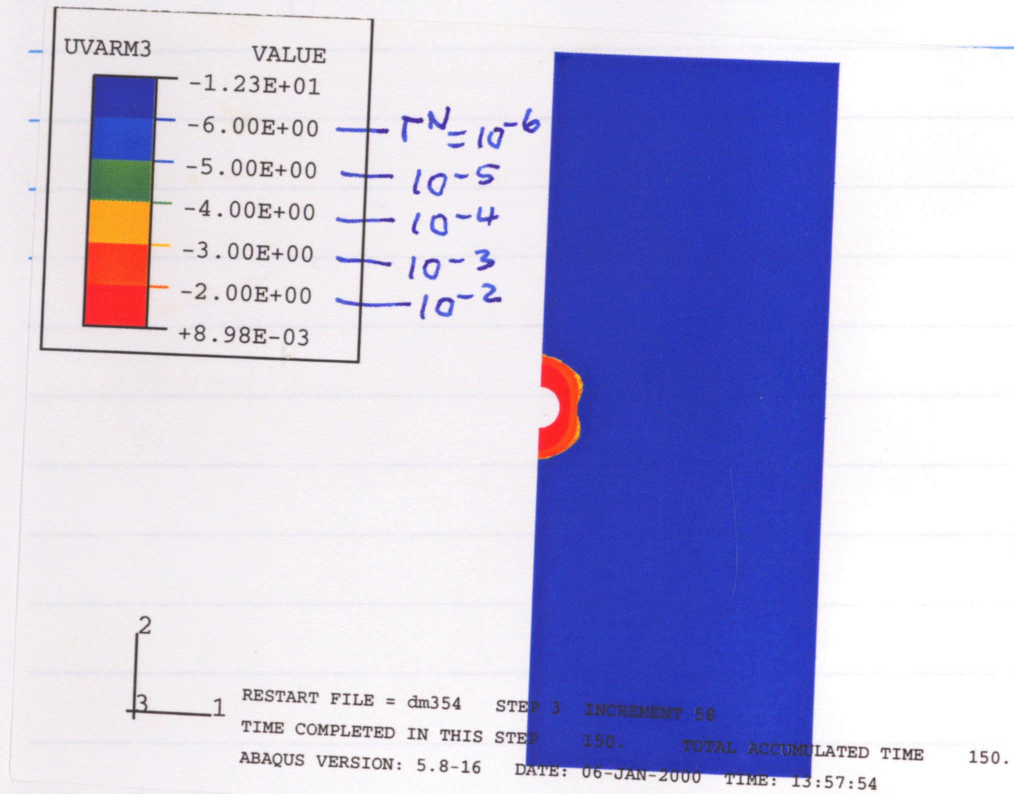


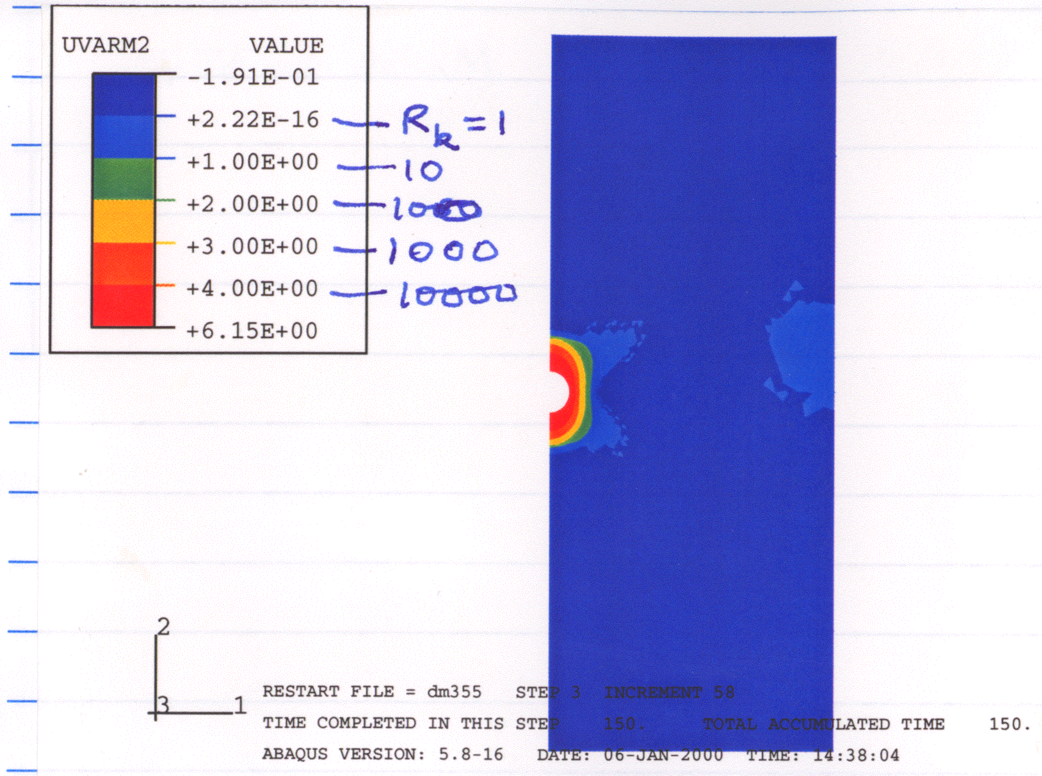
Model dm353

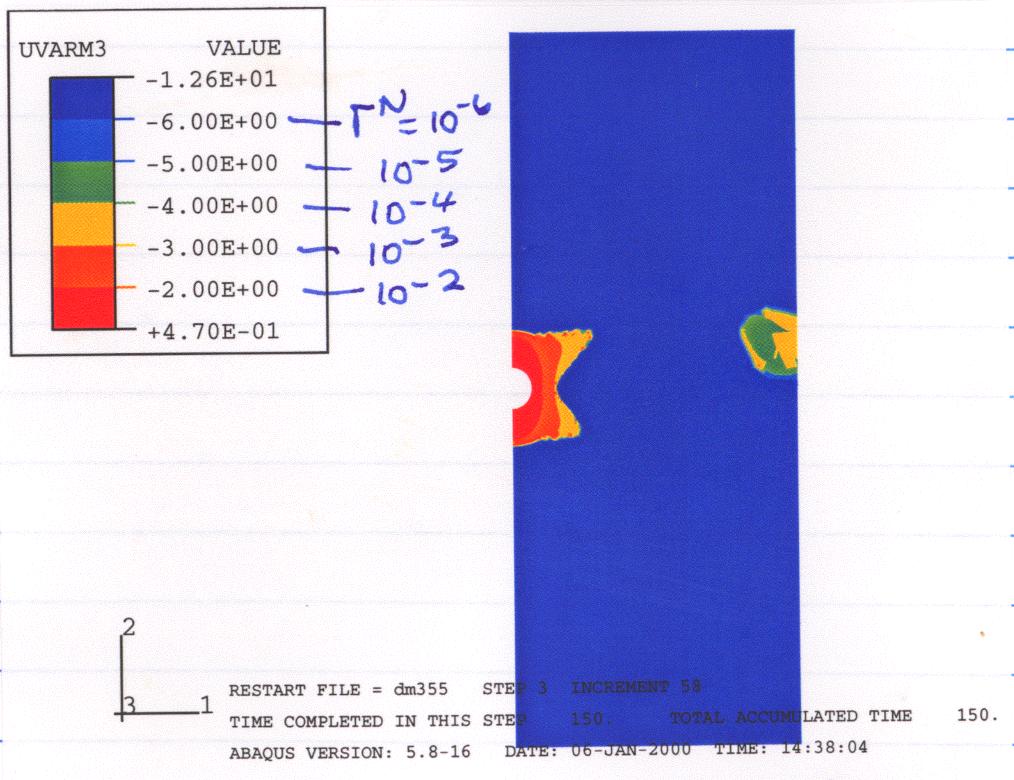




Model dm354







Model dm355

Pages 64-68 entries by GW 1/5/2000

Pages 69-82 entries by GW 1/17/2000

January 17 2000

Develop Models to Evaluate Results from DOE's Ventillation Model

DOE Publication:

Ventillation Model ← title
Hang Yang ← author
Office of Civilian Radioactive Waste Management; Analysis Model Report
ANL-EBS-MD-000030 Rev 00, Nov 9 1999.

The report presents analysis of two ventillation options: Air flow rate of 10 m^3/s or 15 m^3/s for 200 yr. The results are summarized on p. 70-71, which show profiles of driftwall and air temperature at different times.

Evaluation Strategy

Using the DOE-calculated air temperatures and the specified air-flow rates and the thermodynamic properties of air, calculate the rates of heat removal owing to ventillation. Then subtract the ~~net~~ ventillation-removed heat from the waste-package generated heat

(Contd on p. 72)

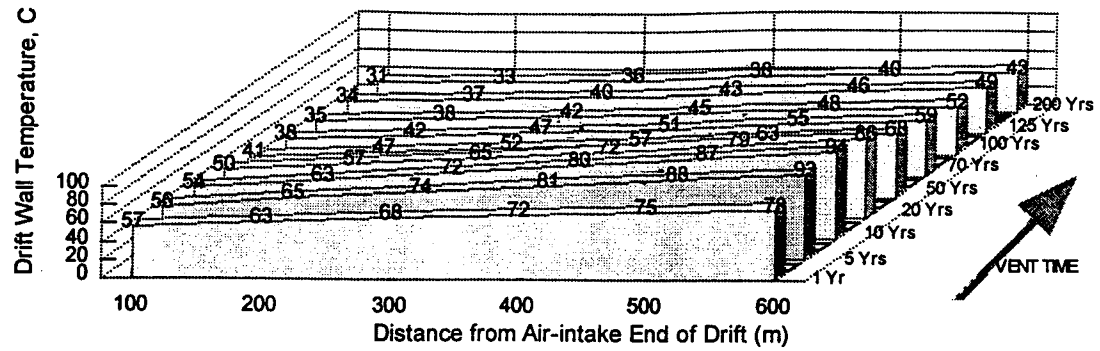


Figure 4 Wall Temperature During Continuous Ventilation at 10 cms.

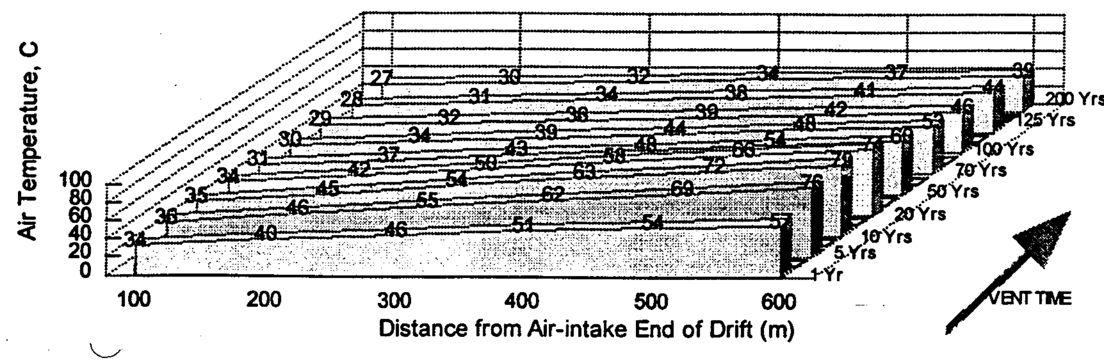


Figure 5 Air Temperature During Continuous Ventilation at 10 cms

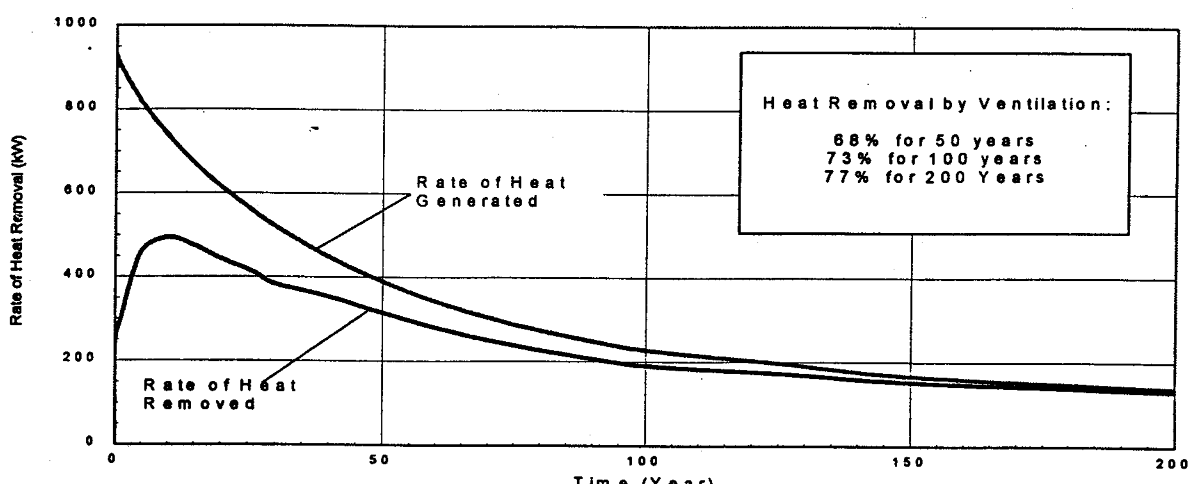


Figure 6 Heat Removed During Continuous Ventilation at 10 ~~cms~~ m³/s

GWO 1/17/2000

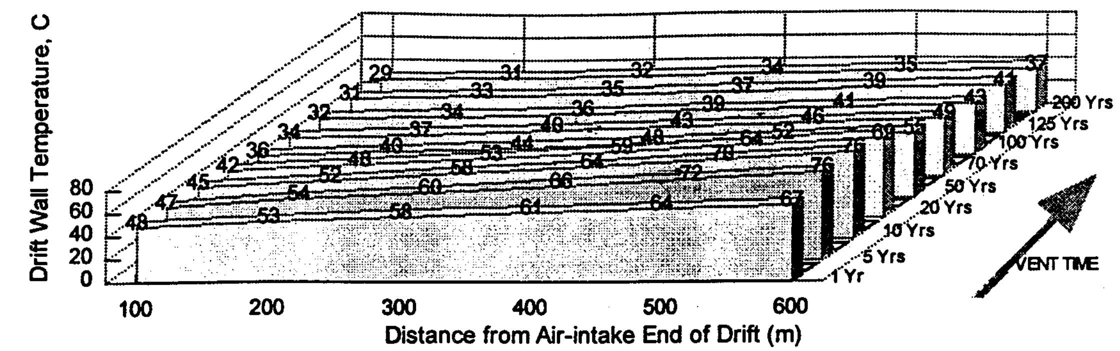


Figure 7 Wall Temperature During Continuous Ventilation at 15 cms

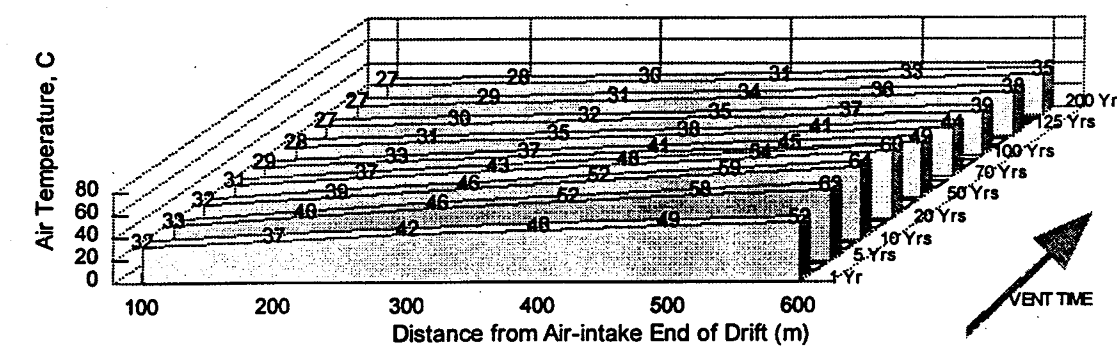


Figure 8 Air Temperature During Continuous Ventilation at 15 cms

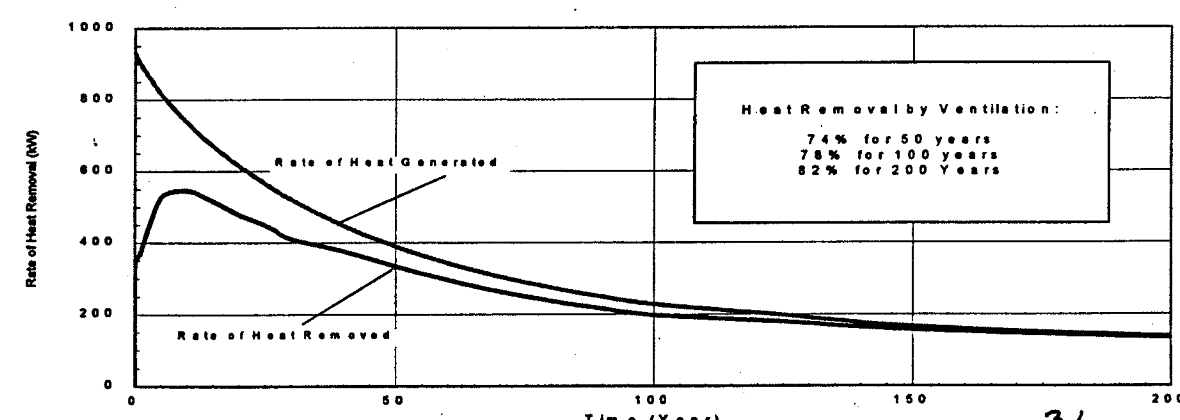


Figure 9 Heat Removed During Continuous Ventilation at 15 ~~cms~~ m³/s

GWO 1/17/00

and apply the result in a heat-conduction analysis to determine the driftwall-temperature history. Compare the resulting driftwall temperature with the DOE-calculated driftwall temperature.

The required thermodynamic properties of air are specific heat C_p and density ρ at constant pressure. Both were obtained from the table on p. 73, which was copied from: "Convective Heat and Mass Transfer, by W. M. Kays and M. E. Crawford. 2d. edition. McGraw-Hill, Inc., 1980.

The heat removed by ventillation, E_v, is given by

$$E_v = m_v C_p \Delta T$$

C_p = specific heat of air J/(kg.K)
m_v = mass flow rate of air = kg/s
ΔT = Temperature change of the air from an entry point to an exit point

$$m_v = V_v \rho_a$$

V_v = ventillation flow rate (m^3/s)
ρ_a = density of air in kg/m^3

With these units, E_v comes out in the unit

388 CONVECTIVE HEAT AND MASS TRANSFER

Table A-1 Properties of air at $P = 101.325$ kPa
$\mathfrak{M} = 28.966$

°K = 273.15 + °C (GWO 1/17/2000)

Information potentially subject to copyright protection was redacted from this location. The redacted material is from a book whose title is listed above.

of J/s; the resulting value of E_v is divided by the drift length over which the temperature change ΔT is measured, which gives the linear ~~rate~~ heat-loss (GWO 1/17/00) rate owing to ventillation in units of J/s/m or W/m.

Note:

(1) Both C_p and ρ_a vary with temperature

(2) For the DOE results, air temperature varies linearly over each 100-m long drift segment. The results are available over 500 m of drift, from the 100 m point to the 600 m point.

(3) The calculation of E_v ~~at~~ for a given time (GWO 1/17/2000) is performed for each of 5 drift segments; the result is accumulated, and divided by 500 to obtain the linear heat-loss rate.

The calculation is implemented through

HeatLoss.cpp

Page 1 of 4

```
/*********************************************************************************
HeatLoss

This code calculates the rate of heat removal by ventillatior
using the values of inlet and outlet air temperatures
and ventillation flow rate specified in DOE's ventillatior
model (Document # ANL-EBS-MD-000030-REV. 00 of November 1999).
Air temperature is assumed to vary linearly over each 100-m drift segment.

Modify the values of the following hard-coded variables as necessary:

tAirFile
airPropFile
outFile
flowRate
driftDiameter
numAirDataPoints
timeVal

Author:                    G. I. Ofoegbu
Date:                      January 200C
System:                    C++ Compiler
                           (Code developed using Borland C++Builder 4.0
*********************************************************************************/

#include <stdio>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <strstrea>
#include <conio>
#include <stdlib>
#include <math>
#include <string>

#pragma hdrstop
#include <condefs>

using namespace std;

char* messageBuffer;
void DumpAndQuit();

int ERROREND = 1;
int NORMALEND = 0;

struct AirProp{
  bool init;
  int numData;
  float *temp;
  float *dens;
  float *sHeat;
  AirProp(int n){
    numData = n;
    temp = new float[n];
    dens = new float[n];
    sHeat = new float[n];
    init = true;
    if (!temp || !dens || !sHeat)
      init = false;
  };
  ~AirProp(){
    if (temp) delete[] temp;
    if (dens) delete[] dens;
    if (sHeat) delete[] sHeat;
  };
};

float LinearInterp(const float* x, const float* fOfX, float atX, int n);
int GetAirProps(AirProp* pAir, char* airPropFile);
int GetTAir(char* tAirFile, float* temp, int numData);

//--------------------------------------------------------------------------------
#pragma argsused
int main(int argc, char* argv[])
{

  int timeVal[] = {1,5,10,20,50,70,100,125};
  int numTimeVal = 8;
  float flowRate = 10.0;
  char* airPropFile = "d:\\Ventillation\\Data\\airProps.dat";
  int numAirDataPoints = 12;
  float ductLength = 500.0;
//
// Name of the input file for air temperature (tAirFile) varies witl
// the ventillation rate RR and the year YYY as follows
// Also, the name of the output file (outFile) varies with RF
//
```

HeatLoss.cpp

Page 2 of 4

```
  char* tAirFile = "d:\\Ventillation\\DOEResults\\taRRYYY.out";
  char* outFile = "d:\\Ventillation\\VLCheck\\vLRR.out";

// Allocate memory for possible error messages

  messageBuffer = new char [151];
  if (!messageBuffer){
    cerr << "Memory allocation error for messageBuffer\n";
    cerr << "Press any key to end: ";
    getch();
    return (1);
  }

// Read Air Property data

  AirProp* pAir = new AirProp(numAirDataPoints);
  if (!pAir || !(pAir->init)){
    sprintf(messageBuffer,"Memory allocation failure on pAir");
    delete pAir;
    DumpAndQuit();
  }
  if (GetAirProps(pAir, airPropFile) == ERROREND){
    delete pAir;
    DumpAndQuit();
  }

// Open output file and print heading informatior

  int vRate = flowRate;
  sprintf(outFile,"d:\\Ventillation\\VLCheck\\vL%02d.out",vRate);
  ofstream Fout(outFile);
  if (!Fout){
    sprintf(messageBuffer,"Unable to open file %s",outFile);
    delete pAir;
    DumpAndQuit();
  }
  Fout << "Consistent Heat Removal Rate\n"
       << "For DOE's Ventillation Case with Flow Rate = "
       << setiosflags(ios::fixed) << setprecision(1)
       << flowRate << " m^3/s" << resetiosflags(ios::fixed)
       << endl;
  Fout << setw(15) << "Time (yr)"
       << setw(17) << "HeatLoss (kW/m)"
       << endl;

  float temp[7];
  int numTemp = 7;
  for (int i=0; i<numTimeVal; i++){
    float year = timeVal[i];
    sprintf(tAirFile,"d:\\Ventillation\\DOEResults\\ta%02d%03d.out",
      vRate,timeVal[i]);
    if (GetTAir(tAirFile,temp,numTemp) == ERROREND){
      delete pAir;
      DumpAndQuit();
    }
    float kWLoss = 0.0;
    for (int j=2; j<numTemp; j++){
      float tAvg = 0.5*(temp[j-1] + temp[j]);
      float tDelta = temp[j] - temp[j-1];
      float sHeat = LinearInterp(pAir->temp,pAir->sHeat,tAvg,pAir->numData);
      float dens = LinearInterp(pAir->temp,pAir->dens,tAvg,pAir->numData);
      kWLoss += (sHeat*tDelta*flowRate*dens);
    }
//
// There is no information for year=0; apply year=1 ventillation loss
// (which is at i=0 position) to year=C
//
    if (i==0)
      Fout << setiosflags(ios::fixed) << setprecision(1)
           << setw(15) << "0.0"
           << setprecision(4)
           << setw(15) << kWLoss/ductLength
           << endl;
    Fout << setiosflags(ios::fixed) << setprecision(1)
         << setw(15) << year
         << setprecision(4)
         << setw(15) << kWLoss/ductLength
         << endl;
  }

// Done

  delete pAir;
  delete[] messageBuffer;
  cout << "\nDone .. press any key to end:";
  getch();
  return 0;
}
```

the computer code "HeatLoss", which is documented on p. 74 and 75. The results of the calculation are given ~~in as~~ in terms of ~~in tab~~ tables of heat-loss rate versus time, one table for each ventillation case, on p. 75 (top right). GW 1/17/00

The resulting ventillation loss is subtracted from the waste-package heat generation rate to obtain a net

HeatLoss.cpp

Page 3 of 4

```
//
// Function LinearInterp evaluates the relationship f(t) at t=atTemp
// using linear-polynomial assumption for f(t). Calling program provides
// pointers to arrays of t and f values that define the f(t) relationship.
// Function LinearInterp assumes that the t array gives values of t ir
// increasing order and that no two t values are equal.
//

float LinearInterp(const float* t, const float* f, float atTemp, int n)
{

  if (atTemp <= t[0])
    return(f[0]);

  if (atTemp >= t[n-1])
    return(f[n-1]);

  for (int i=1; i<n; i++)
    if (atTemp <= t[i])
      return(
        f[i-1] + (f[i] - f[i-1])*(atTemp - t[i-1])/(t[i] - t[i-1])
      );

  return(f[n-1]); // This line is never reached; it was added to avoid
                  // Borland-Compiler warning
}

int GetAirProps(AirProp* pAir, char* airPropFile)
{
  ifstream Fin(airPropFile);
  if (!Fin){
    sprintf(messageBuffer,"Unable to open file %s",airPropFile);
    return (ERROREND);
  }

  int numData = pAir->numData;
  float temp,dens,sHeat;
  char buf[100];
  size_t nBuf = sizeof(buf);
  int nn = 0;
  while (!Fin.eof()){
    Fin.getline(buf,nBuf);
    istrstream inpLine(buf,nBuf);
    inpLine >> temp >> dens >> sHeat;
    if (inpLine.good()){
      pAir->temp[nn] = temp;
      pAir->dens[nn] = dens;
      pAir->sHeat[nn] = sHeat;
      if ((++nn) == numData)
        break;
    }
  }
  if (nn < numData){
    sprintf(messageBuffer,"Only %d of %d air property data found in file %s",
      nn,numData,airPropFile);
    return (ERROREND);
  }
  return (NORMALEND);
}

int GetTAir(char* tAirFile, float* temp, int numData)
{
  ifstream Fin(tAirFile);
  if (!Fin){
    sprintf(messageBuffer,"Unable to open file %s",tAirFile);
    return (ERROREND);
  }

  char buf[100];
  size_t nBuf = sizeof(buf);
  float d,t;
  int numFound = 0;
  while (!Fin.eof()){
    Fin.getline(buf,nBuf);
    istrstream inpLine(buf,nBuf);
    inpLine >> d >> t;
    if (inpLine.good()){
      temp[numFound] = t;
      if ((++numFound) == numData)
        break;
    }
  }
  if (numFound < numData){
    sprintf(messageBuffer,"Only %d of %d temperatures found in file %s",
      numFound,numData,tAirFile);
    return (ERROREND);
  }
```

HeatLoss.cpp

Page 4 of 4

```
  return (NORMALEND);
}

void DumpAndQuit()
{
  cerr << "\n" << messageBuffer << "\n";
  delete[] messageBuffer;
  cerr << "Press any key to end: ";
  getch();
  exit(1);
}
```

linear heat source. The steps are as follows:

Consistent Heat Removal Rate
For DOE's Ventillation Case with Flow Rate = 10.0 m^3/s

Time (yr)	HeatLoss (kW/m)
0.0	0.5130
1.0	0.5130
5.0	0.8650
10.0	0.9489
20.0	0.8703
50.0	0.6469
70.0	0.5193
100.0	0.3886
125.0	0.3675

Consistent Heat Removal Rate
For DOE's Ventillation Case with Flow Rate = 15.0 m^3/s

Time (yr)	HeatLoss (kW/m)
0.0	0.6763
1.0	0.6763
5.0	0.9645
10.0	1.0629
20.0	0.9704
50.0	0.6827
70.0	0.5513
100.0	0.4175
125.0	0.3833

(1) First, the heat-source history on p. 13 of Notebook #263 is normalized (divided by 71.499486) to obtain a relative decay ~~decay curve~~ history, which is ~~then~~ multiplied by 1.55 (kW/m) to obtain a ~~the~~ heat decay history. The DOE ventillation report was based on a maximum heat generation rate of 1.55 kW/m, but the decay history is not specified. GW 1/17/00

(2) Second, the ventillation heat-loss results are

interpolated to time values on the heat generation decay curves ~~and~~ so the net heat waste-package heat source is determined by subtraction.

The calculation of net heat source was implemented through the computer code "DrftSrc2001", which is documented on p. 77-78. The calculation can be summarized as follows:

Waste-generated heat E_{WP}:

$$E_{WP} = 1.55\, f_d(t) \qquad 1 \geq f_d(t) > 0$$

where $f_d(t)$ is the time decay function and 1.55 kW/m is the linear heat generation rate per unit length of drift at zero time.

Then the net heat source ~~is given~~ E_{NET} is given by

GIO 1/17/00

$$E_{NET} = E_{WP} - E_{VL}$$

where $E_{VL} = E_V / \Delta L$ where ΔL is the drift length over which the ventilation loss E_V is calculated ($\Delta L = 500$ m for the DOE ventilation results on p 70-71).

GIO 1/17/00

DrftSrc2001.cpp Page 1 of 4

```
/**************************************************************************
DrftSrc2001

This code calculates the heat source per unit volume of emplacement
drift for the EDA-II thermal loading option: 60 MTU/acre with 81 n
(center-to-center) drift spacing.

The input data are:

 1.          Drift geometry
 2.          Initial heat output per unit length of drift
 3.          Ventillation loss per unit length of drift (decays with time;
 4.          Heat source decay function (Decays from initial value of 1.0;

The calculated source-strength history
(J/yr/m^3) is output following ABAQUS input format.

Modify values of the following variables as necessary

maxVTime
airFlowRate
maxDSrc

Author:                          G. I. Ofoegbu
Date:                            January 200C
System:                          C++ compilez
                                 (Code developed using Borland C++ Builder 4;
*************************************************************************/

#include <stdio>
#include <stdlib>
#include <iomanip>
#include <fstream>
#include <strstrea>
#include <math>
#include <string>
#include <conio>

#pragma hdrstop
#include <condefs>

using namespace std;

char* messageBuffer;
void DumpAndQuit();

int ERROREND = 1;
int NORMALEND = 0;

int GetVentillationData(float* vTime, float* vLoss, int num, char* fileName);
float LinearInterp(const float* x, const float* fOfX, float atX, int n);

//------------------------------------------------------------------------
#pragma argsused
int main(int argc, char **argv)
{
  bool ventillated = true;
  int numVLossData = 9;
  float* vTime = 0;
  float* vLoss = 0;
  float maxVTime = 125.0;
  float airFlowRate = 15.0;
  float maxDSrc = 1550.0; //Heat source per drift length (W/m) at time zerc
  float driftHeight = 5.5;
  float secPerYear = 365.25*24.0*60.0*60.0;
  char* inFileName = "d:\\Ventillation\\allWPSrc.txt";
//
// The name of input file (vLossFile) and the output files (outFileName anc
// plotFileName) vary with the ventillation flow rate RR and ventillatior
// time TTT as follows
//
  char* vLossFile = "d:\\Ventillation\\VLCheck\\vLRR.out";
  char* outFileName = "d:\\Ventillation\\DrftScale\\srcRRTTT.def";
  char* plotFileName = "d:\\Ventillation\\VLCheck\\srcRRTTT.txt";
  int tMax = maxVTime;
  int fRate = airFlowRate;
  sprintf(vLossFile,"d:\\Ventillation\\VLCheck\\vL%02d.out",fRate);
  sprintf(outFileName,"d:\\Ventillation\\DrftScale\\src%02d%03d.def",
    fRate,tMax);
  sprintf(plotFileName,"d:\\Ventillation\\VLCheck\\src%02d%03d.txt",
    fRate,tMax);

  if (ventillated){
    vTime = new float[numVLossData];
    vLoss = new float[numVLossData];
    if (!vTime || !vLoss){
      sprintf(messageBuffer,"Memory allocation failure");
      DumpAndQuit();
```

DrftSrc2001.cpp Page 2 of 4

```
    }
    if (GetVentillationData(vTime,vLoss,numVLossData,vLossFile) == ERROREND){
    delete[] vTime;
    delete[] vLoss;
    DumpAndQuit();
    }
  }

  ifstream Fin(inFileName);
  if (!Fin){
    sprintf(messageBuffer, "Unable to open file %s",inFileName);
    delete[] vTime;
    delete[] vLoss;
    DumpAndQuit();
  }
  ofstream Fout(outFileName);
  if (!Fout){
    sprintf(messageBuffer, "Unable to open file %s",outFileName);
    delete[] vTime;
    delete[] vLoss;
    DumpAndQuit();
  }
  ofstream Fplot(plotFileName);
  if (!Fplot){
    sprintf(messageBuffer, "Unable to open file %s",plotFileName);
    delete[] vTime;
    delete[] vLoss;
    DumpAndQuit();
  }

  Fout << "***\n";
  Fout << "*** Heat source strength function for a "
       << setiosflags(ios::fixed) << setprecision(1)
       << driftHeight
       << "-m diameter drift"
       << resetiosflags(ios::fixed)
       << endl;
  if (ventillated)
    Fout << "*** Heat source data INCLUDES "
         << setiosflags(ios::fixed) << setprecision(0)
         << maxVTime
         << " yr of VENTILLATION LOSS at "
         << airFlowRate
         << resetiosflags(ios::fixed)
         << " m^3/s\n";
  else
    Fout << "*** Heat source data DOES NOT INCLUDE VENTILLATION LOSS\n";
  Fout << "***\n";
  Fout << "*** t1,q1,t2,q2 ... etc\n";
  Fout << "*** t = time in yr; q = Joules/m^3/yr\n";
  Fout << "***\n";

  Fplot << setw(12) << "Year"
        << setw(12) << "WPSrc kW/m"
        << setw(12) << "vLoss kW/m"
        << setw(15) << "netSrc kW/m"
        << endl;

  char buf[151];
  size_t nBuf = sizeof(buf);
  int onThisLine = 0;
  float year,wpQ;
  float maxWPQ = 71.499486;  //Max value of wpQ (at time zero)
  float wpSrc,lostSrc,netSrc;
  float zeroTime = 2.0e-6;
  float pi = 4.0*atan(1.0);
  float xSecArea = driftHeight*driftHeight*(pi/4.0);
  float kiloWatt = 1000.;
  while (Fin){
    Fin.getline(buf,150);
    istrstream inpLine(buf,nBuf);
    inpLine >> year >> wpQ;
    if (inpLine.good()){
      wpSrc = maxDSrc*(wpQ/maxWPQ)*secPerYear/xSecArea;
      lostSrc = 0.0;
      if (ventillated && !(year > maxVTime))
        lostSrc = LinearInterp(vTime,vLoss,year,numVLossData);
      lostSrc *= (kiloWatt*secPerYear/xSecArea);
      netSrc = wpSrc - lostSrc;
      if (netSrc < 0.0) netSrc = 0.0;
      Fplot << setiosflags(ios::fixed) << setprecision(1)
            << setw(12) << year
            << setprecision(3)
            << setw(12) << (xSecArea*wpSrc/secPerYear/kiloWatt)
            << setw(12) << (xSecArea*lostSrc/secPerYear/kiloWatt)
            << setw(12) << (xSecArea*netSrc/secPerYear/kiloWatt)
            << endl;
      if (onThisLine == 3){
```

The ~~k~~ resulting histories of E_{WP}, E_{VL}, and E_{NET} are shown on p. 79. The results indicate that the resulting E_{VL} (estimated based on the DOE-calculated air temperatures) exceed the E_{WP} between 100-125 yr for the 10 m^3/s flow-rate case and between 40-125 yr for the 15 m^3/s case.

DrftSrc2001.cpp Page 3 of 4

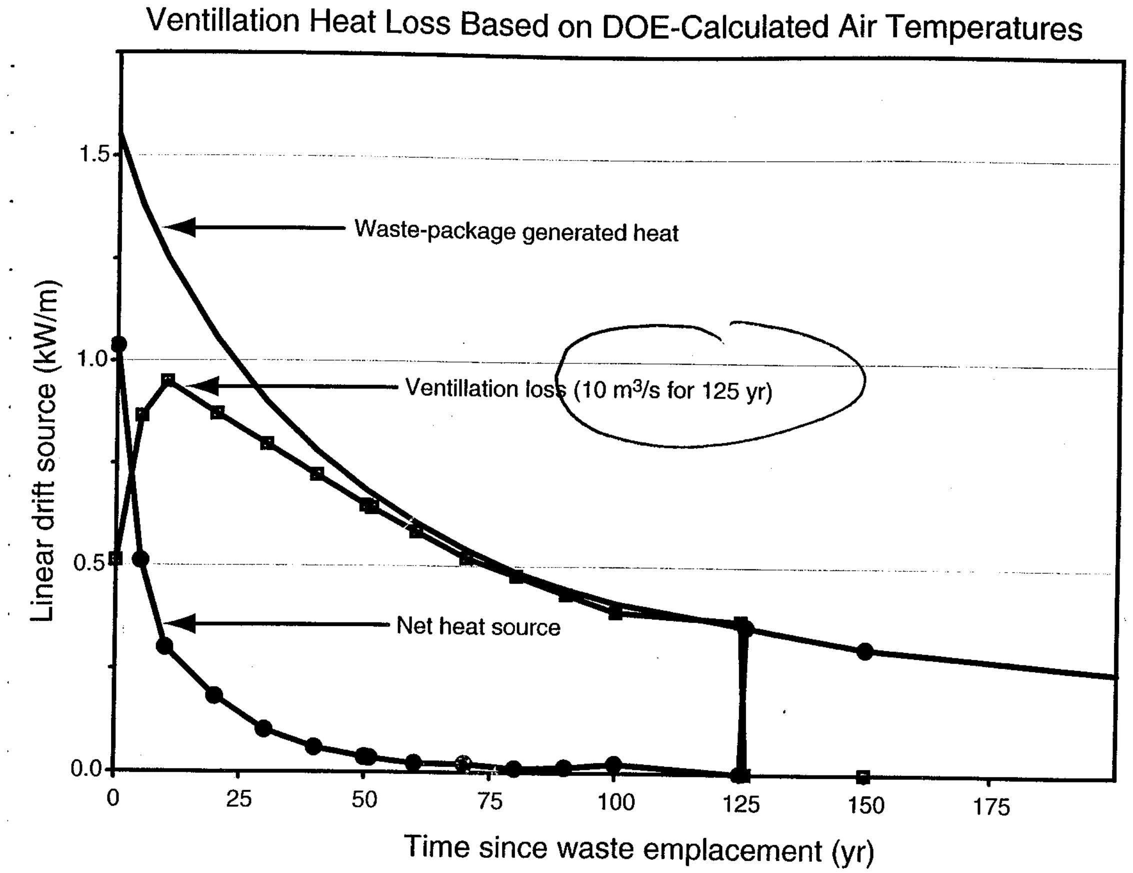
```
      Fout << setiosflags(ios::fixed) << setprecision(3)
           << setw(12) << (year + zeroTime) << ","
           << resetiosflags(ios::fixed)
           << setiosflags(ios::scientific)
           << setprecision(5)
           << setw(12) << netSrc
           << resetiosflags(ios::scientific)
           << endl;
      onThisLine = 0;
    }
    else{
      Fout << setiosflags(ios::scientific) << setprecision(3)
           << setw(12) << (year + zeroTime) << ","
           << resetiosflags(ios::fixed)
           << setiosflags(ios::scientific)
           << setprecision(5)
           << setw(12) << netSrc << ","
           << resetiosflags(ios::scientific);
      onThisLine++;
    }
    if (year < zeroTime){  // Keep flux constant thru first year
      Fout << setiosflags(ios::scientific) << setprecision(2)
           << setw(12) << (year + 1.0 + zeroTime) << ","
           << resetiosflags(ios::fixed)
           << setiosflags(ios::scientific)
           << setprecision(5)
           << setw(12) << netSrc << ","
           << resetiosflags(ios::scientific);
      onThisLine++;
    }
   }
  }

// Done

  delete[] vTime;
  delete[] vLoss;
  cout << "Done ... Press return to end: ";
  getch();
  return 0;
}

int GetVentillationData(float* vTime, float* vLoss, int numData,
  char* vLossFile)
{
  ifstream Fin(vLossFile);
  if (!Fin){
    sprintf(messageBuffer,"Unable to open file %s",vLossFile);
    return (ERROREND);
  }

  float time,loss;
  char buf[100];
  size_t nBuf = sizeof(buf);
  int nn = 0;
  while (!Fin.eof()){
    Fin.getline(buf,nBuf);
    istrstream inpLine(buf,nBuf);
    inpLine >> time >> loss;
    if (inpLine.good()){
      vTime[nn] = time;
      vLoss[nn] = loss;
      if ((++nn) == numData)
        break;
    }
  }
  if (nn < numData){
    sprintf(messageBuffer,"Only %d of %d air property data found in file %s",
      nn,numData,vLossFile);
    return (ERROREND);
  }
  return (NORMALEND);
}

//
// Function LinearInterp evaluates the relationship f(t) at t=atTemp
// using linear-polynomial assumption for f(t). Calling program provides
// pointers to arrays of t and f values that define the f(t) relationship.
// Function LinearInterp assumes that the t array gives values of t in
// increasing order and that no two t values are equal.
//

float LinearInterp(const float* t, const float* f, float atTemp, int n)
{

  if (atTemp <= t[0])
    return(f[0]);
```

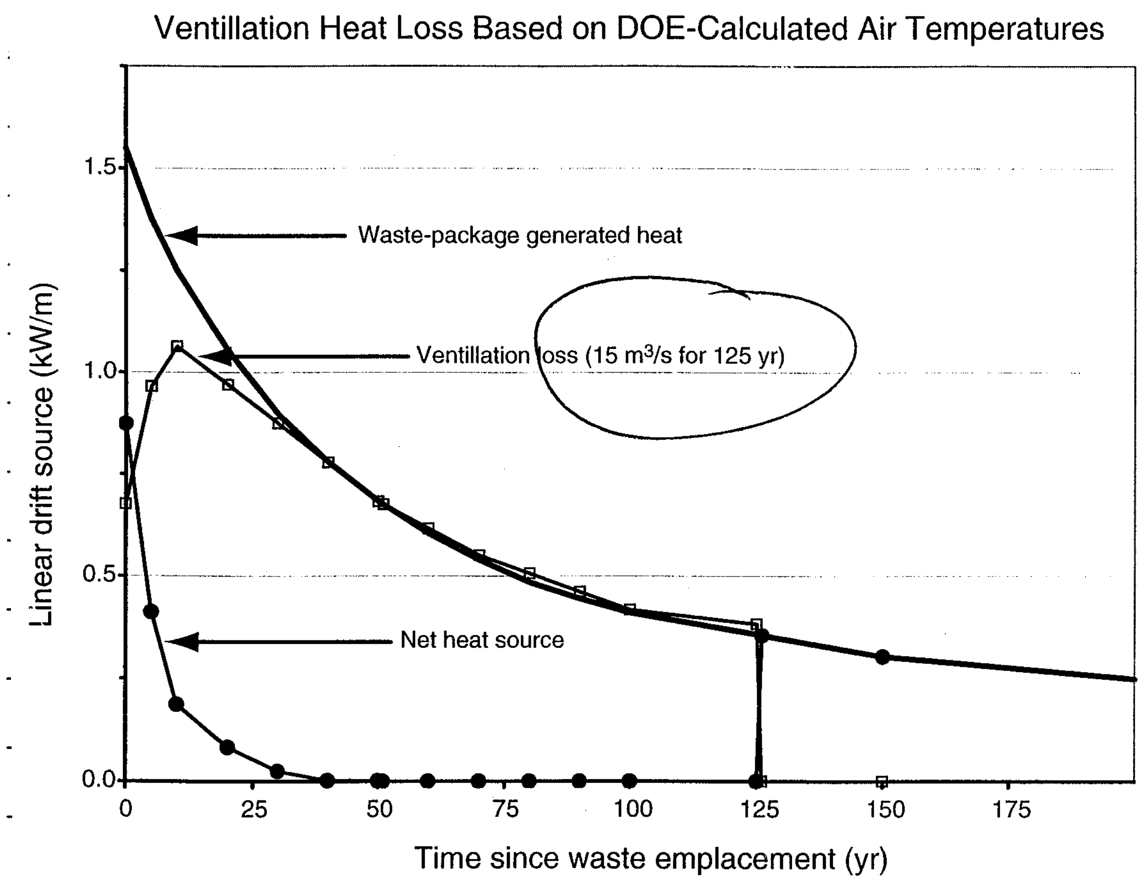
DrftSrc2001.cpp Page 4 of 4

```
  if (atTemp >= t[n-1])
    return(f[n-1]);

  for (int i=1; i<n; i++)
    if (atTemp <= t[i])
      return(
        f[i-1] + (f[i] - f[i-1])*(atTemp - t[i-1])/(t[i] - t[i-1])
      );

  return(f[n-1]); // This line is never reached; it was added to avoid
                  // Borland-Compiler warning
}

void DumpAndQuit()
{
  cerr << "\n" << messageBuffer << "\n";
  delete[] messageBuffer;
  cerr << "Press any key to end: ";
  getch();
  exit(1);
}
```

On the other hand, the DOE results ~~indicate~~ on p. 70–71 indicate that the heat removed by ventillation is always smaller than the generated heat for both ventillation cases. In other words, the heat loss calculated based on the reported air temperatures and the thermodynamic properties of air is not consistent with the heat loss reported by DOE. This result suggests that the air temperatures reported by DOE (p. 70–71) may not be correct. GWO 1/17/00

The net heat source E_{NET} (p. 79) was used as input heat source in ~~a~~ heat-conduction analyses to calculate drift-wall temperature histories. The finite-element model used for the heat conduction analyses is the ~~as~~ same as the model described on p. 48–52 ("Model Geometry"), except for





the following changes:

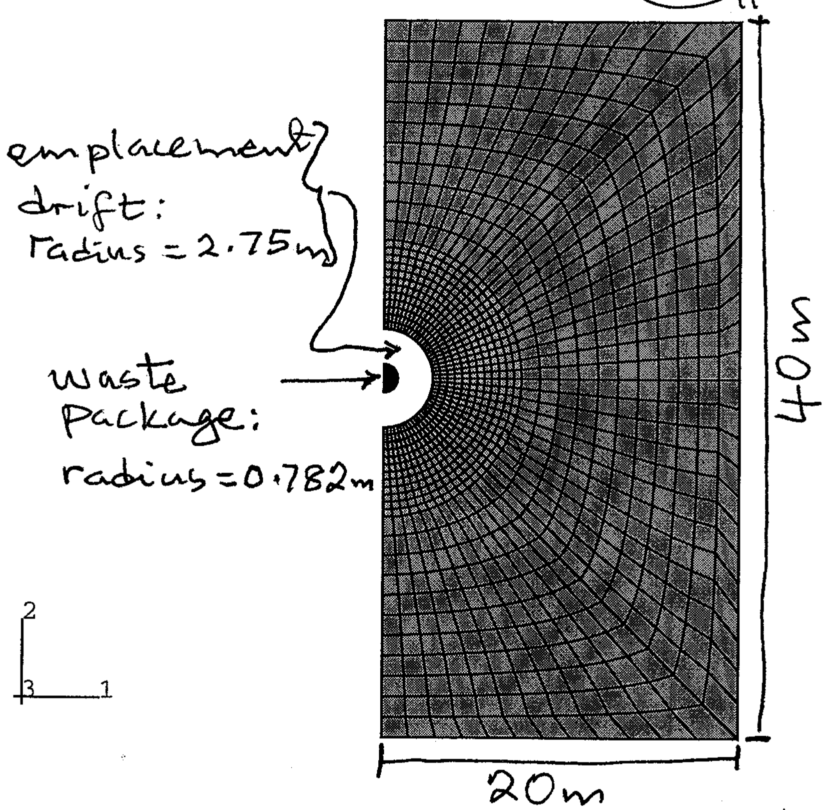
(1) The drift section (i.e., red zone on p. 50) is ~~was~~ taken out, ~~so the drift~~ was represented as an opening. GWO 1/17/00

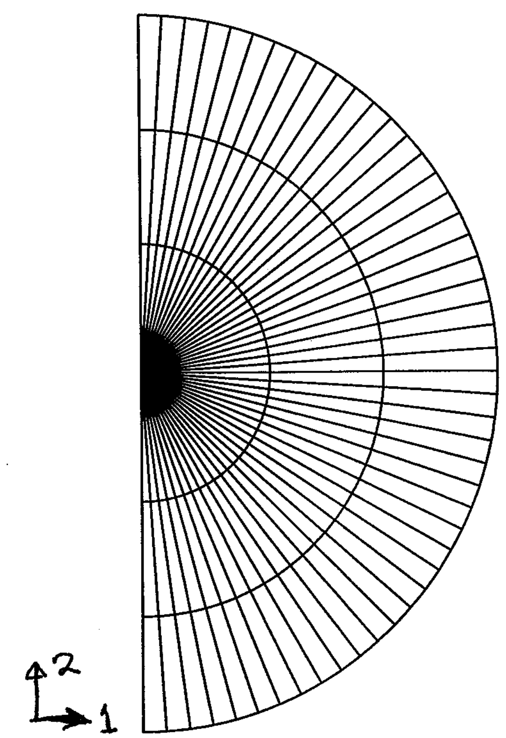
(2) The waste package is ~~made~~ represented explicitly as a (semi) circle of radius 0.782 m

(3) Thermal load is applied as a volumetric heat source in the waste package

(4) Radiation heat transfer between the waste package and the drift wall is accounted for explicitly.

The part of the model within a 20-by-40 m box enclosing the emplacement drift is shown below. ~~A blo~~ An enlarged view of the waste-package model is also shown. (CHO 11/17/00)

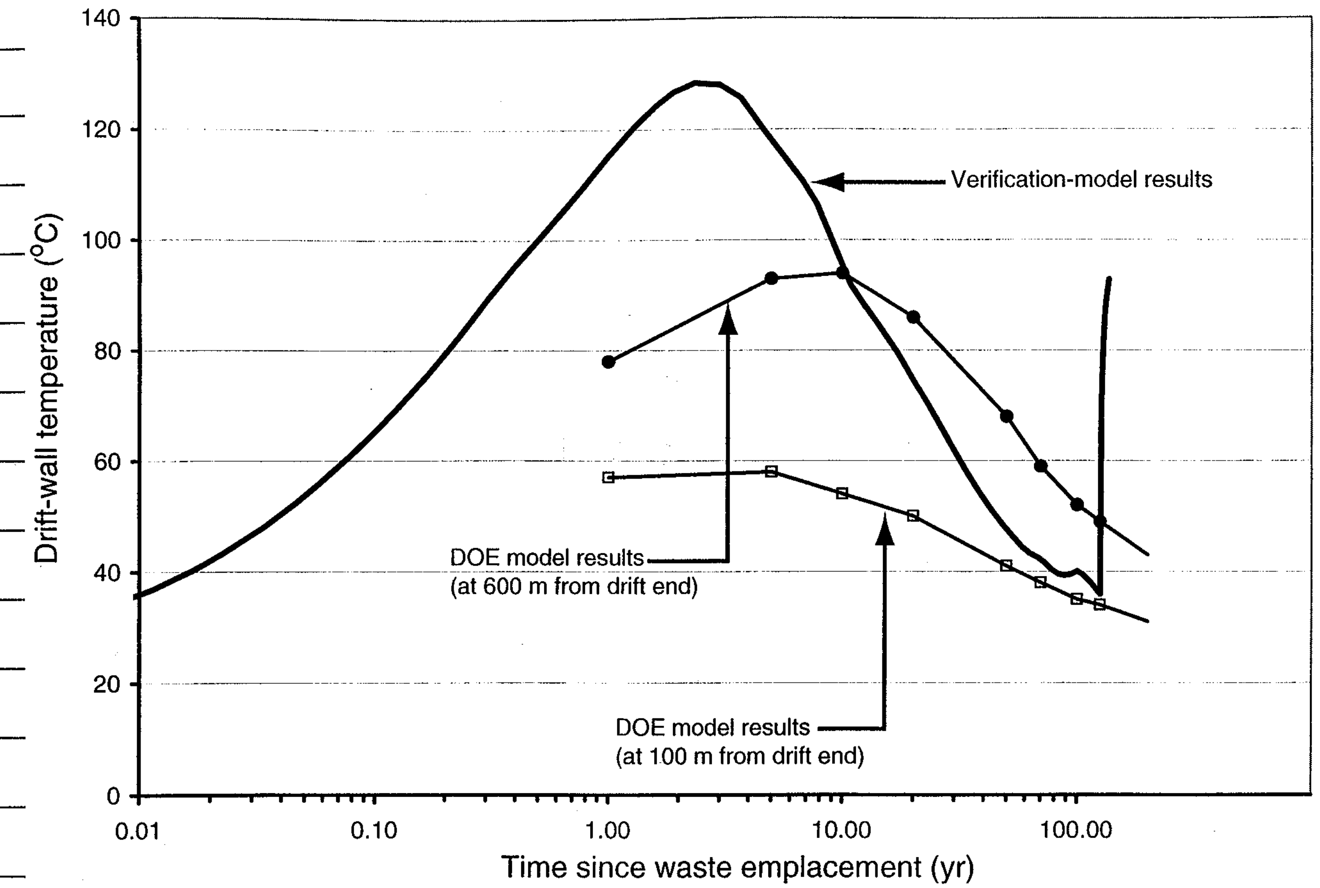




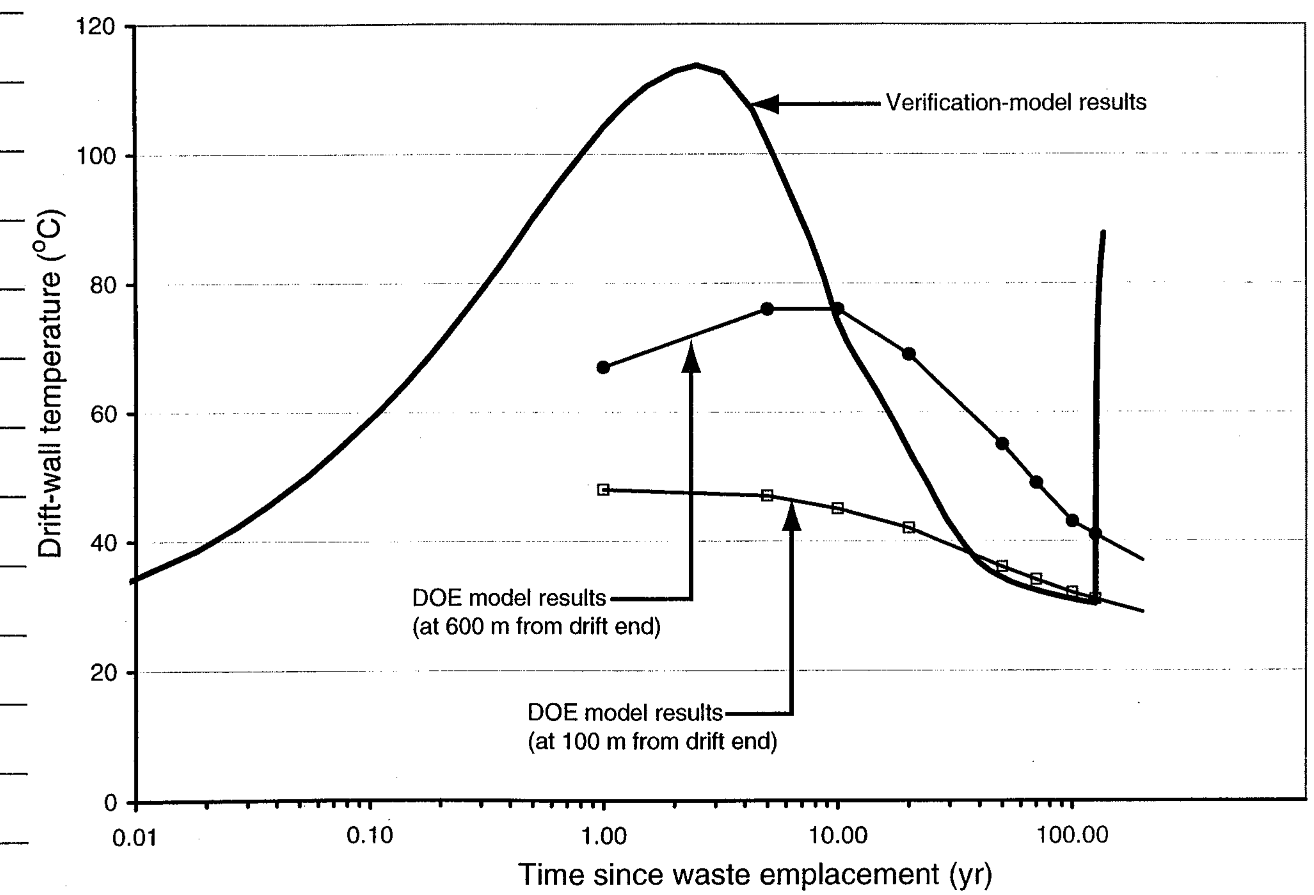
Waste-package model.

The results of the heat-conduction analyses are presented on p. 81: The curve labeled "Verification model results" represent the history of drift-wall temperature calculated in the analyses. The drift-wall temperature histories from the DOE model (p. 70-71) are also shown in the figures on p. 81. If the drift-wall temperatures and air temperatures from the DOE model were consistent the verification-model results would have matched the DOE model drift-wall temperature at 600 m from drift end.

Assessment of Results from DOE's Ventillation Model
(Case of 10 m^3/s ventillation for 125 yr)



Assessment of Results from DOE's Ventillation Model
(Case of 15 m^3/s ventillation for 125 yr)



Conclusion: Results from the DOE ventillation model (air temperatures and drift-wall temperatures) have not passed the consistency test described here. As a result, the DOE should re-examine the model and should ~~provide~~ verifications of the model or any future replacement.

GW 1/17/00

Pages 69-82 entries by GW 1/17/2000

August 23 2000 Pages 83-91 entries by GW 8/23/2000

Additional TM analyses conducted to explain the distributions of failure susceptibility obtained from previous TM analyses. These analyses are being conducted to respond to questions raised by Dr. M. Nataraja through the two letters documented on p.84-85.

A set of linear-elastic analyses are performed to examine the effects of Young's modulus on stress histories around (at points close to) the drift openings and pillars.

Temperature history obtained through thermal-analysis case dt301 (described on p.52). Input files for dt301 are located in the subdirectory

D:\ThermMech\NARMS2000\DrftScale.

dt301.inp	Analysis input file
d20Nodes.def	Node definitions
d20Nsets.def	Node-set definitions
dt301Elem.def	Elem and material definitions
d20InTem.def	Initial-temperature definitions
cDrftSrc.def	Heat source definitions

Contd on p. 86

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555-0001

GWD 8/23/2000

October 22, 1998

Dr. Asadul H. Chowdhury, Manager
Mining, Geotechnical, and Facility Engineering
Center for Nuclear Waste Regulatory Analyses
6220 Culebra Road, Bldg. 189
San Antonio, Texas 78238-5166

SUBJECT: REPOSITORY DESIGN AND THERMAL MECHANICAL EFFECTS (RDTME) KEY TECHNICAL ISSUE INTERMEDIATE MILESTONE NO. 20-1402-671-845: TM DRIFT STABILITY ANALYSIS AT REPOSITORY SCALE

Dear Dr. Chowdhury:

I have reviewed the subject report entitled, "Effects of Spatial and Temporal Variations of Rock-Mass Quality and Integrity of Support on Distributions of Potential Instability at the Proposed Yucca Mountain Repository for High-Level Nuclear Waste," submitted on September 24, 1998, as an Intermediate Milestone (IM). The report programmatically satisfies the requirements of this IM for the fiscal year; however, some technical points related to its contents were previously raised when I reviewed the abstract of the paper (based on this report). The abstract was submitted by the Center for the NRC review sometime back. You may recall that the same issues were also raised during our discussions when I visited the Center during a program review meeting in April 1998.

The major conclusion of the report raises some concerns. The report rightly states that the spatial distribution of potentially unstable areas is primarily controlled by rock-mass quality variations. This statement would lead the reader to expect that the areas of higher rock-mass quality are likely to be more stable than the area of lower (poorer) rock-mass quality. However, the report concludes that the "failure-related ground movements are more intense in areas of relatively high rock-mass quality because of higher induced thermal stresses caused by the greater stiffness of the rock-mass in such areas." This somewhat counter-intuitive conclusion has resulted from the premise of the analysis that there are well-defined relationships between rock-mass quality and stiffness properties of the rock, and rock-mass quality and strength properties of the rock-mass. This seems to be the beginning of the problem. Although the report attempts to explain the counterintuitive result ("unexpected relationship between rock-mass quality and potential for instability..."), it does not explain the validity and defensibility of using the empirical relationships as the premise for the analyses. If one examines Figure 2-6 of the report (relationships between rock-mass quality and friction angle, cohesion, and the Young's modulus), it becomes apparent that there is hardly any increase in the value of friction angle and a very unimpressive increase in cohesion, whereas, the value of Young's modulus increases quite dramatically, with increasing values of rock-mass quality. With cohesion and friction angle being the primary contributors to strength, it seems that the outcome of the analysis would be rather skewed, in that the empirical relationships are not truly representative of the strength parameters at higher rock-mass quality values. If one can argue that higher stiffness leads to more failure in rocks, one can also extend the argument to concrete roof

cc: W. Patrick
Dirs
EMs
MGFE staff

Dr. A.H. Chowdhury 2

support and conclude that it is better to have degraded (poor quality) concrete support than a high-quality (high-stiffness) concrete to support excavations (because, a higher quality/stiffness would result in higher stresses and, thus, lead to higher potential instability).

As discussed in our recent telephone conversation, I understand that you are planning on an independent external review of this report before sending it out for publication in the proceedings of the upcoming Rock Mechanics Conference. I consider this a good idea and encourage you to send the completed paper for an independent external review with particular focus on the issues raised in this letter. However, it is entirely up to the Center to follow up in any manner it deems appropriate.

If you or Mr. G. Ofoegbu have any questions on the contents of this letter, please contact me at (301) 415-6695 or via e-mail (msn1@nrc.gov).

Sincerely,
(orig signed by:)
Mysore S. Nataraja
Program Element Manager
Engineering and Geosciences Branch
Division of Waste Management
Office of Nuclear Material Safety
and Safeguards

GWD 8/23/2000

cc: J. Linehan
B. Meehan

Ticket #: C-980099
DISTRIBUTION:

Central File	NMSS r/f	ENGB r/f	NKStablein
CWReamer	SWastler	DDeMarco	BStiltenpole
DRom			

DOCUMENT NAME: S:\DWM\ENGB\MSN\C980099

OFC	ENGB	ENGB		
NAME	MNataraja/eb	RWeller		
DATE	10/21/98	10/22/98		

OFFICIAL RECORD COPY

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555-0001

1975 25 2000 years

August 11, 2000

SUBJECT: 718.4
PROJECT: 20.01402-671

GWD 8/23/2000

Asadul H. Chowdhury, Manager
Mining, Geotechnical, and Facility Engineering
Center for Nuclear Waste Regulatory Analyses
6220 Culebra Road P.O. Drawer 28510
San Antonio Texas 78228-5166

SUBJECT: RDTME KTI INTERMEDIATE MILESTONE NO. 20-01402.671.050: THERMAL-MECHANICAL EFFECTS ON REPOSITORY DESIGN/PERFORMANCE: DISCONTINUUM MODEL

Dear Dr. Chowdhury:

I have reviewed the Center for Nuclear Regulatory Analyses (CNWRA) report entitled: "Drift Stability and Ground Support Performance Under Thermal and Dynamic Load in Fractured Rock mass at Yucca Mountain Nevada." I concur with the change of title, which better reflects the contents of the report. I also concur with the decision to present the product in the form of a report rather than a conference/journal paper. The subject report documents the results of numerical modeling of rock mass behavior to study drift stability under thermal load, taking into account rock support provided by steel sets and reinforcing rock bolts. The report also documents conclusions on ground support performance subject to vibratory ground motion.

The fact that the effects of ventilation are not factored into the analyses presented in the subject report accounts for the overestimated temperatures and thermally induced stresses. To make the results applicable to pre-closure conditions, the next phase of this modeling exercise should account for the effects of ventilation on thermally induced stresses and drift stability. There are two conclusions in the report that stand out: (1) thermally induced stresses and deformation are greater in higher quality rock mass than in a lower quality rock mass; and (2) the existing experience on ground support design gained from the Exploratory Studies Facility and conventional underground mining and tunneling industry may not be applicable to ground support design under thermal load (particularly at high thermal loads and for higher quality rock mass).

The first conclusion is contrary to the common understanding that a lower quality rock mass would experience greater deformation than a higher quality rock mass under the same loading conditions. I have raised this point before when previous studies by the CNWRA came to the same conclusions. Based on my discussions with Dr. Simon Hsiung and Dr. Rui Chen of your staff, it is clear they understand my concerns, and they have assured me that the results are not an artifact of modeling assumptions or limitations. I would strongly recommend that some analytical verifications be done using simple closed form solutions to convince ourselves that, indeed, the results are not artifacts of numerical modeling. For example, thermal stresses due to a point/line heat source can be superimposed on the readily available solutions for stress distributions around a hole in an elastic plate. The results can be used to verify trends observed in the numerical studies.

cc: W. Patrick
Dirs
EMs
MGFE staff

A. Chowdhury 2

The second conclusion begs the question, "What should the U.S. Department of Energy (DOE) be doing, if neither the ESF experience nor the conventional mining and tunneling experience can be relied upon?" How should the findings of this report be used in reviewing the DOE designs of heated drifts (with respect to reinforcement and roof support of different quality rocks)?

I look forward to further discussions on this study with the author of the report and the other team members. If there are any additional comments on the subject report from other KTI teams, they will be communicated to you through informal discussions or e-mails. If you have any questions on the contents of this letter, please contact me at (301) 415-6695 or via e-mail (msn1@nrc.gov). No written response to this letter is required and the subject report is considered to fulfill the CNWRA's contractual obligations for this Intermediate Milestone.

Sincerely,

Mysore Nataraja
Program Element Manager
Repository Design Thermal-Mechanical
Effect KTI
Division of Waste Management
Office of Nuclear Material Safety
and Safeguards

cc: J. Linehan, PMDA
B. Meehan, ADM/DCPM/CMB2

GWD 8/23/2000

Values of Young's Modulus for Linear-elastic Analyses

Young's Modulus (GPa)	Rock Category
33.0	Intact rock
32.6	RMQ5 rock mass
7.8	RMQ1 rock mass

Input Files for Linear-elastic Analysis

GVG 8/23/9?

Intact Rock

dm500.inp	Analysis input file
d20Nodes.def	Node definitions
d20Nsets.def	Node_set definitions
dm500Elem.def	Element and material properties.
d20InTem.def	Initial temperature definitions

RMQ5 Rock Mass

dm501.inp	Analysis input file
dm501Elem.def	Element and material properties

also files d20Nodes.def, d20Nsets.def,

and d20InTem.def, as listed for "Intact Rock".

RMQ1 Rock Mass

dm502.inp	Analysis input file
dm502Elem.def	Element and material properties

in addition to the common files d20Nodes.def, d20Nsets.def, and d20InTem.def

Each of the linear-elastic TM analyses included the calculation of safety factor (f_s) using the equation

$$f_s = \frac{2c\tan\alpha + \sigma_{min}\tan^2\alpha}{\sigma_{max}}$$

σ_{min} = minimum principal compressive stress
σ_{max} = maximum " " "
c = cohesion parameter
α = $45 + \phi/2$
ϕ = friction angle

Rock Category	C (MPa)	ϕ (°)
Intact rock	37.0	46
RMQ5 rock	5.08	34.4
RMQ1 rock	2.82	27.5

Contd on p. 89

dm500LF.f

```
      SUBROUTINE UVARM(UVAR,DIRECT,T,TIME,DTIME,CMNAME,ORNAME,
     1     NUVARM,NOEL,NPT,LAYER,KSPT,KSTEP,KINC,NDI,NSHR)
C
      INCLUDE 'ABA_PARAM.INC'
      CHARACTER*8 CMNAME,ORNAME,FLGRAY(15)
      DIMENSION UVAR(NUVARM),DIRECT(3,3),T(3,3),TIME(2)
      DIMENSION ARRAY(15),JARRAY(15)
C
C     Code DM500LF
C
C     Computes safety factor (strength/stress ratio) for model dm500
C     following procedure documented in CNWRA Scientific Notebook
C     #324 p. 87
C     Values of principal stress required for the calculation are
C     obtained through ABAQUS user-interface subroutine GETVRM
C
C     Externally supplied input parameter:
C
C     FRIC         Friction angle (degrees);
C     CPAR         Cohesion parameter (MPa);
C
C     The calculated safety factor and its reciprocal
C     are stored in vector UVAR as follows
C
C     Location in UVAR          Stored Variable
C     ----------------          ---------------
C          1                    safety factor, fs
C
C          2                    1/fs
C
      FRIC = 46.0
      CPAR = 37.0
C
      PI = 3.141592654
      ALPHA = PI/4.0 + (FRIC/2.0)*(PI/180.0)
      TA = DTAN(ALPHA)
C
C     Obtain current values of principal stress components
C
      JRCD = 0
      CALL GETVRM('SP',ARRAY,JARRAY,FLGRAY,JRCD)
      IF (JRCD .NE. 0) THEN
        WRITE(6,1000) NOEL,NPT,TIME(2)
        RETURN
      END IF
C
      PSMAX = -ARRAY(1)
      PSMIN = -ARRAY(3)
      FS = (2.0*CPAR*TA + PSMIN*TA*TA)/PSMAX
      UVAR(1) = FS
      UVAR(2) = 1.0/FS
      RETURN
C
C----67--1---------2---------3---------4---------5---------6---------7-*
C
 1000 FORMAT(/,'ERROR IN UVARM-CALL FOR VARIABLE PE',/,
     1       10X,'FOR ELEMENT NUMBER = ',I5,/,
     2       10X,'INTEGRATION POINT  = ',I5,/,
     3       10X,'AT TIME            = ',E12.3)
      END
```

dm502LF.f

```
      SUBROUTINE UVARM(UVAR,DIRECT,T,TIME,DTIME,CMNAME,ORNAME,
     1     NUVARM,NOEL,NPT,LAYER,KSPT,KSTEP,KINC,NDI,NSHR)
C
      INCLUDE 'ABA_PARAM.INC'
      CHARACTER*8 CMNAME,ORNAME,FLGRAY(15)
      DIMENSION UVAR(NUVARM),DIRECT(3,3),T(3,3),TIME(2)
      DIMENSION ARRAY(15),JARRAY(15)
C
C     Code DM502LF
C
C     Computes safety factor (strength/stress ratio) for model dm500 [dm502]
C     following procedure documented in CNWRA Scientific Notebook
C     #324 p. 87
C     Values of principal stress required for the calculation are
C     obtained through ABAQUS user-interface subroutine GETVRM
C
C     Externally supplied input parameter:
C
C     FRIC         Friction angle (degrees);
C     CPAR         Cohesion parameter (MPa);
C
C     The calculated safety factor and its reciprocal
C     are stored in vector UVAR as follows
C
C     Location in UVAR          Stored Variable
C     ----------------          ---------------
C          1                    safety factor, fs
C
C          2                    1/fs
C
      FRIC = 27.5
      CPAR = 2.82
C
      PI = 3.141592654
      ALPHA = PI/4.0 + (FRIC/2.0)*(PI/180.0)
      TA = DTAN(ALPHA)
C
C     Obtain current values of principal stress components
C
      JRCD = 0
      CALL GETVRM('SP',ARRAY,JARRAY,FLGRAY,JRCD)
      IF (JRCD .NE. 0) THEN
        WRITE(6,1000) NOEL,NPT,TIME(2)
        RETURN
      END IF
C
      PSMAX = -ARRAY(1)
      PSMIN = -ARRAY(3)
      FS = (2.0*CPAR*TA + PSMIN*TA*TA)/PSMAX
      UVAR(1) = FS
      UVAR(2) = 1.0/FS
      RETURN
C
C----67--1---------2---------3---------4---------5---------6---------7-*
C
 1000 FORMAT(/,'ERROR IN UVARM-CALL FOR VARIABLE PE',/,
     1       10X,'FOR ELEMENT NUMBER = ',I5,/,
     2       10X,'INTEGRATION POINT  = ',I5,/,
     3       10X,'AT TIME            = ',E12.3)
      END
```

dm501LF.f

```
      SUBROUTINE UVARM(UVAR,DIRECT,T,TIME,DTIME,CMNAME,ORNAME,
     1     NUVARM,NOEL,NPT,LAYER,KSPT,KSTEP,KINC,NDI,NSHR)
C
      INCLUDE 'ABA_PARAM.INC'
      CHARACTER*8 CMNAME,ORNAME,FLGRAY(15)
      DIMENSION UVAR(NUVARM),DIRECT(3,3),T(3,3),TIME(2)
      DIMENSION ARRAY(15),JARRAY(15)
C
C     Code DM501LF
C
C     Computes safety factor (strength/stress ratio) for model dm500 [dm501]
C     following procedure documented in CNWRA Scientific Notebook
C     #324 p. 87
C     Values of principal stress required for the calculation are
C     obtained through ABAQUS user-interface subroutine GETVRM
C
C     Externally supplied input parameter:
C
C     FRIC         Friction angle (degrees);
C     CPAR         Cohesion parameter (MPa);
C
C     The calculated safety factor and its reciprocal
C     are stored in vector UVAR as follows
C
C     Location in UVAR          Stored Variable
C     ----------------          ---------------
C          1                    safety factor, fs
C
C          2                    1/fs
C
      FRIC = 34.4
      CPAR = 5.08
C
      PI = 3.141592654
      ALPHA = PI/4.0 + (FRIC/2.0)*(PI/180.0)
      TA = DTAN(ALPHA)
C
C     Obtain current values of principal stress components
C
      JRCD = 0
      CALL GETVRM('SP',ARRAY,JARRAY,FLGRAY,JRCD)
      IF (JRCD .NE. 0) THEN
        WRITE(6,1000) NOEL,NPT,TIME(2)
        RETURN
      END IF
C
      PSMAX = -ARRAY(1)
      PSMIN = -ARRAY(3)
      FS = (2.0*CPAR*TA + PSMIN*TA*TA)/PSMAX
      UVAR(1) = FS
      UVAR(2) = 1.0/FS
      RETURN
C
C----67--1---------2---------3---------4---------5---------6---------7-*
C
 1000 FORMAT(/,'ERROR IN UVARM-CALL FOR VARIABLE PE',/,
     1       10X,'FOR ELEMENT NUMBER = ',I5,/,
     2       10X,'INTEGRATION POINT  = ',I5,/,
     3       10X,'AT TIME            = ',E12.3)
      END
```

These ABAQUS-user subroutine codes implement the calculation of fs described on p. 87. GW 8/23/2000

The strength parameters c and ϕ for intact rock were taken from the reference on p. 46. The values for RMQ5 and RMQ1 are the same as given on p. 48.

Processing of Results

Stress Histories

Principal-stress Histories

Histories of

- σ_{11} in-plane horizontal direct stress
- σ_{22} vertical direct stress
- σ_{33} out-of-plane horizontal direct stress

were obtained from ABAQUS output files using the ABAQUS/POST command files

StrsRoof.jnl	Roof point
StrsWall.jnl	Wall point
StrsPillar.jnl	Pillar point

which are documented on p. 90.

Principal-stress histories (Pmin and Pmax — minimum and maximum principal stresses) were extracted from ABAQUS output files using the following ABAQUS/POST command files.

~~Cont'd on p. 89~~ 91 GW 8/23/00

strsRoof.jnl

```
results,file=dm502
set, XYPrintFile=dm502RoofStrs.out
readcur,name=sva,var=s22,elem=2036,centroid
readcur,name=sha,var=s11,elem=2036,centroid
readcur,name=ssa,var=s12,elem=2036,centroid
readcur,name=s3a,var=s33,elem=2036,centroid
***
*** Sign-change the stress histories
***
definecur,name=vstrs,operation=multiply,const=-1
sva
definecur,name=hstrs,operation=multiply,const=-1
sha
definecur,name=sstrs,operation=multiply,const=-1
ssa
definecur,name=h3strs,operation=multiply,const=-1
s3a
printcurve
vstrs,hstrs,h3strs,sstrs

end
```

variables

GVO 8/23/2000

For each of these command files, modify values of variables file XYPrintFile as necessary (to process results for dm500, dm501, or dm502).

strsPillar.jnl

```
results,file=dm502
set, XYPrintFile=dm502PillarStrs.out
readcur,name=sva,var=s22,elem=1142,centroid
readcur,name=sha,var=s11,elem=1142,centroid
readcur,name=ssa,var=s12,elem=1142,centroid
readcur,name=s3a,var=s33,elem=1142,centroid
readcur,name=svb,var=s22,elem=614,centroid
readcur,name=shb,var=s11,elem=614,centroid
readcur,name=ssb,var=s12,elem=614,centroid
readcur,name=s3b,var=s33,elem=614,centroid
***
*** Average and sign-change the stress histories
***
definecur,name=vstrs,operation=sum
sva, -0.5
svb, -0.5

definecur,name=hstrs,operation=sum
sha, -0.5
shb, -0.5

definecur,name=sstrs,operation=sum
ssa, -0.5
ssb, -0.5

definecur,name=h3strs,operation=sum
s3a, -0.5
s3b, -0.5

printcurve
vstrs,hstrs,h3strs,sstrs

end
```

variables

GVO 8/23/2000

strsWall.jnl

```
results,file=dm502
set, XYPrintFile=dm502WallStrs.out
readcur,name=sva,var=s22,elem=2113,centroid
readcur,name=sha,var=s11,elem=2113,centroid
readcur,name=ssa,var=s12,elem=2113,centroid
readcur,name=s3a,var=s33,elem=2113,centroid
readcur,name=svb,var=s22,elem=2180,centroid
readcur,name=shb,var=s11,elem=2180,centroid
readcur,name=ssb,var=s12,elem=2180,centroid
readcur,name=s3b,var=s33,elem=2180,centroid
***
*** Average and sign-change the stress histories
***
definecur,name=vstrs,operation=sum
sva, -0.5
svb, -0.5

definecur,name=hstrs,operation=sum
sha, -0.5
shb, -0.5

definecur,name=sstrs,operation=sum
ssa, -0.5
ssb, -0.5

definecur,name=h3strs,operation=sum
s3a, -0.5
s3b, -0.5

printcurve
vstrs,hstrs,h3strs,sstrs

end
```

variables

GVO 8/23/00

psRoof.jnl

```
results,file=dm502
set, XYPrintFile=dm502Proof.out
readcur,name=p1,var=sp1,elem=2036,centroid
readcur,name=p3,var=sp3,elem=2036,centroid
***
*** Sign-change the p1 and p3 histories to obtain
*** pmax and pmin histories
***
definecur,name=pmax,operation=multiply,const=-1
p1
definecur,name=pmin,operation=multiply,const=-1
p3
printcurve
pmin,pmax

end
```

variables

psRoof.jnl Roof point
psFloor.jnl Floor point
psWall.jnl sidewall point
psPillar.jnl Pillar point.

psFloor.jnl

```
results,file=dm502
set, XYPrintFile=dm502Pfloor.out
readcur,name=p1,var=sp1,elem=2257,centroid
readcur,name=p3,var=sp3,elem=2257,centroid
***
*** Sign-change the p1 and p3 histories to obtain
*** pmax and pmin histories
***
definecur,name=pmax,operation=multiply,const=-1
p1
definecur,name=pmin,operation=multiply,const=-1
p3
printcurve
pmin,pmax

end
```

variables

For each of these command files, modify variables file XYPrintFile as necessary to process results for dm500, dm501, or dm502.

psWall.jnl

```
results,file=dm502
set, XYPrintFile=dm502Pwall.out
readcur,name=p113,var=sp1,elem=2113,centroid
readcur,name=p180,var=sp1,elem=2180,centroid
readcur,name=p313,var=sp3,eleme=2113,centroid
readcur,name=p380,var=sp3,elem=2180,centroid
***
*** Average and sign-change the p1 and p3 histories to obtain
*** pmax and pmin histories
***
definecur,name=pmax,operation=sum
p113, -0.5
p180, -0.5

definecur,name=pmin,operation=sum
p313, -0.5
p380, -0.5

printcurve
pmin,pmax

end
```

variables

GVO 8/23/2000

psPillar.jnl

```
results,file=dm502
set, XYPrintFile=dm502Ppillar.out
readcur,name=p1a,var=sp1,elem=1142,centroid
readcur,name=p1b,var=sp1,elem=614,centroid
readcur,name=p3a,var=sp3,eleme=1142,centroid
readcur,name=p3b,var=sp3,eleme=614,centroid
***
*** Average and sign-change the p1 and p3 histories to obtain
*** pmax and pmin histories
***
definecur,name=pmax,operation=sum
p1a, -0.5
p1b, -0.5

definecur,name=pmin,operation=sum
p3a, -0.5
p3b, -0.5

printcurve
pmin,pmax

end
```

variables

Pages 83–91 entries by GVO. 8/23/2000

September 13 2000 — Entries on p. 92–95 by GW

More on Results Processing for Analyses Described on p. 83–89

The outputs from ABAQUS/POST obtained using the command files on p. 90–91, which are listed below, were read into MICROSOFT EXCEL 97 SR-2 worksheets that are also listed below. The files are located in D:\ThermMech\NARMS2000\DrftScale. Each file name indicates dm500, dm501, or dm502; Pillar, Roof, Floor, or Wall; P for principal stress or Strs for stress components; and SPath for stress path or Strs for stress history. The .out files are output of codes on p. 90–91 and the .xls files are EXCEL 97 worksheets.

dm500PillarStrs.xls	31KB	Microsoft Excel Wor...	8/28/2000 3:00 PM
dm500RoofStrs.xls	31KB	Microsoft Excel Wor...	8/29/2000 1:44 PM
dm500SPath.xls	43KB	Microsoft Excel Wor...	8/30/2000 1:37 PM
dm500WallStrs.xls	31KB	Microsoft Excel Wor...	8/29/2000 9:40 AM
dm501PillarStrs.xls	32KB	Microsoft Excel Wor...	8/28/2000 3:07 PM
dm501RoofStrs.xls	31KB	Microsoft Excel Wor...	8/29/2000 1:58 PM
dm501SPath.xls	34KB	Microsoft Excel Wor...	8/30/2000 1:35 PM
dm501WallStrs.xls	31KB	Microsoft Excel Wor...	8/29/2000 9:48 AM
dm502PillarStrs.xls	32KB	Microsoft Excel Wor...	8/28/2000 3:14 PM
dm502RoofStrs.xls	31KB	Microsoft Excel Wor...	8/29/2000 2:04 PM
dm502SPath.xls	43KB	Microsoft Excel Wor...	8/30/2000 1:33 PM
dm502WallStrs.xls	31KB	Microsoft Excel Wor...	8/29/2000 10:02 AM
~~Hist.xls~~	~~26KB~~	~~Microsoft Excel Wor...~~	~~5/12/2000 10:56 AM~~
~~baseNodeList.out~~	~~2KB~~	~~OUT File~~	~~4/21/2000 1:52 PM~~
dm500Pfloor.out	3KB	OUT File	8/24/2000 2:29 PM
dm500PillarStrs.out	4KB	OUT File	8/28/2000 10:01 AM
dm500Ppillar.out	3KB	OUT File	8/24/2000 2:29 PM
dm500Proof.out	3KB	OUT File	8/24/2000 2:29 PM
dm500Pwall.out	3KB	OUT File	8/24/2000 2:29 PM
dm500RoofStrs.out	4KB	OUT File	8/28/2000 10:01 AM
dm500WallStrs.out	4KB	OUT File	8/28/2000 10:01 AM
dm501Pfloor.out	3KB	OUT File	8/24/2000 3:45 PM
dm501PillarStrs.out	4KB	OUT File	8/28/2000 10:01 AM
dm501Ppillar.out	3KB	OUT File	8/24/2000 3:45 PM
dm501Proof.out	3KB	OUT File	8/24/2000 3:45 PM
dm501Pwall.out	3KB	OUT File	8/24/2000 3:45 PM
dm501RoofStrs.out	4KB	OUT File	8/28/2000 10:01 AM
dm501WallStrs.out	4KB	OUT File	8/28/2000 10:01 AM
dm502Pfloor.out	3KB	OUT File	8/24/2000 4:14 PM
dm502PillarStrs.out	4KB	OUT File	8/28/2000 10:02 AM
dm502Ppillar.out	3KB	OUT File	8/24/2000 4:14 PM
dm502Proof.out	3KB	OUT File	8/24/2000 4:14 PM
dm502Pwall.out	3KB	OUT File	8/24/2000 4:14 PM
dm502RoofStrs.out	4KB	OUT File	8/28/2000 10:02 AM
dm502WallStrs.out	4KB	OUT File	8/28/2000 10:02 AM

GW 9/13/00

Postscript figure-files prepared using the EXCEL worksheets are listed below. The postscript files were assembled in Adobe Illustrator (.ai files) to produce the figure files also listed below.

Contents of 'D:\ThermMech\AiRpt2000\Figures'

Name	Size	Type	Modified
dm500PillarPSHist....	12KB	Postscript	8/28/2000 2:59 PM
dm500PillarSHist.ps	15KB	Postscript	8/28/2000 2:58 PM
dm500PillarSPath.ps	9KB	Postscript	8/30/2000 1:37 PM
dm500RoofPSHist...	12KB	Postscript	8/29/2000 1:45 PM
dm500RoofSHist.ps	15KB	Postscript	8/29/2000 1:45 PM
dm500RoofSPath.ps	9KB	Postscript	8/30/2000 1:20 PM
dm500WallPSHist....	12KB	Postscript	8/29/2000 9:39 AM
dm500WallSHist.ps	15KB	Postscript	8/29/2000 9:39 AM
dm500WallSPath.ps	9KB	Postscript	8/30/2000 1:19 PM
dm501PillarPSHist....	12KB	Postscript	8/28/2000 3:09 PM
dm501PillarSHist.ps	15KB	Postscript	8/28/2000 3:08 PM
dm501PillarSPath.ps	9KB	Postscript	8/30/2000 1:36 PM
dm501RoofPSHist...	12KB	Postscript	8/29/2000 1:59 PM
dm501RoofSHist.ps	15KB	Postscript	8/29/2000 1:59 PM
dm501RoofSPath.ps	9KB	Postscript	8/30/2000 1:27 PM
dm501WallPSHist....	12KB	Postscript	8/29/2000 9:48 AM
dm501WallSHist.ps	15KB	Postscript	8/29/2000 9:47 AM
dm501WallSPath.ps	9KB	Postscript	8/30/2000 1:27 PM
dm502PillarPSHist....	12KB	Postscript	8/28/2000 3:16 PM
dm502PillarSHist.ps	15KB	Postscript	8/28/2000 3:15 PM
dm502PillarSPath.ps	9KB	Postscript	8/30/2000 1:34 PM
dm502RoofPSHist...	12KB	Postscript	8/29/2000 2:04 PM
dm502RoofSHist.ps	15KB	Postscript	8/29/2000 2:04 PM
dm502RoofSPath.ps	9KB	Postscript	8/30/2000 1:35 PM
dm502WallPSHist....	12KB	Postscript	8/29/2000 10:02 AM
dm502WallSHist.ps	15KB	Postscript	8/29/2000 10:01 AM
dm502WallSPath.ps	9KB	Postscript	8/30/2000 1:34 PM
dt331TContours.ai	2,136KB	Adobe Illustrator Art...	9/12/2000 3:51 PM
dt331y025.ps	502KB	Postscript	8/31/2000 3:19 PM
dt331y050.ps	505KB	Postscript	8/31/2000 3:19 PM
dt331y100.ps	503KB	Postscript	8/31/2000 3:19 PM
dt331y150.ps	498KB	Postscript	8/31/2000 3:19 PM
pillarSHist.ai	296KB	Adobe Illustrator Art...	9/12/2000 3:49 PM
pillarSPath.ai	254KB	Adobe Illustrator Art...	9/12/2000 3:55 PM
roofSHist.ai	301KB	Adobe Illustrator Art...	9/12/2000 3:45 PM
roofSPath.ai	260KB	Adobe Illustrator Art...	9/12/2000 3:53 PM
wallSHist.ai	299KB	Adobe Illustrator Art...	9/12/2000 3:47 PM
wallSPath.ai	256KB	Adobe Illustrator Art...	9/12/2000 3:54 PM

These postscript files (dm500PillarPSHist through dm502WallSPath.ps) were generated from the EXCEL worksheets listed on p. 92.

These files (dt331TContours.ai through dt331y150.ps) show temperature contours as explained later.

These (pillarSHist.ai through wallSPath.ai) are Adobe Illustrator files that combine various postscript-file outputs from MICROSOFT EXCEL 97 SR-2

Analyses were also conducted to develop contours of temperature and rock-failure susceptibility (reciprocal of safety factor) at 25, 50, 100, and 150 yr. following waste emplacement. The analysis files are:

dt331.inp	Thermal analysis
dm511.inp	TM analysis with RMQ5 parameter set
dm512.inp	TM analysis with RMQ1 parameter set.

These files also use the .def files listed on p. ~~85~~ 86-87. The element property definition files (dt301Elem.def, dm500Elem.def, dm501Elem.def, and dm502Elem.def) each invoke the reading of three files that define the element connectivities. The three files are

GHP 9/13/2000

d20Quads.def	Quadratic elements
d20Triangles.def	Triangular elements
d20Elsets.def	Element sets.

Temperature contours were prepared using ABAQUS/POST and the following

Command files:

```
input,file=nfld50.def
set,fill
set,outline=per
set,clev=5
set,clegendsize=.25
set,cmin=80
set,cmax=120
```

} dt331ContPars.def — sets POST environment variables for contouring.

```
***
*** 25-yr temperature contour
***
restart,step=2
contour,var=nt11
pause
***
*** 50-yr temperature contour
***
restart,step=3
contour,var=nt11
pause
***
*** 100-yr temperature contour
***
restart,step=4
contour,var=nt11
pause
***
*** 150-yr temperature contour
***
restart,step=5
contour,var=nt11
```

} dt331Contours.def — plots temperature contours at 25, 50, 100, and 150 yr using the POST environment variables set in the above command file.

```
elset,elset=host
nrfld50,nrfld20
rr1zone,rr2zone

detail,elset=host
```

} File nfld50.def that was invoked from file dt331ContPars.def. This file defines a set of elements that lie ~~at~~ ±50 m from (above and below) the repository axis, using element sets already defined in the model through file d20Elsets.def (p. 94).

Entries on p. 92-95 by GHP 9/13/2000

September 19 2000

Pages 96-97 entries by [signature] 9/19/2000

Results Processing Cont'd

GVO 9/19/2000

~~Factor~~ Failure-susceptibility contours were prepared using ABAQUS/POST and the following command files:

GVO 9/19/2000

```
input,file=nfld50.def
set,fill
set,outline=per
set,clev=3
set,clegendsize=.15
```

File dm5ContPars.def sets up POST environment variables for contouring. File nfld50.def is described on p. 95.

GVO 9/19/2000

```
restart,step=2
set,cmin
set,cmax
set,cmin=0.5
set,cmax=1.0
cont,v=uvarm2
pause
restart,step=3
set,cmin
set,cmax
set,cmin=0.5
set,cmax=1.0
cont,v=uvarm2
pause
restart,step=4
set,cmin
set,cmax
set,cmin=0.5
set,cmax=1.0
cont,v=uvarm2
pause
restart,step=5
set,cmin
set,cmax
set,cmin=0.5
set,cmax=1.0
cont,v=uvarm2
pause
restart,step=6
set,cmin
set,cmax
set,cmin=0.5
set,cmax=1.0
cont,v=uvarm2
```

This file (named dm511Contours.def or dm512Contours.def) plots failure-susceptibility contours at end of excavation, and 25, 50, 100, and 150 yr following waste emplacement. The environment variables defined in file dm5ContPars.def are set prior to invoking this file.

GVO 9/19/2000

Contents of 'D:\ThermMech\AiRpt2000\Figures'

Name	Size	Type	Modified
failFactor.ai	4,742KB	Adobe Illustrator Art...	9/14/2000 1:54 PM
dm511y000fs.ps	439KB	Postscript	9/14/2000 11:21 AM
dm512y000fs.ps	474KB	Postscript	9/14/2000 11:21 AM
dm512y150fs.ps	469KB	Postscript	9/13/2000 4:45 PM
dm512y025fs.ps	469KB	Postscript	9/13/2000 4:45 PM
dm512y050fs.ps	468KB	Postscript	9/13/2000 4:45 PM
dm512y100fs.ps	469KB	Postscript	9/13/2000 4:45 PM
dm511y050fs.ps	501KB	Postscript	9/13/2000 4:44 PM
dm511y100fs.ps	501KB	Postscript	9/13/2000 4:44 PM
dm511y025fs.ps	502KB	Postscript	9/13/2000 4:44 PM
dm511y150fs.ps	495KB	Postscript	9/13/2000 4:44 PM
pillarSPath.ai	254KB	Adobe Illustrator Art...	9/12/2000 3:55 PM
wallSPath.ai	256KB	Adobe Illustrator Art...	9/12/2000 3:54 PM
roofSPath.ai	260KB	Adobe Illustrator Art...	9/12/2000 3:53 PM
dt331TContours.ai	2,136KB	Adobe Illustrator Art...	9/12/2000 3:51 PM
pillarSHist.ai	296KB	Adobe Illustrator Art...	9/12/2000 3:49 PM
wallSHist.ai	299KB	Adobe Illustrator Art...	9/12/2000 3:47 PM
roofSHist.ai	301KB	Adobe Illustrator Art...	9/12/2000 3:45 PM
dt331y025.ps	502KB	Postscript	8/31/2000 3:19 PM
dt331y050.ps	505KB	Postscript	8/31/2000 3:19 PM
dt331y100.ps	503KB	Postscript	8/31/2000 3:19 PM
dt331y150.ps	498KB	Postscript	8/31/2000 3:19 PM
dm500PillarSPath.ps	9KB	Postscript	8/30/2000 1:37 PM
dm501PillarSPath.ps	9KB	Postscript	8/30/2000 1:36 PM

Failure susceptibility contours.

GVO 9/19/2000

The temperature-contours files are listed on p. 93 and the failure-susceptibility contour files are listed on this page.

Pages 96-97 entries by [signature] 9/19/2000

Element Managers are requested to put the following statement at the conclusion of "manual" Scientific Notebooks:

"I have reviewed this scientific notebook and find it in compliance with QAP-001. There is sufficient information regarding procedures used for conducting tests, acquiring and analyzing data so that another qualified individual could repeat the activity.

9-30-04

(Element Manager signature and date above line, Name of Element beneath line)"